

Introducción a Delivery Media Integration Framework (MPEG-4 parte 6).

José Ignacio Estévez Damas

Abstract. Estos apuntes son una introducción al estándar recomendado por MPEG4 para realizar la distribución del contenido multimedia, el llamado Delivery Multimedia Integration Framework (parte 6 de la especificación de MPEG4). Están en su mayor parte basados en el trabajo [Pourmohammadi et al., 2001], accesible desde internet. Se recomienda bajar el documento referenciado, sobre todo para consultar los diagramas.

1. Introducción.

En la distribución de multimedia hemos de considerar tres contextos típicos.

- Contexto con recuperación y almacenamiento de datos local. En este caso los datos se encuentran en uno o varios archivos en el mismo ordenador que la aplicación que los demanda. Llamaremos a esta situación contexto local.
- Broadcast (difusión de contenidos). Tenemos un servidor de datos al que están conectados diversos clientes. Los clientes conectados reciben continuamente la información difundida por el servidor. Esta información se estructura en un conjunto de canales que el servidor puede aumentar o disminuir dinámicamente. Llamaremos a esta situación contexto de broadcast.

- Escenario con recuperación de datos remota. Tenemos que los datos multimedia son suministrados por una aplicación que se encuentra en un ordenador al que nos conectamos mediante una red de comunicaciones. Esta aplicación se comporta como un servidor que suministra datos por demanda. Llamaremos a esta situación icontexto remoto interactivo.

La arquitectura que se describe en estos apuntes se denomina Delivery Media Integration Framework y constituye la parte número 6 del estándar MPEG-4. Está diseñada para hacer frente a los contextos descritos en un marco de diversificación tecnológica y por lo tanto de heterogeneidad en las redes. Aunque se trata de una parte de MPEG-4, que se integra muy bien con MPEG-4 Systems, DMIF es una arquitectura que se puede aplicar en otros contextos de difusión de datos.

MPEG-4 se diferencia respecto a otros estándares de codificación / distribución de contenidos multimedia en muchos aspectos. Uno fundamental es la abstracción del proceso de distribución de multimedia. Entendamos este proceso en un sentido amplio, esto es, habiendo codificados los componentes multimedia, transmitir y/o almacenar el resultado de la codificación, de modo que un sistema de decodificación pueda obtener y presentar al usuario la información contenida en los datos codificados bajo un conjunto de restricciones técnicas y de calidad previamente establecido.

MPEG-2 se ocupa de la distribución de un modo concreto, estableciendo su propio protocolo de transporte. Sin embargo, debemos recordar que una de las razones de ser de MPEG-4, es hacer frente a la heterogeneidad de los procesos de distribución. Por ello, MPEG-4 no establece en su parte de sistemas un modo concreto de realizar la distribución de la multimedia, sino que recomienda una arquitectura suficientemente abstracta denominada DMIF para poder adaptarse a muchas tecnologías actuales y futuras.

Lógicamente esto podría suponer un problema para los programadores de contenidos multimedia: ¿cómo afrontar la construcción de una aplicación multimedia, incluida la distribución de la misma ante la indefinición de esta capa?. Por ello, DMIF incluye la formalización de una API (Application Interface) denominada DAI. El programador usará las funciones de la DAI y no tendrá entonces que conocer los detalles concretos de la implementación realizada de DMIF.

Otro aspecto del que se preocupa DMIF es del conjunto de señales soportadas por los protocolos de streaming. DMIF establece un protocolo genérico de señales y establece cómo adaptarlo a otros protocolos de señales.

Finalmente, debemos entender que aunque DMIF está diseñado para soportar las características de MPEG-4 Systems, es perfectamente posible usar-

lo para otras aplicaciones ajenas a MPEG-4.

2. Información complementaria para la comprensión de estos apuntes.

Puesto que vamos a tratar el tema de la distribución multimedia, usaremos terminología propia de las redes de comunicaciones. En particular nos referiremos a aspectos del modelo OSI (Open Systems Interconnection Reference Model). Este modelo puede verse como una descripción abstracta de un sistema de comunicaciones. Consiste en siete capas:

- Aplicación.
- Presentación.
- Sesión.
- Transporte.
- Red.
- Enlace de los datos.
- Medios físicos.

