

Control de congestión de vídeo escalable en MANEs (Media Aware Network Element) para aplicaciones de streaming

Jordi Ortiz Murillo

jordi.ortiz@um.es

MEMORIA TESIS DE MÁSTER

Máster en "Tecnologías de la Información y Telemática Avanzadas" – Curso 2008/09
Dpto. Ingeniería de la Información y las Comunicaciones
Dpto. Ingeniería y Tecnología de Computadores
Facultad de Informática. Universidad de Murcia.
Campus de Espinardo. 30100 Murcia. Spain.

Resumen Extendido

Desde que las redes de información tienen capacidad suficiente, uno de los anhelos de los investigadores y desarrolladores ha sido el de transmitir vídeo sobre la misma. Mientras algunos se centraban en la transmisión de vídeo en tiempo real, otros, se dedicaron a mejorar los sistemas de video bajo demanda o streaming de vídeo. Este último tipo de aplicaciones se ha visto altamente potenciada últimamente debido a las mejoras de la capacidad de la red, sobretodo en la denominada 'last mille'. Las operadoras están cada vez más interesadas en evolucionar de proveedores de servicio de red a proveedores de contenidos, en particular, servicios de vídeo bajo demanda que les permitirá diversificar su negocio. Sin duda los servicios de streaming van a suponer un gran reto en los próximos años.

Los servicios de streaming procuran llegar al mayor número de dispositivos, y por tanto de usuarios, posible. Hasta el momento, para lograr este objetivo ha sido necesario hacer uso de métodos de transcodificación mediante los cuales, bien el servidor o bien un punto intermedio (o MANE), se encarga de convertir el vídeo al formato soportado por el cliente. Este método no es para nada deseable, ya que supone un gran consumo de recursos y de energía, por parte del proveedor de contenidos en el primer caso o de la red en el segundo. Cualquiera que sea la solución empleada, no escalará.

Una solución escalable desde el punto de vista de la red es la de tener distintas versiones del mismo vídeo almacenadas. De esta forma el usuario podría seleccionar la versión que más se ajuste a sus necesidades o el propio proveedor podría tomar esta decisión por él. El problema en este caso surge en cuanto a la necesidad de almacenamiento por parte del proveedor del mismo contenido replicado; algo totalmente indeseable.

En Noviembre de 2007, el ITU-T, agregó al estándar H.264/AVC su extensión para escalabilidad conocida como H.264/SVC. Este anexo supone una revolución para los sistemas de streaming puesto que se centra en ofrecer escalabilidad a la hora de codificar, transmitir y decodificar un vídeo. Aunque esta no es la primera vez que se intenta producir un codec de vídeo escalable, sí que es la primera que se aborda el problema de una forma tan directa. Con este codec se permite mejorar las propiedades de un vídeo mediante adiciones incrementales sobre el mismo, es decir, permite lo que se ha denominado un sistema de capas de mejora. Así, un proveedor de servicios podrá tener una única versión del vídeo con múltiples capas que permitirán emplear el servicio a distintos tipos de usuario. De forma adicional, este tipo de codificación permite la adaptación del contenido por la red con un bajo coste computacional; un punto intermedio de la red puede desechar al vuelo la información que no podrá ser empleada por los clientes, lo que producirá además un ahorro de ancho de banda en el último tramo de la red. También está contemplada la transmisión de distintas capas por distintos flujos. La aplicación directa de esta capacidad es la de emplear flujos multicast a los que el cliente se podría suscribir dependiendo de sus necesidades o posibilidades. La cruda realidad es que a día de hoy no todas las redes soportan este tipo de transmisión, de ahí que sea la red la encargada de adaptar el contenido al usuario.

Con esto podría quedar solucionado el problema de cómo suministrar al usuario los contenidos que desea en el formato que necesita. No obstante, existe otro problema relacionado con la transmisión de flujos multimedia en general y de los servicios de streaming en particular. Estos servicios son altamente acaparadores de ancho de banda, lo que produce problemas en el resto de la red. Estos problemas son debidos, en gran medida, a la congestión que se produce debido a la falta de control de flujo de los protocolos empleados para la transmisión de vídeo. Controlar la congestión provocada por este tipo de servicios debe ser una prioridad de ahora en adelante.

