

ANEXO A

Tabla de Contenido

Anexo A	1
A.1 Red HFC	1
A.1.1 Cabecera	1
A.1.2 Interconexiones de Fibra.....	2
A.1.3 Nodos de Fibra.....	2
A.1.4 Red de distribución Coaxial.....	2
A.1.5 Equipos Terminales	2
A.2 Televisión Digital Interactiva – TDi	3
A.2.1 Historia de la interactividad.....	4
A.2.2 Interactividad.....	5
A.2.2.1 MHP	5
A.2.2.2 DVB-MHP.....	6
A.2.2.3 Tipos de aplicaciones MHP.....	6
A.3. MPEG	6
A.3.1 Compresión de vídeo MPEG.....	7
A.3.2 Compresión de audio MPEG.....	10
A.3.2.1 Audio MPEG layer 1	12
A.3.2.2 Audio MPEG layer 2	12
A.3.2.3 Audio MPEG layer 3 (MP3).....	12
A.4. MPEG-2.....	13
A.4.1 Paquetes MPEG-2.....	14
A.4.1.1 MPEG-2 Elementary Streams (ES)	16
A.4.1.2 MPEG-2 Packetised Elementary Streams (PES)	17
A.4.1.3 Flujo de Programa (PS, Program Stream).....	18
A.4.2 MPEG-2 Transport Stream (TS).....	19
A.4.2.1 Program Specific Information (PSI)	21
A.4.3 Trama de Transporte DVB	22
A.4.3.1 TABLAS DVB	24
A.4.4 Protocolo DSM-CC	25
Referencias	27

Lista de Figuras

Figura A.1. Arquitectura básica de una red HFC.....	1
Figura A.2. Pila software MHP	6
Figura A.3. Formato 4:2:0.....	7
Figura A.4. Formato 4:2:2.....	7
Figura A.5. Proceso de cuantificación MPEG.....	8
Figura A.6. Proceso de escaneo de coeficientes.....	8
Figura A.7. Proceso de escaneo de coeficientes.....	8
Figura A.8. Desplazamiento de macrobloques	9
Figura A.9. Grupo de imágenes (GOP).....	10
Figura A.10. Límite mínimo para percepción sonora	10
Figura A.11. Enmascaramiento de frecuencias	11
Figura A.12. Paquetes MPEG-2.....	14
Figura A.13. Multiplexación de paquetes MPEG-2	15
Figura A.14. Ampliación del sistema de multiplexación de paquetes	16
Figura A.15. MPEG-2 Elementary Streams	16
Figura A.16. Audio Elementary Stream	17
Figura A.17. Paso de ES a PES	17
Figura A.18. Estructura de un paquete PES	18
Figura A.19. Generación de paquetes TS a partir de los paquetes PES.....	20
Figura A.20. Ejemplo gráfico de MPEG-2.....	20
Figura A.21. Organización lógica de los servicios	23
Figura A.22. Diferentes servicios y elementary streams	23
Figura A.23. Tablas DVB.....	25

Anexo A

A.1 Red HFC

Las redes HFC son el resultado de un proceso evolutivo que se ha llevado a cabo sobre las redes de distribución de televisión por cable - CATV (Community Antenna Television – Televisión por cable). La transformación de las redes de cable se debe a que la tendencia al aumento en la oferta de canales como estrategia de la expansión del negocio, requiere aumento del ancho de banda disponible. Para resolver esta limitación se introdujo la tecnología óptica dando origen a las redes HFC multiservicio en las que se hace posible la creación de topologías de red susceptibles de transportar señales bidireccionales (Berrocal, y otros, 2003), (Andueza, 2004). Las redes CATV deben tener en cuenta las siguientes modificaciones para transformarse en una red multiservicio (España, 2003), (ETSI_EN_300_744, 2004-11), (ETSI_EN_300_429_V1.2.1, 1998).

- ✓ Debe habilitarse en la red HFC la transmisión de la información en ambos sentidos, la información que es enviada al suscriptor (downstream) y la que proviene de éste (upstream), son alojadas en diferentes bandas de frecuencia dentro del espectro que utiliza el cable (CableLabs, 2006).
- ✓ Para la recepción de los servicios digitales, se dota a los usuarios de un equipo denominado cable modem – CM y en la cabecera es necesario añadir equipos llamados sistema de terminación de los cable módems – CMTS
- ✓ Establecer mecanismos de seguridad y privacidad, disponer de un sistema de autorización de acceso y de tarificación.
- ✓ Capacidad de comunicarse con abonados de otras redes.

Las redes HFC suponen un paso intermedio respecto a la *FTTH* (Fiber To The Home - Fibra hasta la casa), puesto que utilizan la fibra óptica en la parte troncal hasta ‘la manzana’ del usuario y en el último tramo (hasta la vivienda del usuario), se utiliza el cable coaxial. A continuación se presente una breve descripción de los elementos físicos que componen la Arquitectura básica de una red HFC típica. Ver Figura A.1

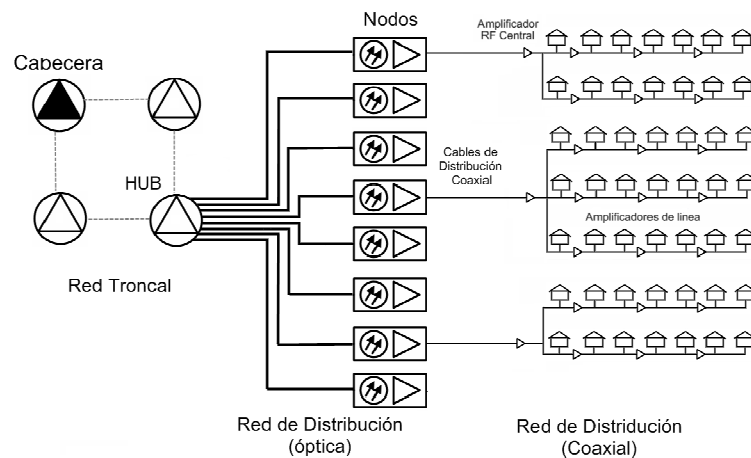


Figura A.1. Arquitectura básica de una red HFC

A.1.1 Cabecera

La principal función de las Cabeceras es combinar distintas fuentes de información para introducirlas en la red, en el caso de la televisión por cable, la cabecera es encargada de combinar las señales provenientes de distintos lugares y medios físicos e incluso tipos de información y formas de codificación. Antiguamente los operadores de cable combinaban señales provenientes de satélites, cables, e incluso antenas radioeléctricas. En la actualidad los nuevos operadores de cable son, en su mayoría operadores multiservicio, proporcionando telefonía e Internet de alta velocidad (España, 2003).

Para la conexión a Internet, es necesario una comunicación bidireccional y ésta se obtiene asignando una parte relativamente pequeña del ancho de banda con un tasa de datos de 500 kbps a 1.5 Mbps que se conoce como canal upstream (o canal de subida) de comunicación de datos del suscriptor al operador. Por otro lado el ancho de banda de descarga suele ser más grande, con una tasa de datos de hasta 39 Mbps, que se conoce como comunicación de datos del canal downstream (o canal de bajada) del operador al suscriptor. En esta técnica se utilizan dos dispositivos básicos: un módem en el extremo del usuario conocidos como Cablemódems, CM y en la cabecera del operador un Sistema de Terminación de Cablemódems, CMTS. El operador conecta su cabecera a un Proveedor de Servicios de Internet, ISP. Los CMTS pueden manejar la conexión a Internet de entre 4.000 y 30.000 usuarios. Una cabecera puede tener más de un CMTS. El CMTS es el dispositivo encargado de realizar la codificación, modulación y gestión de acceso al medio compartido por los CM. En la cabecera se encuentran además otros equipos como el Switch Ethernet, un servidor AAA (Authentication, Authorization and Accounting), para control de acceso y tarificación y un servidor de contenidos locales y de caching para las páginas más visitadas.

Los proveedores de cable también tendrán servidores para la contabilidad y registrando, protocolo de configuración dinámica de host, DHCP para asignar y administrar las direcciones IP de todos los usuarios de sistema del cable, y control de los servidores. Las redes HFC más evolucionadas proponen el servicio telefónico sobre IP. Esto es posible gracias al estándar DOCSIS en el que se especifican mecanismos para garantizar la calidad de servicio en tiempo real. En este caso, la voz y los datos comparten los mismos canales en la red HFC (Wolff, 2008).

A.1.2 Interconexiones de Fibra

La fibra óptica se utiliza para transmitir a lo largo de la red señales seleccionadas de todos los tipos y comúnmente en formato digital. En las redes HFC provee la conectividad desde donde se genera el espectro FDM y el punto central de distribución coaxial. También es utilizada para comunicar las Cabeceras y con proveedores externos de información (como Broadcasters y proveedores de Internet) (España, 2003), (Wolff, 2008).

A.1.3 Nodos de Fibra

Los nodos de fibra son básicamente la interconexión entre una línea de fibra óptica y la red de distribución coaxial. Un nodo esta compuesto básicamente de un receptor óptico que alimenta a un amplificador de downstream y opcionalmente, puede contener un transmisor de upstream, también conectado a la red coaxial.

Algunos nodos tienen la opción de tener múltiples entradas de fibras para downstream que son alimentadas por receptores ópticos diferentes cuyas salidas son combinadas mediante el uso de filtros, mejorando la calidad de la señal y la flexibilidad de la red. A esta configuración se le conoce con el nombre de Entradas de Fibra Múltiple Split-Band. Una alternativa para utilizar múltiples entradas de fibra para downstream es la redundancia para la fibra óptica en redes del tipo Trunking (función para conectar dos switches, enrutadores o servidores, mediante dos cables en paralelo en modo Full-Duplex). Consiguiendo un ancho de banda del doble para la comunicación entre los switches. A esta configuración se le conoce con el nombre de Entradas de Fibra Múltiple Redundante (Wolff, 2008).

A.1.4 Red de distribución Coaxial

El parámetro más elemental en una red coaxial es el Ancho de Banda, el cual es determinado por los equipos de amplificación, que son utilizados en dicha red. Algunos operadores de redes HFC sugieren utilizar redes coaxiales pequeñas para evitar utilizar amplificadores entre la parte óptica y el usuario. El segundo parámetro importante a considerar en una red HFC es el tamaño del área de la red coaxial. El total de usuarios conectados a la misma red coaxial comparten el ancho de banda (González, 2002),(Wolff, 2008).

A.1.5 Equipos Terminales

Los equipos terminales proveen la interfaz entre los dispositivos de aplicaciones del usuario final y el sistema de distribución compartido. Dependiendo del servicio, los equipos terminales pueden realizar funciones de transformación de formato de los datos de la red, así como testeo y otras funciones de seguridad. Por lo general son la cara más visible de la red y proveer a los usuarios de estos equipos representa de una gran inversión debido principalmente al alto número de equipos que puede tener una red (González, 2002),(Wolff, 2008).

Los equipos más importantes son: Equipos In-Home, que se encuentran al interior de los hogares conectados directamente a la red o como equipos intermedios entre el Drop y el receptor del subscriber; Equipos Punto-de-entrada (Point-of-Entry), que proporcionan medios para aislar las redes de cable direccionales de las señales generadas al interior de los hogares, ese equipo puede manejar un canal de Ethernet; y Equipos Terminales Compartidos que corresponden a equipos con multipuertos para ser compartidos por más de un usuario. La más comúnmente utilizada es la configuración híbrida que utiliza equipos Punto-de-entrada o equipos terminales compartidos coordinados con equipos In-Home. Esta configuración es más costosa, pero brinda a la red una gran flexibilidad y al operador la facilidad de elegir entre distintos vendedores de equipos (Wolff, 2008).

A.2 Televisión Digital Interactiva – TDi

El manejo y procesamiento de las señales que varían de manera continua en el tiempo (como ondas de audio y video) se transforma en información digital es decir, en bits, brindando las ventajas de un código simbólico preciso (un alfabeto) sobre un viejo sistema analógico. Una vez se hacen las representaciones por un número limitado de símbolos (0,1), se pueden aplicar procesos de compresión y corrección de errores, lo que permitirá disfrutar de una mejor calidad de imagen y de sonido. Por ejemplo una señal puede ser protegida por el envío de códigos especiales, que permiten que el decodificador digital pueda funcionar cuando los errores hayan ocurrido, brindando las capacidades a un televisor digital de que puede clasificar la interferencia impulsiva de la señal deseada (Perales, Benoit, & Días, 2004).

Así, la TD es la evolución de la Televisión analógica surgiendo como un nuevo concepto que proporcionara la posibilidad, no sólo de recibir muchos más canales con mayor calidad en su receptor, sino también de añadir aplicaciones que brinden interactividad, esto es la Televisión Digital Interactiva (TDi), lo que permitirá a los espectadores constituirse en elemento activo dentro del mundo audiovisual. Las nuevas potencialidades inherentes a la televisión digital, en especial, la interactividad, permitirán instaurar al televisor como vía alternativa de acceso a los servicios de la Sociedad de la Información (Brice, 2003).

No obstante la TDi está ligada a la elección de un estándar de TD que para el caso de esta investigación ha sido el europeo, DVB-C (Digital Video Broadcasting-Cable) (ETSI, EN 300 744 V1.5.1, 2004-11) (ETSI_EN_300_429_V1.2.1, 1998), puesto que la Universidad de Oviedo donde se recogen muestras de tráfico es Europea y el estándar escogido para Colombia es el DVB, además el estándar escogido presenta las siguientes ventajas respecto al americano y japonés (Marianov, Oberli, & Rios, 2006) (Simoneta, 2002):

- ✓ Trabaja con dos modos de transmisión: transmisión no jerárquica y jerárquica. Esto le da flexibilidad para configurar las transmisiones en función de tasas de datos deseadas y cobertura requerida.
- ✓ Mejor inmunidad a propagación de multitrayectoria.
- ✓ El tamaño del mercado mundial de equipos DVB es superior, garantizando gran variedad de dispositivos a costos competitivos.
- ✓ El estándar DVB fue el escogido por Colombia.

DVB-C se basa en el empleo de la fibra óptica, que hace posible la transmisión de todo tipo de mensajes y contenidos (voz, datos, imágenes fijas, vídeo, sonidos, música y otros formatos), a altísima velocidad, en gran volumen al mismo tiempo y con excelente resolución. Así pues, el usuario no se va a limitar a recibir información, sino que puede definir qué clase de contenidos quiere, o influir en los mismos gracias al concepto de interactividad, en función de su propia libertad de elección (MITC, 2008). La Televisión Digital por Cable es entonces el resultado de la aplicación de la tecnología digital a la señal de televisión, para luego distribuirla por medio de redes híbridas de fibra óptica y cable coaxial.

La Interactividad en la TD puede entenderse como la evolución de la televisión convencional debido a la confluencia de las tecnologías de televisión digital y las TIC. La nueva posibilidad introducida por la Interactividad es la “emisión interactiva”, donde gracias al uso de un canal de retorno (como por ejemplo una conexión telefónica o de cable) se posibilita una interacción real con el programa de televisión. La red HFC brinda un canal de retorno que permite la interactividad sin la necesidad de usar ninguna otra red (CableLabs, 2006). La emisión interactiva habilita

también nuevas áreas de aplicación como: servicios de información, servicios de entretenimiento, video bajo demanda, comercio electrónico, educación virtual, servicios de salud, etc siendo el usuario el que decide si quiere o no verlos, y cuando verlos.

A.2.1 Historia de la interactividad

La difusión de televisión tradicional es un medio pasivo de un solo sentido. El concepto de telespectador que interactúa con los contenidos de televisión no es extraño incluso para los pioneros en la difusión de contenidos audiovisuales en el modelo tradicional de televisión. Entre 1952 y 1957, el canal CBS emitió la serie infantil *Winky Dink and You*, la cual podría considerarse como el primer programa interactivo de la historia de la televisión. Esta interacción se proporcionaba por medio de unos pedazos de plástico que se colocaban en la pantalla de la televisión y que debido a la electricidad estática se mantenían en dicha posición. El objetivo del programa era que los niños ayudasen al protagonista a superar los retos que se iban presentando por medio de dibujos que se solaparían con el contenido presentado en pantalla. Por ejemplo, el programa solicitaba ayuda a los telespectadores cuando el protagonista llegaba a un desfiladero que debía cruzar, lo cual debía traducirse en el acción por parte de los niños de dibujar un puente que le permitiese cruzar. El hecho de que muchos niños no hiciesen uso del 'software adicional' (el plástico) y realizasen su interacción directamente sobre la pantalla de la televisión hizo que los padres forzasen a la CBS la retirada del programa.

De cualquier forma, el concepto introducido en ese momento ya no desaparecería. Otra forma de interactividad es la conversación, tal y como demostrarían Silvester (Pat) Weaver en 1950 a través del programa *The Tonight Show with Steve Allen* de la NBC lo que convertiría a los Shows en una de las killer-app de la televisión tradicional.

El primer gran proveedor de TDi mediante una estructura de red basada en cable en EEUU fue la Warner Communications Qube, la cual comenzó su actividad en diciembre de 1977 en Columbus-Ohio. El sistema Qube constaba de un set-top box integrado por una pequeña CPU y un módulo de memoria en la que se almacenaban preferencias del cliente. Qube ofrecía un total de 30 canales divididos entre canales de difusión, pago por visión (PPV), e incluso canales interactivos propios. La interfaz entre el usuario y el STB se realizaba mediante un mando propietario de control 'remoto', conectado por cable al STB, y que permitía al usuario la selección de canales, el acceso a los canales PPV, y proporcionar las respuestas en la programación interactiva. Los botones de propósito general del mando de control tomaban 'significado' dependiendo del programa interactivo que se visualizaba en cada momento. Y aunque el nuevo sistema tuvo un éxito aceptable, el modelo de negocio en el que se sustentaba no pudo hacer frente al éxito de los VCRs, y las películas para vídeo doméstico que tan rápidamente se hicieron populares y que forzaron el fin de toda actividad del sistema Qube en 1984.

En 1979 el Gobierno Británico comienza a ofrecer a través de la BBC el servicio Teletext, mediante el que se permitía a los televidentes del canal el intercambio de mensajes de texto enviados mediante el teléfono. El sistema Teletext aprovechaba los Vertical Blanking Interval (VBI) entre líneas de vídeo para la transmisión de los mensajes que se visualizaban en pantalla como si de una página de texto se tratase. El modelo Teletext británico fue ampliamente adoptado en múltiples países y todavía persiste en la actualidad (aunque con un uso diferente).

En 1980 TCI y Time Warner llevaron a cabo sus primeras experimentaciones con sus propias versiones de TDi sobre algunos mercados de prueba. Los clientes de estos servicios podían comprar online, juegos entre vecinos y muchos otros servicios de los considerados como diferenciales en la actualidad. Nuevamente los servicios ofrecidos tuvieron una buena acogida entre los usuarios seleccionados para las pruebas. Sin embargo, la escalabilidad total del sistema estaba bastante limitada debido a la red de baja capacidad sobre la que se sustentaba y los costos de mantenimiento de aquellos sistemas eran muy elevados principalmente debido al proceso por parte de los servidores de la época de todo el tráfico simultáneo generado por cada programa interactivo. Por ello, tanto TCI como Time Warner decidieron abandonar sus proyectos al no ser capaces de cubrir los costos de operación manteniendo unos precios razonablemente competitivos hacia el consumidor.

En 1994 Time Warner lo intenta de nuevo en Orlando por medio de Full Service Network (FSN) en el que se ofrecían servicios online de compra, juegos, canales temáticos, guía de programación electrónica e incluso vídeo bajo demanda. La infraestructura de FSN era extremadamente compleja necesitando un elevado número de servidores en los que almacenar digitalmente los contenidos ofrecidos e interconectados mediante circuitos ATM

directamente a los STB en los hogares de los clientes a los que se les ofrecía un caudal de 30 imágenes por segundo que necesitaban ser decodificadas por una CPU 5 veces más potente a sus homólogas habituales de los PCs de la época. Aunque el potencial de FSN era enorme, Time Warner decidió ofrecer demasiado en muy poco tiempo y se vio desbordada por el incremento de complejidad en la integración de todos los servicios lo que nuevamente desembocó en el abandono del proyecto.

En 2000 WebTV (MicrosoftTV hoy) comenzó la realización de pruebas de su sistema en Baltimore y San Diego. Los teleespectadores podían pulsar el botón 'Educational', que se ofrecía sobre la pantalla, para acceder a contenido parecido al Web para la televisión, canales locales como PTV y KPBS. La compañía que se encontraba detrás de este sistema en sus orígenes era Liberate Technologies, hoy líder en el sector de desarrollo software para los STB de cable.

Dos implementaciones de los tradicionales servicios interactivos sobreviven todavía en la actualidad. Éstos son el mencionado Teletexto y los Captions (Subtítulos). Ambos sistemas se basan en una forma de interactividad local con el STB no necesitando por tanto infraestructura de retorno. Este aspecto podría considerarse un indicador para los proveedores de los futuros servicios de TDi de cómo deben afrontarse los nuevos servicios, difundiendo los contenidos interactivos e implementando la interactividad con el usuario de forma local con el STB

A.2.2 Interactividad

Básicamente, el nivel de la interactividad en la TDi lo marca el elemento con el cuál finalmente se interactúa. De este modo, se diferencian dos tipos de interactividad que se corresponden con el Set-top box (STB) o interactividad local y la interactividad Remota que puede ser con el operador de red o con el Proveedor de contenidos.(GNU, 2008). DVB ha creado diversos grupos de trabajo específicos para estas tareas de interactividad que son englobadas bajo un nuevo término: La Plataforma Multimedia del Hogar o MHP (Multimedia Home Platform) (GTCF, 2005).

La ITU-T define a la televisión interactiva (iTV, Interactive Television) como "un servicio en el que el usuario puede enviar solicitudes, en un ambiente de navegación, al proveedor del servicio con el fin de obtener información adicional" (González, 2008). En consecuencia, se crea la posibilidad de personalizar e influir en el contenido que se muestra en el televisor, bien sea accediendo a información enviada durante el proceso de emisión, pero que solo se hace visible si el espectador lo desea, o bien accediendo a servidores con los que puede intercambiar información a través de un canal de retorno utilizando el televisor como interfaz de salida(Jokipelto, 2006) (Meritxell, 2002). El middleware de desarrollo del proyecto EDiTV está basado en las recomendaciones del DVB por lo tanto los niveles de interactividad que se consideran son los que define esta plataforma (MHP T. M., 2006).

A.2.2.1 MHP

DVB proporciona un entorno basado en MPEG-2 para transmitir señal de TV a través de diferentes medios como Satelite, Terrestre y Cable. MHP que se encuentra dentro de la misma definición añade la posibilidad técnica de ejecutar aplicaciones en el receptor permitiendo además el uso de un canal de retorno que puede basarse en diferentes soportes.

La definición de MHP es independiente además del fabricante de hardware, del emisor y del que implementa la especificación. Es decir, diferentes redes con diferentes fabricantes de STBs y de implementaciones podrán ejecutar las mismas aplicaciones MHP(MHP T. M., 2006).

• Niveles de Interactividad MHP

- ✓ Broadcast only: aplicaciones de solo difusión, tales como teletexto digital y grabación de video personal (PVR), obteniendo los datos requeridos desde el flujo broadcast. Solo soporta una interactividad local, donde el usuario puede interactuar localmente con los datos almacenados en el dispositivo final
- ✓ Interactividad unidireccional: las aplicaciones con interactividad unidireccional, tales como votación en línea y respuestas a publicidad, permiten al usuario proveer datos de respuesta en una simple dirección, no obteniendo una respuesta directa desde un canal de retorno del servidor.

- ✓ Interactividad Bidireccional: las aplicaciones con interactividad bidireccional, tales como e-mails, navegación web, y juegos en línea, permiten al usuario adquirir datos desde una fuente fuera del flujo broadcast, a través de un canal de retorno desde el servidor.

A.2.2.2 DVB-MHP

MHP define una plataforma común para las aplicaciones interactivas de la televisión digital, independiente tanto del proveedor de servicios interactivos como del receptor de televisión utilizado. De este modo, MHP favorece la creación de un mercado horizontal donde aplicaciones, red de transmisión y set-top box pueden ser suministrados por proveedores o fabricantes independientes.

DVB selecciona Java como lenguaje de desarrollo de aplicaciones, definiendo una serie de APIs, ver Figura A.2, en las que se incluyen a JavaTV, HAVi (interfaz de usuario), y DAVIC (Digital Audio Video Council). Estas aplicaciones java serán descargadas y ejecutadas directamente en el STB del usuario por medio de la aplicación especial Navigator que será la que permita el acceso a las mismas.

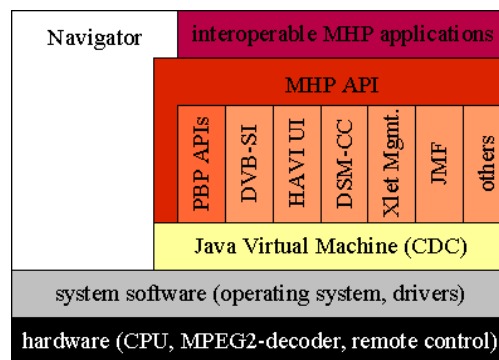


Figura A.2. Pila software MHP

A.2.2.3 Tipos de aplicaciones MHP

Las aplicaciones MHP se pueden dividir en 2 tipos. El primero de ellos es DVB-HTML. Este tipo de aplicaciones no son muy populares, en parte debido a la especificación de DVB-HTML sólo se completa con MHP 1.1, y en parte debido a que muchos organismos de radiodifusión, fabricantes de dispositivos e industria de contenidos les resulta demasiado complejo y difícil de aplicar. Las aplicaciones DVB-HTML se constituyen de un conjunto de páginas HTML que se emiten mediante difusión como parte de un servicio. La especificación se basa en torno a una versión de XHTML 1,1, y también incluye CSS 2.0 (Cascading Style Sheets), DOM 2.0 (Document Object Model), y ECMAScript (European Computer Manufacturers Association).

El segundo tipo más popular es DVB-J (DVB-Java). Estas serán las aplicaciones escritas en Java utilizando la API MHP API e integradas como un conjunto de archivos de clase (.class) que se emiten mediante difusión como parte de un servicio. A este tipo de aplicaciones se las conoce comúnmente como Xlets. Se trata de un concepto similar a los applets para páginas web o las aplicaciones J2ME para móviles que, por supuesto, ha sido introducido por Sun en su JavaTV. Al igual que los applets, las xlet deben implementar una determinada interfaz que permite a una fuente externa (el gestor del ciclo de vida de las aplicaciones o Navigator) iniciar y detener la aplicación.

A.3. MPEG

En este apartado se exponen los conceptos generales sobre comprensión de video y audio y se estudian en detalle los conceptos relacionados con MPEG-2 relativos a la TD, las secciones 3.1 y 3.1 han sido extraídas de (Álvarez, 2008)

A.3.1 Compresión de video MPEG

De forma general un video consiste en un número variable de grupos de imágenes (GOP, group of pictures). Matemáticamente, cada imagen es realmente una unión de los valores que representan a un píxel: una componente de luminancia y dos de crominancia; es decir, tres matrices de píxeles. Cada imagen se divide en bloques de píxeles de 8x8 no solapados. Cuatro de estos bloques se organizan adicionalmente en un bloque mayor de tamaño 16x16, llamado macrobloque.

Debido a que el ojo humano no es muy sensible a los cambios de información cromática comparada con la información de luminancia (redundancia psicovisual), las matrices de información cromática son reducidas en tamaño por un factor de dos en ambas direcciones horizontal y vertical. Por tanto hay una cuarta parte de números de píxeles de crominancia para procesar con los píxeles de luminancia. Este formato, denominado formato (4:2:0), se emplea en MPEG-1, ver Figura A.3.

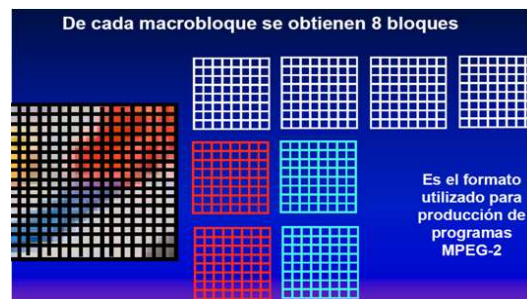


Figura A.3. Formato 4:2:0

MPEG-2 adicionalmente permite la posibilidad de no diezmar o sólo diezmar horizontalmente la componente cromática consiguiendo formatos 4:4:4 y 4:2:2 respectivamente, ver Figura A.4.

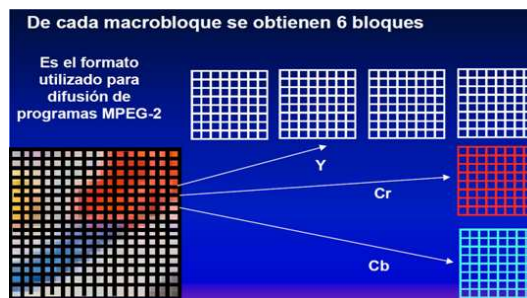


Figura A.4. Formato 4:2:2

La Transformada Discreta Coseno (CDT) se aplica a cada bloque de 8x8 individualmente. La transformada explota la correlación espacial de los píxeles convirtiéndolos en un conjunto de coeficientes independientes en el que cada uno representa la contribución de las frecuencias horizontales y verticales a la imagen. La DCT no reduce directamente la cantidad de información a transmitir, de hecho, se necesitaría una DCT con coeficientes de 11 bits para recuperar la imagen original. La compresión surge de transmitir los coeficientes de manera inteligente (cuantificación, Zig-Zag, Run-Lenght Code, VLC). Una vez que los coeficientes de la CDT han sido calculados. Éstos deben ser cuantificados (divididos entre los coeficientes de la matriz de cuantificación) de forma que se obtendrán coeficientes próximos a 0, ver Figura A.5.

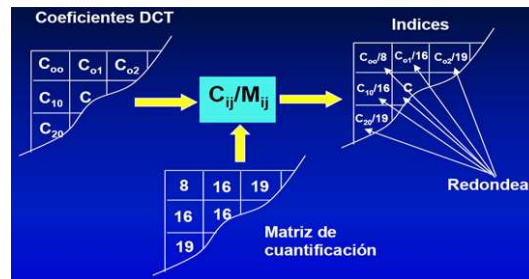


Figura A.5. Proceso de cuantificación MPEG

Una vez cuantificados, los coeficientes son escaneados siguiendo un patrón en zig-zag para crear una secuencia de una dimensión que prioriza los coeficientes más significativos, ver Figura A.6.

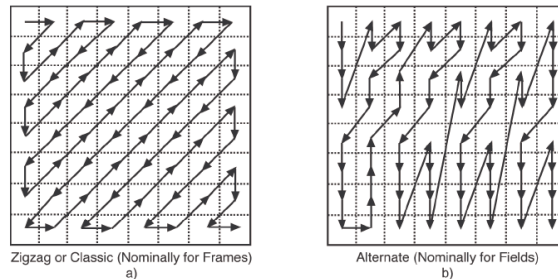


Figura A.6. Proceso de escaneo de coeficientes

En la información de video, no todas las frecuencias espaciales están presentes simultáneamente por lo que parte de los coeficientes de la matriz serán cero. La codificación RLC (Run Length Coding) surge para un manejo óptimo de los coeficientes nulos con respecto a aquellos que sí son significativos. Cada coeficiente diferente de cero se asocia con un par de apuntadores. El primer apuntador será su posición en el bloque que se indica por el número de ceros entre él y el coeficiente anterior diferente de cero (run length). El segundo apuntador será el valor del propio coeficiente cuantificado. Además, la probabilidad de ocurrencia de los posibles coeficientes no es equiprobable con los que resulta beneficioso el uso de codificaciones VLC (Variable Length Coding) mediante las cuales se asigna códigos con un inferior número de bits a aquellos coeficientes más probables mientras que a los menos probables se le asignarán códigos de longitud superior. Los códigos (datos comprimidos) serán posteriormente almacenados en un buffer que, gracias a una vía de realimentación hacia el cuantificador, es capaz de mantener un flujo de datos constante gracias al ‘ajuste’ en tiempo real de los valores de la matriz de cuantificación y con ello la cantidad de códigos que en él es necesario almacenar, ver Figura A.7.

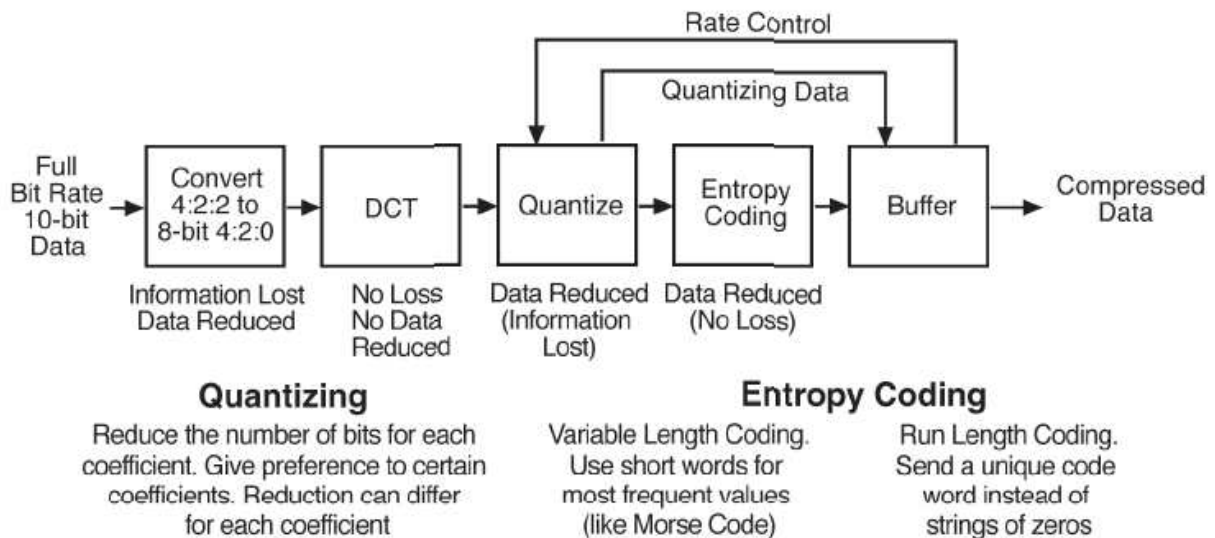


Figura A.7. Proceso de escaneo de coeficientes

La DCT explota adecuadamente la redundancia espacial, incluyendo en su sistema de compresión a la redundancia psicovisual y de codificación. Pero no resulta adecuada para detectar la redundancia temporal de información (sucesión de imágenes). Esta redundancia puede ser eliminada sencillamente mediante la codificación exclusiva de las diferencias entre imágenes individuales. Estas “imágenes diferencia” pueden ser tratadas como una imagen común y pueden ser codificadas mediante las técnicas de compresión espacial comentadas anteriormente. Un supuesto decodificador simplemente debería almacenar la codificación espacial inicial e ir añadiendo a ésta las imágenes diferencia para obtener las sucesivas imágenes finales. El esquema propuesto presenta un problema debido a que resultaría imposible decodificar las imágenes si no se recibe la imagen inicial (este fenómeno sucede parcialmente en los sistemas actuales de televisión en lo que cuando el usuario cambia de canal, éste no se reproduce automáticamente como ocurría en los sistemas de televisión analógica tradicional). Este problema puede ser resuelto no haciendo al sistema completamente diferencial, sino que periódicamente se transmite una imagen completa para poder rearmar el sistema. Estas imágenes completas se denominan Intra-coded (I) y sólo es comprimida espacialmente.