La capa de medios físicos considera los elementos tecnológicos que posibilitan la comunicación, y que se orientan a la representación y transmisión en los sistemas de comunicaciones de la unidad de información más básica, el bit.

La capa de enlace de los datos se ocupa de que existan los elementos adecuados en el sistema para poder establecer el concepto de nodo de comunicación. Proporciona por lo tanto la idea de identificación física o dirección física del elemento de comunicación. En este nivel la unidad de información es el frame, que podemos asociar al conjunto de bits involucrados en las operaciones más básicas de transferencia de información entre dos nodos conectados físicamente.

El nivel de red se asocia a conceptos del tipo origen y destino de la comunicación. En un sistema de nodos interconectados podremos realizar la comunicación entre un nodo etiquetado como nodo origen y otro como nodo destino. Hay muchos aspectos relacionados con esta idea: dirección lógica, calidad del servicio, control del flujo de la información, segmentación de la

red. El protocolo IP es un ejemplo de protocolo de nivel de red. La unidad de datos en el nivel de red se denomina paquete.

La capa de transporte se ocupa de preparar los datos para conseguir una comunicación host a host de forma transparente, y con valor añadido en cuanto a robustez y eficiencia de la comunicación. El adjetivo transparente se refiere a que no se considera la semántica de los datos: todos son tratados por igual. El valor añadido al que nos referimos se concentra en aspectos como control del flujo, recuperación ante errores, asegurar la transmisión completa de los datos, etcétera. TCP y UDP son ejemplos de protocolos en el nivel de transporte. La unidad de datos en el nivel de transporte es el segmento.

El nivel de sesión establece los medios para organizar el trabajo “cooperativo” de los agentes que se comunican. Aquí el énfasis está en la palabra “cooperativo” y abarca aspectos que van desde el establecimiento de una conexión entre dos aplicaciones, hasta la adecuada preparación de la información transferida para lograr que diferentes streams de datos mantengan una relación temporal (sincronización). Son ejemplos de protocolos a este nivel, SSH o RTCP.

La capa de presentación se ocupa de procesar los datos recibidos de manera que se adapten al formato de entrada requerido por la aplicación particular que tendrá que usarlos. Hablamos aquí de encriptación, juegos de caracteres, etcétera.

Finalmente tenemos la capa de aplicación, que proporciona servicios a los procesos de los usuarios. Un ejemplo podría ser el de un fichero virtual, al que los procesos de usuario acceden para obtener los datos.

El orden de las diferentes capas del modelo es importante. Una capa solicitará servicios de la capa inmediatamente inferior y suministrará servicio a la capa inmediatamente superior. Por ello, la implementación de los protocolos asociados a los niveles se denomina pila de protocolos.

Los conceptos de *control plane* y *user plane* también son referenciados en los apuntes. El término control plane, o signalling se refiere a aquella información que circula en el sistema de comunicaciones referida a la gestión del propio sistema y de las comunicaciones. Por contra, el user o data plane se refiere a los datos que los procesos usuarios del sistema de comunicaciones transfieren mediante éste.

3. Abstracción de la distribución multimedia.

Para tener una idea más clara de qué es lo que se pretende con esta abstracción, pensemos primero en los estándares MPEG-1 y MPEG-2. ¿Qué es lo que establecen respecto a la distribución multimedia?. MPEG-1 simple-

mente define cómo almacenar localmente el contenido multimedia (video). Por otra parte, MPEG-2 afronta tanto el almacenamiento local (MPEG2-PS (Program Stream)), como la distribución en redes dedicadas (MPEG2-TS (Transport Stream)), con la filosofía de cubrir las necesidades de la televisión por demanda.

El problema de MPEG2-TS es la dificultad que presenta en su adaptación a redes donde ya se ha establecido parte de un protocolo de comunicaciones diferente. MPEG-2 TS incluye aspectos de la capa de enlace de datos (data link) (longitud de paquetes fija y byte de sincronía), junto a aspectos de la sincronización multimedia (reloj de referencia y time stamps), siendo estas informaciones difícilmente separables, aunque en realidad se encuentren en niveles diferentes. Así, cuando se trata de implementar un sistema para la distribución de MPEG-2 sobre una red basada en IP o ATM nos encontramos funcionalidades replicadas, lo que lleva a sistemas de comunicación no eficientes.

El enfoque de MPEG-4 es diferente. Desde el principio se separó lo que debía ser uniforme en la implementación (MPEG4-Systems) de lo que debido a la heterogeneidad de las tecnologías de comunicación sería claramente particular de la implementación concreta. En este segundo contexto, nace MPEG4-DMIF. La frontera entre ambos mundos sería el DMIF-Application Interfaces (DAI). Resumamos entonces la estructura del estándar MPEG-4:

- *Capa de compresión.* La capa de compresión o codificación está formada por los estándares internacionales ISO/IEC 14496 Visual e ISO/IEC 14496-3 Audio. Esta capa debe tener en cuenta los medios audiovisuales y no considera la distribución de la multimedia.
- *Capa de sincronización.* En esta capa se añade información al stream de datos, de modo que se establece la sincronización de los diferentes ESs. Esta capa se define en ISO/IEC 14496-1 Systems. Es independiente tanto de los medios audiovisuales que se quieren distribuir, como de la propia distribución. La frontera entre la capa de compresión y la capa de sincronización se denomina interfaz ES.
- *Capa de distribución.* Esta capa es independiente de los medios audiovisuales, pero sí debe tener en consideración la implementación concreta del soporte de distribución. La implementación particular puede fundamentarse, aunque no es obligatorio, en el ISO/IEC 14496-6 DMIF. La frontera entre la capa de sincronización y la capa de distribución es el interfaz de aplicación del DMIF denominado DAI.

4. Los requerimientos de DMIF.

DMIF debe afrontar la heterogeneidad de los soportes y protocolos de distribución. En particular debe adaptarse a tres contextos:

- Almacenamiento y recuperación local del contenido multimedia (denominado en la introducción contexto local).
- Interacción peer to peer entre aplicaciones remotas (contexto remoto interactivo).
- Difusión de contenidos en broadcast (contexto broadcast).

Debemos además tener en consideración dos grandes clases de redes:

- *Redes basadas en el concepto de “mejor esfuerzo” (best effort delivery).* A la hora de llevar la información de un punto a otro el servidor y el cliente simplemente realizan el “mejor esfuerzo” para aprovechar las características de la red, pero no hay ninguna garantía sobre la calidad del servicio que se va a obtener. Es el caso del servicio postal. Ante el envío de una carta no se puede garantizar el tiempo. El sistema no acondiciona recursos para dar unas prestaciones particulares. Simplemente se trata de dar el servicio de la mejor forma posible para todos y no hay una adaptación ante circunstancias cambiantes. Por contra en una red de telefonía convencional, cuando se emprende una llamada se realizan reservas de recursos que garantizan una calidad uniforme durante la prestación del servicio. Los routers IP convencionales trabajan bajo el principio del mejor esfuerzo, lo que proporciona simplicidad y abaratamiento de costes en este tipo de redes. ATM proporciona en alguno de sus servicios el paradigma del “mejor esfuerzo” aunque en otros proporciona garantías sobre calidad.
- *Redes basadas en “calidad de servicio” (quality of service (QoS)).* En este caso hablamos de tecnologías que proporcionan mecanismos de control atendiendo al tipo de usuario, clase de flujos de datos, o son capaces de garantizar una calidad de servicio determinada para un flujo de datos por la demanda de una aplicación. No debemos confundir QoS con un alto nivel de desempeño (alto bit rate, latencia baja o probabilidad de error baja), ya que QoS tiene que ver en realidad con la capacidad de asegurar niveles de calidad. Hoy en día, los routers IP modernos pueden proporcionar una calidad de servicio garantizada a determinados flujos de datos (protocolos IntServ o DiffServ). De esta

manera estamos hablando hoy en día de telefonía IP o IP-TV, campos en los que es esencial garantizar niveles de calidad en términos de tasa de bit rates o latencias.