Otro problema adicional del sistema basado en imágenes diferencia surge de la propia naturaleza del movimiento que hace que se reduzcan las similitudes entre imágenes y hace necesario introducir el concepto de “compensación de movimiento” que reduzca la cantidad necesaria de información en la imagen diferencia. Cuando un objeto se mueve por la pantalla, éste aparece en diferentes posiciones pero sin cambiar mucho su aspecto. Este movimiento es medido por el codificador como un vector, denominado vector de movimiento, y que el decodificador utilizará para mover partes de la imagen previa hacia las nuevas posiciones en la imagen actual. El medidor de movimiento del codificador utiliza la diferencia de luminancia entre imágenes sucesivas. Un macrobloque de la imagen previa es utilizado como referencia, moviéndolo sobre su posición inicial y buscando la correlación entre este desplazamiento y la imagen posterior ver Figura A.8. Si se alcanza un alto grado de correlación en alguno de estos movimientos, ésta se asume como una buena estimación del vector de movimiento. La carga computacional de buscar el macrobloque que más se parezca es muy alta por lo que se limita la búsqueda a una zona determinada (Hierarchical block matching).

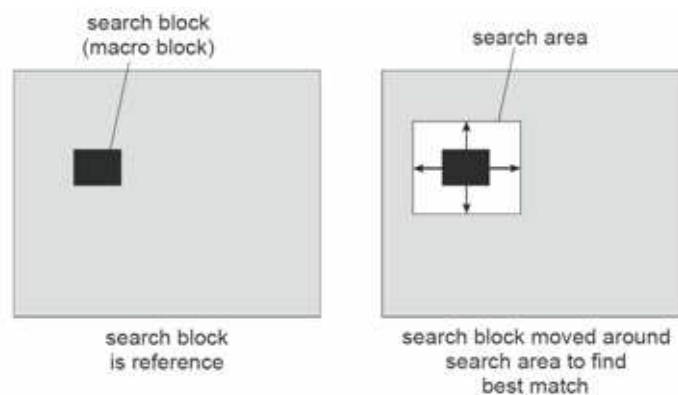


Figura A.8. Desplazamiento de macrobloques

Además, cuando un objeto se mueve, éste va ocultando la imagen de fondo que se encuentra inmediatamente delante y va haciendo aparecer la imagen de fondo inmediatamente posterior al mismo. Esta imagen de fondo ‘revelada’ necesitará de transmisión de información ya que la imagen posterior carece de información relativa a dicha área. Si un fondo está siendo revelado por un objeto en movimiento, dicho fondo será visible por completo en una imagen posterior por lo que parte de la información de esta imagen posterior puede ser retrasada temporalmente (en la sucesión teórica de imágenes) para ser usada en la composición de la imagen en la que dicho fondo está siendo revelado. A este concepto se le conoce como codificación bidireccional.

Por tanto, las imágenes pueden clasificarse principalmente en tres tipos basados en sus esquemas de compresión:

- I (Intraframes) o intra cuadros: Estas imágenes pueden ser decodificadas por si solas. Necesitan de mucha más información para ser codificadas y por tanto serán usadas en su justa medida. Ya que consisten mayoritariamente de la codificación de coeficientes DCM, no necesita de imagen previa para ser reconstruida y por tanto sirve de punto de arranque a los decodificadores.

- P(Predictive) o cuadros predichos: estas imágenes son predichas a partir de la imagen anterior, que puede ser una imagen I o incluso otra P, por medio de un número de vectores de compensación de movimiento, añade cierta información de corrección sobre la imagen predcada, y que requieren de mucha menos información para ser transmitidos. Cuando el proceso de predicción falla (no se encuentra un movimiento del macrobloque correlacionado con la imagen siguiente) esta imagen se codifica como I.
- B(Bi-directional) o cuadros bidireccionales. Estas imágenes son predichas a partir de imágenes anteriores o futuras por medio de vectores de compensación de movimiento a los que se añade cierta información de corrección sobre la imagen predcada.

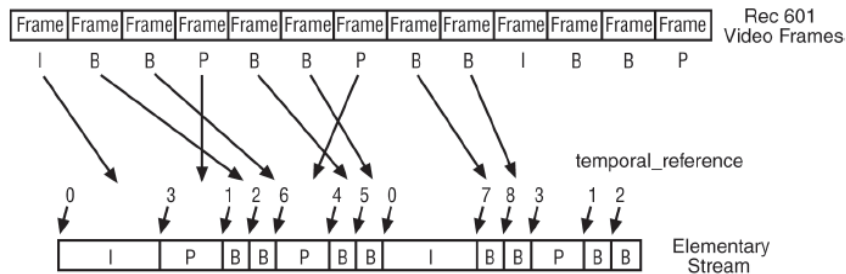


Figura A.9. Grupo de imágenes (GOP)

La Figura A.9 introduce el concepto de GOP (Grupo de Imágenes-Group Of Pictures). Un GOP estructura una secuencia de imágenes I, P y B sin una longitud predeterminada (aunque en la práctica suele constar de entre 12 y 15 imágenes). Debido a la compresión temporal que se acaba de describir, las imágenes en la secuencia a transmitir no tienen por qué aparecer en el mismo orden en el que éstas serán reproducidas. Toda imagen P debe ser precedida por la imagen I a partir de la que es predcada. Sin embargo, las imágenes B no pueden ser decodificadas hasta que no se hayan recibido todas las imágenes referencia (en la que pueden existir referencias a imágenes futuras).

A.3.2 Compresión de audio MPEG

Antes de pasar a describir las técnicas de compresión de audio de MPEG se introducen las características de funcionamiento del oído humano, ya que permitirán fijar los conceptos en los que se basan las estrategias de compresión.

El oído humano percibe un rango de frecuencias entre 20 Hz. y 20 KHz. En primer lugar, la sensibilidad es mayor en la zona alrededor de los 2-4 KHz, de forma que el sonido resulta más difícilmente audible cuanto más cercano a los extremos de la escala, ver Figura A.10.

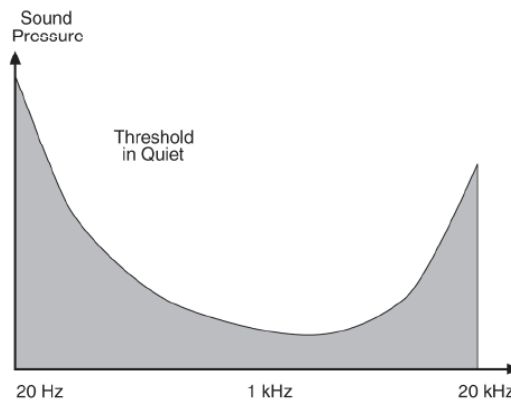


Figura A.10. Límite mínimo para percepción sonora

En segundo lugar está el enmascaramiento producido cuando la componente a cierta frecuencia de una señal tiene una energía elevada, el oído no puede percibir componentes de menor energía en frecuencias cercanas, tanto

inferiores como superiores. A una cierta distancia de la frecuencia enmascaradora, el efecto se reduce tanto que resulta despreciable; el rango de frecuencias en las que se produce el fenómeno se denomina banda crítica (critical band). Las componentes que pertenecen a la misma banda crítica se influyen mutuamente y no afectan ni se ven afectadas por las que aparecen fuera de ella. La amplitud de la banda crítica es diferente según la frecuencia en la que se situó y viene dada por unos determinados datos que demuestran que es mayor con la frecuencia, ver Figura A.11.

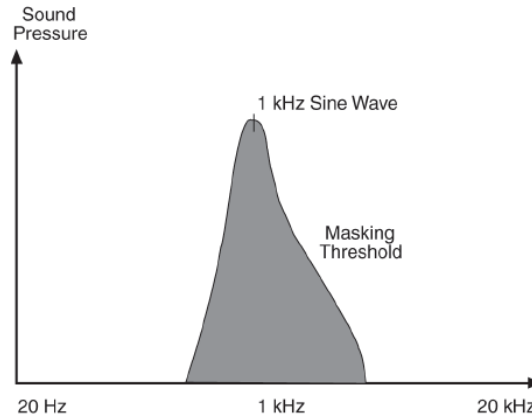


Figura A.11. Enmascaramiento de frecuencias

La codificación sub-banda o SBC (sub-band coding) es un método potente y flexible para codificar señales de audio eficientemente. A diferencia de los métodos específicos para ciertas fuentes, el SBC puede codificar cualquier señal de audio sin importar su origen, ya sea voz, música o sonido de tipos variados. El estándar MPEG Audio es el ejemplo más popular de SBC. El principio básico del SBC es la limitación del ancho de banda por descarte de información en frecuencias enmascaradas. El resultado simplemente no es el mismo que el original, pero si el proceso se realiza correctamente, el oído humano no percibe la diferencia. La mayoría de los codificadores SBC utilizan el mismo esquema. Primero, un filtro o un banco de ellos, o algún otro mecanismo descompone la señal de entrada en varias subbandas. A continuación se aplica un modelo psicoacústico que analiza tanto las bandas como la señal y determina los niveles de enmascaramiento utilizando los datos psicoacústicos de que dispone. Considerando estos niveles de enmascaramiento se cuantifican y codifican las muestras de cada banda y si en una frecuencia dentro de la banda hay una componente por debajo de dicho nivel, se elimina. Si lo supera, se calculan los bits necesarios para cuantificarla y codificarla posteriormente.

Concretamente la codificación MPEG se caracteriza por:

1. El banco de filtros: realiza el mapeado del dominio del tiempo al de la frecuencia. Existen dos tipos: el polifase y el híbrido polifase/MDCT. Estos bancos proporcionan tanto la separación en frecuencia para el codificador como los filtros de reconstrucción del decodificador. Las muestras de salida del banco están cuantificadas.
2. El modelo psicoacústico: calcula el nivel a partir del cual el ruido comienza a ser perceptible, para cada banda. Este nivel se utiliza en el bloque de asignación de bit/ruido para determinar la cuantificación y sus niveles. De nuevo, se utilizan dos diferentes. En ambos, los datos de salida forman el SMR (signal-to-mask ratio) para cada banda o grupo de bandas.
3. Asignación de bit/ruido: examina tanto las muestras de salida del banco de filtros como el SMR proporcionado por el modelo psicoacústico, y ajusta la asignación de bit o ruido, según el esquema utilizado, para satisfacer simultáneamente los requisitos de tasa de bits y de enmascaramiento.
4. El formateador de bitstream: toma las muestras cuantificadas del banco de filtros, junto a los datos de asignación de bit/ruido y otra información lateral para formar la trama final (elementary stream).

La complejidad en el proceso de codificación anterior del audio MPEG ha originado la definición de 3 esquemas de codificación (layers) que se muestra a continuación en función del costo de dicho proceso.

A.3.2.1 Audio MPEG layer 1

- El mapeado tiempo-frecuencia se realiza con un banco de filtros polifase con 32 subbandas. Los filtros polifase consisten en un conjunto de filtros con el mismo ancho de banda con interrelaciones de fase especiales que ofrecen una implementación eficiente del filtro subbanda. Se denomina filtro subbanda al que cubre todo el rango de frecuencias deseado. En general, los filtros polifase combinan una baja complejidad de computación con un diseño flexible y múltiples opciones de implementación.
- El modelo psicoacústico utiliza una FFT (Fast Fourier Transform) de 512 puntos para obtener información espectral detallada de la señal. El resultado de la aplicación de la FFT se utiliza para determinar los enmascaramientos en la señal, cada uno de los cuales produce un nivel de enmascaramiento, según la frecuencia, intensidad y tono. Para cada subbanda, los niveles individuales se combinan y forman uno global, que se compara con el máximo nivel de señal en la banda, produciendo el SMR que se introduce en el cuantificador.
- El bloque de cuantificación y codificación examina las muestras de cada subbanda, encuentra el máximo valor absoluto y lo cuantifica con 6 bits. Este valor es el factor de escala de la subbanda. A continuación se determina la asignación de bits para cada subbanda minimizando el NMR (noise-to-mask ratio) total. Es posible que algunas subbandas con un gran enmascaramiento terminen con cero bits, es decir, no se codificará ninguna muestra. Por último las muestras de subbanda se cuantifican linealmente según el número de bits asignados a dicha subbanda concreta.
- El trabajo del empaquetador de trama es sencillo. La trama, según la definición ISO, es la menor parte del bitstream decodificable por sí misma. Cada trama empieza con una cabecera para sincronización y diferenciación, así como 16 bits opcionales de CRC para detección y corrección de errores. Se emplean, para cada subbanda, 4 bits para describir la asignación de bits y otros 6 para el factor de escala. El resto de bits en la trama se utilizan para la información de muestras, 384 en total, y con la opción de añadir cierta información adicional. A 48 KHz, cada trama lleva 8 ms de sonido.

A.3.2.2 Audio MPEG layer 2

- El mapeado de tiempo-frecuencia es idéntico al del esquema I
- El modelo psicoacústico es similar, salvo que utiliza una FFT de 1024 puntos para obtener mayor resolución espectral. En los demás aspectos, es idéntico.
- El bloque de cuantificación y codificación también es similar, generando factores de escala de 6 bits para cada subbanda. Sin embargo, las tramas del esquema II son tres veces más largas que las del esquema I, de forma que se concede a cada subbanda tres factores de escala, y el codificador utiliza uno, dos o los tres, según la diferencia que haya entre ellos. La asignación de bits es similar a la del esquema I.
- El formateador de trama: la definición ISO de trama es la misma que en el punto anterior. Utiliza la misma cabecera y estructura de CRC que el esquema I. El número de bits que utilizan para describir la asignación de bits varía con las subbandas: 4 bits para las inferiores, 3 para las medias y dos para las superiores, adecuándose a las bandas críticas. Los factores de escala se codifican junto a un número de dos bits que indica si se utilizan uno, dos o los tres. Las muestras de subbanda se cuantifican y a continuación se asocian en grupos de tres, llamados gránulos. Cada uno se codifica con una palabra clave, lo que permite interceptar mucha más información redundante que en el esquema I. Cada trama contiene, pues, 1152 muestras PCM. A 48 KHz. cada trama lleva 24 ms de sonido.

A.3.2.3 Audio MPEG layer 3 (MP3)

- El mapeado de tiempo-frecuencia añade un nuevo banco de filtros, el DCT (Discrete Cosine Transform), que con el polifase forman el denominado filtro híbrido. Proporciona una resolución en frecuencia variable, 6x32 o 18x32 subbandas, ajustándose mucho mejor a las bandas críticas de las diferentes frecuencias.
- El modelo psicoacústico es una modificación del empleado en el esquema II mediante la utilización un método denominado predicción polinómica. También incluye los efectos del enmascaramiento temporal.
- El bloque de cuantificación y codificación emplea algoritmos muy sofisticados que permiten tramas de longitud variable. La gran diferencia con los otros dos esquemas es que es el ruido el factor controlado a través de bucles iterativos que sistemáticamente lo reducen al mínimo posible en cada paso.

- El formateador de trama: la definición de trama para este esquema varía con respecto de la definición de los niveles anteriores ya que en ésta es la "mínima parte del bitstream decodificable mediante el uso de información principal adquirida previamente". Las tramas contienen información de 1152 muestras y empiezan con la misma cabecera de sincronización y diferenciación, pero la información perteneciente a una misma trama no se encuentra generalmente entre dos cabeceras. La longitud de la trama puede variarse en caso de necesidad. Además de tratar con esta información, el esquema III incluye codificación Huffman de longitud variable, un método de codificación entrópica que, sin pérdida de información, elimina redundancia. Los métodos de longitud variable se caracterizan, en general, por asignar palabras cortas a los eventos más frecuentes, dejando las largas para los más infrecuentes.

A.4. MPEG-2

Por MPEG-2 se designa un grupo normas de codificación y compresión para audio y video. MPEG es el acrónimo de Moving Picture Experts Group, el cual fue formado en 1988 para establecer una norma internacional para representar en forma codificada imágenes en movimiento en asociación al audio correspondiente, sobre un método de almacenamiento digital. MPEG-2 se publica a través de la Recomendación H.262 de la UIT y la norma ISO/IEC 13818 y fue desarrollada en respuesta a la creciente demanda de un método de codificación genérico de imágenes en movimiento y audio asociado para diversas aplicaciones como almacenamiento digital, teledifusión y comunicaciones en general. Esta especificación permite que el video en movimiento pueda ser manipulado como datos de computadora, almacenado en diversos medios, transmitido y recibido en redes y distribuido en canales de broadcasting (Trigal, 2007). El estándar de TD DVB adopto como su sistema de codificación para la transmisión a MPEG2, el cual es válido también para DVB-C que es el estándar de TD tenido en cuenta dentro de este trabajo de grado.

El sistema DVB actualmente utiliza las especificaciones MPEG-2 de (ISO/IEC-13818, 1995) para la compresión y codificación de los flujos de audio y video; como se describe en (ISO/IEC-13818-3, 1994) y en (ISO/IEC-13818-2, 1995) respectivamente, y en la recomendación (UIT-T-H.262, 2007), en (ISO/IEC, IS 13818-1, 2000) se normaliza el múltiplex que agrega los audios y videos de varios programas para generar un régimen binario constante, el MPEG-2/ Transport Stream (MPEG-2/TS). DVB también ha establecido este procedimiento a través del estándar MPEG-4 para codificación de audio(ISO/IEC-14496, 2002) y codificación de video(ISO/IEC-14496-3, 1998)(ITU-T-H.264, 2003) (ETSI_TS_101_154, 2007). Estos sistemas de compresión digital permiten que se pueda transmitir múltiples programas de TV de definición estándar (SDTV¹, Standard Definition Television) o de alta definición (HDTV, High Definition Television) en el ancho de banda ocupado por un canal de televisión analógico. En la Tabla 1 se muestra cada una de las 9 normas que componen el MPEG-2 (Pérez, 2009).

Hay dos tipos de multiplexación especificados por los sistemas MPEG-2, uno que corresponde al Tren de Programa (program stream, PS) que está formado por la multiplexación de un solo programa y es utilizado, por ejemplo, por el DVD y otro, el tren de transporte (transport stream, TS) define como se multiplexan varios programas y es el que utiliza DVB, entre otros. Estas dos multiplexaciones facilitan la inclusión de la PSI (Program Specific Information), que brinda información de los datos que se multiplexan.

Además, los sistemas MPEG-2 aportan unas referencias temporales para que los datos se representen en el momento adecuado puesto que, por ejemplo, el sonido y las imágenes no viajan en paralelo, pero el usuario final las tiene que percibir en el mismo momento. Además, los sistemas MPEG-2 dan flexibilidad para la inclusión de nuevas sintaxis, añadir información de control de acceso condicional, datos etc. La normativa MPEG-2 no especifica como se realiza esta multiplexación ni como protegerla. A título orientativo, sólo menciona que los dos tipos de multiplexación que se están manejando actualmente son o bien TDM (Multiplexación por División de Tiempo), o bien estadística.

La multiplexación estadística, a diferencia de la TDM, no asigna un espacio de tiempo determinado y concreto, sino que los diferentes programas van pasando información de cuánto ancho de banda requieren para la transmisión. Así pues, un programa que necesite mucho podrá beneficiarse de uno que tenga espacio libre y que, de otra

¹ SDTV se usa por denominar las señales analógicas de 480 líneas (Americano) o 575 (Europeo). Su relación de aspecto siempre es de 4:3, mientras que la exploración es entrelazada

manera, se desaprovecharía utilizando bits de relleno (stuffing bits). En realidad ambas multiplexaciones (TDM y estadística) son multiplexaciones en el dominio del tiempo (en contraposición a las multiplexaciones en frecuencia). La diferencia entre ambas radica en que la multiplexación estadística reserva las ranuras de tiempo de manera dinámica. En DVB el proceso de obtención de los paquetes MPEG-2 se lo conoce como códificación de fuente mediante el cual se añade redundancia y protección a la señal para hacerla susceptible de utilización de técnicas FEC (Forward Error Correction). La descripción de la formación de los paquetes MPEG-2 que se muestra a continuación corresponden básicamente a la información contenida en la norma (ISO/IEC, IS 13818-1, 2000).

ISO/IEC 13818-1:2000	Part 1. Elementary Streams transport definition
ISO/IEC 13818-2:2000	Part 2. Video Compression
ISO/IEC 13818-3:1998	Part 3. Audio Compression (Compatible con MPEG-1)
ISO/IEC 13818-4:1998	Part 4. Audio. Conformance testing
ISO/IEC 13818-4:1998/Cor 2:1998	Part 4. Audio.
ISO/IEC 13818-4:1998/Amd 1:1999	Part 4. Audio. Advanced Audio Coding (AAC) conformance testing
ISO/IEC 13818-4:1998/Amd 2:2000	Part 4. Audio. System target decoder model
ISO/IEC 13818-4:1998/Amd 3:2000	Part 4. Audio. Additional audio conformance bitstreams
ISO/IEC TR 13818-5:1997	Part 5. Audio. Software simulation
ISO/IEC TR 13818-5:1997/Amd 1:1999	Part 5. Audio. Advanced Audio Coding (AAC)
ISO/IEC 13818-6:1998	Part 6. Extensions for DSM-CC. Digital Storage Media Command and Control
ISO/IEC 13818-6:1998/Cor 1:1999	Part 6. Extensions for DSM-CC
ISO/IEC 13818-6:1998/Amd 1:2000	Part 6. Extensions for DSM-CC Additions to support data broadcasting
ISO/IEC 13818-6:1998/Amd 2:2000	Part 6. Extensions for DSM-CC Additions to support synchronized download services, opportunistic data services and resource announcement in broadcast and interactive services
ISO/IEC 13818-7:1997	Part 7. Advanced Audio Coding (AAC)
ISO/IEC 13818-7:1997/Cor 1:1998	Part 7. Advanced Audio Coding (AAC)
Part 8	Part 8. Se ceso su definición. Codificación de Video con input en 10bits
ISO/IEC 13818-9:1996	Part 9. Extension for real time interface for systems decoders
ISO/IEC 13818-10:1999	Part 10. Conformance extensions for Digital Storage Media Command and Control (DSM-CC)

Tabla 1. Normas MPEG-2

A.4.1 Paquetes MPEG-2

Como ya se ha comentado anteriormente, MPEG-2 dispone de varios sistemas diferentes para la transmisión u almacenamiento de datos, pero todos ellos coinciden en la utilización de paquetes para la realización de esa tarea. Un paquete es una colección de bits de un mismo tipo (video, audio, datos...). Los paquetes pueden ser de tamaño fijo o variable, en función de la aplicación para la que estén destinados.

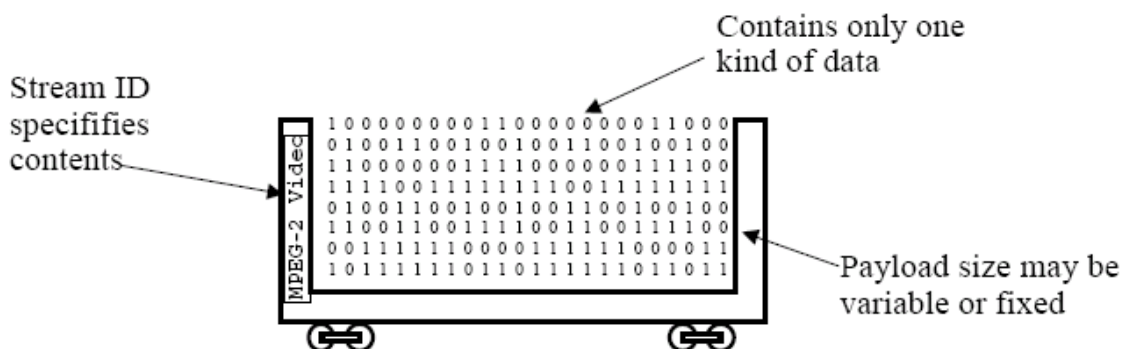


Figura A.12. Paquetes MPEG-2

De esa forma, los paquetes de tamaño variable serán utilizados para aplicaciones en las que el medio este libre de errores (DVD, CD-ROM, Disco Duro...). Además, la longitud de dichos paquetes podrá llegar a ser muy grande, minimizando el espacio reservado para cabeceras y demás mecanismos de señalización los cuales reducen la eficiencia del sistema.

Los paquetes de longitud fija, normalmente serán de pequeño tamaño y se utilizarán para aplicaciones en las que el medio sea susceptible de introducir errores (por ejemplo en broadcast). Nótese que al tener paquetes de longitud fija y pequeña se simplifica muchísimo la corrección de errores en los mismos, además de reducir el efecto de dichos errores sobre nuestro sistema.

Los paquetes son la llave a la flexibilidad y a la extensibilidad de los sistemas MPEG-2. Un sistema basado en paquetes es flexible, ya que permite una sencillísima multiplexación basada en el tiempo y al mismo tiempo la optimización de nuestros recursos a través de una asignación dinámica de los mismos basada en la prioridad de los diferentes paquetes.

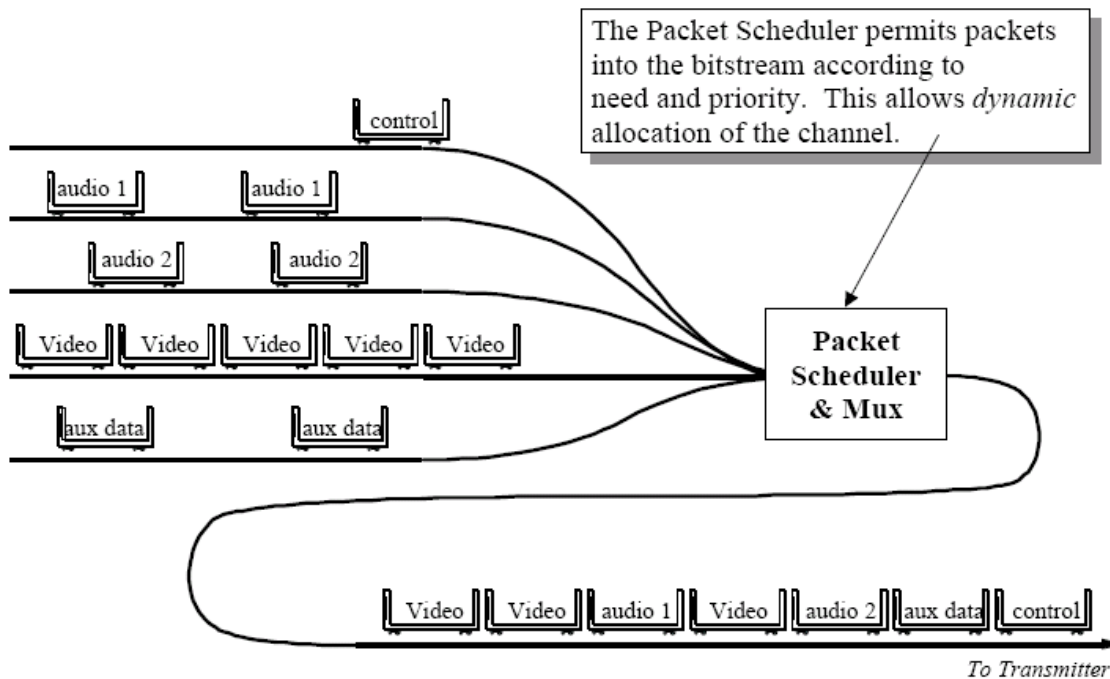


Figura A.13. Multiplexación de paquetes MPEG-2

Un sistema basado en paquetes es fácilmente ampliable, por lo que se garantiza la extensibilidad del mismo. Dicha extensibilidad se consigue haciendo que el receptor descarte aquellos paquetes que no es capaz de decodificar. De esa forma, la ampliación de las prestaciones de nuestro sistema no afecta a los receptores antiguos, ya que simplemente descartan lo que desconocen y siguen siendo capaces de decodificar el resto.

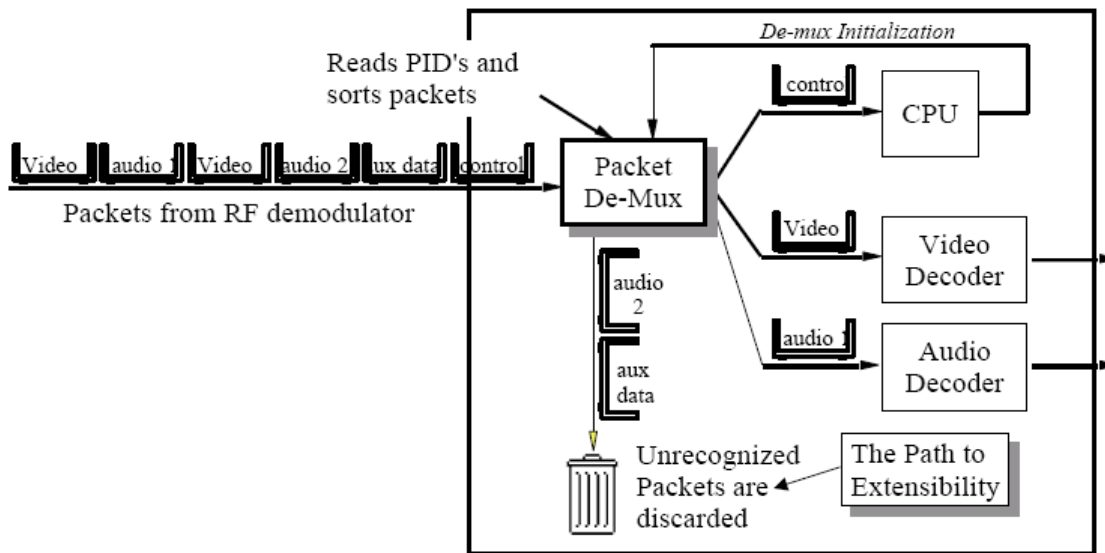


Figura A.14. Ampliación del sistema de multiplexación de paquetes

A.4.1.1 MPEG-2 Elementary Streams (ES)

Los contenidos que transporta un sistema MPEG-2 pueden ser muy variados. Como ejemplo se puede pensar en un DVD, en el cual además de las tramas de vídeo y audio se puede encontrar subtítulos e incluso juegos interactivos. Además, es posible que disponga de más de una opción para cada uno de esos contenidos (varias tramas de audio diferentes, subtítulos en distintos idiomas, etc). Cada uno de ellos conforman Flujos Elementales (ES – Elementary Stream) de datos independientes entre si, aunque pueden estar relacionadas de alguna manera.

Los ES son los datos tal y como salen del codificador MPEG-2. Los datos que entran al codificador, como por ejemplo las imágenes, se denominan unidades de presentación y los datos comprimidos son las unidades de acceso. En la Figura A.15 se puede ver la diferencia entre unidades de acceso y unidades de representación en la compresión de una secuencia de vídeo.

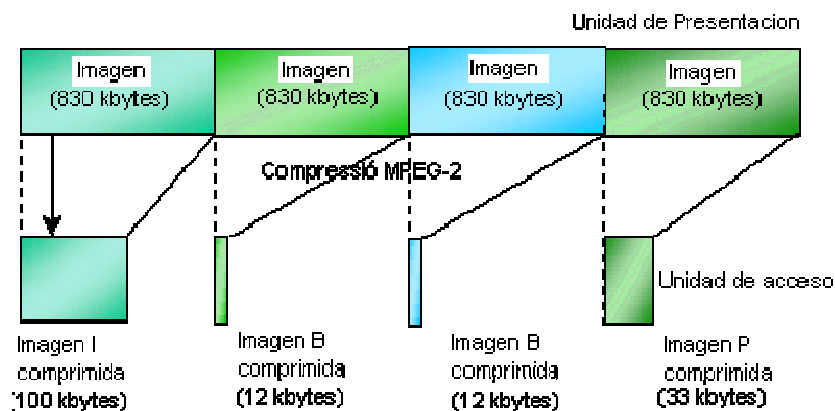


Figura A.15. MPEG-2 Elementary Streams

El resultado de la codificación de una secuencia de vídeo o de audio es una sucesión de unidades de acceso llamadas Flujos Elementales de Vídeo (Video Elementary Stream) o Flujos Elementales de Audio (Audio Elementary Stream) respectivamente ver Figura A.16, u otro tipo de flujo elemental como por ejemplo de datos privados, de teletexto, de subtítulos, de aplicaciones interactivas,..

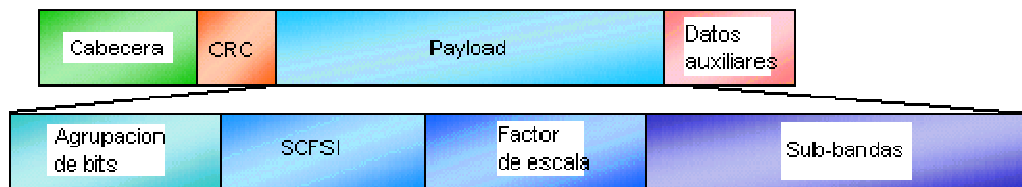


Figura A.16. Audio Elementary Stream

A.4.1.2 MPEG-2 Packetised Elementary Streams (PES)

El siguiente punto en el proceso de multiplexación es dividir cada ES en paquetes conocidos como Flujo Elemental Empaquetado (PES – Packetised Elementary Stream). Un paquete PES está formado por cabecera y payload. El payload está formado por los datos tomados secuencialmente del ES. No hay necesidad de alinear las unidades de acceso y el comienzo de los PES payloads. Es más, una nueva unidad de acceso puede empezar en cualquier punto del payload. A continuación se presenta un ejemplo de como se inserta la información del Elementary Stream dentro de los paquetes PES ver.