Otro requerimiento de DMIF es la ocultación de los detalles del “contexto operacional” donde se va ejecutar la aplicación. La idea es usar un interfaz común para acceder a los streamas en contextos muy diferentes: almacenamiento local, acceso remoto, difusión por broadcast, multicast, unicast, etcétera. Los beneficios de este interfaz común se centran en facilitar el proceso de diseño y producción de la aplicación multimedia.

5. Aspectos de la arquitectura de referencia DMIF.

Distinguimos en la arquitectura DMIF cuatro bloques fundamentales:

- La aplicación de origen.
- La implementación de DMIF origen.
- La implementación de DMIF destino.
- La aplicación destino.
- Elementos de control adicionales: DMIF Filter, módulos de señales.

Cuando hablamos de origen, lo hacemos desde el punto de vista del consumidor del contenido multimedia y nos referiremos a él como terminal. El DMIF origen y el DMIF destino trabajan juntos para proporcionar un servicio de nivel de sesión. La aplicación de origen es la aplicación en el terminal, por ejemplo un browser de MPEG-4 o una aplicación de videoconferencia. La aplicación de origen, interactúa con la aplicación destino mediante DMIF. En el caso de interacción remota, las dos aplicaciones residirán típicamente en hosts diferentes, y la comunicación entre ambas será regulada por un protocolo de señales (como se ha mencionada, el DMIF ha elaborado un protocolo genérico para este sistema de señales, el DMIF Default Signalling Protocol). En el caso de almacenamiento local o multicast/broadcast la aplicación de destino residirá en el mismo host que la aplicación remota y probablemente no será una aplicación en sentido usual, sino un proceso más del host.

El mundo de las aplicaciones se comunica con las implementaciones de DMIF mediante un interfaz de aplicación, denominado DMIF Application

Interface (DAI). Es importante comprender, que la utilización del DAI implica la separación de las capas de compresión y sincronización de MPEG-4, de la capa de distribución. Por ello, hay elementos que al ser comunicados mediante DAI, serán opacos para el DMIF, es decir no serán tratados de forma especial en función de su semántica, su significado sólo tendrá relevancia para la aplicación destino. Es el caso de los ES_IDs, o de los comandos emitidos por el usuario de la aplicación. De esta manera mantenemos la independencia de DMIF respecto a la posible evolución de las aplicaciones.

El elemento que hace de puerta de entrada del DMIF (lo podemos imaginar en contacto directo con el DAI es el denominado DMIF Filter. El DMIF Filter, está relacionado con la capacidad de instanciación del DMIF. Es decir, podemos tener en el sistema diferentes instancias de DMIF cada una con diferentes capacidades en función del contexto de la aplicación multimedia. El DMIF Filter permite seleccionar aquella instancia de DMIF más acorde a nuestras necesidades. La selección se realizará a partir de un URL que determina la instancia de DMIF que deseamos seleccionar, además de otros datos sobre las características del servicio.

La arquitectura DMIF establece una diferencia importante entre dos clases de contexto: el contexto interactivo remoto, y los contextos de almacenamiento local y el broadcast/multicast. El primero supone la existencia de una red entre la aplicación destino y la aplicación origen, además de un mecanismo de señales para gobernar la interactividad remota. Esta característica la proporciona el módulo de señales (DDSP), que será de aplicación sólo en aquellas instancias de DMIF que contemplan el contexto de interactividad remota. En este escenario DMIF además separa las características que no dependen de la tecnología de transporte de datos utilizada, de aquellas que son específicas de redes particulares. El interfaz entre la parte genérica de la particular se denomina DNI (DMIF-Network Interface).

Resumimos entonces los principales bloques de la arquitectura: aplicación origen, DMIF-Application Interfac (DAI), DMIF Filter, instancias de DMIF origen de dos tipos: almacenamiento local o broadcast/multicast e interacción remota, DMIF destino, módulo de señales (en caso de interacción remota) y aplicación remota.

6. El DMIF-Application Interface (DAI).

La misión del DAI es la de un API que oculta la tecnología de distribución de la multimedia y el contexto operacional. Un browser MPEG-4 que use el API será capaz de acceder al contenido multimedia de manera uniforme e independientemente del contexto operacional. Como se ha dicho, en MPEG-

4 el DAI es la línea divisoria entre la capa de sistemas (parte 1 del estándar) y el DMIF (parte 6).

El DAI comprende un conjunto de primitivas que podemos clasificar en tres tipos:

- *Primitivas de servicio.* Trabajan en el *control plane* (parte de la red que lleva información de control) y realizan la gestión de las sesiones de servicios (DA_ServiceAttach, DA_ServiceDetach).
- *Primitivas de canal.* Trabajan en el control plane. Realizan la gestión de los canales (DA_ChannelADD() y DA_ChannelDelete()).
- *Primitivas de datos.* Trabajan en el *user plane* (parte de la red que lleva el tráfico de los usuarios (datos)). Su propósito es la transferencia de datos por los canales establecidos (DA_Data() y DA_UserCommand()).

Además de estas primitivas, la implementación de un DAI debe incluir otras funciones como la inicialización, el reset, la consulta del estado etcétera. El conjunto de primitivas mencionado es suficiente para describir el comportamiento de una instancia de DMIF.

Recordemos ahora qué ocurre cuando los streams elementales cruzan a la capa de distribución. Para cruzar el DAI deben usar los canales previamente creados. Entonces pueden sufrir multiplexación / demultiplexación de la capa de distribución generando los streams FlexMux (los streams FlexMux son una herramienta de MPEG-4 especificada en MPEG-4 Systems). Estos streams son llevados a los canales TransMux donde son multiplexados en función de las pilas de protocolo de transporte elegidas (puertos IP, ATM VCs, MPEG-2 PIDs, etcétera).

Cada instancia de DMIF está a cargo de configurar la pila de protocolos que debe ser asociada a cada canal, así como realizar el seguimiento de la asociación entre los canales y los recursos de transporte.

7. El protocolo de señales de DMIF (DMIF Default Signalling Protocol).

El DMIF Default Signalling Protocol (DDSP) es un protocolo genérico en el nivel de sesión diseñado para el streaming de multimedia. El (DDSP) se usa entre otras cosas para configurar las pilas de protocolos de transporte conforme a los requerimientos de sincronización y calidad del contenido multimedia. En la pila de protocolos puede incluirse la utilidad FlexMux

(la podemos considerar en el nivel de sesión), que se especifica en MPEG4-Systems, lo que da la posibilidad a la implementación de DMIF de optimizar el consumo de recursos de la red (como por ejemplo el número de sockets usados, así como los requerimientos de QoS en cada conexión de red). La especificación de DMIF no concreta como realizar esta optimización pero da los medios para hacerla posible.

El DDSF se deriva del protocolo MPEG-2 DSM-CC User-to-Network-Protocol. Este último se diseñó para resolver el problema de establecer la sesión, así como la conexión en servicios de video por demanda entre un cliente y un servidor. Por el contrario, el DDSF está diseñado para cubrir el caso más general de una comunicación peer-to-peer, con interacción remota. En su primera versión, sólo se aborda el caso de redes homogéneas.

Como hemos mencionado, se trata de un protocolo de nivel de sesión. Por lo tanto, el primer paso consiste en abrir una sesión con un elemento de comunicación. Este proceso puede incluir mecanismos de autenticación y seguridad, partes que de momento caen fuera del alcance del DDSF. Una vez se ha establecido la sesión, la aplicación origen solita un número de streams particulares. Gracias al DMIF Signalling Protocol se creará un canal para cada stream solicitado, pudiendo multiplexar varios canales en un socket mediante la herramienta FlexMux. Será la aplicación la que se encargue entonces de comenzar la recepción de los streams.