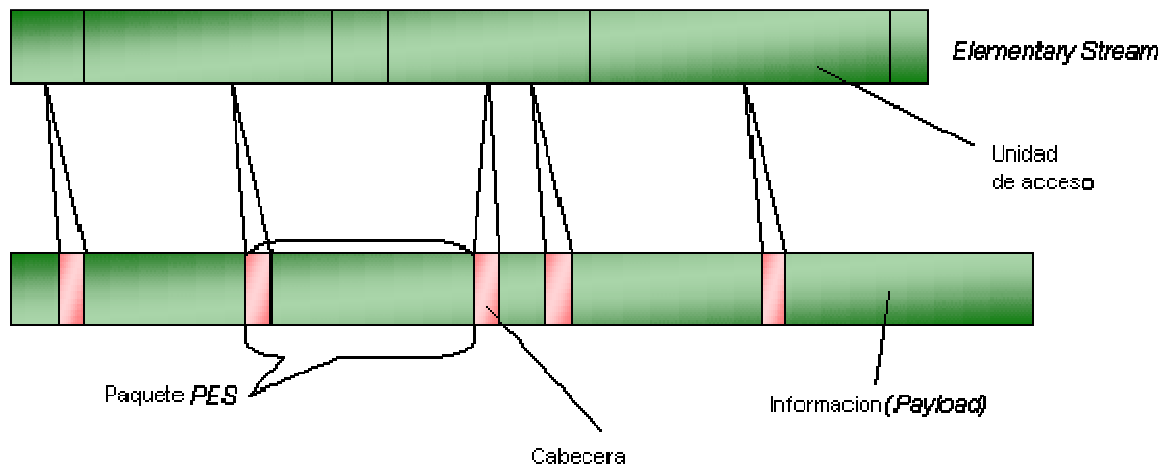


Figura A.17. Paso de ES a PES

El encabezado del paquete PES incluye una estampa de tiempo (time stamp), un ID de flujo que identifica el tipo del contenido y la forma en que está codificado, y alguna información de sincronización para el audio, video y datos que es usada en el receptor. Los paquetes PES pueden ser de longitud variable hasta un máximo de 64 Kbytes. La estructura de un paquete PES es el que se muestra en la Figura A.18.

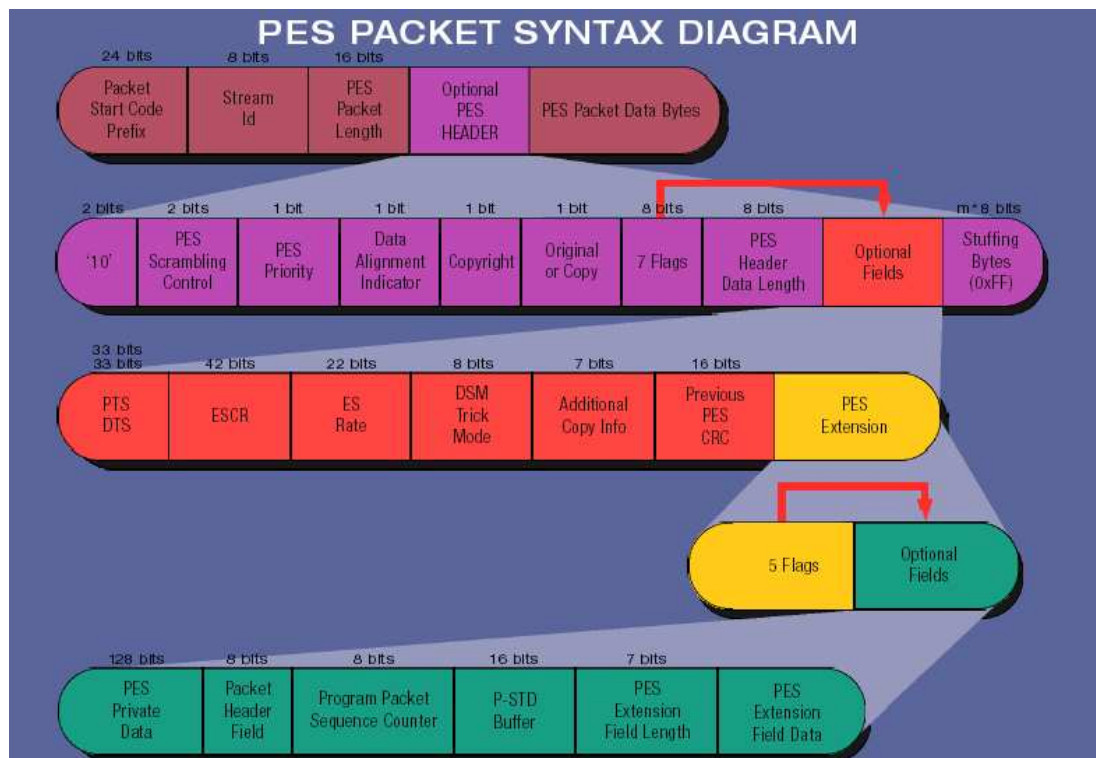


Figura A.18. Estructura de un paquete PES

Los primeros 4 bytes de la cabecera forman el código de inicio del paquete PES. Evidentemente, durante todo el paquete PES esta secuencia de bits no se puede repetir, puesto que si no, se podría interpretar como el inicio de un nuevo paquete.

El campo `stream_id` permite distinguir paquetes PES de un mismo programa. El estándar MPEG-2 especifica los valores permitidos para este campo, incluyendo 32 para streams elementales de audio y 16 para streams elementales de vídeo.

Los flags 1 y 2 son bits que indican la presencia o ausencia de campos opcionales en la cabecera. Estos paquetes opcionales dan idea de, por ejemplo, si la información contenida está encriptada, tiene alguna prioridad, trae información de copyright y un campo opcional de control de errores para el paquete PES.

De los mencionados flags, hay dos que son de gran importancia. Son los que están marcados como PTS y DTS en la Figura A.18. Cuando estos bits están activos indican la presencia del Presentation Time Stamp (PTS) y el Decoding Time Stamp (DTS). Estos serán los elementos que permitirán la correcta sincronización de los diferentes ES de vídeo y audio en recepción.

A continuación está el paquete de longitud de cabecera, que indica cuántos bytes opcionales hay en la cabecera antes de que empiece el primer byte de payload. Para más detalles relativos a la estructura del paquete PES lo más acertado es recurrir a la propia especificación MPEG-2 (ISO/IEC 13818-1 1996).

A.4.1.3 Flujo de Programa (PS, Program Stream)

Los flujos de programa son el resultado de combinar uno o más flujos de paquetes PES, que tengan una base de tiempo común, en un flujo único. Un PS está diseñado para la comunicación o almacenamiento de un programa de datos codificados (audio y vídeo) y otros datos (aplicaciones software) en ambientes relativamente libre de errores y servir para aplicaciones que puedan implicar un procesamiento software del sistema de información como las aplicaciones interactivas multimedia donde esta consideración es muy importante. Los paquetes de los flujos de programa pueden ser de longitud variable y relativamente grande. Un PS es la estructura que define MPEG-2 para el almacenamiento de datos en medios libres de errores, como pueden ser los DVDs, discos duros, CD-ROMs, etc

A.4.2 MPEG-2 Transport Stream (TS)

Los paquetes PES constituidos por flujos elementales – ES, que comparten la misma base de tiempo forman un programa, con lo cual se crea un canal virtual. Los Flujos de Transporte (Transport Stream – TS) combinan uno o más programas con una o más bases de tiempo independientes en un flujo único. El flujo de transporte (TS - Transport Stream) está diseñado para el uso en ambientes donde los errores (por ejemplo, pérdida de paquetes) son probables, tales como el almacenamiento o la transmisión en medios ruidos y con pérdidas.

La aplicación más importante del formato TS es la transmisión de televisión digital en los estándares DVB y ATSC². MPEG-2 define dos maneras de construir la trama de datos. Por un lado se tiene el Transport Stream y por otro el Program Stream. El TS es la estructura que se define para la transmisión en medios susceptibles de errores (por ejemplo, pérdida de paquetes), tales como el almacenamiento o la transmisión en medios ruidos y con pérdidas. Por lo tanto todas las aplicaciones destinadas a broadcast se realizarán en este formato. La información de señalización en un Transport Stream se realiza mediante la (Información Específica de programa (PSI – Program Specific Information) que se transmite mediante tablas, las cuales son encapsuladas en secciones MPEG-2.

El TS soluciona los problemas críticos presentes en los trenes de programa con respecto a la pérdida de un paquete, puesto que limita la longitud de estos a 188 bytes. Por ello, su uso está más orientado a la difusión por broadcast. Un TS dispone de mecanismos de detección de errores, pese a que hace falta añadir todavía métodos de corrección de dichos errores. Además, el Transport Stream permite multiplexar varios programas dentro de un mismo flujo binario de datos.

Se denomina programa, dentro de un TS, a una agrupación lógica de Elementary Streams. La manera más sencilla de comprender este concepto es pensando en un canal de televisión, en el cual se tiene tramas de video, audio, datos, etc, todas esas tramas conforman el programa. En cualquier caso, es conveniente advertir de que no todas las tramas del programa se presentan al mismo tiempo en pantalla (o por los altavoces del televisor). Un programa de televisión podría tener varias tramas de video asociadas, pero en un momento dado se puede estar presentando sólo una de ellas por pantalla.

Al igual que ocurre en cualquier protocolo que divida su complejidad en capas, los paquetes de transporte (el conjunto de paquetes de transporte forman el TS) están formados por una cabecera seguida de una carga o información útil. Los paquetes PES son divididos entre los payloads de los paquetes transporte ver Figura A.19 . El proceso de división de PES tiene que seguir dos premisas. Una es que el primer byte de cada paquete PES tiene que ser el primero del payload del paquete de transporte. La segunda premisa es que cada paquete de transporte sólo puede traer información de un PES.

Es bastante probable que no haya un número entero de paquetes de transporte para traer todo el PES, con lo cual, nos quedan paquetes sin acabar de llenar. Para que se cumplan las premisas anteriores, existe un campo de adaptación que permite acabar de llenar el paquete. Son los denominados bits de relleno. Su uso se puede minimizar, utilizando longitudes de paquetes PES grandes, puesto que así se asegura que una gran parte de los paquetes de transporte estén completamente llenos.

² ATSC: Advanced Television System Committee es un grupo que se encarga del desarrollo de los estándares de la televisión digital en los Estados Unidos, y en base a estos estándares Canadá, México, Corea del Sur, Guatemala, Honduras y Costa Rica adoptaron esta normativa.

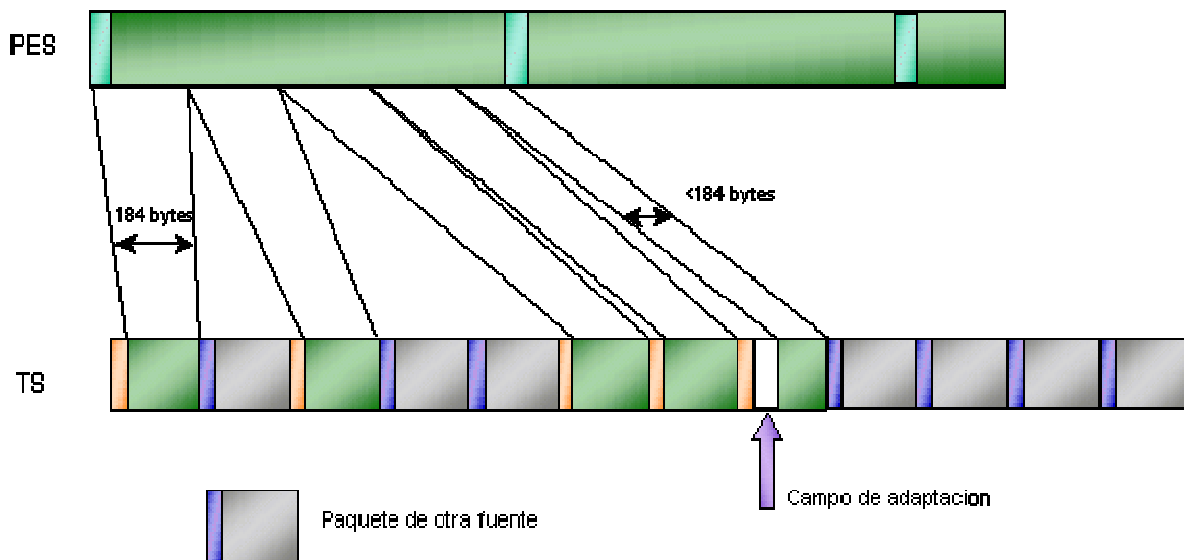


Figura A.19. Generación de paquetes TS a partir de los paquetes PES

El conjunto de paquetes que resulten de este proceso forman el Transport Stream, en una solo TS se puede tener varios programas diferentes (y dentro de cada uno de ellos varios ES diferentes). No se especifica el orden en qué los paquetes de transporte llegan al multiplexor. Lo único que sí que se especifica es que los paquetes de un mismo PES han de enviarse secuencialmente.

No todos lo paquetes de transporte contienen paquetes PES. Por poner un par de ejemplos, la Información Específica de programa (PSI) o las aplicaciones interactivas son transmitidas en secciones MPEG-2 que posteriormente se encapsulan en paquetes de transporte. En la Figura A.20 se tiene un ejemplo gráfico.

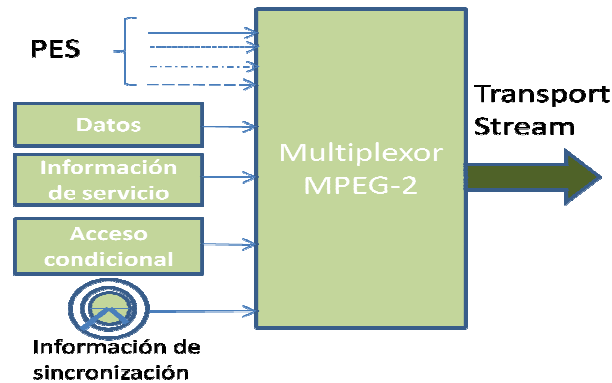


Figura A.20. Ejemplo grafico de MPEG-2

La multiplexación del transport stream consiste, en pequeños paquetes de longitud constante. Un paquete de transporte es siempre de 188 bytes, de los cuales 4 se destinan a una cabecera de inclusión obligatoria tras la que se puede encontrar un campo de adaptación opcional. El resto de bytes, hasta completar los 188, son de información (payload). El significado de uno de los campos más importantes y a tener en cuenta para la generación de contenidos de EDiTV es el campo Identificador de Paquete (PID).

Los paquetes de TS pueden traer información de programas diferentes, además de datos para la reconstrucción de la información. Aparece un campo de 13 bits que se denomina PID que permite la distinción de paquetes de diferentes elementary streams. El multiplexor tiene que garantizar que cada ES tenga un único PID. La normativa MPEG no especifica que valores de PID se tienen que dar a los ES.

A.4.2.1 Program Specific Information (PSI)

Una trama de transporte MPEG-2 está formada por varios programas diferentes. Un programa MPEG-2 está compuesto por diferentes Elementary Streams. El decodificador MPEG-2, además de ser capaz de decodificar cada uno de los Elementary Streams que conforman un programa ha de ser capaz de encontrarlos dentro de una trama de transporte. El Program Specific Information (PSI) es lo que permite que el decodificador realice esa tarea.

Toda la señalización necesaria para la correcta recepción de la trama de transporte se da mediante tablas de información de servicio (Service Information Tables en inglés). Estas tablas se introducen en la propia trama de transporte divididas en secciones MPEG-2 y embutidas en paquetes de transporte. Estas tablas de señalización conforman Elementary Streams, tal y como lo hacen las tramas de video y audio. Por lo tanto, cada una viajará en paquetes de transporte con un PID único para cada Transport Stream.

Sin embargo, el mecanismo mediante el que dichos Elementary Streams se introducen en los paquetes de transporte es diferente al usado para las tramas que se han visto hasta este momento. Una tabla de PSI no se introduce en un paquete PES antes de encapsularse en paquetes de transporte. Las tablas se introducen en secciones MPEG-2, las cuales pueden ser directamente introducidas en los paquetes de transporte. Al contrario que en los PES, las secciones no empiezan y acaban forzosamente con un paquete de transporte. Cuando una sección o un PES empieza en un paquete, el indicador payload_unit_star_indicator (PUSI) se pone a "1".

Cuando se trata de una sección, el paquete puede empezar al final de otra sección, precedida o no de un campo de adaptación (adaptation_field). El primer byte de la "carga útil" (payload) es un indicador llamado pointer_field el que da el desplazamiento (offset) del comienzo de la nueva sección con respecto a este byte.

Además de la tablas de PSI, MPEG-2 define una estructura de secciones privadas (private sections) mediante la cual el estándar es extensible (se puede definir otras tablas diferentes además de las ya existentes en MPEG-2). Además, estas tablas permiten la transmisión de datos privados. Cada tabla está constituida, según su importancia, por una o varias secciones (256 como máximo, con una longitud máxima de 1024 bytes, salvo para la tabla private que puede alcanzar los 4026 bytes).

A continuación se da una breve descripción del objetivo de cada una de las tablas obligatorias en MPEG-2.

- **PAT (Program Association Table):** La PAT nos da información sobre todos los programas presentes en un transport stream. A través de ella, se sabe en que PID viajan las tablas PMT que nos dan información sobre cada uno de los programas. La PAT siempre viaja en paquetes de transporte con PID = 0. Evidentemente la tabla PAT será única para cada trama de transporte.
- **PMT (Program Map Table):** Existe una PMT por cada programa presente en el transport stream. En ella se da información sobre todos los elementary streams asociados a un programa, de tal forma que el receptor es capaz de localizarlos y decodificarlos. Por lo tanto para cada elementary stream indica:
 - ✓ PID en el que viaja la trama fundamental.
 - ✓ Tipo de trama fundamental (vídeo, audio, datos...).
 - ✓ Descriptores asociados a la trama fundamental.

El PID en el que viaja cada una de las PMTs (una para cada programa presente en la trama de transporte) es asignado por la PAT.

- **CAT (Conditional Access Table):** Nos da información sobre el sistema de acceso condicional presente en el transport stream. Sólo es obligatoria en caso de que algún programa del transport stream esté codificado. La tabla PAT siempre viaja en paquetes de transporte de PID = 1.
- **NIT (Network Information Section):** Transporta información de red. Esta red puede estar formada por varios canales físicos diferentes, que a su vez transporten tramas de transporte independientes entre si. El

PID en el que viaja la NIT es asignado por la PAT. La NIT es una tabla opcional, pero en caso de estar presente, conforma el programa número 0 en la PAT.

Cada una de estas tablas fue construida para el desarrollo del presente trabajo su implementación se muestran en el anexo B.

A.4.3 Trama de Transporte DVB

Una trama de transporte DVB tiene la misma estructura básica de una trama de transporte MPEG-2, pero añade algunas restricciones y funcionalidades. Sin embargo lo expuesto anteriormente respecto a MPEG2 es aplicable sin pérdida de generalidad al estándar DVB. A continuación se introducen una serie de conceptos nuevos o a renombrar algunos que ya se conocían .

- **Elementary Stream:** Un elementary stream (ES) es un stream (en castellano podría traducirse como trama) de video, audio o datos binarios codificados en MPEG2. En una trama DVB cada elementary stream viaja en un PID diferente.
- **Event:** Un evento es lo que en lenguaje coloquial se conoce como un “programa de televisión”. En cada evento pueden existir varios elementary streams, que pueden ser o no de tipos diferentes. En cualquier caso el protocolo DVB permite la transmisión de eventos de varios tipos, por lo que además de los “programas de televisión” anteriormente mencionados se puede tener programas de radio, o simplemente tramas de datos.
- **Service:** Un servicio es lo que en lenguaje coloquial se conoce como un “canal de televisión”. Por lo tanto, un servicio ofrece varios eventos uno detrás de otro (en el tiempo). El número de elementary streams presentes en un servicio puede variar a lo largo del tiempo, ya que algunos eventos pueden tener más elementary streams que otros. Al igual que ocurre con los eventos los servicios no tienen porqué ser exclusivamente canales de televisión, pudiendo ser cadenas de radio, o tramas de datos. En definitiva un servicio es un conjunto de elementary streams agrupados lógicamente. En la Figura A.21 se expresa la idea anterior.

Es importante destacar que DVB llama servicio (service) a lo que MPEG2 llama programa (program), ya que en algunas fuentes bibliográficas estos dos términos se usan indistintamente.

- **Multiplex:** Un multiplex es un conjunto de servicios multiplexados en MPEG2. Cada multiplex viaja en una frecuencia diferente, siendo su velocidad máxima de transmisión de 40 Mbps (Megabits por segundo).

En la Figura A.22 se muestran los diferentes servicios y elementary streams de un multiplex, así como el PID en el que viajan y su velocidad de transmisión:

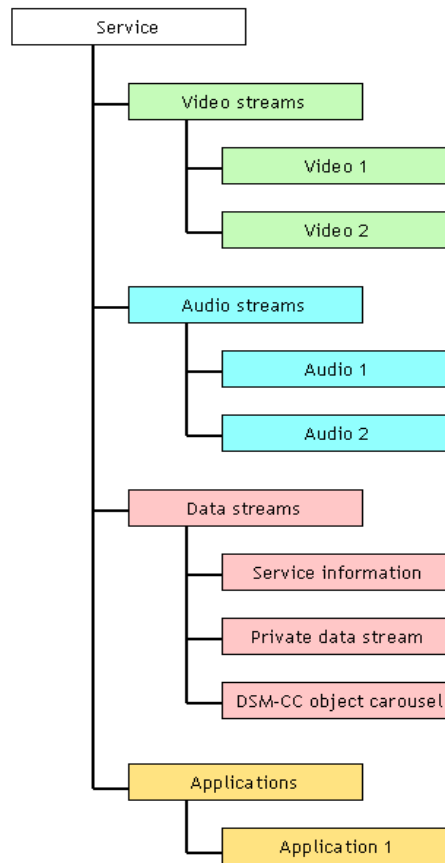


Figura A.21. Organización lógica de los servicios

Program	PID	Multiplex Occupancy (Mbps)	Current
Program 11: Astra - ASTRA-SDT 1			
11:Priv	55		0.06
Program 12000: RTL World - RTL CH			
12000:Priv	32		0.26
12000:UPriv	331		0.01
12000:Audio	80		0.20
12000:Video	160		4.87
Program 12001: RTL World - RTL A			
12001:Priv	32		0.26
12001:UPriv	361		0.01
12001:Audio	84		0.20
12001:Video	161		4.87
Program 12003: RTL World - RTL Television			
12003:Priv	32		0.26
12003:Audio	104		0.20
12003:Video	163		3.61
12003:Priv	1100		0.20
12003:11?	1102		0.97
12003:Priv	11031		0.02
Program 12010: RTL World - RTLradio-OLDIESENDER			
12010:Audio	1053		0.27
Program 12015: RTL World - RTL World			
12015:Priv	1100		0.20
12015:Priv	1101		3.30
Program 12020: RTL World - RTL2			
12020:Priv	68		0.30
12020:Audio	128		0.20
12020:Video	166		4.81
12020:Priv	1100		0.20
12020:11?	1102		0.97
12020:Priv	11031		0.02

Figura A.22. Diferentes servicios y elementary streams

Como se puede observar el ancho de banda (o la velocidad de transmisión) ocupado por los diferentes elementary streams es muy diferente. De esta forma, un elementary stream de video suele ocupar alrededor de los 3.5 Mbps, un ES de audio unos 0.2 Mbps, un ES de datos unos 1 Mbps... etc.

En la figura también se observa como algunos servicios (en la figura se les llama program siguiendo la terminología MPEG2) contienen varios ES, mientras que otros están compuestos por un solo elementary stream. Normalmente en cada multiplex suelen viajar alrededor de 6 o 7 canales de televisión más alguno de datos y de radio.

- **Bouquet:** Un bouquet es un grupo de servicios agrupados lógicamente (paquete de fútbol, paquete de cine...). De esta forma se puede tener en cada bouquet más (o menos) servicios de los que caben en un multiplex, y al mismo tiempo seguir teniéndolos ordenados sin bajar la eficiencia.

A.4.3.1 TABLAS DVB

DVB dispone de un mecanismo de información basado en tablas. El conjunto de tablas presentes en una trama de transporte constituyen el Service Information de la trama. Como se ha visto anteriormente, MPEG-2 define una serie de tablas obligatorias, además de la tabla NIT de carácter opcional.

DVB, además de las tablas MPEG-2 añade otro conjunto de tablas obligatorias más otro grupo de tablas opcionales. Nótese que la tabla NIT en el estándar DVB pasa a ser obligatoria y que se fija su PID a 0x10. En la Figura A.23 se describe lo anterior gráficamente.

En caso de que se quiera construir una trama DVB completa, al menos las tablas señaladas en rojo y las señaladas en azul (primera dos columnas de la tabla, de izquierda a derecha) deberían estar presentes en dicha trama de transporte.

Todas las tablas DVB obligatorias, excepto la tabla PMT, viajan en PIDs fijados por el estándar. El caso de las tablas PMT es diferente, ya que el PID en el que viajan se señala dentro de la tabla PAT. En la Figura A.23 anterior también se pueden observar dichos PIDs.

Las tablas MPEG-2 ya fueron definidas en el punto anterior, por lo que solamente se añade las descripciones de las tablas obligatorias DVB que además no estuvieran definidas por MPEG-2.

- **SDT (Service Description Table):** En esta tabla se tiene una lista de los servicios asociados a cada transport stream. En ella se muestran los nombres y los parámetros asociados a cada servicio. Esta información se transmite mediante descriptores, principalmente el "service descriptor". Además, la tabla SDT puede transportar información de temporización de los diferentes servicios.
- **EIT (Event Information Table):** Esta tabla nos da información sobre los eventos presentes o futuros de la trama de transporte. De esa forma el receptor conoce el momento de inicio del evento y su duración, entre otras cosas.
- **TDT (Time and Date Table):** Esta tabla se utiliza para transmitir la hora y fecha actual, de tal forma que el receptor pueda sincronizarse a ella.

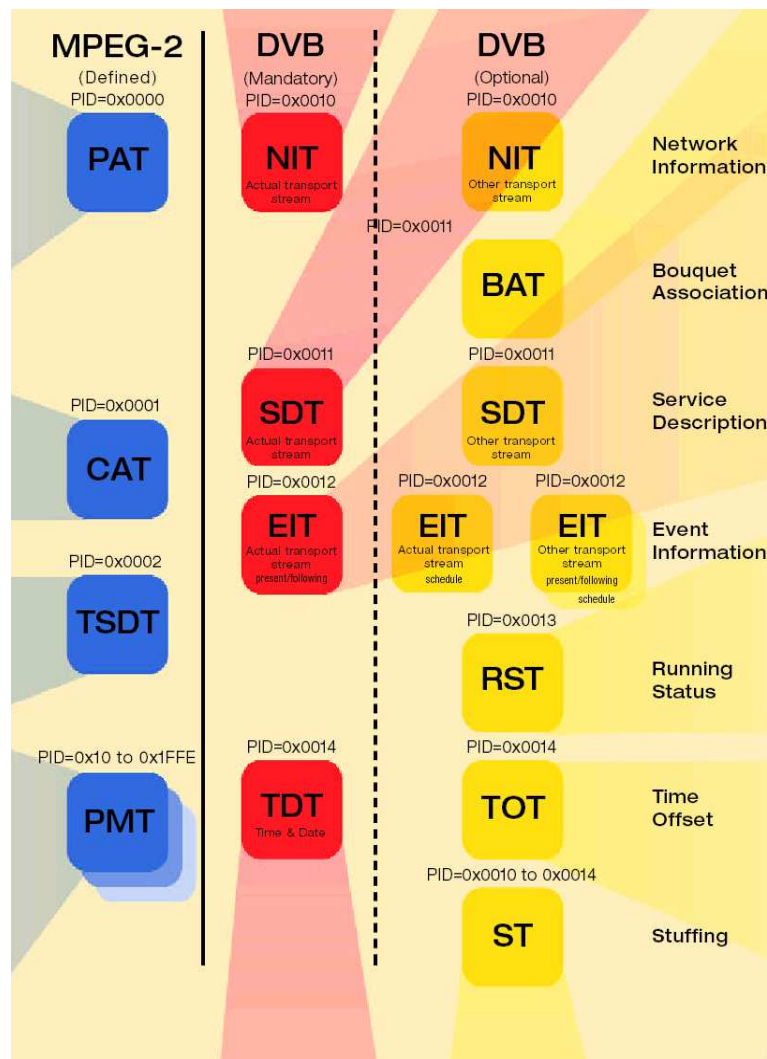


Figura A.23. Tablas DVB

A.4.4 Protocolo DSM-CC

El Digital Storage Media – Command and Control, es una especificación complejísima que forma parte del estándar MPEG-2 parte 6, (ISO/IEC-13818-6, 2001). DSM-CC está definido en todas las versiones de la especificación MHP y generalmente se utiliza para difundir las aplicaciones interactivas, por lo que todos los perfiles de MHP deben soportarlo (ETSI_TS_102_812, 2006). DSM-CC es una definición inmensa de protocolos de comunicación sobre redes heterogéneas. MPEG-2 se apoya en ella para facilitar el transporte de datos en redes de broadcasting, incluyendo aplicaciones.

La transmisión de datos y eventos por un canal de difusión se realiza en forma de carrusel, mecanismo que se describe en el protocolo DSM-CC. Los datos a ser transmitidos se dividen en partes llamadas módulos para que pueda transmitirse en forma de carrusel. Un carrusel es un mecanismo en el que difusor transmite periódicamente todos los datos, uno de tras del otro, hasta que todos ellos hayan sido difundidos momento en el cual el proceso se inicia desde el principio y el primer módulo se difunde de nuevo. Para acceder a un archivo el receptor debe esperar hasta que reciba los fragmentos de datos que está buscando para analizar el módulo y acceder al archivo propiamente dicho; DSM-CC soporta dos clases de carrusel (ISO/IEC-13818-6, 2001).

- **Carrusel de Datos**

Consta de un número de módulos que contiene un archivo. Este módulo puede ser dividido en bloques para hacer más fácil su transmisión. Todos los bloques de todos los módulos dentro de un carrusel de datos son del mismo

tamaño, excepto para el último bloque, el cual puede ser de tamaño más pequeño para permitir al receptor procesar más fácilmente los módulos.

- **Carrusel de Objetos DSM-CC**

Amplia el carrusel de datos DSM-CC añadiendo los conceptos de archivo, directorio y flujo, con unas funcionalidades parecidas a las de los sistemas de archivos convencionales.

Referencias

- Alvarez, Q. D. (2008). Estudio y Análisis de Sistemas de Televisión Digital Interactiva. Trabajo de Investigación, Universidad de Oviedo.
- Aarreniemi, P. (2006). *Modelling and content production of distance learning concept for interactive digital television*. Recuperado el 8 de septiembre de 2008, de portal lib.tkk.fi: <http://lib.tkk.fi/Diss/2006/isbn9512285428/isbn9512285428.pdf>
- Abi-Nassif, F., Lee, W. C., & Stavrakakis, I. (1999). Offered Load Estimation in a Multimedia Cable Network. *IEEE International Conference on Communications, ICC, Vol. 1*, 140-146.
- ADB. (2007). *ADB Ref. Q75DEV Soporte a MHP*. Recuperado el 25 de marzo de 2008, de portal adbglobal.com: <http://www.adbglobal.com/?q=node/193>
- Adobe. (2008). *Adobe Premier*. Recuperado el 12 de diciembre de 2008, de portal adobe.com: <http://www.adobe.com/es/products/premiere/>
- Al-Khatib, W., Rajeswari, A., & Gunavathi, K. (2007). Bandwith Allocation Algorithm for Docsis Based HFC Broadband Networks. *IEEE-ICSCN*, 452-458.
- Alzate, M. (2001). Introducción al Tráfico autosimilar en redes de comunicaciones. *Revista Ingeniería vol 6 No 2, Universidad Distrital; Colombia*, 6 a 17.
- Andueza, A. P. (2004). *Redes de acceso de Banda Ancha en la comunidad Foral de navarra*. Pamplona.
- Astaiza, E., Bermúdez, H. F., & Muñoz, P. A. (2007). *Simulación de sistemas de telecomunicaciones*. Armenia: Arte Imagen.
- Avalpa, E. D. (2008). *OpenCaster*. Recuperado el 5 de marzo de 2008, de portal avalpa.com: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- Balakrishnan, H., Padmanabhan, V. N., & Katz, R. H. (1999). The effects of asymmetry on TCP performance. *Mobile Networks Applicat., vol. 4*, 219-241.
- Balaluishnan, H., & Padmanabhan, V. N. (2001). How Network Asymmetry Affect TCP. *IEEE Communications Magazine, Vol. 4, 2001, pág. 2*, 2.
- Bansal, D., & Balakrishnan, H. (2001). Binomial Congestion Control Algorithms. *Proceedings INFOCOM*, Alaska.
- Barakat, N., & Darcie, T. E. (2007). Delay characterization of cable access networks. *IEEE Communications Letters, vol. 11, no. 4*, 357 - 359.
- Berrocal, J., Vázquez, E., González, F., Álvares, C., Vinyes, J., Mandinabeitia, G., y otros. (2003). *Redes de Acceso de Banda Ancha: Arquitectura, Prestaciones, Servicios y Evolución*. Madrid, España: Ministerio de Ciencia y Tecnología.
- Brice, R. (2003). *Newnes Guide to Digital TV*. Second edition, Newnes.
- Bushmitch, D., Mukherjee, S., Narayanan, S., Ratty, M., & Shi, Q. (2000). Supporting MPEG Video Transport on DOCSIS-Compliant Cable Networks. *IEEE Journal on Selected Areas in Communications, Vol. 18, N° 9*.
- CableLabs. (2006). *Radio Frequency Interface Specification DOCSIS 2.0*. CableLabs, Inc.
- CableLabs, C. T. (1996). *Data Over Cable Interface Specification, Cable Modem Termination System - Network Side Interface Specification*. SP-CMTS-NSII01-960702.
- CableLabs, C. T. (2003). *Data Over Cable Service Interface Specifications, Cable Modem to Customer Premise Equipment Interface*. SP-CMCI-109-030730.
- Cen, S., & Walpole, J. (1998). Flow and congestion control for internet streaming applications. *Proceedings Multimedia Computing and Networking (MMCN98)*.
- Chandrasekaran, G., Hawa, M., & Petr, D. (2003). Preliminary Performance Evaluation of QoS in DOCSIS 1.1. *Information and Telecommunication Technology Center, University of Kansas, Technical Report 22736-01*.
- Chatterjee, S. (1999). TCP Behavior over HFC Cable Modem Networks. *Computer Communications, Vol. 22, n° 14*, 1321-1332.