No debemos olvidar que la cuestión del protocolo de transporte particular a ser utilizado queda abierta, por lo que es posible que el protocolo genérico de señales se solape con el mecanismo de señales particular del protocolo elegido. El protocolo genérico se hace específico para cada tecnología de red, integrando los mecanismos de señales particulares de cada tecnología. No se trata de duplicar funcionalidades, por lo que en caso de que la tecnología elegida ya contemple mecanismos del DMIF Signalling Protocol, estos se mapean en lugar de duplicarse.

El DMIF Signalling Protocol soporta el intercambio de información sobre QoS de la red, lo que le permite configurar el MPEG-4 FlexMux para mejorar en eficiencia. Este hecho debe tenerse en cuenta a la hora de elegir otras alternativas para la capa de distribución de los datos, como el Real Time Streaming Protocol (RTSP).

El módulo de señales se descompone en dos partes. La denominada DMIF Service Layer (DS), que se encarga de interactuar con el filtro DMIF y proporcionar los servicios requeridos por la aplicación y la DMIF Network Access Layer (DNA) que se encarga de todo lo relativo a los mensajes de control de la red e implementa el DDSF. Entre la capa DS y la DNA tenemos el DMIF Network-Interface (DNI), de manera que la capa DS accede a la capa DNI usando el interfaz DNI.

8. Descripción de la escena y localización de los componentes en MPEG-4.

Las tres herramientas usadas en MPEG-4 para describir la escena y localizar sus componentes son:

- *BIFS*. *El stream de descripción de la escena*. Especifica la escena y contiene punteros a los descriptores de objetos.
- *Streams descriptores de objetos (ODs)*. Describe cada ES en la escena, incluyendo su identificador.
- *Stream Map Table*. Esta tabla especifica para cada identificador de ES su localización física.

Ya se han explicado en otras partes del tema, las dos primeras herramientas. Éstas se caracterizan por ser completamente independientes de la capa de distribución. Por contra la tercera de las herramientas es totalmente dependiente de esta capa de distribución, por lo que será diferente en función de la implementación concreta.

9. Responsabilidades de una instancia de DMIF.

Sea cual sea la tecnología que implemente la instancia de DMIF, debe acometer las siguientes áreas:

- Herramientas de señales en el control plane.
- Pila de protocolos para el transporte de MPEG-4 ESs.
- Mecanismo para transportar el descriptor de objetos.
- Mecanismo para llevar la información de la Stream Map Table.

10. Funcionamiento del DMIF.

La aplicación de origen accede al contenido multimedia usando el DAI. Las peticiones de la aplicación son procesadas por el DMIF Filter, que determina qué instancia del DMIF debe servir las peticiones. Este proceso es

completamente transparente a la aplicación, ésta no conocerá qué instancia de DMIF es finalmente asignada. Una aplicación podrá tener asociada diferentes instancias de DMIF.

Por lo tanto, el proceso comienza con la selección de un servicio a activar. Este servicio se identifica por un URIL de DMIF. Para solicitar el servicio, la aplicación usa la primitiva del DAI denominada `DA_ServiceAttach`. El DMF Filter, examina el DMIF URL pasado por la aplicación y determina la instancia de DMIF que debe proporcionar el servicio. Este módulo contacta con la instancia de DMIF destino (usando el DMIF Signalling Protocol o mecanismo equivalente en caso de un escenario de interactividad remota). Por su parte, la instancia de DMIF destino tendrá que identificar y contactar con la aplicación destino. También tendrá que establecer una sesión de servicio con ella. En caso de escenarios remotos interactivos, tendrá que establecer una sesión de red también. Una vez localizado el servicio, la aplicación destino devuelve una respuesta positiva que es transmitida a la aplicación origen. Si es una aplicación de MPEG4-Systems, esta respuesta incluirá el descriptor del objeto inicial de ese servicio. Las aplicaciones usarán entonces la sesión de servicio para crear conexiones que son usadas para transportar los datos.