- Chu, K. C., Tai, Y. G., Lee, W. T., & Chung, P. C. (2002). A Novel Mechanism for Providing Service Differentiation over CATV Network. *Chu, K. C., Tai, Y. G., Lee, W. T. and Chung, P. C., Computer Communications, Vol. 25, , 1214-1229.*
- Chunpeng, X., & Benny, B. (2004). *Measured QoS Performance of the DOCSIS Hybrid-Fiber Coax Cable Network*. Georgia: School of ECE, Georgia Institute of technology.
- CNTV. (02 de diciembre de 2008). *Televisión Digital Terrestre*. Recuperado el 02 de diciembre de 2008, de portal cntv.org.co: http://www.cntv.org.co/cntv_bop/tdt/
- Code4tv. (2009). *Curso Multimedia Home Platform 1.1.2*. Recuperado el 3 de enero de 2009, de portal code4tv.com/: <http://www.code4tv.com/res/mhp112course/16-CODE4TV-MHP-SI-DVB.pdf>
- Cohen, R., & Ramanathan, S. (1998). TCP for high performance in hybrid fiber coaxial broad-band access networks. *IEEE/ACM Trans. Networking , vol. 6, , 15-29.*
- Cruz, B. C. (2008). *Buenas Prácticas con Xlets*. Recuperado el 15 de Noviembre de 2008, de Portal <http://www.cesnavarra.net/>: <http://www.cesnavarra.net/cesdigital/Lists/Noticias%20CESDigital/DispFormCES.aspx?List=5ec0dfc7%2D7911%2D470b%2D8b6b%2D71ba72783fdd&ID=63>
- DekTec. (2007). *DVB-T Modulador with UHF Upconverter*.
- Díaz, J., Pazos, J., Fernández, A., & López, M. (2007). Objetos adaptativos de aprendizaje para T-learning. *IEEE Latin America Transactions, , vol. 5, Nro. 6.,*
- diSessa, A. (2004). *Changing minds: computer, learning, and literacy*. Massachusetts: The MIT Press.
- Dosi, A. P. (2004). New Frontiers of T-Learning: the Emergence of Interactive Digital Broadcasting Learning Services in Europe. *ED-Media World Conference on Educational Multimedia, Hypermedia & Telecommunications* (págs. 4831-4836). Lugano, Switzerland: AACE.
- Droubi, M., Idirene, N., & Chen, C. (2000). Dynamic bandwidth allocation for the HFC DOCSIS MAC protocol. *Proc. IEEE IC3N, .*
- EDiTV, P. (2008). *Educación Virtual basada en Televisión Digital Interactiva para apoyar procesos educativos a distancia*. Recuperado el 15 de Octubre de 2008, de Portal: <http://www.unicauca.edu.co/> <http://www.unicauca.edu.co/EDiTV>
- Elloumi, O., Golmie, N., Afifi, H., & Su, D. A. (2000). Simulation-based study of TCP Dynamics over HFC networks. *Computer Networks (21), , , 307-323.*
- España, M. C. (2003). *Servicios Avanzados de Telecomunicación*. España: Ediciones Díaz Santos.
- ETSI_EN_300_429_V1.2.1. (1998). *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems*.
- ETSI_EN_300_744. (2004-11). *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television, EBU-UER. V1.5.1.*
- ETSI_EN_300_744. (2004-11). *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television, EBU-UER. V1.5.1.*
- ETSI_TR_101_154. (2007). *Digital Video Broadcasting (DVB) Implementation guidelines for the use of MPEG-2 System, Video and Audio in satellite, cable and terrestrial broadcasting applications*.
- ETSI_TS_101_154. (2007). *Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream. V1.8.1.*
- ETSI_TS_102_812. (2006). *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1. V1.2.2.*
- Feamster, N., Bansal, D., & Balakrishnan, H. (2001). On the Interactions between Layered Quality Adaptation and Congestion Control for Streaming Video. *N. Feamster, D. Bansal, and H. Balakrishnan. . In 11th International Packet Video Workshop , Kyongiu, Korea.*
- FFMPEG. (2007). *FFMPEG*. Recuperado el 15 de enero de 2009, de portal [ffmpeg.mplayerhq.hu:](http://ffmpeg.mplayerhq.hu/) <http://ffmpeg.mplayerhq.hu/download.html>

- Floyd, S., & Fall, K. (1999). Promoting the use of end-to-end congestion control in the Internet,. *IEEE/ACM Trans. Networking*, vol. 7, n°. 4 , 458-472.
- Freshmeat. (2008). *mplex13818*. Recuperado el 223 de Enero de 2009, de Portal freshmeat.net: <http://freshmeat.net/projects/mplex13818/>
- García, M. (2002). “*Modelado de prestaciones de Redes de área metropolitana para la transmisión de datos, basadas en tecnología hibridad fibra-coaxial*”. Tesis Doctoral, Área de Ingeniería Telemática. Gijón - España: Universidad de Oviedo.
- García, M., García, D. F., García, V. G., & Bonis, R. (2004). Analysis and Modeling of Traffic on a Hybrid Fiber-Coax Networ. *Journal on Select Areas in Communications, Volume 22, Inssue 9. IEEE* , 1718 - 1730.
- García, M., García, D. F., García, V. G., & Bonis, R. (2004). Analysis and Modeling of Traffic on a Hybrid Fiber-Coax Network. *Journal on Select Areas in Communications, Volume 22, Inssue 9. IEEE* , 1718 - 1730.
- García, M., García, D. F., García, V. G., & Bonis, R. (2004). Analysis and Modeling of Traffic on a Hybrid Fiber-Coax Network. *Journal on Select Areas in Communications, Volume 22, Inssue 9. IEEE* , 1718 - 1730.
- García, M., García, D. F., García, V. G., García, X. G., & Bonis, R. (2002). Traffic Analysis of Data Transmission on Hybrid Fiber Coax Network. *IASTED International Conference on Communication Systems and Networks*. Málaga - España.
- García, R. (2005). “*Desarrollo de un modelo de red con tecnología FTTH para transmisión de voz y datos. Análisis del protocolo de acceso al medio y validación del tráfico generado*”. Tesis Doctoral, Área de Ingeniería Telemática. Gijón - España: Universidad de Oviedo.
- García, R., García, V. G., Pañeda, X. G., Melendi, D., & Neira, A. (2004). Aggregated traffic generation in FTTH networks. *International Conference on Applied Computing*. Lisboa - Portugal.
- García, X. (2004). *Análisis, modelado y configuración de servicios de video bajo demanda sobre redes de cable*, Tesis Doctoral, Área de Ingeniería Telemática. Gijón - España: Universidad de Oviedo.
- GNU, L. F. (25 de 12 de 2008). *Televisión Interactiva*. Recuperado el 1 de enero de 2009, de portal es.wikipedia.org: http://es.wikipedia.org/wiki/Televisi%C3%B3n_interactiva#Tipos_de_interacci.C3.B3n
- Golmie, N., Mouveau, F., & Su, D. A. (1999). A Comparison of MAC Protocols for Hybrid Fiber/Coax Networks: IEEE 802.14 vs. MCNS. *IEEE International Conference on Communications, ICC '99* , 266 - 272.
- Golmie, N., Saintillan, Y., & Su, D. H. (1999). A Review of Contention Resolution Algorithms for IEEE 802.14 Networks. *IEEE Communications Surveys* , 2-12.
- González, A. A. (2008). *La televisión digital interactiva y sus aplicaiones educativas*. Recuperado el 1 de septiembre de 2008, de portal dialnet.unirioja.es: http://dialnet.unirioja.es/servlet/fichero_articulo?codigo=1985787&orden=0
- González, C. (2002). *Sistemas de televisión por cable: una visión integral*. Recuperado el 24 de 12 de 2008, de portal cinit.org.mx: <http://www.cinit.org.mx/articulo.php?idArticulo=6>
- GTCF, G. d. (2005). *Docuemnto informativo sobre la TV digital. TELEVISIÓN DIGITAL: ADAPTARSE HOY PARA LA NUEVA TELEVISIÓN*. España: Colegio Oficial de Ingenieros de Telecomunicaciones.
- ImageMagick. (2008). *ImageMagick*. Recuperado el 10 de Enero de 2009, de portal imagemagick.org: <http://www.imagemagick.org/script/download.php>
- ISO/IEC. (2000). *Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile*., ISO/IEC 14496-2/FPDAM4.
- ISO/IEC_13818-1. (2000). *IS 13818-1, International Standard, Information Technology – Generic coding of moving pictures and associated audio information – Part 1: Systems*.
- ISO/IEC_2. (1995). *International Standard, Information Technology – Generic Coding of moving pictures and associated audio – Part 2: Video*.
- ISO/IEC-11172-3. (1996). *JTC1/SC29/WG11 MPEG. International Standard IS. Coding of moving pictures and associated audio for digital storage media at up to 1-5 Mbit/s*.
- ISO/IEC-13818. (1995). *13818, Information Technology – Generic coding of moving pictures and associated audio information (MPEG-2)*.

- ISO/IEC-13818-3. (1994). *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information. Part 3: Audio.*
- ISO/IEC-13818-6. (2001). *Extensions for DSM-CC. Digital Storage Media Command and Control.*
- ISO/IEC-14496. (2002). *Information Technology – Coding of audio-visual objects (MPEG-4)*”.
- ISO/IEC-14496-3. (1998). *Information Technology – Very low bitrate audio-visual coding – Part 3: Audio.*
- ITU-T_G.1010. (2001). *Categorías de calidad de servicio para los usuarios de extremo de servicios multimedia.*
- ITU-T_H.262. (1995). *International Standard, Information Technology – Generic Coding of moving pictures and associated audio – Part 2: Video.*
- ITU-T_IPTV. (2007). *FG IPTV-DOC-0199 IPTV vocabulary of terms. Focus Group On IPTV.*
- ITU-T-H.264. (2003). *Recommendation H.264 - ISO/IEC 14496-10, Advanced Video Coding.*
- Jokipielto, P. (2006). *Modelling and Content Production of Distance Learning Concept for Interactive Digital Television.* Recuperado el 20 de Noviembre de 2008, de portal lib.tkk.fi: <http://lib.tkk.fi/Diss/2006/isbn9512285428/isbn9512285428.pdf>
- Ju, H. J., & Liao, W. (2002). Fast request transmission (FRT) for DOCSIS based CATV networks. *IEEE Int. Conf. Multimedia and Exp (ICME), 2002* , 543-547.
- Kalampoukas, L., Varma, A., & Ramakrishnan, K. K. (1996). Two-way TCP traffic over ATM: Effects and analysis. *IEEE Infocom* .
- Kaza, T. (2001). *Application Performance and System Impact over DOCSIS Networks in a Lossy System.* Kansas City: Master’s thesis, Dept. of Computer Science, University of Missouri Kansas City,.
- Krasic, C., Li, K., & Wapole, J. (2001). The Case for Streaming Multimedia with TCP. *8th International Workshop on Interactive Distributed Multimedia Systems (iDMS)* .
- Lakshman, T. V., Madhow, U., & Suter, B. (1997). Window-based error recovery and flow control with a slow acknowledgment channel: A study of TCP/IP performance. *IEEE INFOCOM '97* , 38–47.
- Lee, W. T., Chung, K. C., Chu, K. C., & Pan, J. Y. (2006). DOCSIS performance analysis under high traffic conditions in the HFC networks. *IEEE Transactions on Broadcasting, vol. 52, no. 1* , 21-30.
- Li, J. R., Dwyer, D., & Bharghavan, V. (1999). A transport protocol for heterogeneous packet flows. *IEEE Infocom'99* .
- Liao, W., & Ju, H. J. (2004). Adaptive Slot Allocation in DOCSIS - based CATV Networks. *IEEE Transactions On Multimedia, Vol. 6, N° 3* .
- Lin, Y. D., Huang, C. Y., & Yin, W. M. (1998). Allocation and scheduling algorithm for IEEE 802.14 and MCNS in hybrid fiber coaxial networks. *IEEE Trans. on Broadcasting, vol. 44, n° 4* , 427– 435.
- Lopez, M., Pazos, J., Blanco, Y., Lopez, M., & Duque, J. (2005). Solutions for Personalized T-learning . *in 3rd European Conference on Interactive Television, Aalborg, Denmark* , 235-236.
- Lu, D.-J., Wang, Y.-C., Ho, J.-M., Ko, M.-T., & Chen, M.-C. (1997). Experience in designing a TCP/IP based VOD system over a dedicated network. *IEEE International Symposium on Consumer Electronics* .
- Main_Concept. (2009). *MainConcept Reference.* Recuperado el 2 de febrero de 2009, de portal mainconcept.com: <http://www.mainconcept.com/site/prosumer-products-4.html>
- Marianov, V., Oberli, C., & Rios, M. (2006). *Análisis de los Estándares de Transmisión de Televisión Digital Terrestre y su Aplicabilidad al Medio Nacional.* Recuperado el 12 de 11 de 2008, de portal subtel.cl (DICTUC, Pontificia Universidad Católica de Chile): http://www.subtel.cl/prontus_tvd/site/artic/20070315/asocfile/20070315173311/estudio_uc.pdf
- Marketing ICT, a. (2009). *Terisias.* Recuperado el 28 de Enero de 2009, de portal tiresias.org: <http://www.tiresias.org/about/index.htm>
- Martin, J. (2004). The Interaction Between the DOCSIS 1.1/2.0 MAC Protocol and TCP Application Performance. *Proceedings of the International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks*, (págs. P57/1-10). Ikley, UK.

- Martin, J., & Shrivastav, N. (2003). Modeling the DOCSIS 1.1/2.0 MAC Protocol. *Proceedings International Conference on Computer Communications and Networks*. Dallas.
- Martínez, S. (2005). *Transmisión de voz sobre IP en redes de cable*. Tesis de Maestría. , Área de Ingeniería Telemática. Gijón - España: Universidad de Oviedo.
- Meritxell, E. (2002). *Interactividad e interacción*, Departamento de Pedagogía, Universidad de Girona. España. Documento PDF disponible en: Recuperado el 30 de noviembre de 2008, de portal dialnet.unirioja.es: http://dialnet.unirioja.es/servlet/fichero_articulo?codigo=1252603&orden=89124
- MHP. (2005). *KnowledgeDatabase*. Recuperado el 15 de octubre de 2008, de portal mhp-knowledgebase.org: http://www.mhp-knowledgebase.org/publ/mhp-kdb_d12.pdf
- MHP, T. M. (2006). *The MHP-Guide*. KnowledgeDatabase.
- Mimundo. (2008). *MimundoTV Player*. Recuperado el 10 de enero de 2009, de portal mimundotv.de: <http://www.mimundotv.de/parking.php?ses>
- Ming, I. T. (1999). Performance evaluation of web browsing over hybrid fiber coaxial broad-/band networks. *Proc. ICON* , 372–382.
- MITC. (2008). *Televisión Digital por Cable (MINISTERIO DE INDUSTRIA, TURISMO Y COMERCIO)*. Recuperado el 15 de Septiembre de 2008, de portal televisiondigital.es: <http://www.televisiondigital.es/Cable/>
- MJPEG_tools. (2008). *mjpeg*. Recuperado el 15 de Enero de 2009, de portal mjpeg.sourceforge.net: <http://mjpeg.sourceforge.net/>
- MPlayer. (2008). *MEncoder*. Recuperado el 28 de diciembre de 2008, de portal mplayerhq.hu: <http://www.mplayerhq.hu/design7/news-es.html>
- Nortel. (2003). *Introduction to Quality of Service (QoS)*. EE.UU: Nortel Networks.
- OPNET Technologies, I. (2005). *General Tutorials*. USA: Bethesda MD 20814-7904.
- Perales, B., Benoit, H., & Días, F. (2004). *Televisión Digital*. Thomson Paraninfo.
- Pérez, E. (2009). *01 INTRO2 TRANSMISION - Introducción a la TV Digital II La Transmisión*. Recuperado el 1 de enero de 2009, de code4tv.com: <http://www.code4tv.com/res/mhp112course/01-CODE4TV-MHP-INTRO2-TRANSMISION.pdf>
- Puche, W. S., Amaya, F., & Cárdenas, A. M. (2008). Simulación de la subcapa PMD en sentido de retorno según la especificación DOCSIS 3.0 para redes HFC. *Congreso Colombiano de Comunicaciones – Colcom, Popayán* .
- Pulkkinen, J. (2006). *The paradigms of E-Education, An analysis of the communication structures in the research on information and communication technology integration in education in the years 2000-2001*. Oulu: Oulun yliopisto.
- Rabbat, R., & Siu, K. (1999). QoS Support for Integrated Services over CATV. *IEEE Communications Magazine* , 64-68.
- Rege, K., & Sun, D. (Noviembre de 2005). A simple analytical model to estimate VoIP signaling delays in an HFC access network. *GLOBECOM - IEEE Global Telecommunications Conference* , 1344 – 1348.
- Rejaie, R., Handley, M., & Estrin, D. (1999). An end-to-end rate-based congestion control mechanism for realtime streams in the internet. *Proceedings of IEEE Infocomm* .
- Reyes, M. (2001). *Estudio de Transmisión Digital en el Retorno para la aplicación de Redes HFC del Sistema de Televisión por Cable de TELEDINAMICA S. A*. Universidad Industrial de Santander.
- Reyes, M. (2002). *Estudio y Diseño del Procesamiento Digital de Señales para sus uso en Aplicaciones del forward del sistema de Televisión por Cable de TELEDINAMICA S. A* . Universidad Industrial de Santander.
- Sala, D., & Limb, J. (1998). Comparison of Contention Resolution Algorithms for a Cable Modem MAC Protocol. *Seminar on Broadband Communications*, (págs. 83-90). Zurich.
- Sala, D., Limb, J. O., & Khaunte, S. U. (1997). Adaptive MAC protocol for a Cable Modem. *Georgia Tech Technical Report GIT-CC-97/14* .
- Sdralia, C., Smythe, P., Tzerefos, & Cvetkovic, S. (1999). Performance characterization of the MCNS docsis 1.0CATV protocol with prioritised first come first served scheduling. *IEEE Trans. Broadcasting*, vol. 45, , 161–205.

- Sdralia, V., Tzerefos, P., & Smythe, C. (2000). Recovery Analysis of the DOCSIS protocol after service disruption. *submitted to IEEE Transactions on Broadcasting*, .
- Sdralia, V., Tzerefos, p., Cvetkovic, S., & Smythe, C. (2000 a). Fault Recovery of the DOCSIS protocol. *Contribution to IEEE 802.16 WG, IEEE 802.16.lmc-00/1 I*, .
- Seung-Eun, H., & Oh-Hyeong, K. (2006). Performance Analysis of Single- and Multi-Channel Contention Resolution Algorithm for the DOCSIS MAC Protocol. *IEEE ICC Proceedings*, 1083-1088.
- Shah, N., Kouvatso, D., Martin, J., & Moser, S. (2005). A tutorial on DOCSIS: protocol and performance models. *Proceedings of the international working conference on performance modeling and evaluation of heterogeneous networks*.
- Shimonishi, H., Hama, T., & Murase, T. (2007). TCP congestion control enhancements for streaming media. *in Proceedings of IEEE Consumer Communications and Networking Conference* .
- Siahaan, M., & Hamdani, A. (2002). Determination Consideration for DOCSIS Implementation. *IEEE*, 431- 434.
- Sierra, L. G. (Noviembre de 2001). *DOCSIS. El Conjunto de Estándares Cable Módem de CableLabs*. Recuperado el 2 de septiembre de 2008, de Portal Cinit.org.mx: <http://www.cinit.org.mx/articulo.php?idArticulo=4>
- Sierra, L. G. (1 de Febrero de 2003). *Midiendo la calidad de señales digitales: Una orientación práctica*. Recuperado el 15 de Noviembre de 2008, de portal: [www.cinit.org.mx: http://www.cinit.org.mx/articulo.php?idArticulo=16](http://www.cinit.org.mx/articulo.php?idArticulo=16)
- Simoneta, J. (2002). *Televisión Digital Avanzada*. Buenos Aires: 1 edición, Intertel.
- Taimal, I. R., & Rueda, D. F. (2009). *Análisis y evaluación de una red de acceso HFC para la distribución de contenidos educativos de iTV con QoS*. Popayán: Universidad del Cauca.
- Tan, D., & Zakhor, A. (1999). Real-time internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Trans. Multimedia*, vol. 1, n°. 2, 172-186.
- Tan, W., & Zakhor, A. (1998). Internet video using error resilient scalable compression and cooperative transport protocol. *Proc. ICIP, volumen 1*, 17-20.
- Televés. (2006). *Televisión y Radio, analógica y digital*. Santiago de Compostela-La Coruña: Telvés.
- TMPGEnc. (2008). *TMPGEnc Xpress*. Recuperado el 20 de Diciembre de 2008, de portal [tmpgenc.net](http://www.tmpgenc.net): <http://www.tmpgenc.net/es/index.html>
- Trigal, O. A. (2007). *Televisión Digital en Argentina*. Buenos Aires: Instituto Tecnológico de Buenos Aires.
- Tzerefos, P., Sdralia, V., Smythe, C., & Cvetkovic, S. (1999). P. Tzerefos, V. Delivery of Low Bit Rate Isochronous Streams of the DOCSIS 1 .O Cable Television Protocol. *IEEE Transactions on Broadcasting*, Vol. 45, N°. 2, 206-214.
- UIT-T-H.262. (2007). *SISTEMAS DE AUDIOVISUALES Y MULTIMEDIA: infraestructura de los servicios audiovisuales - Codificación de imágenes vídeo en movimiento*.
- Vera, D. P. (2002). *Análisis y Diseño de una Red HFC para la ciudad de Popayán. Tesis de pregrado* . Popayán: Universidad del Cauca.
- Wanjiun, L. (2006). The behavior of TCP over DOCSIS-based CATV networks. *IEEE Transactions on Communications*, vol. 54, no. 9, 1633 – 1642.
- Wen, K. K., Sunil, K., & Jay, K. C. (2002). Dynamic Collision Resolution and Traffic Scheduling for DOCSIS Systems with QoS Support. *IEEE GLOBECOM'02, Vol.2* .
- Wen, K. K., Sunil, K., & Jay, K. C. (2003). Improved Priority Access, Bandwidth Allocation and Traffic Scheduling for DOCSIS Cable Networks. *IEEE Transactions On Broadcasting*, Vol. 49, n° 4 .
- Widmer, J., Denda, R., & Mauve, M. (2001). A survey on TCP-friendly congestion control. *IEEE Network*, vol. 15, 28-37.
- Wireshark. (2008). *wireshark*. Recuperado el 15 de marzo de 2008, de portal [wireshark.org](http://www.wireshark.org): <http://www.wireshark.org>
- Wolff, P. A. (2008). *Diseño e Implementación de un curso de Servicios sobre Redes HFC de Nueva Generación*. Santiago de Chile: Universidad de Chile.

Yan, J., Katrinis, K., May, M., & Plattner, B. (2006). Media- and tcp-friendly congestion control for scalable video streams. *IEEE Transactions on Multimedia*, vol. 8, n°. 2 , 196-206.

Yin, W., & Lin, Y. (2000). Statistically Optimised Minislot Allocation for Initial and Collision Resolution in Hybrid Fiber Coaxial Networks. *IEEE Journal on Selected Areas in Communications*, Vol. 18, n°. 9, .

Zhang, L., Shenker, S., & Clark, D. D. (1991). Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. *ACM Computer Communication Review* .

Anexo B

Tabla de Contenido

Anexo B	1
B.1 Configuración de herramientas.....	1
B.1.1 Configuración del entorno de desarrollo Eclipse	1
B.1.2 Ejecución de aplicación directamente en el STB de desarrollo	9
B.1.3 Simulación de la aplicación en mimundoTV Player.....	11
B.2 Ejemplos de generación y distribución de contenidos	16
B.2.1 Codificación de audio y video con FFMPEG	16
B.2.1.1 Codificación de Video	16
B.2.1.2 Codificación del Audio	17
B.2.2 Codificación de audio y video con Mainconcept Reference.....	17
B.2.3 Conversión de imágenes en Videodrips.....	20
B.2.4 Creación de tablas de información de servicio	23
B.2.5 Creación de eventos DSMCC.....	27
B.2.6 Generación de flujos de transporte	28
B.2.6.1 Creación de un flujo de transporte de video.....	28
B.2.6.2 Creación de un flujo de transporte de audio.....	28
B.2.6.3 Creación de un flujo de transporte para carrusel de objetos.....	29
B.2.6.4 Creación de flujos de transporte para tablas de señalización	30
B.2.7 Multiplexación de flujos de transporte	30
B.2.8 Modulación de un flujo de transporte MPEG2	32
B.3 Ejecución de la aplicación en la cadena de EDiTV del laboratorio	32
Referencias.....	36

Lista de Figuras

Figura B.1. Secuencia de Creación de un proyecto	2
Figura B.2: Secuencia de Configuración de las librerías de compilación	4
Figura B.3. Secuencia de creación de un paquete y una clase	7
Figura B.4. Proyecto piloto con sus clases.....	7
Figura B.5. Clases generadas	8
Figura B.6. Ejemplo de la ejecución de una aplicación en el STB	9
Figura B.7. Progreso de la descarga de archivos	11
Figura B.8. Registro de ejecución	11
Figura B.9. Ejecución de una aplicación en mimundoTV Player – Paso uno	12
Figura B.10. Ejecución de una aplicación en mimundoTV Player – Paso dos	13
Figura B.11. Ejecución de una aplicación en mimundoTV Player – Paso tres	14
Figura B.12. Ejecución de una aplicación en mimundoTV Player – Paso cuatro	14
Figura B.13. Ejecución de una aplicación en mimundoTV Player – Paso cinco	15
Figura B.14. Ejecución de una aplicación en mimundoTV Player – Paso seis.....	15
Figura B.15. Ejecución de una aplicación en mimundoTV Player – Paso siete	16
Figura B.16. Perfil de video.....	18
Figura B.17. Parámetros de vídeo	18
Figura B.18. Parámetros avanzados de Vídeo	19
Figura B.19. Parámetros de Multiplexación	19
Figura B.20. Generación del flujo de Transporte.....	20
Figura B.21. Tabla de Información de Red.....	23
Figura B.22. Tabla de asociación de programas	24
Figura B.23. Tabla de descripción de Servicios.....	24
Figura B.24: Taba de descripción de Eventos DSMCC	25
Figura B.25. Tabla de Mapeo de Programas.....	25
Figura B.26. Continuación tabla de mapeo de Programas	26
Figura B.27. Tabla de Información de aplicación.....	26
Figura B.28. Configuración de archivos especiales del carrusel	27
Figura B.29. Sistema de ficheros del carrusel de objetos.....	27
Figura B.30. Salida de un flujo de transporte	29
Figura B.31. Script que transformación de tablas.	30
Figura B.32. Tasas de bits disponibles	31
Figura B.33. Ejecución de la aplicación en la cadena de EDiTV del laboratorio – Paso uno.....	33
Figura B.34. Ejecución de la aplicación en la cadena de EDiTV del laboratorio – Paso dos	33
Figura B.35. de la aplicación en la cadena de EDiTV del laboratorio – Paso tres	34
Figura B.36. Salida del comando python	34

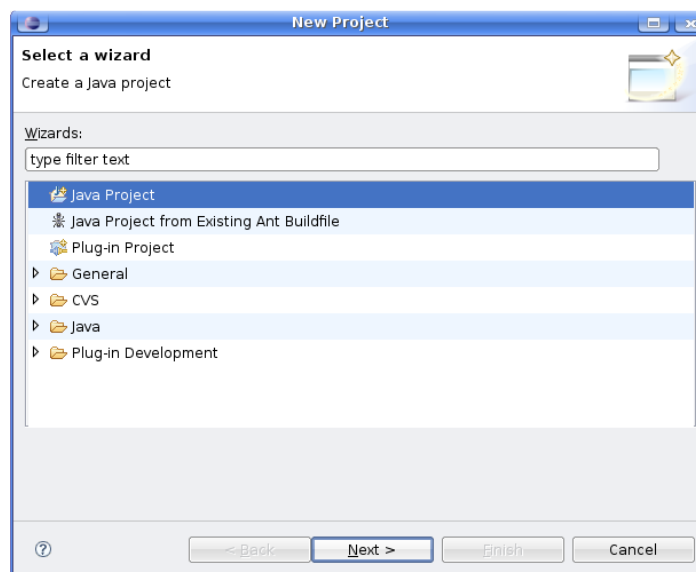
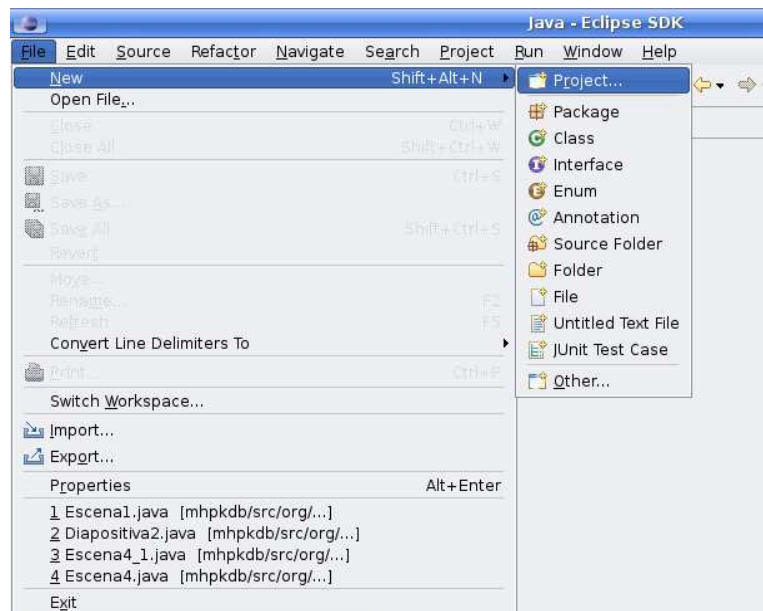
Anexo B

B.1 Configuración de herramientas

B.1.1 Configuración del entorno de desarrollo Eclipse

El entorno integrado de desarrollo Open Source Eclipse es el usado para desarrollar las clases que componen la aplicación piloto. La configuración de este entorno para aplicaciones de televisión digital bajo MHP es la siguiente (Eclipse, 2008):

- Creación de un proyecto, ver Figura B.1.



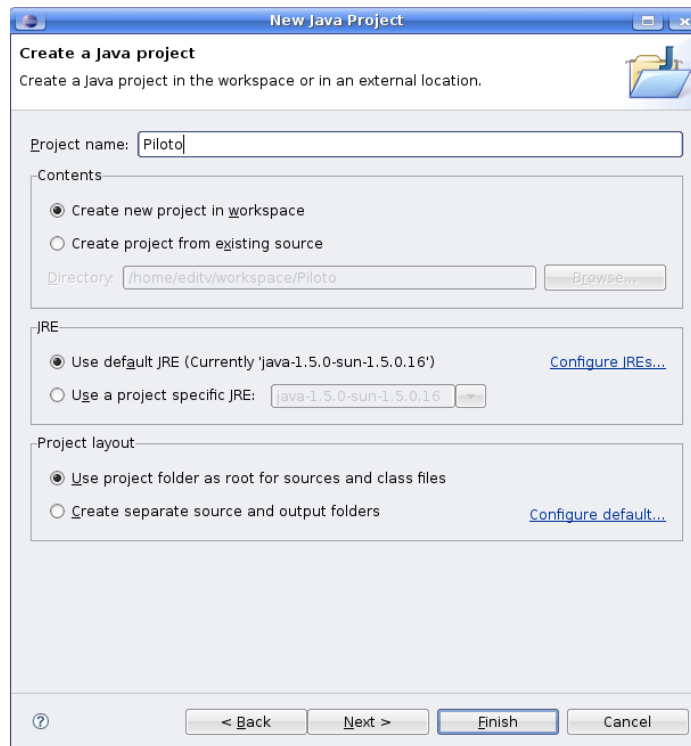
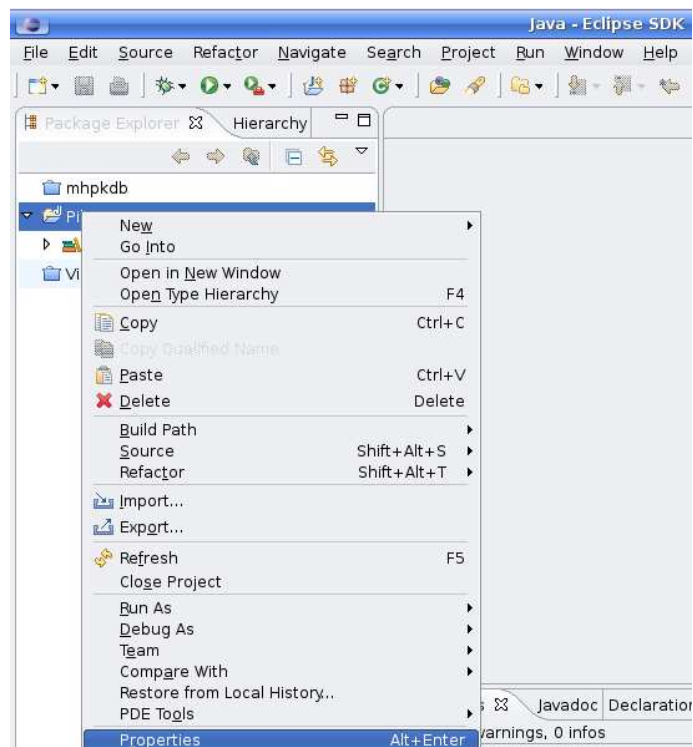
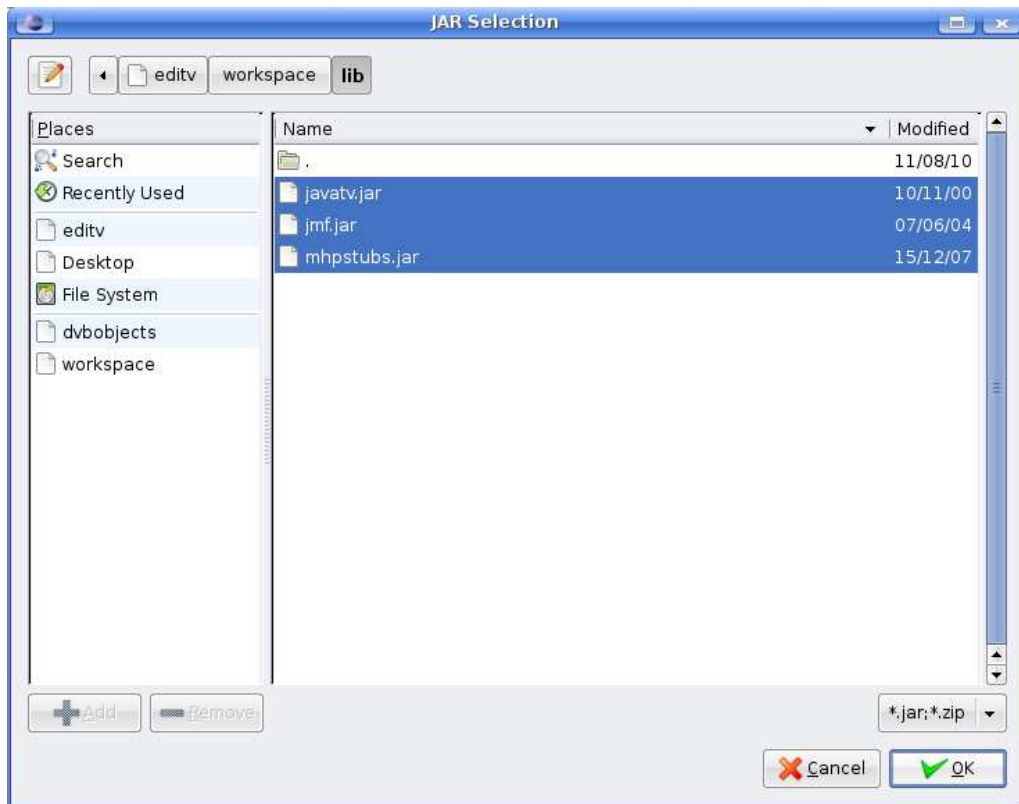
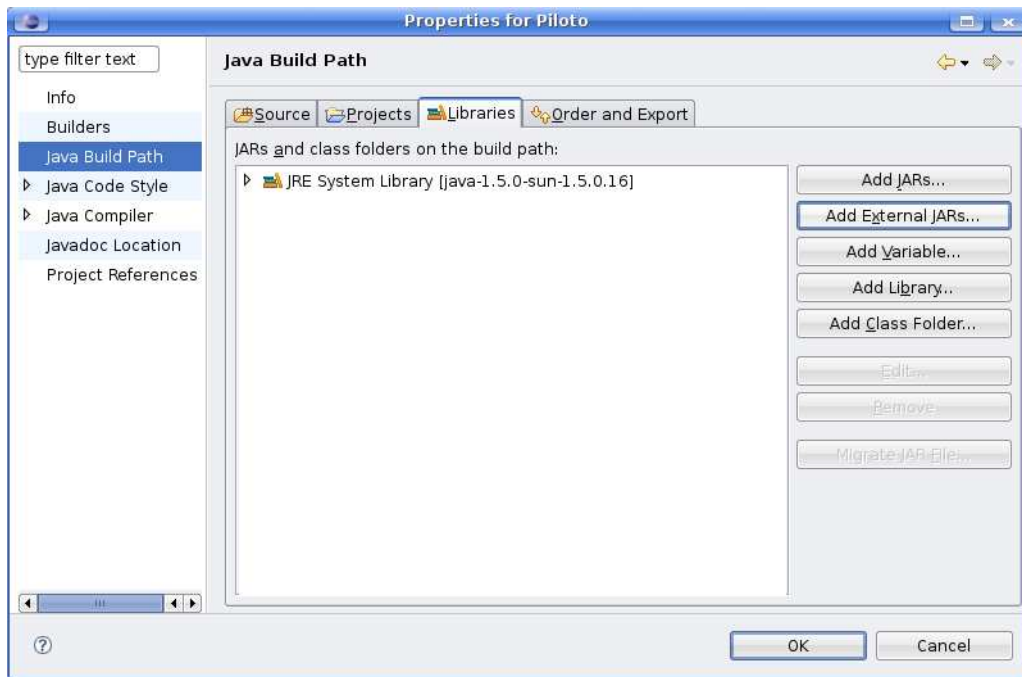


Figura B.1. Secuencia de Creación de un proyecto

- Configuración de las librerías de compilación, ver Figura B.2.





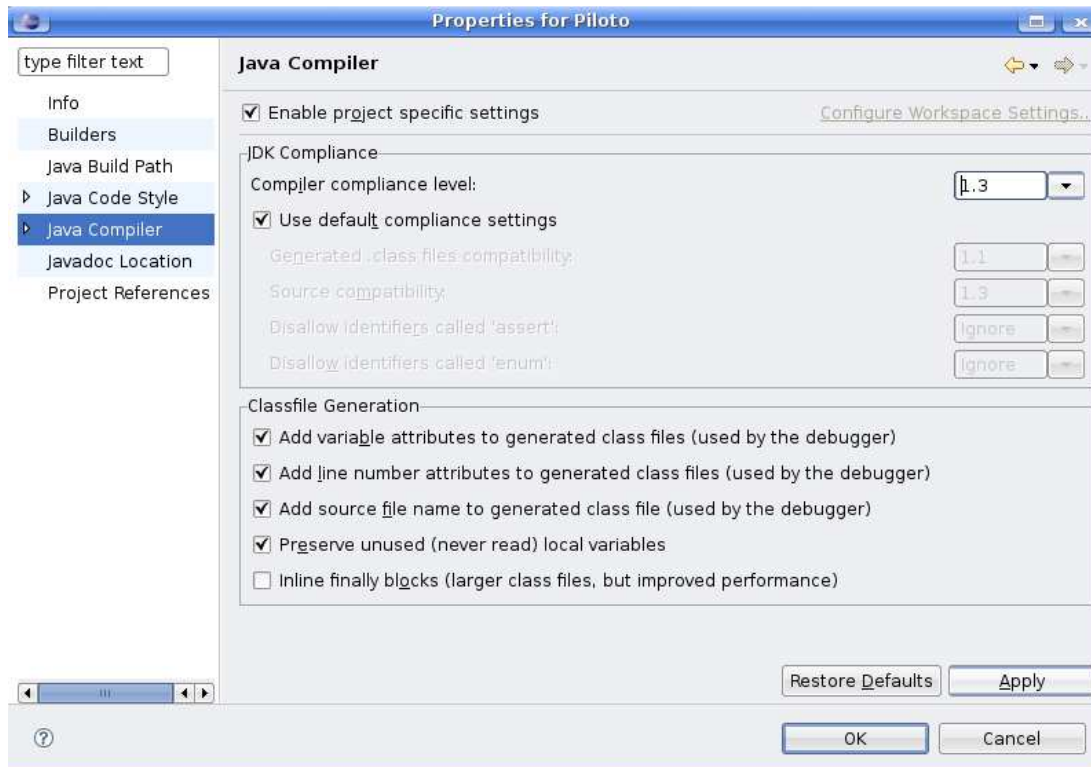
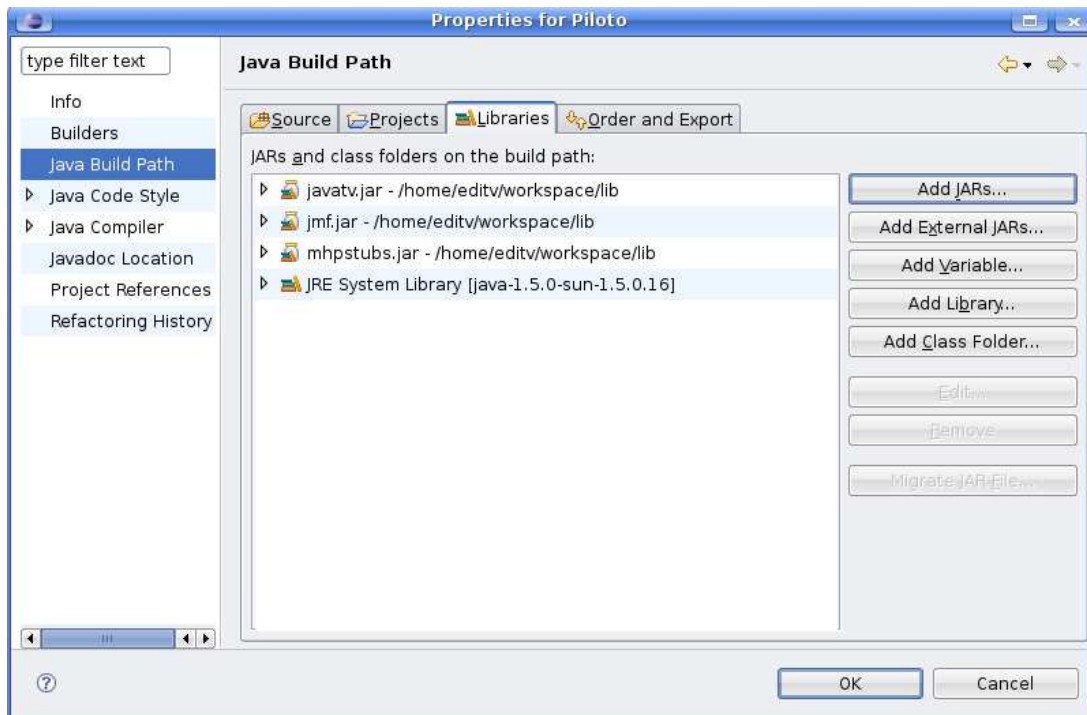
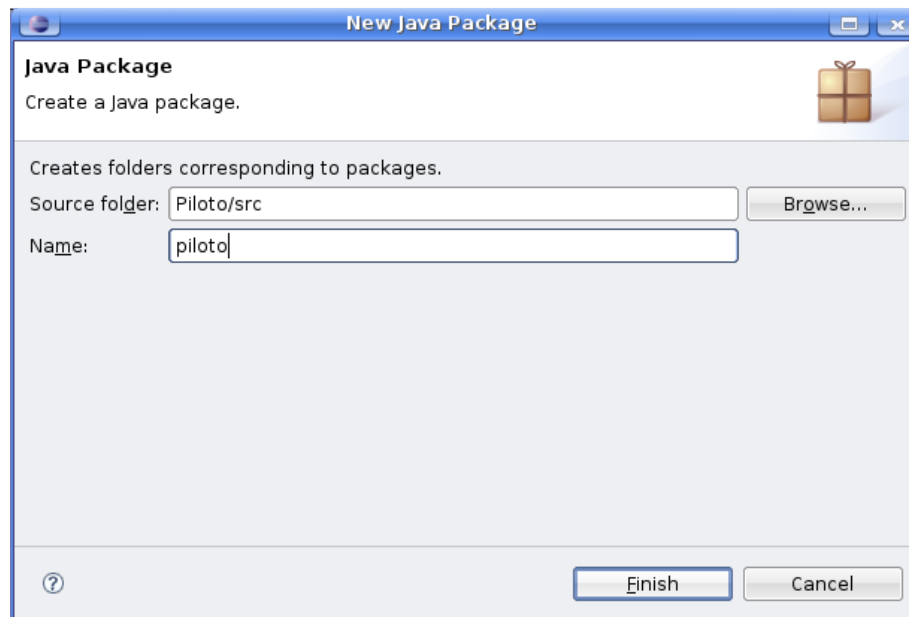
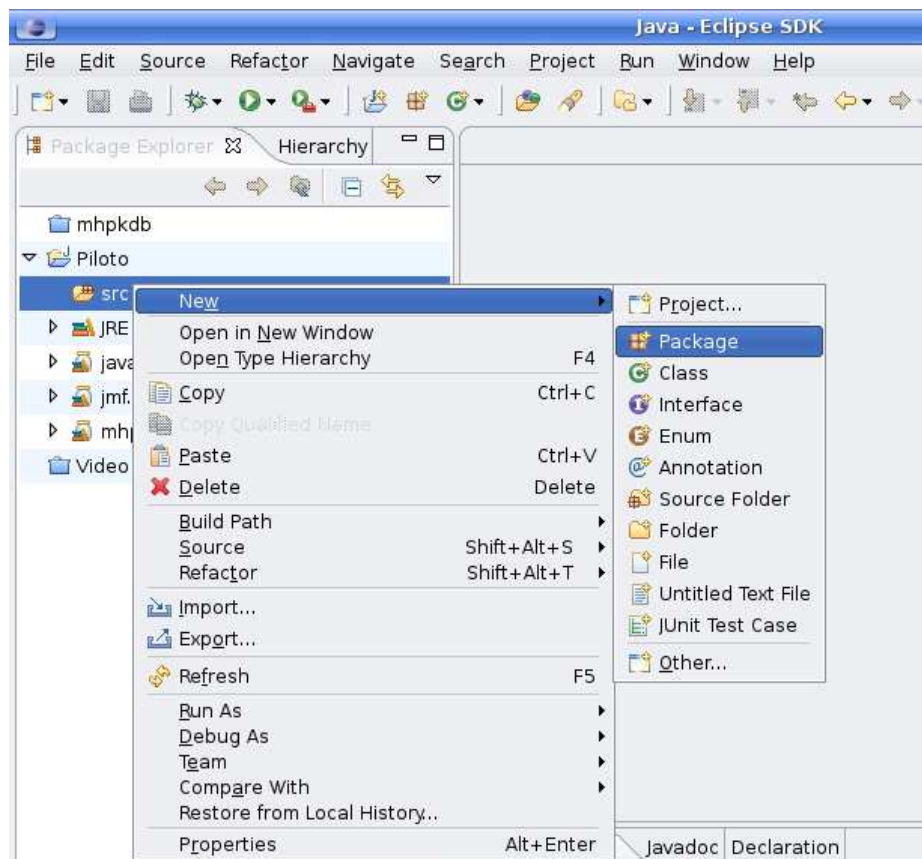
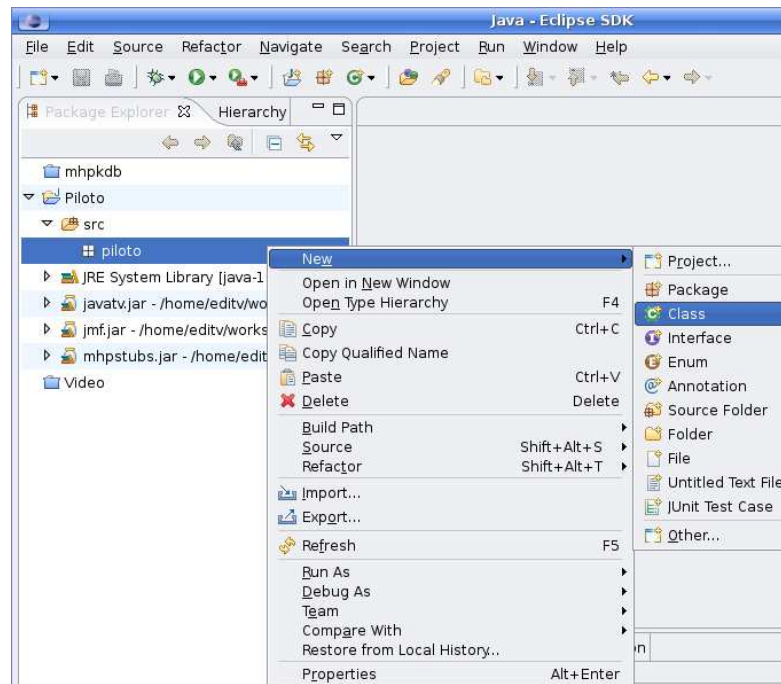


Figura B.2: Secuencia de Configuración de las librerías de compilación

- Creación de un paquete y una clase, ver Figura B.3.





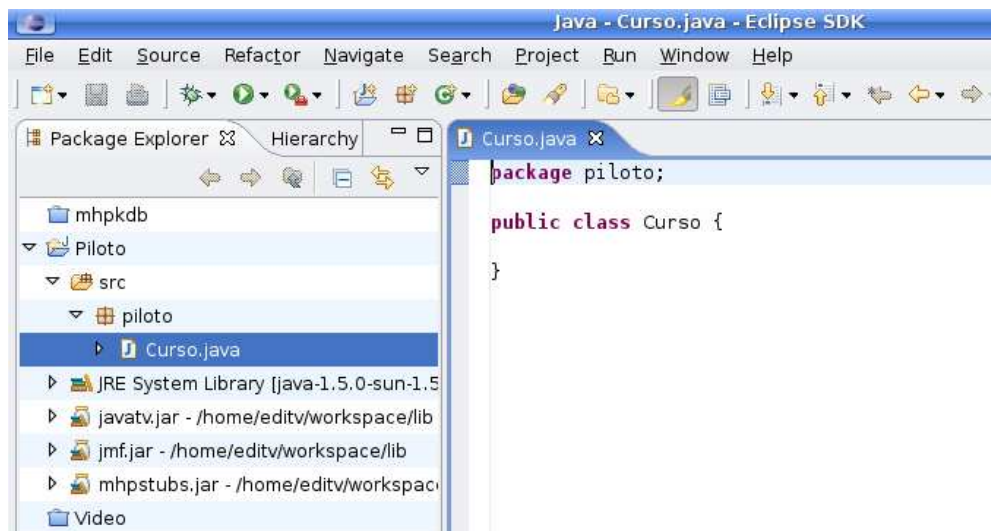


Figura B.3. Secuencia de creación de un paquete y una clase

- El proyecto piloto con sus clases, Figura B.4.

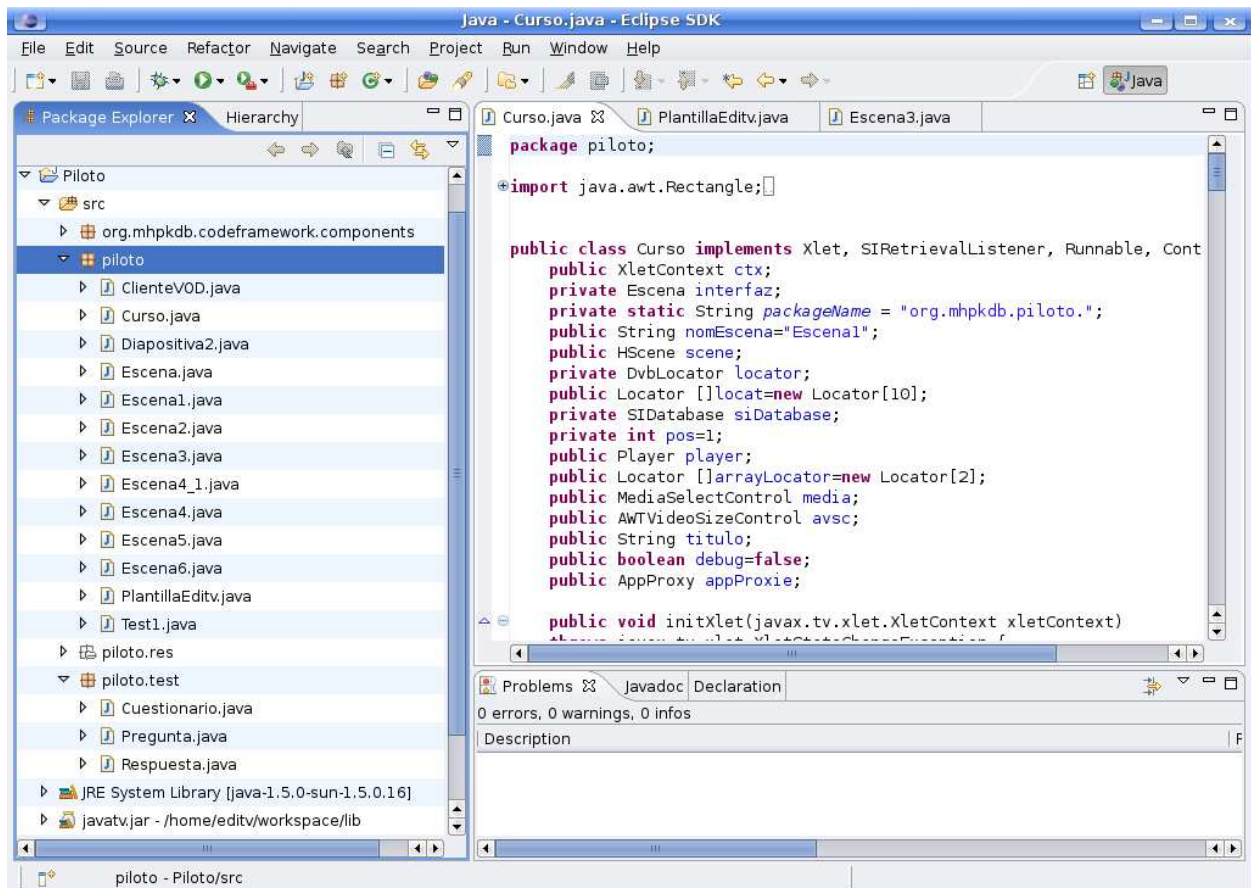


Figura B.4. Proyecto piloto con sus clases

- Clases generadas

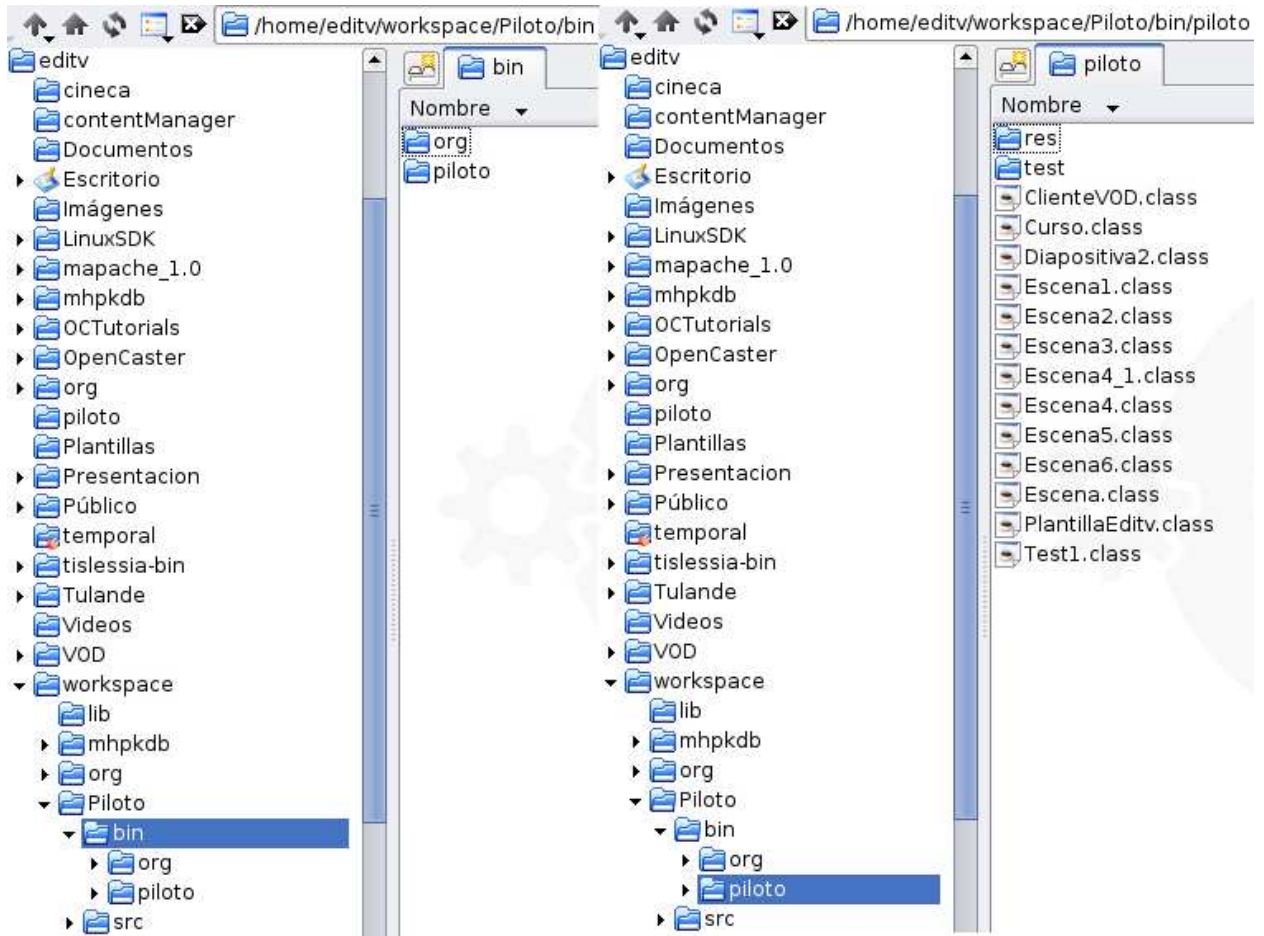


Figura B.5. Clases generadas

B.1.2 Ejecución de aplicación directamente en el STB de desarrollo

El STB de desarrollo permite ejecutar una aplicación MHP con este conectado directamente a un puerto serial del computador. Para esto cuenta con un conjunto de aplicaciones denominadas stbtools (ADB, 2008), (MHP_Stubs, 2008).

- STBProxy es una aplicación que se encarga de realizar la conexión entre el computador y el STB a través del puerto serial. La sintaxis para ejecutarlo es
- `stbproxy.exe -com numero_de_puerto -log ruta_archivo_log`

ej. `stbproxy.exe -com 1 -log c:\stb.log`, ver Figura B.6.

En el ejemplo el `numero_de_puerto` indica que se usará el puerto COM1 y `ruta_archivo_log` es el argumento para indicar la ruta completa al archivo donde se escribirá el registro de las comunicaciones. Aparece un icono en la barra inferior con opciones para parar o activar el Proxy. Además el archivo `stb.log` empieza a ser llenado con información enviada desde el STB.

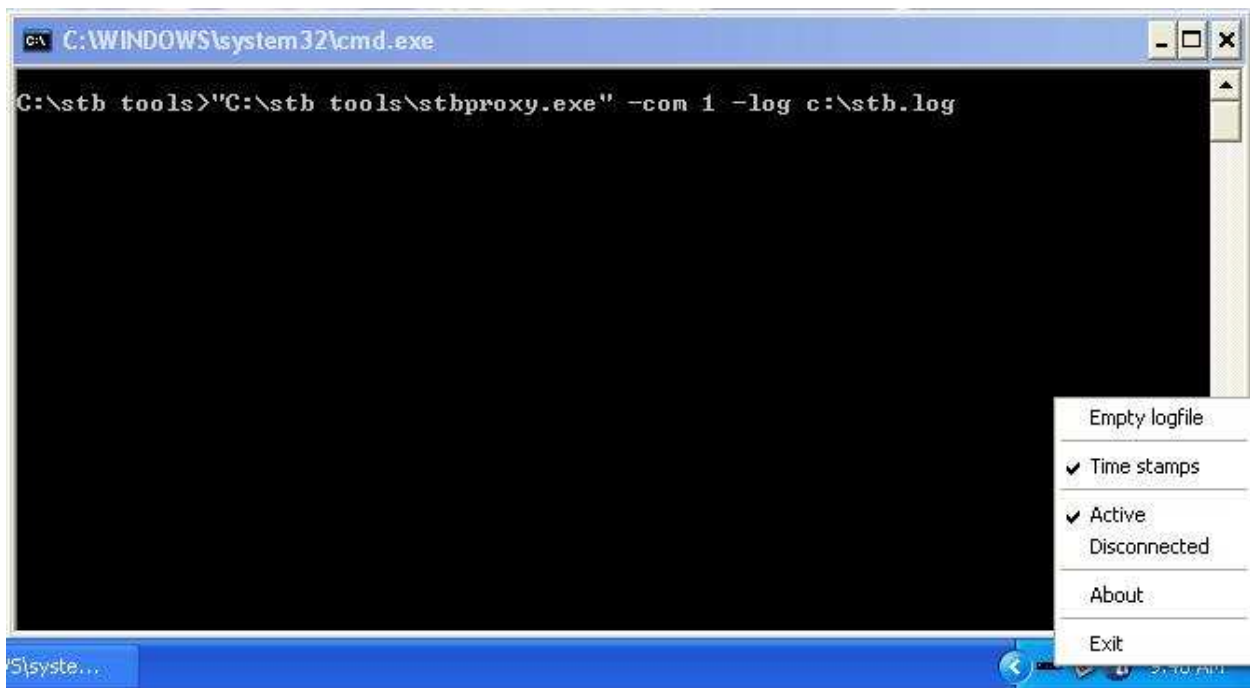


Figura B.6. Ejemplo de la ejecución de una aplicación en el STB

Después de ejecutar el programa `stbproxy`, ya puede descargarse una aplicación al STB mediante el programa `stbupload`. Se usa con la siguiente sintaxis:

```
stbupload host archivo_descripcion ruta_clases -exec
```

ej. `stbupload localhost xlets_description_file c:/piloto/clases -exec`

Donde `localhost` es la dirección IP o nombre del host desde donde se ejecuta el `stbproxy`, en este caso el mismo equipo. `Xlets_description_file` es el archivo con los parámetros de ejecución de la aplicación. `c:/piloto/clases` es la ruta a la carpeta donde se encuentran las clases java compiladas. `-exec` indica al stb que debe ejecutar la aplicación sin necesidad de tener un servicio al aire.

El archivo `xlets_description_file` es análogo a la tabla AIT de MPEG-2. Al menos debe contener los siguientes parámetros:


```
#app <Identificador de la aplicación> <Identificador de la organización>
app      0x00 0x00

#name eng <Nombre de la aplicación>
name     eng "Curso"

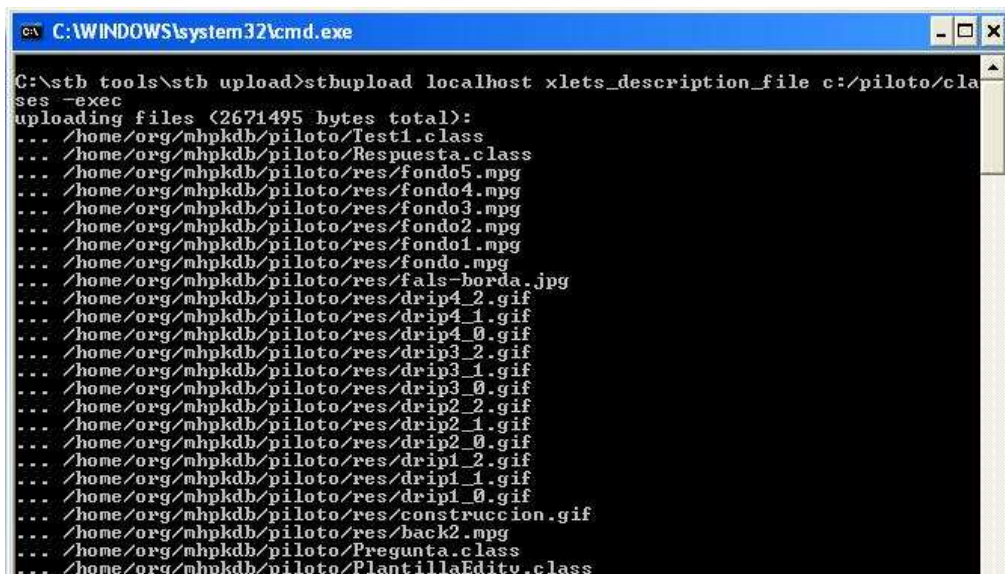
#control <Bandera de control de la aplicación: 1=autoarranque 2=presente
3=destruida 4=muerta>
control  1

#bound <Bandera de asociación al servicio (0 o 1)>
bound    0

#basedir <directorio de la aplicación en el STB, relativo a /home >
basedir  "/home"

#class <clase inicial>
class    "org.mhpkdb.piloto.Curso"
```

Al ejecutar el stbupload aparece una ventana de DOS con el progreso de la descarga de archivos. Al terminar la descarga de los archivos pude verse como la aplicación se ejecuta en el STB., ver Figura B.7.



```
C:\WINDOWS\system32\cmd.exe
C:\stb tools\stb upload>stbupload localhost xlets_description_file c:/piloto/cla
ses -exec
uploading files (2671495 bytes total):
... /home/org/mhpkdb/piloto/Test1.class
... /home/org/mhpkdb/piloto/Respuesta.class
... /home/org/mhpkdb/piloto/res/fondo5.mpg
... /home/org/mhpkdb/piloto/res/fondo4.mpg
... /home/org/mhpkdb/piloto/res/fondo3.mpg
... /home/org/mhpkdb/piloto/res/fondo2.mpg
... /home/org/mhpkdb/piloto/res/fondo1.mpg
... /home/org/mhpkdb/piloto/res/fondo.mpg
... /home/org/mhpkdb/piloto/res/fals-borda.jpg
... /home/org/mhpkdb/piloto/res/drip4_2.gif
... /home/org/mhpkdb/piloto/res/drip4_1.gif
... /home/org/mhpkdb/piloto/res/drip4_0.gif
... /home/org/mhpkdb/piloto/res/drip3_2.gif
... /home/org/mhpkdb/piloto/res/drip3_1.gif
... /home/org/mhpkdb/piloto/res/drip3_0.gif
... /home/org/mhpkdb/piloto/res/drip2_2.gif
... /home/org/mhpkdb/piloto/res/drip2_1.gif
... /home/org/mhpkdb/piloto/res/drip2_0.gif
... /home/org/mhpkdb/piloto/res/drip1_2.gif
... /home/org/mhpkdb/piloto/res/drip1_1.gif
... /home/org/mhpkdb/piloto/res/drip1_0.gif
... /home/org/mhpkdb/piloto/res/construccion.gif
... /home/org/mhpkdb/piloto/res/back2.mpg
... /home/org/mhpkdb/piloto/Pregunta.class
... /home/org/mhpkdb/piloto/PlantillaEditv.class
```

```

... /home/org/mhpkdb/codeframework/template/KDBTemplate.class
... /home/org/mhpkdb/codeframework/template/KDBTemplate$1.class
... /home/org/mhpkdb/codeframework/template/KDBLauncher.class
... /home/org/mhpkdb/codeframework/template/KDBLauncher$3.class
... /home/org/mhpkdb/codeframework/template/KDBLauncher$2.class
... /home/org/mhpkdb/codeframework/template/KDBLauncher$1.class
... /home/org/mhpkdb/codeframework/template/KDBExampleRunner.class
... /home/org/mhpkdb/codeframework/components/MyResourceClient.class
... /home/org/mhpkdb/codeframework/components/MyBackgroundImageListener.class
... /home/org/mhpkdb/codeframework/components/KDBTextField.class
... /home/org/mhpkdb/codeframework/components/KDBTextField$1.class
... /home/org/mhpkdb/codeframework/components/KDBNumberField.class
... /home/org/mhpkdb/codeframework/components/KDBNumberField$1.class
... /home/org/mhpkdb/codeframework/components/KDBNavigable.class
... /home/org/mhpkdb/codeframework/components/KDBMensaje.class
... /home/org/mhpkdb/codeframework/components/KDBMensaje$1.class
... /home/org/mhpkdb/codeframework/components/KDBLook.class
... /home/org/mhpkdb/codeframework/components/KDBList.class
... /home/org/mhpkdb/codeframework/components/KDBList$1.class
... /home/org/mhpkdb/codeframework/components/KDBLabel.class
... /home/org/mhpkdb/codeframework/components/KDBLabel$1.class
... /home/org/mhpkdb/codeframework/components/KDBImage.class
... /home/org/mhpkdb/codeframework/components/KDBImage$1.class
... /home/org/mhpkdb/codeframework/components/KDBComponent.class
... /home/org/mhpkdb/codeframework/components/KDBComponent$1.class
... /home/org/mhpkdb/codeframework/components/KDBButton.class
... /home/org/mhpkdb/codeframework/components/KDBButton$1.class
uploading xlets description...

C:\stb tools\stb upload>pause
Press any key to continue . . .