Una aplicación basada en MPEG-4 Systems, realizará el parsing del Initial OD, y realizará peticiones para abrir los canales de los streams que se incluyen en el objeto inicial (esto streams suelen ser el BIFS y streams descriptores de objetos).

Cuando la aplicación de origen realiza la petición de los streams, usa la primitiva `DA_ChannelAdd` del DAI, donde se debe indicar el servicio al que pertenecen. También debe indicar los streams a transmitir en ese canal. Esto se transmite a la aplicación destino. El DMIF tratará la información sobre los streams de manera transparente, es la aplicación destino quien interpreta estos datos.

La aplicación de destino tendrá que localizar los streams pedidos por la aplicación origen. En el caso de escenarios remotos interactivos, las instancias de DMIFs comunicantes, deben configurar los canales conforme a los protocolos apropiados (por ejemplo, el DMF Signalling Protocol). Los parámetros sobre QoS para los streams solicitados y pasados al DAI influenciarán la configuración de las conexiones de red, dependiendo de los criterios que el emisor adopte para agregar los streams elementales en un socket (por medio de MPEG-4 FlexMux).

La aplicación emisora usará el canal para enviar los streams (primitiva `DA_DATA` del DAI). Esta emisión se realizará conforme a los comandos de controles enviados por la aplicación receptora de los datos (por ejemplo, mediante la primitiva `DA_UserCommand` del DAI).

Así pues, la aplicación origen basada en MPEG-4 Systems, recibirá los

streams de BIFS y OD, los decodificará y en función de los mismos, realizará la petición de nuevos streams elementales, usando la primitiva `DA_ChannelAdd` del DAI.

Si un stream ya no es necesario, la aplicación de origen usará la primitiva `DA_ChannelDelete` del DAI para liberar el canal. Si un servicio ya no es necesario, la aplicación origen usará la primitiva `DA_ServiceDetach` para liberar el servicio.

Veamos como funcionaría esto en un escenario donde se accede localmente a la información. En este caso el servicio es representado por un conjunto de ficheros, que de forma cooperativa constituyen el contenido multimedia requerido. Cuando la aplicación origen requiere un servicio particular (identificado con un DMIF URL), la aplicación destino, hará el parsing del URL y determinará el servicio que se está pidiendo. Entonces obtendrá el Initial OD para el servicio (que será enviado a la aplicación origen) así como la Stream Map Table. Con el Stream Map Table, la aplicación destino será capaz de satisfacer peticiones posteriores para nuevos canales, ya que podrá buscar en la tabla por `ES_ID`.

Cuando se trata de un escenario con un servidor remito, esto cambia ya que el servicio es organizado por el servidor. Ante una petición de servicio, la instancia de DMIF origen lee la parte inicial del URL y determina la dirección del servidor. El DMIF destino por su parte identificará la aplicación que es reponsable del servicio que se solicita y se establecerá una conexión lógica entre la aplicación origen y la aplicación destino. EL servidor proporcionará a la aplicación origen el Initial OD. En este escenario, el Stream Map Table es generado incrementalmente según las demandas de la aplicación origen. Esto es así porque el servidor establecerá como disponible los recursos en función de la demanda.

El escenario de broadcast es similar al escenario local. La diferencia es que en el escenario local, puede usarse una Stream Map Table estática, ya que la localización de los streams va a permanecer sin cambios. Sin embargo en el broadcast, los canales pueden cambiar (se eliminan y añaden canales). Hay que tener en cuenta que en el escenario de broadcast un servidor debe enviar periódicamente a los clientes el objeto inicial y el Stream Map Table conveniente actualizados, de forma que cada cliente pueda iniciar la sesión correctamente.

11. Bibliografía

Referencias

[Pourmohammadi et al., 2001] Pourmohammadi, Y., Haghghi, K., Mohamed, A., y Alnuweiri, H. (2001). Streaming mpeg-4 over ip and broadcast networks: Dmif based architectures. Proceedings of the 11th International Packet Video Workshop. Accesible desde internet en: <http://lan.ece.ubc.ca/DMIF-PV2001.pdf>.