```

Figura B.7. Progreso de la descarga de archivos

Puede observarse el registro de ejecución en el archivo stb.log para realizar corrección de errores y poner a punto la aplicación, ver Figura B.8 .

```

Tail for Win32 - [Non-Workspace Files - C:\stb.log]
File Edit View Settings Window Help

10 ms PNGDecoder: Starting decoding PNGDecoder took 10 ms
PNGDecoder: Starting decoding PNGDecoder took 9 ms PNGDecoder: Starting decoding PNGDecoder took 11 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 10 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 11 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 12 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 11 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 9 ms
PNGDecoder: Starting decoding PNGDecoder took 10 ms PNGDecoder: Starting decoding PNGDecoder took 10 ms
[1] System country code = esp DRV_DEVICE_MODEM set country to 10 [1] PreferredLang for PID = -1, lang: eng

System_DrvAV2_SetBackgroundImage(image=0x41490cc0, size=6332) [2] java.lang.NullPointerException [2]
at java.util.Hashtable.put (Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object;(Unknown Source) [2]
at com.adb.kdb.server.settings.changed(I)V(Unknown Source) [2]
at com.adb.ag.properties.SettingsConverter.saveSettings(Lcom/adb/ag/properties/PropertiesManager;Ljava/lang/String;I)V(Unknown Source) [2]
at com.adb.ag.properties.PropertiesManager.run(I)V(Unknown Source) [2]
at java.lang.Thread.startup(Z)V(Unknown Source)
System_DrvAV2_SetBackgroundImage(image=0x0, size=0)

[1] Registering Pincode Dialog

PFR: /system/fonts/Tires-o_802.pfr 21 ([0] Tiresias-Regular, 350 glyphs)

PFR: /system/fonts/Tires-o_802.pfr 18 ([0] Tiresias-Regular, 350 glyphs) [1] adbSIDatabase, notification service selected onid = -1, tsid = -1, svid = -1 SVC CONNECT DATA 4153dd00 STOPWATCH: MHP time profile - System ready: 21380 ms (14152 ms)

Demod status = LOCK OK

[1] FocusHistory: total len = 0

APPMAN:ait: new service: tsid:0x1 onid:0x1 svid:0x1 no app [1] [AG_CORE]:
PinCodeMessageGroup.validatePinCode() error: null [1] [AG_CORE]: !!! pin validated? = true [1]
AdbSIDatabase, notification service selected onid = -1, tsid = -1, svid = -1 [1] >>>
getServiceContextEvent returned com.adb.tv.service.selection.XSelectionFailedEventImpl [source=com.adb.tv.service.selection.BroadcastServiceContextImpl@807a2d]

Last updated: 10:15:29

```

Figura B.8. Registro de ejecución

B.1.3 Simulación de la aplicación en mimundoTV Player

La aplicación también puede ser ejecutada en el computador a través de un emulador llamado mimundoTV Player. Este emulador es una aplicación comercial que permite ser probada durante 30 días, tiempo después del cual se deshabilita. El precio de su licencia es muy económico lo cual lo hace una herramienta imprescindible para cualquiera que quiera desarrollar aplicaciones de iTV. La versión aquí trabajada es la 1.03.

Para ejecutar una aplicación en mimundoTV Player se realizan los siguientes pasos (mimundo.tv, 2007):

1. Crear una carpeta en la ruta "C:\Archivos de Programa\mimundoTV Player DEMO\windows\mhpri\runtime\fileio\dsmcc\" con el nombre 0.0.X donde X= número en hexadecimal que no esté ya presente, en la figura de ejemplo la carpeta 0.0.7, ver Figura B.9.

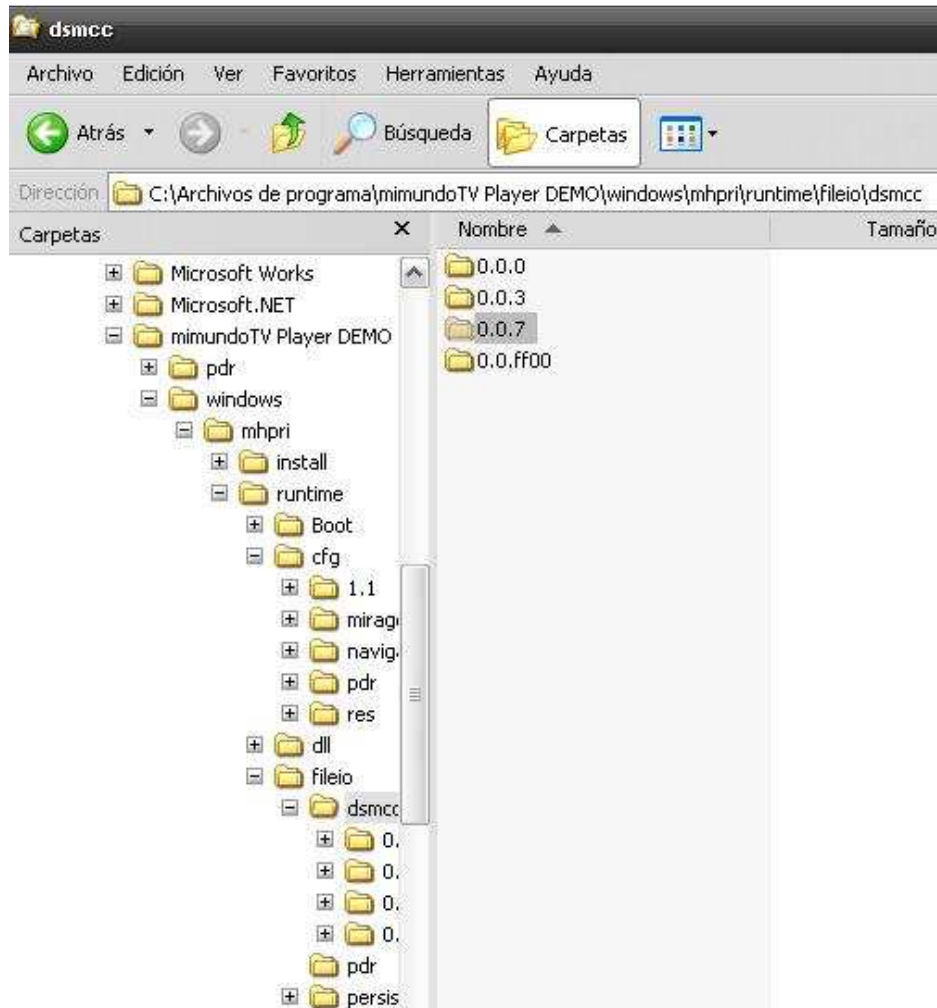


Figura B.9. Ejecución de una aplicación en mimundoTV Player – Paso uno

2. Copiar las clases compiladas de la aplicación a emular en la carpeta creada con la ruta completa del paquete al que pertenecen.

En el ejemplo las clases se encuentran dentro del paquete org\mhpkdb\ ver Figura B.10.

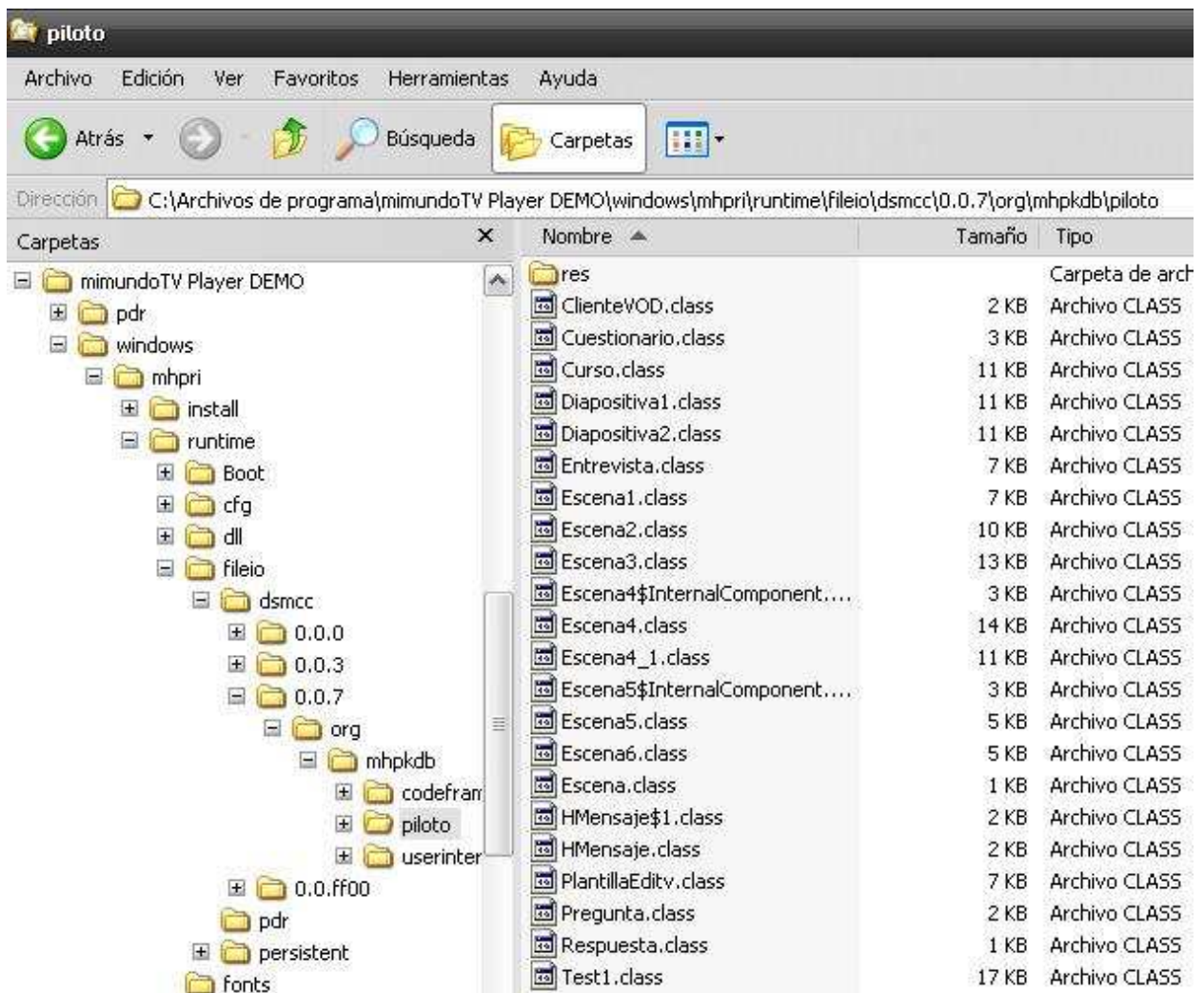


Figura B.10. Ejecución de una aplicación en mimundoTV Player – Paso dos

3. Crear una nueva línea en el archivo “C:\Archivos de programa\mimundoTV Player DEMO\windows\mhpri\runtime\cfg\Applications.cfg”, ver Figura B.11, con la siguiente información:

Nombre_de_la_clase;Id_aplicacion;Id_organizacion = Nombre_de_la_aplicacion;Localizador

En el ejemplo:

Nombre_de_la_clase = "org.mhpkdb.piloto.Curso" (incluye paquetes)
 Id_aplicacion = "192" (este debe ser irrepitable)
 Id_organizacion = "9" (identificador único)
 Nombre_de_la_aplicacion = "TEST: Curso"
 Localizador = "dvb://0.0.7" (0.0.7 debe coincidir con la carpeta creada)

```

Applications.cfg - Bloc de notas
Archivo Edición Formato Ver Ayuda
# This file contains the list of built-in applications that are shown when
# the MHP RI is started. These applications may reside on the local file
# system or on the network (transport stream, internet, etc.).
#
# The format is as follows (entries in brackets [] are optional):
# <class name>[;<app id>[;<org id>[;<visibility>]]] = <application name>[;<service locator>[;<application
# locators>[;<application args>]]]
#
# the <app id> / <org id>, when not specified, is 1 / 1
# the <visibility>, when not specified, is "public"
# "public" means visible to user and other applications
# "protected" means visible to other applications only
# "private" means not visible at all
# the <service locator>, when not specified, is dvb://0.0.0
# the <application locators>, when not specified, is <service locator>
# the format for the <application locators> is plain URLs separated by spaces
# the format for the <application args> is a set of strings separated by space;
# if you need to enclose a space in an application argument escape it by a "\"
# if you need to enclose a newline in an application argument write "\n"
# all digits are hexadecimal (e.g. app id, org id, locator parts)

##### TEST Applications

#de.irt.testing.havi.HSoundXlet;102;9 = TEST: HSound;dvb://0.0.3
#de.irt.testing.ixc.IxcRemoteXlet;135;9 = TEST: Ixc Remote;dvb://0.0.3
#de.irt.testing.ixc.IxcInvokationTestXlet;136;9 = TEST: Ixc Test;dvb://0.0.3
#de.irt.testing.jmf.MpegAudioPlayerXlet;142;9 = TEST: MPEG Audio Player;dvb://0.0.3
#de.irt.testing.jmf.Mpeg4PlayerXlet;143;9 = TEST: MPEG4 Player;dvb://0.0.3
#de.irt.testing.jmf.WMPPlayerDemoXlet;144;9 = TEST: WMPPlayer;dvb://0.0.3
#de.irt.testing.havi.uiter.UITester;171;9 = TEST: HAVi Widget Tester;dvb://0.0.3
#de.irt.testing.net.FileAccessXlet;180;9 = TEST: File Access;dvb://0.0.3
#de.irt.testing.net.HttpTestXlet;181;9 = TEST: HTTP Test;dvb://0.0.3;http://www.irt.de
#de.irt.testing.net.HttpsTestXlet;182;9 = TEST: HTTPS Test;dvb://0.0.3;https://meine.deutsche-bank.de
#de.irt.testing.returnchannel.RCTest;185;9 = TEST: Return Channel;dvb://0.0.3
#de.irt.testing.service.ListInstalledServiceXlet;190;9 = TEST: List Installed Services;dvb://0.0.3
#de.irt.testing.service.ListCurrentServiceAppsXlet;191;9 = TEST: List Current Service Apps;dvb://0.0.3
org.mhp.kdb.piloto.Cursor;192;9 = TEST: Cursor;dvb://0.0.7
    
```

Figura B.11. Ejecución de una aplicación en mimundoTV Player – Paso tres

4. Ejecutar el emulador ver Figura B.12, corriendo el archivo mhp.bat de la ruta “C:\Archivos de Programa\mimundoTV Player DEMO\windows\mhpr\runtime”.

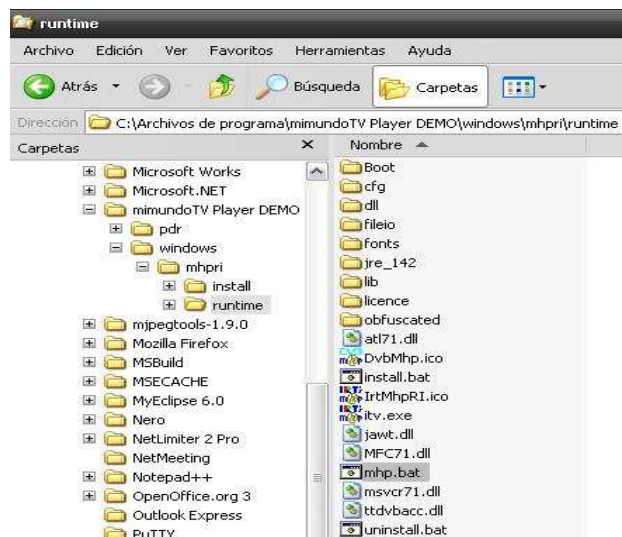


Figura B.12. Ejecución de una aplicación en mimundoTV Player – Paso cuatro

5. Ir al menú My Application de mimundoTV Player, ver Figura B.13.



Figura B.13. Ejecución de una aplicación en mimundoTV Player – Paso cinco

6. Ejecutar la aplicación con el nombre colocado en el archivo Applications.cfg, ver Figura B.14

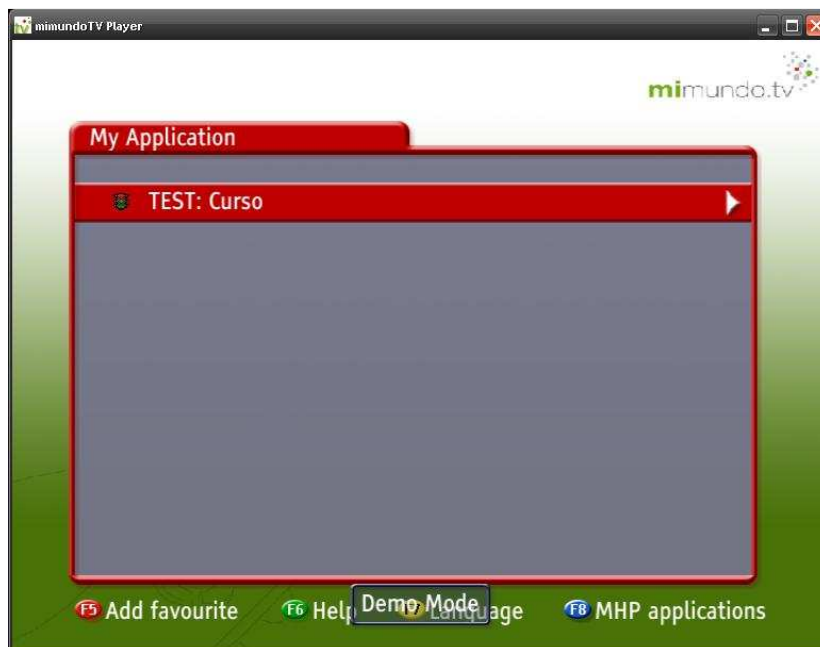


Figura B.14. Ejecución de una aplicación en mimundoTV Player – Paso seis

7. Se puede realizar debug de la aplicación a través de la consola DOS de mimundoTV, Figura B.15.



Figura B.15. Ejecución de una aplicación en mimundoTV Player – Paso siete

B.2 Ejemplos de generación y distribución de contenidos

En este apartado se darán mini-tutoriales acerca de cómo generar diferentes tipos de contenidos con herramientas de libre distribución y algunas comerciales, todos estos ejemplos han sido realizados en base a la experiencia obtenida al realizar los contenidos del proyecto.

B.2.1 Codificación de audio y video con FFMPEG

Un codificador MPEG2 Open Source se encuentra dentro de la librería libavcodec del paquete ffmpeg, con el cual se puede codificar un flujo de video elemental de tipo CBR (Tasa de bits constante). El mismo paquete ofrece un codificador MPEG1 para el manejo de flujos elementales de audio (FFMPEG, 2007).

B.2.1.1 Codificación de Video

La codificación del video con ffmpeg se muestra a continuación:

```

ffmpeg -i archivo_entrada -an -vcodec mpeg2video -f mpeg2video -b
tasa_de_bits -maxrate tasa_de_bits -minrate tasa_de_bits -bf
numero_de_frames_b -bufsize 224 video.mp2
  
```

- -i: archivo_entrada es un video en cualquier formato que se encuentre soportado por el decodificador del paquete ffmpeg.
- -an: indica que el audio será ignorado para la codificación.
- -vcodec: representa el codec de video a ser usado, en este caso mpeg2video representa la codificación de video MPEG2
- -f: representa el formato de salida del archivo de video, en este caso mpeg2video representa un flujo elemental de video MPEG2

- -b, -maxrate, -minrate tasa_de_bits: representa la tasa de bits que tendrá el video codificado en unidades de bits/segundo que dependen de la versión de ffmpeg. La tasa_de_bits debe ser igual en los tres parámetros para asegurar que la codificación sea con tasa de bits constante. Los valores más comunes van desde 2000 kbps a 10000 kbps.
- -bf: número de frames b a usar. Entre mayor sea su número se obtendrá un video con mayor calidad.
- -bufsize 224: es el máximo buffer de video soportado por los decodificadores MPEG2 de los STB

B.2.1.2 Codificación del Audio

La codificación del audio se muestra a continuación:

```
ffmpeg -i archivo_de_entrada -ac 2 -vn -acodec mp2 -f mp2 -ab tasa_de_bits audio.mp2
```

- -i: archivo_de_entrada, cualquier formato de audio soportado por ffmpeg.
- -ac 2: opción para elegir canales estéreo.
- -vn: indica que solo codifique el audio, no el video.
- -acodec mp2: para escoger el codec de audio MPEG2.
- -f: para escoger como formato de salida del archivo un flujo elemental MPEG2.
- -ab tasa_de_bits: que puede estar en kbps o bps dependiendo de la versión de ffmpeg. Los valores más comunes son: 128, 196, 224 kbps.

B.2.2 Codificación de audio y video con Mainconcept Reference

Este programa posee un codificador con múltiples perfiles para codificar audio y video en prácticamente cualquier formato. Entre sus perfiles contiene uno específico para generar flujos de transporte DVB MPEG-2 con múltiples parámetros configurables. Aunque los buenos codificadores son costosos, este programa es muy económico y vale la pena tenerlo entre las herramientas para televisión digital. El proceso a seguir mediante el Mainconcept Reference versión 1.1.1 es el siguiente (Main_concept, 2009):

- En MainConcept Reference cargar un archivo de video a través de la opción File -> Open Media.
1. Cargar el video a ser codificado. Seleccionarlo desde el directorio donde ha sido almacenado.

La herramienta despliega tres ventanas. En la ventana Input se muestran los flujos elementales que contiene el video, así como los parámetros con los que fueron inicialmente codificados. En la ventana Preview se puede ver el video cargado, una barra para moverse a través de la secuencia de frames del video y unos botones con los que se puede seleccionar partes del video a codificar. En la ventana Output es donde se configurarán los parámetros del video de salida.
 2. Se selecciona el perfil del video a usar en la ventana Output, para este trabajo DVB/ATSC Digital TV. Ver Figura B.16

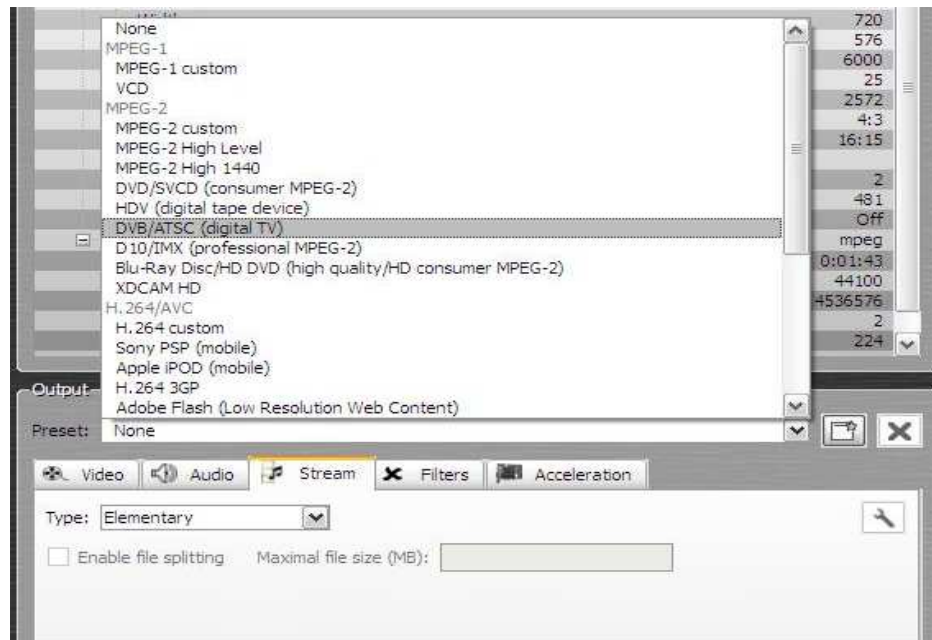


Figura B.16. Perfil de video

3. Se configuran los parámetros principales del video en la pestaña Video, los parámetros básico son: formato PAL en 16:9 con una tasa de bits de 6000 kbps, ver Figura B.17.

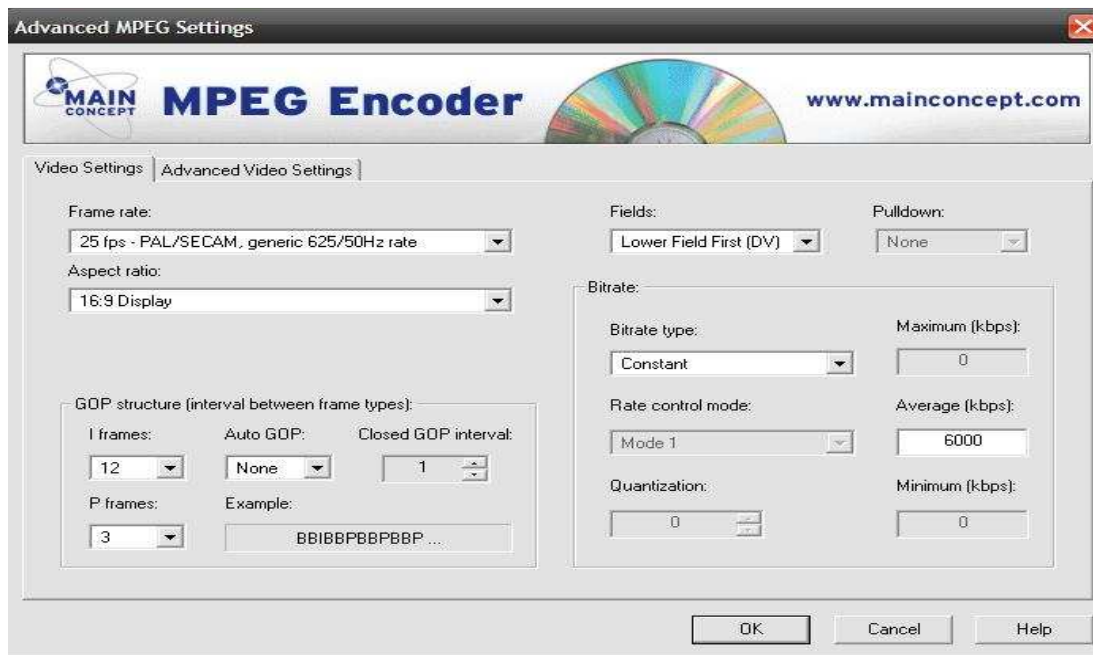


Figura B.17. Parámetros de vídeo

- El siguiente paso es configurar los parámetros avanzados del video mediante la la pestaña Advanced Video Settings ver Figura B.18

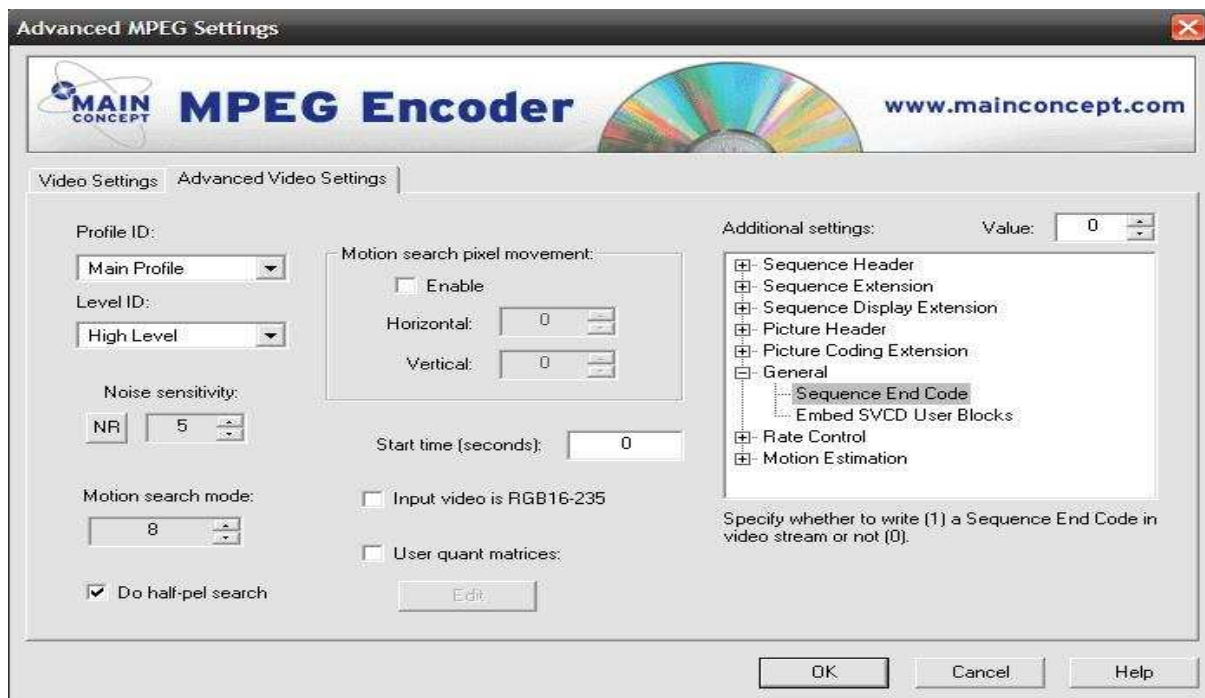


Figura B.18. Parámetros avanzados de Video

4. Se debe configurar los parámetros de Audio, sobre la ventana Output, en la pestaña Audio, los parámetros seleccionados para la aplicación son:
 - ✓ Codec: MPEG Audio,
 - ✓ Channels: 2,
 - ✓ Bitrate: 224 kbps,
 - ✓ Samplerate (frecuencia de Muestreo): 44100 Hz.
5. Se configura el Mux donde le parámetro Mux rate se calcula como al menos un 15 % más que la tasa de bits usada en la codificación del video. En la pestaña Stream se seleccionan las opciones avanzadas con el botón de la llave y se configuran como en la Figura B.19.

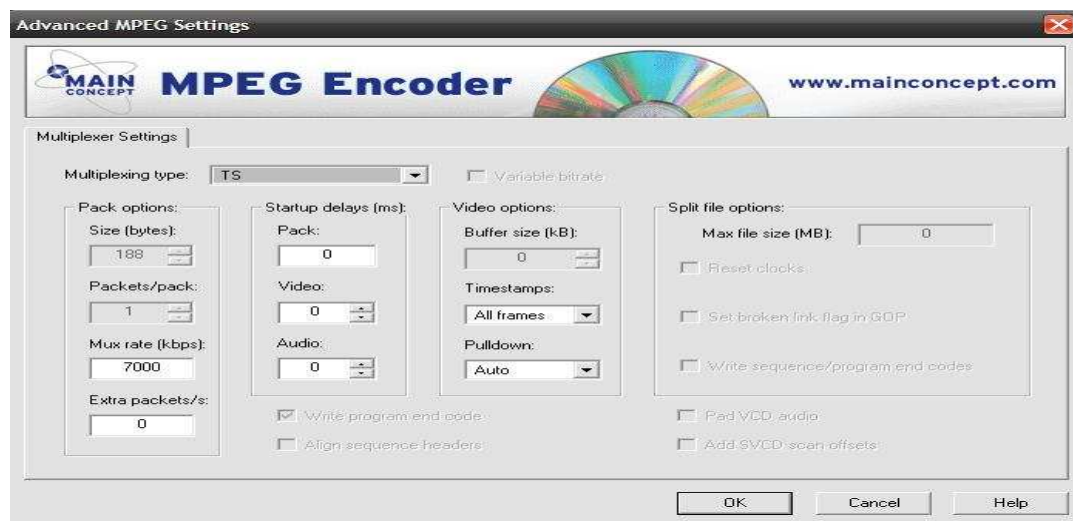


Figura B.19. Parámetros de Multiplexación

6. Por último se selecciona la ruta donde se creará el archivo de salida. Con el botón en forma de engranaje y se comienza la generación del flujo de transporte con el botón triangular. Ver Figura B.20

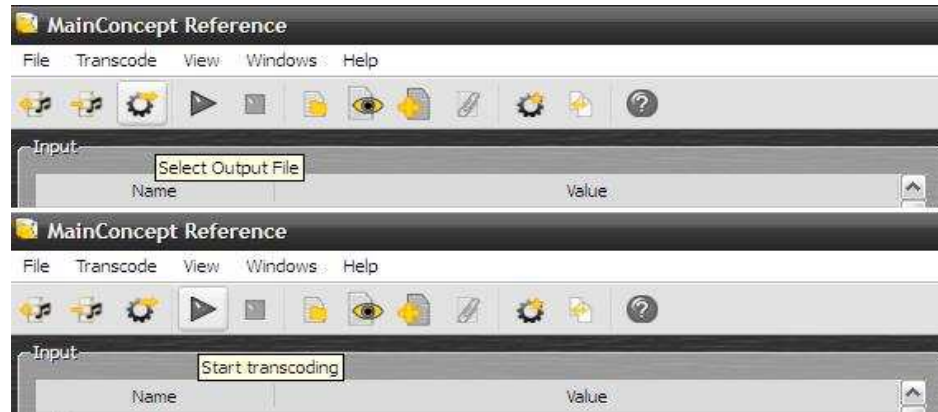


Figura B.20. Generación del flujo de Transporte

7. El archivo generado es un flujo de transporte TS que contiene información de PMT¹ (Program Map table). Los PID de los flujos de audio y video ya se encuentran fijos, por lo que es necesario utilizar dos herramientas de OpenCaster para modificarlos.
 - a. Tsloop: permite ver los identificadores de los flujos presentes en el TS.
 - b. Tspidmapper: permite cambiar un identificador por otro, en este caso mapeando todos los identificadores no deseados al identificador 8191 o nulo como se muestra a continuación;

```
tsloop archivots > /dev/null
tspidmapper archivo.ts 0 to 8191 and PMT_PID to 8191 > nuevo_archivo.ts
```

B.2.3 Conversión de imágenes en Videodrips

La conversión de imágenes en videodrips se puede realizar en Windows o Linux a través de dos paquetes: Netpbm (Netpbm, 2008) y MJPEG Tools (MJPEGTools, 2005).

Netpbm es un conjunto de utilidades para la manipulación de imágenes, incluye conversión de imágenes entre diferentes formatos. Netpbm usa un formato intermedio para la transformación de imágenes llamado PPM/PNM.

MJPEG Tools es un conjunto de herramientas que permiten realizar grabación y reproducción de videos, edición simple y compresión de audio/video en formato MPEG.

Para realizar la transformación de una imagen al formato PPM/PNM se debe primero conocer el formato de la imagen, ya que para cada uno existe una herramienta especial de conversión:

- JPEG usa la herramienta jpegtopnm
- TIFF usa la herramienta tifftopnm
- BMP usa la herramienta bmptopnm
- PNG usa la herramienta pngtopnm
- GIF usa la herramienta giftopnm

En el caso de una imagen PNG el comando a usar es:

```
pngtopnm imagen.png > imagen_convertida.pnm
```

¹ PMT: Existe una PMT por cada programa presente en el TS. En ella se da información sobre todos los elementary streams asociados a un programa, de tal forma que el receptor es capaz de localizarlos y decodificarlos

Una vez se tiene la imagen en este formato pueden usarse diferentes herramientas para adaptar la imagen al formato NTSC o PAL adecuado. Pnmscale (Netpbm, 2008) permite cambiar el tamaño de la imagen a la escala deseada. Pnmpad (pnmpad, 2009) añade bordes a la imagen para adaptarla al tamaño deseado. Ppmntsc modifica los colores de la imagen a la paleta permitida en NTSC o PAL. Ppmttoy4m convierte la imagen PPM/PNM en formato YUV4MPEG2 de video.

Para la imagen de ejemplo la cadena de comandos es:

```
pnmscale factor_de_escala imagen_convetida.pnm | \  
pnmpad -black -width ancho -height alto | \  
ppmntsc --pal | \  
ppmttoy4m -n numero_de_frames -F tasa_de_frames | \  

```

La salida de este comando es un frame o una secuencia de frames de video listos para ser convertidos al formato MPEG2 a través de la herramienta mpeg2enc del paquete MJPEG Tools por el comando:

```
mpeg2enc -f perfil_mpeg -T tamaño_archivo_deseado -F tasa_de_frames --  
video-norm -a tasa_de-aspecto -o archivo.m2v
```

Para optimizar el proceso de conversión se ha creado un script para Linux que analiza el archivo para obtener automáticamente algunos parámetros requeridos por los comandos anteriores y permite configurar otros a opción del usuario. El script se llama conversor.sh y se ejecuta de la siguiente forma:

```
conversor.sh archivo_entrada archivo_salida
```

Donde el archivo de entrada puede ser por ejemplo imagen.jpg y el archivo de salida drip1.m2v. El script puede verse a continuación:

```

#!/bin/bash

# Formato del video. pal o ntsc
FORMATO=pal
# Ancho/Alto de la imagen (teniendo en cuenta el area visible de la pantalla)
if [ "$FORMATO" = "ntsc" ]; then
    TW=704
    TH=420
else
    TW=704
    TH=576
fi
TMP=/tmp/imagen_convertida.$$ .pmn
IMG=$1
MPG=$2
DIR=`carpeta "$MPG"`
if [ ! -d "$DIR" ]; then
    mkdir -p "$DIR"
fi

# Se obtiene el tipo de archivo para seleccionar el conversor adecuado
TIPO_ARCHIVO=`file -i -L -b "$IMG" 2>/dev/null | cut -f2 -d/`
case "$TIPO_ARCHIVO" in
    jpg | jpeg)
        A_PNM=jpegtopnm
        ;;
    tiff)
        A_PNM=tifftopnm
        ;;
    bmp | x-bmp)
        A_PNM=bmptoppm
        ;;
    png | x-png)
        A_PNM=pngtopnm
        ;;
    Netpbm | pnm | x-portable-pixmap)
        A_PNM=cat
        ;;
    gif)
        A_PNM=giftopnm
        ;;
    *)
        echo "el tipo de archivo '$TIPO_ARCHIVO' no está soportado"
        exit 1
        ;;
esac

# Se extrae el tamaño de la imagen y se calcula el valor de escala
LANG=C # para obtener el punto decimal correctamente
$A_PNM "$IMG" >$TMP 2>/dev/null
S=`pnmfile $TMP | awk '{ printf "%d %d ",$4,$6 }'`
S=`echo $S $TW $TH | awk '{ sw=$3/$1; sh=$4/$2; s=(sw<sh)?sw:sh; printf "%.4f\n", (s>1)?1.0:s; }'`

# Se realiza la conversion
if [ "$FORMATO" = "ntsc" ]; then
    pnmscale $S $TMP | \
    pnmppad -black -width 704 -height 480 | \
    ppmntsc | \
    ppmtty4m -v 0 -n 1 -r -F 30000:1001 | \
    mpeg2enc -f 7 -T 200 -F 4 -nn -a 2 -v 0 -o "$MPG"
else
    pnmscale $S $TMP | \
    pnmppad -black -width 704 -height 576 | \
    ppmntsc --pal | \
    ppmtty4m -v 0 -n 1 -r -F 25:1 | \
    mpeg2enc -f 7 -T 200 -F 3 -np -a 2 -v 0 -o "$MPG"
fi

```

B.2.4 Creación de tablas de información de servicio

El paquete OpenCaster (Avalpa, 2008) puede realizar la generación de un flujo de transporte para las tablas de Información de Programa y Servicio PSI (la descripción de las tablas se encuentra en el anexo A) a través de la creación de un archivo en Python que contiene los parámetros requeridos por cada tabla según las especificaciones del consorcio DVB. Para cada tabla se tienen en cuenta las siguientes variables globales:

```

editv_transport_stream_id = 1 # valor demo
editv_original_transport_stream_id = 1 # valor demo
editv_service_id = 1 # valor demo
editv_pmt_pid = 1031
ait1_pid = 2001
dsmccB_association_tag = 0xB
dsmccB_carousel_id = 1
ste1_pid = 2002
ste2_pid = 2004

```

En las siguientes imágenes se muestra la configuración para cada tabla del Curso Piloto, descrito en el Capítulo 2

```

#
# Network Information Table (Tabla de Información de Red)
#
nit = network_information_section(
    network_id = 1,
    network_descriptor_loop = [
        network_descriptor(network_name = "EDiTV",),
    ],
    transport_stream_loop = [
        transport_stream_loop_item(
            transport_stream_id = editv_transport_stream_id,
            original_network_id = editv_original_network_id,
            transport_descriptor_loop = [
                service_list_descriptor(
                    dvb_service_descriptor_loop = [
                        service_descriptor_loop_item(
                            service_ID = avalpa1_service_id,
                            service_type = 1, # Servicio de TV digital
                        ),
                    ],
                ),
            ],
        ),
    ],
    version_number = 1, # numero de version a cambiar cada vez que se actualiza la tabla
    section_number = 0,
    last_section_number = 0,
)

```

Figura B.21. Tabla de Información de Red

```

#
# Program Association Table (Tabla de Asociacion de Programas)
#
pat = program_association_section(
    transport_stream_id = editv_transport_stream_id,
    program_loop = [
        program_loop_item(
            program_number = editv_service_id, # programa para EDiTV
            PID = editv_pmt_pid,
        ),
        program_loop_item(
            program_number = 0, # programa especial para la tabla NIT
            PID = 16,
        ),
    ],
    version_number = 1, # numero de version a cambiar cada vez que se actualiza la tabla
    section_number = 0,
    last_section_number = 0,
)

```

Figura B.22. Tabla de asociación de programas

```

#
# Service Description Table (Tabla de Descripcion de Servicios)
#
sdt = service_description_section(
    transport_stream_id = editv_transport_stream_id,
    original_network_id = editv_original_network_id,
    service_loop = [
        service_loop_item(
            service_ID = editv_service_id,
            EIT_schedule_flag = 0, # 0 ninguna informacion de eventos es difundida, 1 difundida
            EIT_present_following_flag = 0, # 0 no es difundida informacion de eventos siguientes, 1 es difundida
            running_status = 4, # 4 el servicio se esta ejecutando, 1 no se esta ejecutando, 2 arranque en unos segundos, 3 pausado
            free_CA_mode = 0, # 0 el servicio no esta encriptado, 1 al menos un flujo esta encriptado
            service_descriptor_loop = [
                service_descriptor(
                    service_type = 1, # servicio de television digital
                    service_provider_name = "EDiTV",
                    service_name = "EDiTV 1",
                ),
            ],
        ),
    ],
    version_number = 1, # numero de version a cambiar cada vez que se actualiza la tabla
    section_number = 0,
    last_section_number = 0,
)

```

Figura B.23. Tabla de descripción de Servicios

```

#
# Tablas de Descripción de Eventos DSMCC
#
ste = stream_event_section(
    event_id = 1,
    stream_event_descriptor_loop = [
        stream_event_do_it_now_descriptor(
            event_id = 1,
            private_data = "evento 1",
        ),
    ],
    version_number = 1,
    section_number = 0,
    last_section_number = 0,
)

ste2 = stream_event_section(
    event_id = 2,
    stream_event_descriptor_loop = [
        stream_event_do_it_now_descriptor(
            event_id = 2,
            private_data = "evento 2",
        ),
    ],
    version_number = 1,
    section_number = 0,
    last_section_number = 0,
)

```

Figura B.24: Taba de descripción de Eventos DSMCC

```

#
# Program Map Table (Tabla de Mapeo de Programas)
#
pmt = program_map_section(
    program_number = editv_service_id,
    PCR_PID = 481,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # tipo de flujo de video mpeg2
            elementary_PID = 481,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 3, # tipo de flujo de audio mpeg2
            elementary_PID = 482,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 2, # tipo de flujo de video mpeg2
            elementary_PID = 2065,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 3, # tipo de flujo de audio mpeg2
            elementary_PID = 2069,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5, # tipo de flujo de aplicacion AIT
            elementary_PID = ait1_pid,
            element_info_descriptor_loop = [
                application_signalling_descriptor(
                    application_type = 1, # 1 aplicacion DVB-J, 2 aplicacion DVB-HTML
                    AIT_version = 1,
                ),
            ],
        ),
    ],
)

```

Figura B.25. Tabla de Mapeo de Programas


```

stream_loop_item(
    stream_type = 12, # tipo de flujo de eventos
    elementary_PID = ste1_pid,
    element_info_descriptor_loop = [
        stream_identifier_descriptor(
            component_tag = 0xD,
        ),
    ],
),
stream_loop_item(
    stream_type = 11, # tipo de flujo DSMCC
    elementary_PID = 2003,
    element_info_descriptor_loop = [
        association_tag_descriptor(
            association_tag = dsmccB_association_tag, # etiqueta de asociacion que referencia el carrusel
            use = 0,
            selector_lenght = 0,
            transaction_id = 0x80000000,
            timeout = 0xFFFFFFFF,
            private_data = "",
        ),
        stream_identifier_descriptor(
            component_tag = dsmccB_association_tag, # la misma etiqueta de asociacion
        ),
        carousel_identifier_descriptor(
            carousel_ID = dsmccB_carousel_id, # identificador del carrusel
            format_ID = 0,
            private_data = "",
        ),
        data_broadcast_id_descriptor(
            data_broadcast_ID = 240, # codigo que especifica que es un carrusel de objetos MHP
            ID_selector_bytes = "",
        ),
    ],
),
],
),
],
version_number = 1, # numero de version a cambiar cada vez que se actualiza la tabla
section_number = 0,
last_section_number = 0,

```

Figura B.26. Continuación tabla de mapeo de Programas

```

#
# Application Informaton Table (Tabla de Informacion de Aplicaciones)
#
ait = application_information_section(
    application_type = DVB_J_application_type,
    common_descriptor_loop = [],
    application_loop = [
        application_loop_item(
            organisation_id = 10, # valor demo, dvb.org asigna valores unicos
            application_id = 1001, # identificador de las aplicaciones, unico para cada organizacion dentro de un programa
            application_control_code = 2, # 2 esta PRESENTE, el decodificador añade la aplicacion pero la ejecuta el usuario
            # 1 AUTOARRANQUE, la aplicacion se ejecuta inmediatamente con el servicio
            # 3 DESTRUIR, le envia señal de peticion de parada a la aplicacion si se esta ejecutando
            # 4 MATAR, para la aplicacion
            application_descriptors_loop = [
                transport_protocol_descriptor(
                    protocol_id = MHP_OC_protocol_id, # la aplicacion es difundida en un carrusel MHP
                    transport_protocol_label = 1, # identificador de carrusel
                    remote_connection = 0,
                    component_tag = 0xB, # etiqueta de asociacion del carrusel
                ),
                application_descriptor(
                    application_profile = 0x0001, # Perfil y version MHP de la aplicacion (MHP 1.0.2)
                    version_mayor = 1,
                    version_minor = 0,
                    version_micro = 2,
                    service_bound_flag = 1, # 1 significa que la aplicacion muere en cambios de servicio, 0 espera estar señalizada en el nuevo servicio
                    visibility = 3, # 3 la aplicacion es visible al usuario, 1 la aplicacion es visible solo a otras aplicaciones
                    application_priority = 1, # 1 es baja, usada cuando se esta ejecutando mas de una aplicacion
                    transport_protocol_labels = [1], # identificador del carrusel
                ),
                application_name_descriptor(application_name = "Curso"),
                dvb_j_application_descriptor(parameters = [""]), # parametros pasados al Xlet
                dvb_j_application_location_descriptor(
                    base_directory = "/", # directorio base
                    class_path_extension = "", # directorios adicionales de clases dentro del carrusel
                    initial_class = "piloto.Curso", # la clase que implementa la interfaz Xlet
                ),
            ],
        ),
    ],
),

```

Figura B.27. Tabla de Información de aplicación

B.2.5 Creación de eventos DSMCC

Los eventos DSMCC no solo deben señalizarse a través de las tablas PSI, además deben ser registrados por las aplicaciones que van a recibirlos a través del carrusel de objetos. Para esto se deben añadir unos archivos especiales al carrusel, que son configurados a través del código de la Figura B.28:

```
import os
import string
from dvbobjects.DSMCC.BIOP.Tap import *
from dvbobjects.DSMCC.STEO import *
from dvbobjects.utils import *

tap = str_event_use_tap0
tap.set(
    id = 0,
    assocTag = 0xD, # Etiqueta de asociacion definida en la tabla PMT para el flujo de eventos
)

taps = Taps (
    taps_count = 1,
    tap_loop = [ tap, ],
)

event_count = 3 # Numero de eventos

event_names = Event_names (
    eventnames_count = event_count,
    event_name_loop = [ "event 1", "event 2", "event 3"], # Nombre de los eventos
)

event_ids = Event_ids (
    eventids_count = event_count,
    event_id_loop = [ 1, 2, 3, ], # Identificador de los eventos
)

out = open(".tap", "wb")
out.write(taps.pack0)
out.close
out = open(".eid", "wb")
out.write(event_ids.pack0)
out.close
out = open(".ename", "wb")
out.write(event_names.pack0)
out.close
```

Figura B.28. Configuración de archivos especiales del carrusel

En el ejemplo de la Figura B.28 se crean tres archivos: .tap .eid y .ename . Estos contienen la información de 3 eventos denominados event 1, event 2 y event 3, que tienen los identificadores 1, 2 y 3 respectivamente. Estos archivos deben estar presentes en una subfichero dentro del fichero que va a ser usada como carrusel de objetos. Como se muestra en la Figura B.29.

```
editv@editvserver: ~/piloto/oc
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ cd occdir1/
editv@editvserver:~/piloto/occdir1$ ls
org piloto
editv@editvserver:~/piloto/occdir1$ mkdir test.event
editv@editvserver:~/piloto/occdir1$ cd test.event
editv@editvserver:~/piloto/occdir1/test.event$ cp ../../.
./      ./      .eid   .ename .tap
editv@editvserver:~/piloto/occdir1/test.event$ cp ../../.eid .
editv@editvserver:~/piloto/occdir1/test.event$ cp ../../.ename .
editv@editvserver:~/piloto/occdir1/test.event$ cp ../../.tap .
editv@editvserver:~/piloto/occdir1/test.event$ ls -la
total 20
drwxr-xr-x 2 editv editv 4096 2009-02-27 10:30 .
drwxr-xr-x 5 editv editv 4096 2009-02-27 10:30 ..
-rw-r--r-- 1 editv editv  7 2009-02-27 10:30 .eid
-rw-r--r-- 1 editv editv 29 2009-02-27 10:30 .ename
-rw-r--r-- 1 editv editv  8 2009-02-27 10:30 .tap
editv@editvserver:~/piloto/occdir1/test.event$
```

Figura B.29. Sistema de ficheros del carrusel de objetos

Para el ejemplo del Curso Piloto el fichero cdir1 contiene los archivos del carrusel de objetos, dentro de esta se crea un subfichero denominado test.event y dentro de esta son copiados los tres archivos anteriormente generados.

De esta forma una aplicación accede a través de las librerías de MHP a la información contenida en estos archivos y así poder registrarse a un evento en particular.

B.2.6 Generación de flujos de transporte

Dependiendo del tipo de flujo de transporte, existe una herramienta del paquete OpenCaster (Avalpa, 2008) adecuada para su generación.

B.2.6.1 Creación de un flujo de transporte de video

A partir de un flujo elemental de video se debe construir primero un flujo de programa elemental PES (para más detalles ver anexo A), a través de la herramienta esvideo2pes.

```
esvideo2pes video.mp2 > video.pes
```

Un PES de video puede ser empaquetado en un flujo de transporte TS con la herramienta pesvideo2ts, de la siguiente manera:

```
pesvideo2ts video.pes identificador_de_programa frames_por_segundo
buffer_video tasa_bits_por_segundo > video.ts
```

- `identificador_de_programa`: es un número único que identifica al flujo de video. Un valor típico es 2064.
- `frames_por_segundo` del video: dependiendo del estándar de video 25 para PAL, 30 para NTSC.
- `buffer_video`: va de 112 a 224 dependiendo de la capacidad de decodificación del STB.
- `tasa_bits_por_segundo` del flujo de transporte: este valor debe ser mayor a la tasa de bits con que se codificó el video elemental. Aproximadamente un 15% mayor.

B.2.6.2 Creación de un flujo de transporte de audio

- A partir de un flujo elemental de audio se debe construir primero un flujo de programa elemental PES, a través de la herramienta esaudio2pes.

```
esaudio2pes audio.mp2 tasa_de_muestreo tamaño_frame_audio
tamaño_primer_frame_video > audio.pes
```

- ✓ `tasa_de_muestreo` del audio elemental: Generalmente 44100 o 48000 Hz.
- ✓ `tamaño_frame_audio` en bytes: puede ser leído con la herramienta esaudioinfo.
- ✓ `Tamaño_primer_frame_video`: necesario para la sincronización del audio y el video. Puede ser leído con la herramienta esvideoinfo.
- Un PES de audio puede ser empaquetado en un flujo de transporte TS con la herramienta pesaudio2ts.

```
pesaudio2ts audio.pes identificador_de_programa tasa_de_muestreo
tamaño_frame_audio tamaño_primer_frame_video > audio.ts
```

- ✓ `identificador_de_programa`: es un número único que identifica al flujo de audio. Un valor típico es 2068.
- ✓ `tasa_de_muestreo` del audio elemental: Generalmente 44100 o 48000 Hz.
- ✓ `tamaño_frame_audio` en bytes: puede ser leído con la herramienta `esaudioinfo`.
- ✓ `Tamaño_primer_frame_video`: necesario para la sincronización del audio y el video. Puede ser leído con la herramienta `esvideoinfo`.

La `tasa_de_bits` del archivo final está dada por el tamaño del flujo de transporte en bits dividido entre la longitud del audio en segundos.

B.2.6.3 Creación de un flujo de transporte para carrusel de objetos

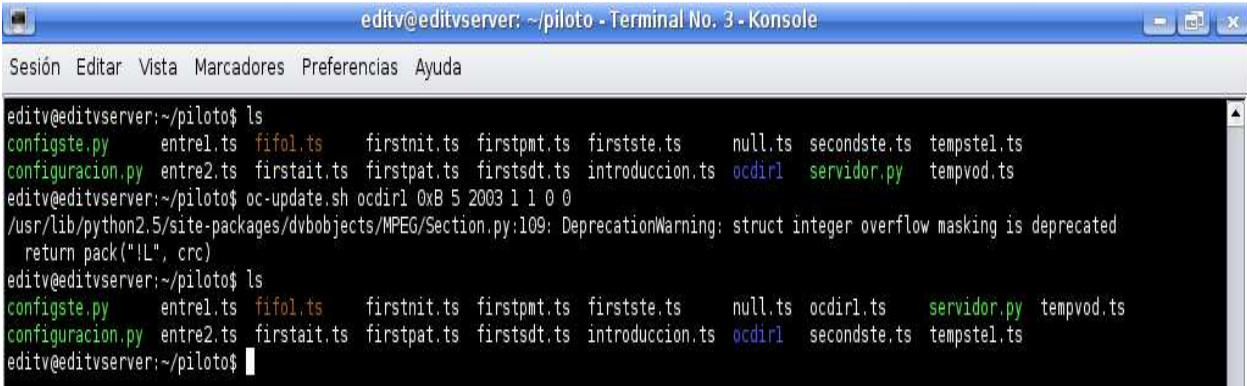
Para implementar un carrusel de objetos es necesario convertir una carpeta de archivos en un flujo de transporte. Para realizar esto se usa el script de OpenCaster `oc-update.sh`.

```
oc-update.sh      carpeta_archivos_del_carrusel      etiqueta_de_asociacion
version_del_modulo  identificador_de_programa      identificador_del_carrusel
```

```
oc-update.sh      carpeta_archivos_del_carrusel      etiqueta_de_asociacion      version_del_modulo
identificador_de_programa  identificador_del_carrusel
```

- `carpeta_archivos_del_carrusel`: la carpeta que será convertida en un carrusel de objetos.
- `etiqueta_de_asociacion`: en hexadecimal, como aparece referenciada en las tablas PMT y AIT, única por carrusel. Por ejemplo `0xB (11)`.
- `version_del_modulo`: necesario para que los cambios en el carrusel sean notificados al Set-top box, va de 0 a 15.
- `Identificador_de_programa`: único para cada flujo de transporte. Un valor típico es 2003.
- `Identificador_del_carrusel`: referenciado por la tabla PMT, cada carrusel tiene su propio identificador.

En la Figura B.30 se muestra un ejemplo de la salida del comando al crear el flujo de transporte `ocdir1.ts` desde los archivos en la carpeta `ocdir1`:



```
editv@editvserver: ~/piloto - Terminal No. 3 - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ ls
configste.py  entrel.ts  fifol.ts  firstnit.ts  firstpmt.ts  firstste.ts  null.ts  secondste.ts  tempstel.ts
configuracion.py  entre2.ts  firstait.ts  firstpat.ts  firstsdt.ts  introduccion.ts  ocdirl  servidor.py  tempvod.ts
editv@editvserver:~/piloto$ oc-update.sh ocdirl 0xB 5 2003 1 1 0 0
/usr/lib/python2.5/site-packages/dvbobjects/MPEG/Section.py:109: DeprecationWarning: struct integer overflow masking is deprecated
  return pack("IL", crc)
editv@editvserver:~/piloto$ ls
configste.py  entrel.ts  fifol.ts  firstnit.ts  firstpmt.ts  firstste.ts  null.ts  ocdirl.ts  servidor.py  tempvod.ts
configuracion.py  entre2.ts  firstait.ts  firstpat.ts  firstsdt.ts  introduccion.ts  ocdirl  secondste.ts  tempstel.ts
editv@editvserver:~/piloto$
```

Figura B.30. Salida de un flujo de transporte

B.2.6.4 Creación de flujos de transporte para tablas de señalización

Las tablas de información de programa y servicio PSI una vez están configuradas con todos sus parámetros son creadas a través de la herramienta de OpenCaster sec2ts.

En el script de la Figura B.31 para cada tabla se crea un archivo con extensión .sec que toma la configuración de cada tabla y la transforma en una sección a través de la función pack() de las librería de OpenCaster. Los archivos de sección son transformados en flujos de transporte con el programa sec2ts.

```
#
# Creacion de los flujos de transporte para las tablas PSI
#
out = open("./nit.sec", "wb")
out.write(nit.pack0)
out.close
os.system('/usr/local/bin/sec2ts 16 < ./nit.sec > ./firstnit.ts')

out = open("./pat.sec", "wb")
out.write(pat.pack0)
out.close
os.system('/usr/local/bin/sec2ts 0 < ./pat.sec > ./firstpat.ts')

out = open("./sdt.sec", "wb")
out.write(sdt.pack0)
out.close
os.system('/usr/local/bin/sec2ts 17 < ./sdt.sec > ./firstsdt.ts')

out = open("./pmt.sec", "wb")
out.write(pmt.pack0)
out.close
os.system('/usr/local/bin/sec2ts ' + str(editv_pmt_pid) + ' < ./pmt.sec > ./firstpmt.ts')

out = open("./ait.sec", "wb")
out.write(ait.pack0)
out.close
os.system('/usr/local/bin/sec2ts ' + str(ait1_pid) + ' < ./ait.sec > ./firstait.ts')

out = open("./ste.sec", "wb")
out.write(ste.pack0)
out.close
os.system('/usr/local/bin/sec2ts ' + str(ste1_pid) + ' < ./ste.sec > ./firstste.ts')

out = open("./ste2.sec", "wb")
out.write(ste2.pack0)
out.close
os.system('/usr/local/bin/sec2ts ' + str(ste1_pid) + ' < ./ste2.sec > ./secondste.ts')
```

Figura B.31. Script que transformación de tablas.

B.2.7 Multiplexación de flujos de transporte

Para realizar la transmisión de los flujos de tablas de señalización, eventos, carrusel de objetos, audio, video y demás es necesario unirlos en un único flujo de transporte MPEG-2 a una tasa de bits dependiente del esquema de modulación a usar y las capacidades del receptor.

- Para DVB-C

El cálculo de la tasa de bits para transmisión bajo el estándar DVB-C se realiza con la fórmula:

$$\text{Tasa_de_bits (bits/segundo)} = \text{tasa_en_baudios (símbolos/segundo)} * \text{numero_bits_por_simbolo} * (188/204)$$

Donde:

Tasa_de_baudios = Configurada en el STB para sintonización.

Numero_bits_por_simbolo = depende del esquema de modulación usado.

204/188 = relación entre los 16 bits añadidos a cada paquete enviado.

En el proyecto se cuenta con un STB que soporta una tasa máxima de 6890000 símbolos/segundo y permite usar un esquema de modulación 64QAM (6 bits/símbolo), por lo tanto la tasa de bits para transmisión sería:

$$\text{Tasa_de_bits} = 6890000 \text{ baudios} * 6 \text{ (bits/symbol)} * (188/204) = 38097647 \text{ bps}$$

De acuerdo a los parámetros máximos soportados por el STB y el modulador pueden darse las diferentes combinaciones de la Figura B.32.

Available bitrates (Mbit/s) for a DVB-T system in 8 MHz channels					
Modulation	Coding rate	Guard interval			
		1/4	1/8	1/16	1/32
QPSK	1/2	4.976	5.529	5.855	6.032
	2/3	6.635	7.373	7.806	8.043
	3/4	7.465	8.294	8.782	9.048
	5/6	8.294	9.216	9.758	10.053
	7/8	8.709	9.676	10.246	10.556
16-QAM	1/2	9.953	11.059	11.709	12.064
	2/3	13.271	14.745	15.612	16.086
	3/4	14.929	16.588	17.564	18.096
	5/6	16.588	18.431	19.516	20.107
	7/8	17.418	19.353	20.491	21.112
64-QAM	1/2	14.929	16.588	17.564	18.096
	2/3	19.906	22.118	23.419	24.128
	3/4	22.394	24.882	26.346	27.144
	5/6	24.882	27.647	29.273	30.160
	7/8	26.126	29.029	30.737	31.668

Figura B.32. Tasas de bits disponibles

De acuerdo a las características del STB del laboratorio de EDiTV éste soporta COFDM (Coded Orthogonal Frequency Division Multiplexing); Modulaciones QPSK/16/64 QAM; y FEC: 1/2,2/3,3/4,5/6,7/8. Por lo cual puede usarse cualquiera de las tasas de bits disponibles en la tabla de la Figura B.32.

Una vez determinada la tasa de bits a la cual se va a transmitir, el siguiente paso es la multiplexación de los flujos con el programa tschrmuxer de OpenCaster.

```
tschrmuxer b:tasa_de_bits_pat pat.ts b:tasa-de_bits_video video.ts
b:tasa_de_bits_pmt pmt.ts ... > archivo_de_salida
```

Donde:

bitrate b:tasa_de_bits_... archivo.ts: está en bits por segundo
archivo_de_salida: la salida del comando debe redirigirse a un archivo o una fifo.

Por ejemplo para DVB-C con una tasa de 38097647 bits/segundo se tiene:

```
tschrmuxer b:7000000 video.ts b:125000 audio.ts b:3008 pat.ts b:3008
pmt.ts b:1400 nit.ts b:1500 sdt.ts b:2000 ait.ts b:4000000 ocdid.ts
b:26961731 null.ts > archivo.ts
```

Donde la tasa de bits en conjunto, de los flujos con información es 11135916 y para completar la tasa de bits de transmisión se añaden 26961731 bits/segundo de un flujo de transporte nulo.

B.2.8 Modulación de un flujo de transporte MPEG2

La modulación de un flujo de transporte MPEG-2 se realiza a través del programa DtPlay del software que acompaña la tarjeta Dektec 110T (Dektec, 2007). Los parámetros de la aplicación son dependientes del estándar con el que se va a transmitir y de las capacidades del receptor. En el caso del proyecto la tarjeta 110T soporta modulación bajo el estándar DVB-C, DVB-T y DVB-H.

Transmisión DVB-C

Para la modulación de un flujo en DVB-C se tienen las opciones:

```
DtPlay archivo.ts -r 38097647 -t 110 -mt QAM64 -mf 474 -ma A
```

```
DtPlay archivo.ts -r 38097647 -t 110 -mt QAM64 -mf 474 -ma A
```

Donde:

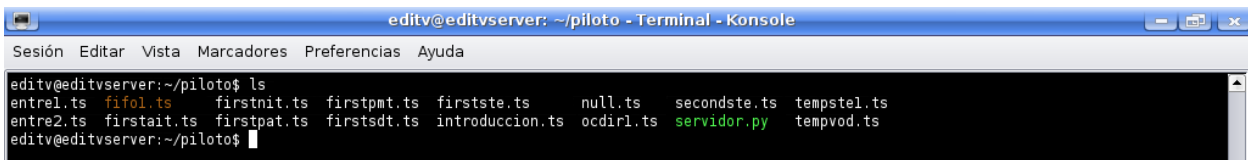
- archivo.ts: es el archivo donde se encuentra el TS multiplexado.
- -r: se usa para indicar la tasa de bits con la que se va a trabajar, debe coincidir con la suma de las tasas de bits de los flujos de transporte multiplexados.
- -t: referencia la tarjeta a usar. En este caso el modelo 110T.
- -mt: indica el tipo de modulación, (64QAM).
- -mf: la frecuencia de la portadora en Mhz (474).
- -ma A: se usa para indicar el uso del anexo J.83 correspondiente a DVB-C.

B.3 Ejecución de la aplicación en la cadena de EDiTV del laboratorio

Para la ejecución de la aplicación generando un flujo de transporte modulado y cargado por el STB es necesario hacer uso de varias herramientas que se describen a continuación.

- Tscbrmuxer: multiplexador de flujos de transporte presente en el paquete OpenCaster.
- DtPlay: Programa de modulación de un flujo de transporte presente con los drivers de la tarjeta moduladora Dektec.
- Servidor.py: Servidor de video bajo demanda creado en el lenguaje Python.
- Flujos de transporte a multiplexar.

Ver Figura B.33.



```

editv@editvserver: ~/piloto - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ ls
entrel.ts  fifo1.ts    firstnit.ts  firstpmt.ts  firstste.ts  null.ts    secondste.ts  tempstel.ts
entre2.ts  firstait.ts firstpat.ts  firstsdt.ts  introduccion.ts  ocdirl.ts  servidor.py    tempvod.ts
editv@editvserver:~/piloto$

```

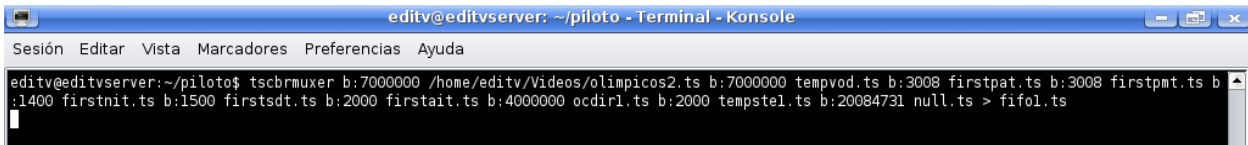
Figura B.33. Ejecución de la aplicación en la cadena de EDiTV del laboratorio – Paso uno

Una vez se tienen los flujos de transporte de audio, video, tablas de señalización, eventos y carrusel de objetos (descrito en el apartado dos de este documento), se usa el programa tscbrmuxer (tscbrmuxer, 2008) para realizar la multiplexación de todos los flujos independientes en un único flujo de transporte MPEG-2 con una tasa de bits acorde al esquema de modulación a usar. La multiplexación se realiza con el comando:

```
tscbrmuxer b:bitrate_archivo1 archivo1.ts ... b:bitrate_archivoN archivoN.ts
```

```
tscbrmuxer b:7000000 /home/editv/Videos/olimpicos2.ts b:7000000 tempvod.ts b:3008 firstpat.ts b:3008
firstpmt.ts b:1400 firstnit.ts b:1500 firstsdt.ts b:2000 firstait.ts b:4000000 ocdirl.ts b:2000 tempste1.ts b:20084731
null.ts > fifo1.ts
```

ver Figura B.34.



```

editv@editvserver: ~/piloto - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ tscbrmuxer b:7000000 /home/editv/Videos/olimpicos2.ts b:7000000 tempvod.ts b:3008 firstpat.ts b:3008 firstpmt.ts b
:1400 firstnit.ts b:1500 firstsdt.ts b:2000 firstait.ts b:4000000 ocdirl.ts b:2000 tempstel.ts b:20084731 null.ts > fifo1.ts

```

Figura B.34. Ejecución de la aplicación en la cadena de EDiTV del laboratorio – Paso dos

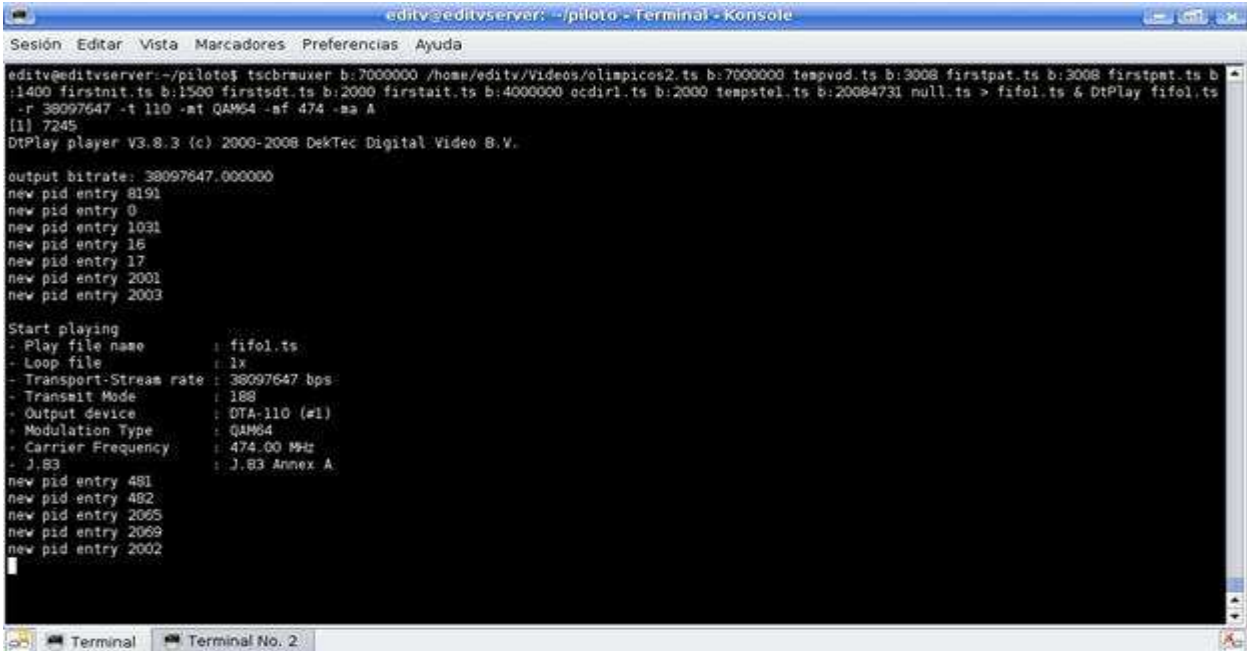
La aplicación Piloto contiene los siguientes flujos de transporte:

- olimpicos2.ts con un bitrate de 7000000 bits/segundo usado como flujo de audio/video para broadcast.
- tempvod.ts con un bitrate de 7000000 bits/segundo usado como flujo de audio/video para video bajo demanda. Este es un flujo con paquetes nulos que es llenado con paquetes de otros flujos a través de las peticiones del usuario al servidor de video bajo demanda.
- firstpat.ts con un bitrate de 3008 bits/segundo que contiene la tabla de aplicación de programas.
- firstpmt.ts con un bitrate de 1400 bits/segundo que contiene la tabla de mapeo de programas.
- firstnit.ts con un bitrate de 1500 bits/segundo que contiene la tabla de información de red.
- firstsdt.ts con un bitrate de 1500 bits/segundo que contiene la tabla de descripción de servicios.
- firstait.ts con un bitrate de 2000 bits/segundo que contiene la tabla de información de aplicaciones.
- ocdirl.ts con un bitrate de 4000000 bits/segundo que contiene el carrusel de objetos con la aplicación piloto.
- tempste1.ts con un bitrate de 2000 bits/segundo que es usado para inyectar eventos. Este es un flujo con paquetes nulos que es llenado con paquetes de otros flujos a través de las peticiones del usuario al servidor de video bajo demanda.
- null.ts con un bitrate de 20084731 bits/segundo que es usado para rellenar el flujo con paquetes nulos hasta alcanzar la tasa de bits de transmisión.

La salida de este comando es redireccionada a un archivo fifo llamado fifo1.ts el cual se va llenando con los flujos multiplexados. Una vez se tiene este archivo se ejecuta el programa DtPlay de la tarjeta moduladora para modular el flujo de transporte con los parámetros dados para la modulación deseada en este caso DVB-C.

```
DtPlay archivo.ts -r tasa_de_bits -t modelo_tarjeta -mt Modulacion -mf frecuencia_portadora -ma Anexo_Modulacion
```

```
tscbrmuxer b:7000000 /home/editv/Videos/olimpicos2.ts b:7000000 tempvod.ts b:3008 firstpat.ts b:3008
firstpmt.ts b:1400 firstnit.ts b:1500 firstsdt.ts b:2000 firstait.ts b:4000000 ocdirl.ts b:2000 tempste1.ts b:20084731
null.ts > fifo1.ts & DtPlay fifo1.ts -r 38097647 -t 110 -mt QAM64 -mf 474 -ma A
```

```

editv@editvserver: ~/piloto - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ tschrmuxer b:7000000 /home/editv/Videos/olimpicos2.ts b:7000000 tempvod.ts b:3008 firstpat.ts b:3008 firstpat.ts b
:1400 firstpat.ts b:1500 firstpat.ts b:2000 firstpat.ts b:4000000 ocdin1.ts b:2000 tempstel.ts b:20084731 null.ts > fifol.ts & DtPlay fifol.ts
-r 38097647 -t 110 -mt QAM64 -mf 474 -ma A
[1] 7245
DtPlay player V3.8.3 (c) 2000-2008 DekTec Digital Video B.V.

output bitrate: 38097647.000000
new pid entry 8191
new pid entry 0
new pid entry 1031
new pid entry 16
new pid entry 17
new pid entry 2001
new pid entry 2003

Start playing
- Play file name      : fifol.ts
- Loop file          : 1x
- Transport-Stream rate : 38097647 bps
- Transmit Mode      : 188
- Output device       : DTA-110 (#1)
- Modulation Type     : QAM64
- Carrier Frequency   : 474.00 MHz
- J.83                : J.83 Annex A
new pid entry 481
new pid entry 482
new pid entry 2065
new pid entry 2069
new pid entry 2002

```

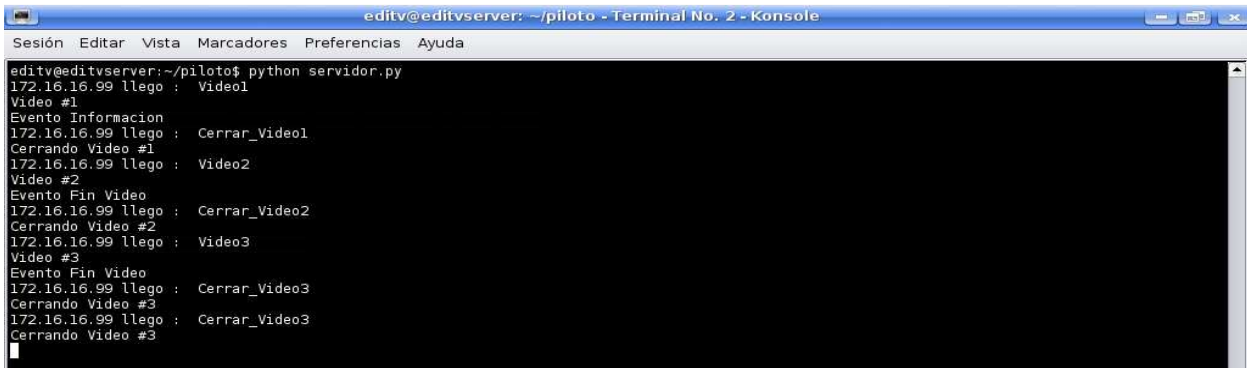
Figura B.35. de la aplicación en la cadena de EDiTV del laboratorio – Paso tres

Para la ejecución de la aplicación en un STB con interfaz de cable se tienen como parámetros:

- -r 38097647 tasa de bits para DVB-C con una tasa de 6900MBaudios.
- -t 110 que indica una tarjeta Dektec 110T.
- -mt QAM64 indica que se usará la modulación QAM de 64 bits.
- -mf 474 representa la frecuencia de la portadora en la que el STB sintonizará el servicio.
- -ma A indica que se usará el anexo A que representa a DV-C.

Todos estos parámetros han de estar acordes con los soportados por el STB, por lo que hay que verificar primero sus características. La salida de la línea de comandos indica que la modulación se ejecuto sin problemas y muestra los identificadores de todos los flujos modulados, así como los parámetros principales de modulación. A la vez debe ejecutarse el programa servidor.py que hace las veces de servidor de video bajo demanda, con el comando:

python servidor.py, ver Figura B.36.



```

editv@editvserver: ~/piloto - Terminal No. 2 - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
editv@editvserver:~/piloto$ python servidor.py
172.16.16.99 luego : Video1
Video #1
Evento Informacion
172.16.16.99 luego : Cerrar_Video1
Cerrando Video #1
172.16.16.99 luego : Video2
Video #2
Evento Fin Video
172.16.16.99 luego : Cerrar_Video2
Cerrando Video #2
172.16.16.99 luego : Video3
Video #3
Evento Fin Video
172.16.16.99 luego : Cerrar_Video3
Cerrando Video #3
172.16.16.99 luego : Cerrar_Video3
Cerrando Video #3

```

Figura B.36. Salida del comando python

El servidor de video bajo demanda es un programa escrito en el lenguaje Python que crea un socket servidor UDP en el puerto 5000 y espera la conexión de los clientes en este caso la aplicación en el Set-top Box para recibir mensajes. La salida de este programa indica cuando llega un mensaje de petición o parada de un video bajo demanda, así como los eventos de información o de fin de video programados. El código fuente de la aplicación es el siguiente:

```

#Servidor UDP

import socket
import os
import time
import threading

PORT = 5000
HOST = '172.16.200.24'
global fin
fin = 0

serv = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serv.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
serv.bind((HOST, PORT))

class Timer(threading.Thread):
    def __init__(self, seconds, titulo):
        self.runTime = seconds
        self.titulo = titulo
        threading.Thread.__init__(self)
    def run(self):
        for l in range(self.runTime*10):
            if fin == 1:
                print 'abortado'
                break
            time.sleep(0.1)
        if fin == 0:
            if self.titulo in ['evento1']:
                os.system('cp firstste.ts tempstel.ts &')
                print "Evento Informacion"
                time.sleep(1)
                os.system('cp null.ts tempstel.ts &')
            if self.titulo in ['evento2']:
                os.system('cp secondste.ts tempstel.ts &')
                print "Evento Fin Video"
                time.sleep(1)
                os.system('cp null.ts tempstel.ts &')

while 1:
    data, address = serv.recvfrom(256)
    print address[0], "llego : ", data
    if data in ['Video1']:
        fin = 0
        os.system('cp introduccion.ts tempvod.ts &')
        t = Timer(10,"evento1")
        t.start()
        print "Video #1"
    if data in ['Cerrar_Video1']:
        fin = 1
        os.system('cp null.ts tempvod.ts &')
        print "Cerrando Video #1"
    if data in ['Video2']:
        fin = 0
        os.system('cp entre1.ts tempvod.ts &')
        t = Timer(90,"evento2")
        t.start()
        print "Video #2"
    if data in ['Cerrar_Video2']:
        fin = 1
        os.system('cp null.ts tempvod.ts &')
        print "Cerrando Video #2"
    if data in ['Video3']:
        fin = 0
        os.system('cp entre2.ts tempvod.ts &')
        t = Timer(40,"evento2")
        t.start()
        print "Video #3"
    if data in ['Cerrar_Video3']:
        fin = 1
        os.system('cp null.ts tempvod.ts &')
        print "Cerrando Video #3"

```

Referencias

- ADB. (2007). *ADB Ref. Q75DEV Soporte a MHP*. Recuperado el 25 de marzo de 2008, de portal adbglobal.com: <http://www.adbglobal.com/?q=node/193>
- Avalpa, E. D. (2008). *OpenCaster*. Recuperado el 5 de marzo de 2008, de portal avalpa.com: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>
- DekTec. (2007). *DVB-T Modulador with UHF Upconverter*.
- Eclipse, (2008) Open Source Java IDE. <http://www.eclipse.org>
- FFMPEG. (2007). *FFMPEG*. Recuperado el 15 de enero de 2009, de portal ffmpeg.mplayerhq.hu: <http://ffmpeg.mplayerhq.hu/download.html>
- Main_Concept. (2009). *MainConcept Reference*. Recuperado el 2 de febrero de 2009, de portal mainconcept.com: <http://www.mainconcept.com/site/prosumer-products-4.html>
- MHP Stubs (2008). *MHPStubs*. Recuperado el 20 de enero de 2009, de portal interactivetvweb: http://interactivetvweb.org/resources/code_samples
- Mimundo.tv, 8 (2007). Recuperado el 4 de febrero de 2009 de portal mimundo.tv: <http://www.mimundotv.de/>
- MJPEG-Tools, (2005), Recuperado el 25 de enero de 2009 de portal mjpeg: <http://mjpeg.sourceforge.net>
- Netpbm, (2008), Netpbm, recuperado el 20 de enero de 2009 de portal netpbm: <http://netpbm.sourceforge.net>
- OpenCaster (2008). Recuperado el 20 de diciembre de 2008 de portal cs-svn.cs.surrey.sfu.ca: <https://cs-svn.cs.surrey.sfu.ca/ns1/browser/DVB-H/src/OpenCaster/tools/tscbrmuxer?rev=1969>.
- Pnmpad (2009). Recuperado el 15 de enero de 2009 de portal netpbm: <http://netpbm.sourceforge.net/doc/pnmpad.html>

Anexo C

Tabla de Contenido

Anexo C	1
C.1 Montaje del laboratorio de ITV de la Universidad de Oviedo	1
C.1.1 Configuración de La red de Cable	1
C.1.2 Registro del cable módem	2
C.1.3 Configuración del CMTS	2
C.1.4 Configuración del servidor DHCP.....	2
C.1.5 Configuración del servidor TFTP	3
C.1.6 Configuración del IPCop	4
C.2 Archivos de Configuración	5
C.2.1 CMTS	5
C.2.2 Servidor DHCP.....	7
C.2.3 Servidor TFTP	10
C.2.4 Cable Modem	10
Referencias.....	11

Anexo C

C.1 Montaje del laboratorio de ITV de la Universidad de Oviedo

En la universidad de Oviedo existen los elementos Hardware y Software de un laboratorio de Televisión Digital, cuyo primer paso fue poner en funcionamiento, para brindar las condiciones mínimas necesarias que permitan el contacto con elementos de una red de TD. Una vez montada esta red existen las condiciones para llevar este escenario hasta un modelo de simulación. De esta manera el trabajo se desarrolla de manera incremental, entrando a trabajar primeramente en la instalación, configuración y puesta a punto de una red de cable, que soportara la TDi, con cada uno de sus elementos como el CMTS (Cable Modem Termination System), el servidor de aplicaciones con sus servicios de DHCP (Dynamic Host Configuration Protocol) y TFTP (Trivial File Transfer Protocol) para el direccionamiento de la red y la trasferencia de los parámetros de configuración de los CM respectivamente. A continuación se describe el proceso necesario para la puesta a punto del laboratorio de ITV de la universidad de Oviedo

C.1.1 Configuración de La red de Cable

En la Figura.1 se muestra la red de cable del laboratorio de la universidad de Oviedo.

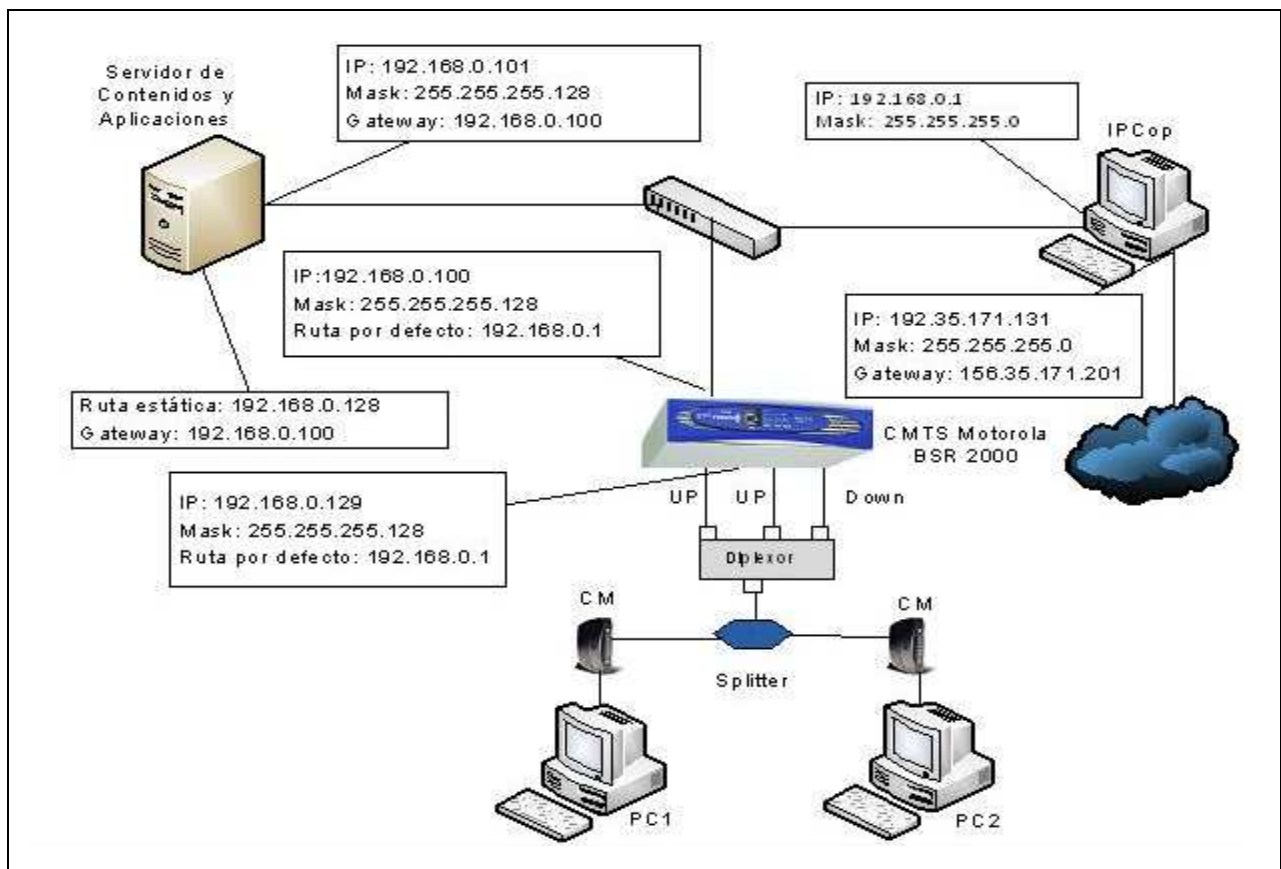


Figura 1. Red del laboratorio de ITV- Universidad de Oviedo

En la red de cable, los usuarios podrán conectarse a Internet desde su computador a través de un cable módem, que sirve de “intermediario” entre la tarjeta ethernet del PC y la propia red de cable. La otra parte de la topología de esta red está compuesta por todos los servicios necesarios para el correcto funcionamiento de los cable módem, esto es: servidor DHCP, servidor TFTP y un punto de salida a Internet.

C.1.2 Registro del cable módem

Cuando el CM (Motorola, 200\$) se conecta a una red, pueden ocurrir dos escenarios:

- Que sea la primera vez que intenta logearse en esa red (o que la frecuencia haya sido cambiada). En este caso, el CM escanea todo el espectro de frecuencias para localizar la frecuencia de los datos, busca señales moduladas digitalmente bajo el formato 64 ó 256 QAM (Quadrature Amplitude Modulation)
- Si el CM ya ha accedido previamente a esa red, se enganchará a la frecuencia de datos automáticamente, está almacenada de la sesión anterior.

Una vez que el CM encuentra la señal de datos, busca el mensaje que contiene los parámetros básicos del canal de subida (frecuencia, modulación, tasa de símbolos....). Una vez hecho esto, el CM estará en condiciones para poder transmitir un mensaje al CMTS pidiendo información adicional que le permita conectarse a la red. Entonces, intercambiando una serie de mensajes, el CM va a establecer conexión IP a través de DHCP y recibirá el archivo de configuración, que contiene parámetros adicionales que el CM necesita para auto configurarse a través de TFTP.

C.1.3 Configuración del CMTS

El CMTS (Motorola, 2006) es el equipo encargado de proporcionar los servicios de Internet por cable. A grandes rasgos, es un router con conexiones ethernet (hacia La Internet) y con conexiones de radiofrecuencia en las interfaces de los cables coaxiales (hacia la zona de abonado). De esta forma, se puede hacer un “puente” entre el tráfico que llega de Internet a través de la interfaz ethernet y entre la red de cable a la que están conectados los CM de los abonados.

Para una configuración básica del CMTS se debe tener en cuenta lo siguiente:

- Asignar IPs a la interfaz ethernet y la interfaz de cable
- Establecer el tipo de CMTS como “euro” o Americano según corresponda, en este caso tipo “euro”
- Permitir el reenvío de información DHCP en la interfaz de cable, y definir cuál es el servidor DHCP al cual deberá reenviar esas peticiones provenientes de los CM.
- Establecer una ruta por defecto, cuyo gateway será la dirección IP de nuestro IPCop.

C.1.4 Configuración del servidor DHCP

El servidor DHCP será el encargado de proporcionar una dirección a los CM para que puedan establecer una comunicación IP. Se instala en el Servidor de Aplicaciones cuyo sistema operativo es Linux .

La puesta a punto del servidor DHCP con la configuración de red que se muestra en la Figura 1 exige tener en cuenta las siguientes condiciones.

- El servidor DHCP está en una red distinta a la red de cable pero las direcciones que tiene que servir este servidor DHCP serán direcciones del rango al que pertenece la red de cable.
- El servidor DHCP debe servir direcciones que no pertenecen a su red, para esto se configuran las máscaras de red, de manera que la red de cable y la red Ethernet pertenezcan a redes distintas, pero el servidor DHCP tendrá una máscara de red que las englobará a las dos. (ver el esquema de la red, Figura 1.)
- Además, se coloca en el computador donde estén instalados estos servicios, una ruta estática hacia la red de cable, a través de la interfaz ethernet del CMTS mediante el siguiente comando

```
$ route add -net 192,168,0,128 netmask 255,255,255,128 gw 192,168,0,100,
```

Entonces la ruta por defecto será hacia el IPCop. Teniendo en cuenta las anteriores condiciones para el servidor DHCP, es necesario realizar los siguientes pasos para su puesta en funcionamiento.

- Instalación: se realiza mediante el siguiente comando:

```
$ sudo apt-get install dhcp3-server
```

- Configuración: es necesario editar mediante el siguiente comando el archivo de configuración dhcpd.conf:

```
$ nano /etc/dhcp3/dhcpd.conf
```

- Iniciación: una vez se han guardado los cambios mediante el siguiente comando se inicia el servicio:

```
$ /etc/init.d/dhcp3-server restart
```

C.1.5 Configuración del servidor TFTP

El TFTP se instala en el Servidor de Aplicaciones. En el servidor TFTP se tiene guardado el archivo de configuración del CM. Es necesario realizar los siguientes pasos para su puesta en funcionamiento.

- Instalación: se realiza mediante el siguiente comando.

```
$ sudo apt-get install xinetd tftpd tft.
```

- Configuración: se crea el fichero tftp

```
$ nano /etc/xinetd.d/tftp
```

- Edición: en el fichero anteriormente creado se escribe:

```
service tftp
{
  protocol = udp
  port = 69
  socket_type = dgram
  wait = yes
  user = nobody
  server = /usr/sbin/in.tftpd
  server_args = /tftpboot
  disable = no
}
```

- Se crea un directorio tftpboot donde estarán los archivos que son necesarios descargar:

```
$ mkdir /var/lib/tftpboot
```

Se le asignan permisos de ejecución, lectura y escritura:

```
$ sudo chmod -R 777 /tftpboot
```

```
$ sudo chown -R nobody /tftpboot
```

- Iniciación: se reinicia el servicio:


```
$sudo /etc/init.d/xinetd restart
```

Si se tiene problemas con el servicio TFTP (ya que el firewall podría estar bloqueándolo), una solución sencilla sería desactivar el servicio xinetd y correr solamente el proceso "in.tftp". En este caso, el tftpboot por defecto estará en el directorio raíz.

- Para subir archivos al servidor TFTP, primero se tiene que crear un fichero vacío con el mismo nombre del fichero que se desea subir

```
$touch /tftpboot/running-config.txt
```

Aplicar los permisos correspondientes al fichero:

```
$chmod 777 /tftpboot/running-config.txt
```

Para el archivo de configuración del cable módem, se debe tener instalado un editor de DOCSIS para crear un archivo de configuración válido para un CM. En la página <http://docsis.sourceforge.net/> se puede encontrar un programa (en Windows) que generará el archivo de configuración binario. Además, también se puede encontrar numerosos ejemplos de archivos de configuración, desde los más sencillos a los más completos.

Para generar el archivo binario a partir de un archivo de configuración con este editor DOCSIS: Se abre una consola y desde el directorio donde esté instalado el programa, se digita "docsis" y aparece una pequeña ayuda con el modo de utilizar el programa. Para codificar un archivo de configuración:

```
docsis -e <archivo_de_confi> <clave de autenticación> <archivo_de_salida>.
```

Esa clave de autenticación será un secreto compartido que usará en CMTS en caso de ser necesario.

C.1.6 Configuración del IPCop

El último paso, para conseguir conectividad a Internet, será instalar una herramienta que proteja la red interna y que proporcione un acceso seguro a Internet. IPCop (Administrative, 2004) es una herramienta de código abierto y distribuida bajo términos de GNU General Public License. Básicamente es un firewall basado en Linux que proporciona una serie de servicios adicionales como DHCP y puede establecer dos zonas bien diferenciadas: la zona verde (nuestra red interna) y la zona roja (el exterior).

La instalación del IPCop es muy sencilla, y sólo habrá que seguir los menús de instalación, teniendo en cuenta las IPs que se asignen cuando se configure la zona verde y la roja.

Una vez hecho esto, y para que todo funcione correctamente, se debe asignar una ruta estática hacia la red de cable (para poder reenviar el tráfico de Internet)

```
$route add -net 192,168,0,128 netmask 255,255,255,128 gw 192,168,0,100
```

Y una ruta por defecto, a través de la interfaz externa:

```
$route add -net 0,0,0,0 netmask 0,0,0,0 gw 156,35,171,201
```

Con estos pasos realizados correctamente, ya se puede navegar desde un computador conectado a través de un CM a la interfaz de cable.

C.2 Archivos de Configuración

A continuación se presentan los archivos de configuración necesarios para el correcto funcionamiento de la red.

C.2.1 CMTS

```
!  
! Last configuration change at FRI JUN 25 15:51:18 2066  
!  
!  
!  
enable password 0 cmts  
!  
hostname CMTS  
!  
!  
!  
no service password-encryption  
!  
!  
!  
logging buffered 51200  
no logging control docsis  
logging admin-status unconstrained  
logging rate-limit 100 1  
logging buffered notifications  
logging console errors  
!  
!  
cable cmts type euro  
!  
!  
interface ethernet 0/0  
ip address 192.168.0.100 255.255.255.128  
ip helper-address 192.168.0.1  
no shutdown  
!  
interface ethernet 0/1  
ip address 10.0.0.1 255.0.0.0  
no shutdown  
!  
interface ethernet 0/2  
shutdown  
!  
interface ethernet 0/3  
shutdown  
!  
interface gigaether 0/0  
shutdown  
!  
interface gigaether 0/1  
shutdown  
!
```

```
interface cable 0/0
ip address 192.168.0.129 255.255.255.128
ip helper-address 192.168.0.1
cable helper-address 192.168.0.101 cable-modem
cable helper-address 192.168.0.101 host
no shutdown
cable downstream 0 frequency 535000000
no cable downstream 0 shutdown
cable upstream 0/0 channel-type atdma
no cable upstream 0 shutdown
no cable upstream 0/0 shutdown
cable upstream 0/1 shutdown
cable upstream 0/2 shutdown
cable upstream 0/3 shutdown
cable upstream 1 shutdown
cable upstream 1/0 shutdown
cable upstream 1/1 shutdown
cable upstream 1/2 shutdown
cable upstream 1/3 shutdown
cable upstream 2 shutdown
cable upstream 2/0 shutdown
cable upstream 2/1 shutdown
cable upstream 2/2 shutdown
cable upstream 2/3 shutdown
cable upstream 3 shutdown
cable upstream 3/0 shutdown
cable upstream 3/1 shutdown
cable upstream 3/2 shutdown
cable upstream 3/3 shutdown
cable dynamic-service authorization-mode unauthorize
ip dhcp relay information option
cable downstream schedule priority-wfq
no cable wireless downstream modulation qpsk
!
!
!
!
!
router rip
!
!
ip route 0.0.0.0 0.0.0.0 192.168.0.1
!
!
!
!
!
packet-cable
dqos shutdown
no em shutdown
no es shutdown
!
!
ipsec
ipsec shutdown
```

```
!  
password telnet 0 cmts  
password console 0 cmts  
!  
session-timeout console 0  
session-timeout telnet 0  
!  
telnet session-limit 3  
!  
!  
!  
!  
!  
!  
!  
no ssh enable  
!  
ip forward-protocol udp 546  
ip forward-protocol udp 547  
!  
no logging trap  
no logging snmp-trap  
logging facility local7  
!  
!  
!  
!  
!  
enable rdn-process  
!  
!  
!  
cable service-class  
!  
!  
!  
!  
!  
!  
!
```

C.2.2 Servidor DHCP

Editar: /etc/dhcp3/dhcpd.conf

```
#  
# Sample configuration file for ISC dhcpd for Debian  
#  
# $Id: dhcpd.conf,v 1.1.1.1 2002/05/21 00:07:44 peloy Exp $  
#
```

```
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

#default-lease-time 600;
#max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

#subnet 10.254.239.0 netmask 255.255.255.224 {
# range 10.254.239.10 10.254.239.20;
# option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
# range dynamic-bootp 10.254.239.40 10.254.239.60;
# option broadcast-address 10.254.239.31;
# option routers rtr-239-32-1.example.org;
#}
option option-122 code 122 = string;

# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.130 192.168.0.134;
option subnet-mask 255.255.255.128;
option domain-name-servers 156.35.14.2;
option domain-name "internal.org";
option routers 192.168.0.129;
option broadcast-address 192.168.0.127;
allow bootp;
default-lease-time 600;
max-lease-time 7200;
```

```
option tftp-server-name "192.168.0.101";
next-server 192.168.0.101;
filename "config.cm";
}
option option-122 code 122 = string;

#subnet 10.3.0.0 netmask 255.255.0.0
#{
# option routers 10.3.0.1;
# option subnet-mask 255.255.0.0;
# option broadcast-address 10.3.255.255;
# range 10.3.0.20 10.3.0.25;
# default-lease-time 21600;
# max-lease-time 43200;
#}

# Hosts which require special configuration options can be listed in
# host statements. If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

host cablemodem {
#hardware ethernet 00:14:04:e2:80:9e;
#fixed-address 109.168.0.131;
#filename "vmunix.passacaglia";
#server-name "toccata.fugue.com";
}
host pclab
{
hardware ethernet 00:15:c5:aa:08:96;
fixed-address 10.0.0.3;
}

# Fixed IP addresses can also be specified for hosts. These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP. Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.
#host fantasia {
# hardware ethernet 08:00:07:26:c0:a5;
# fixed-address fantasia.fugue.com;
#}

# You can declare a class of clients and then do address allocation
# based on that. The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
# match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}

#shared-network 224-29 {
```

```
# subnet 10.17.224.0 netmask 255.255.255.0 {
# option routers rtr-224.example.org;
# }
# subnet 10.0.29.0 netmask 255.255.255.0 {
# option routers rtr-29.example.org;
# }
# pool {
# allow members of "foo";
# range 10.17.224.10 10.17.224.250;
# }
# pool {
# deny members of "foo";
# range 10.0.29.10 10.0.29.230;
# }
#}
```

C.2.3 Servidor TFTP

Editar: /etc/xinetd.d/tftp

```
service tftp
{
protocol=udp
port=69
socket_type=dgram
wait=yes
user=nobody
server=/usr/sbin/in.tftpd
server_args=-l -c -s /etc/xinetd.d/tftpboot
disable=no
}
```

Editar: /etc/default/tftpd-hpa

```
#Defaults for tftpd-hpa
RUN_DAEMON="yes"
OPTIONS="-l -c -s /var/lib/tftpboot/tftp"
#OPTIONS="-l -c -s /etc/xinetd.d/tftp"
```

C.2.4 Cable Modem

```
Main {
NetworkAccess 1;
MaxCPE 3;
ClassOfService {
ClassID 1;
MaxRateDown 20000000;
MaxRateUp 10000000;
PrivacyEnable 0;
}}
```

Sobre esta infraestructura de red se tomaron las medidas de las aplicaciones de Video bajo Demanda, que son presentados en el capítulo tres de la monografía.

Referencias

Motorola, Inc. "MOTOROLA SB5101 Cable Modem Overview", Noviembre de 2004.

Motorola, Inc. "BSR 2000 Configuration and Management Guide" release 1.0. 2006.

Administrative Guide, Chris Clancey, Harry Goldschmitt, John Kastner, Eric Oberlander, Peter , Walker, septiembre de 2004.