Nuestro objetivo es el de diseñar y estudiar un algoritmo de adaptación guiado por las preferencias de cliente y responsable con el uso de los recursos de red. Este objetivo difiere a lo ya expuesto en otras publicaciones[1][2] en el énfasis en las preferencias del cliente. El requisito surge en el momento en que se intentan diseñar mecanismos de decisión para la adaptación en la red. Un ejemplo de estos mecanismos son las entidades denominadas ADTE (Adaptation Decision Taking Engine) diseñadas en proyectos como DANAE[3] o SCALNET[4]. Este último, todavía en desarrollo y del que la Universidad de Murcia forma parte, pretende hacer un análisis exhaustivo de la tecnología SVC. Es aquí precisamente donde esta tesis encuentra su razón de ser, suponiendo una primera aproximación a lo que, finalmente, verá la luz en dicho proyecto.

1. Introducción

En esta tesis de máster se pretende introducir al lector a las herramientas existentes para la codificación y transmisión de vídeo escalable. Después se propondrá una solución al problema que supone la adaptación de contenidos en la red. A su vez, se procurará centrar esa adaptación en dos ideales, el uso responsable del ancho de banda disponible y las preferencias del usuario.

Así, en el capítulo 2, se describirá el codec del ITU-T H.264/AVC junto con su extensión de escalabilidad SVC. También se introducirán brevemente algunos de los protocolos más empleados en servicios de streaming. En cuanto al control de congestión, se explicará brevemente el concepto de flujo compatible con TCP (TCP Friendly). Finalmente se describirán algunas técnicas existentes para la descripción de las capacidades y preferencias del usuario.

En el capítulo 3, se procurará hacer un análisis de cómo emplear SVC para adaptar los contenidos y finalmente ofrecer un algoritmo de adaptación basado en preferencias de usuario y compatible con TCP.

En el capítulo 4, se mostrarán los resultados de las simulaciones del algoritmo propuesto para un escenario simple.

En el capítulo 5, se darán unas conclusiones al trabajo realizado y vías futuras por donde poder continuarlo.

2. Estado del arte

2.1. Compresión de vídeo escalable

MPEG-4 Parte 10 o H.264/AVC

H.264/AVC[5][6] (Advanced Video Coding) es un estándar para la compresión de vídeo del ITU-T Video Coding Experts Group (VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG). Estos dos grupos juntos forman el Joint Video Team (JVT).

El objetivo del estándar es proveer un codec flexible capaz de ejecutarse en diferentes aplicaciones, redes y/o sistemas sin incrementar demasiado la complejidad de diseño, y consiguiendo una buena calidad de vídeo con tasas de bits (bitrates) menores que en estándares previos. De hecho, H.264/AVC ha conseguido una mejora importante en cuanto a eficiencia ratio-distorsión (relativa a la cantidad de entropía necesaria para enviar la información sin superar un umbral de distorsión) respecto a otros estándares existentes. La primera versión del trabajo final de prototipado del estándar fue finalizada en Mayo 2003.

De manera similar a otros codecs de vídeo, un vídeo codificado con AVC consiste en una secuencia de imágenes, es decir, unidades de acceso. Cada unidad de acceso, a su vez, contiene todos los datos necesarios para decodificar exactamente una imagen. Existen básicamente tres tipos de imágenes. Imágenes con decodificación intra (I), las cuales no emplean ninguna información anterior a su ubicación en el flujo de datos, por tanto proveen de la capacidad de tener acceso aleatorio sobre el flujo de datos. Las imágenes codificadas de forma predictiva (P) y las imágenes codificadas de forma bi-predictiva (B) aprovechan la redundancia temporal en un vídeo, codificando únicamente aquellos pixels que difieren comparados con alguna imagen de referencia. Un hecho adicional importante es que existen distintas reglas de dependencia para los distintos tipos de imágenes. Específicamente, las imágenes P podrían únicamente depender de otras imágenes P o I para aprovechar la redundancia temporal. Las imágenes tipo B, por otra parte, podrían depender de cualquier tipo de imagen a la hora de aprovechar la redundancia temporal. El concepto de imágenes B dependiendo de otras imágenes B permite imágenes B jerárquicas. Las hojas de esta jerarquía no son necesarias por ninguna otra imagen en el flujo de datos.

Aplicaciones y Resumen de mejoras de diseño.

En un principio, el estándar estaba enfocado a vídeo de baja calidad para aplicaciones de Internet y vídeo-conferencia incluyendo áreas de aplicación. El estándar, debía ser independiente de la aplicación y de la red. Para lograr esta necesidad, el diseño del H.264/AVC especifica una capa denominada 'Video

Coding Layer (VCL)', que fue diseñada para la codificación del vídeo, y una capa denominada 'Network Abstraction Layer (NAL)', que convierte la representación VCL del vídeo en el formato correcto de acuerdo a la capa de transporte o medio de almacenamiento en que vayan a ser usados.

Vídeo Coding Layer (VCL)

Emplea una aproximación de codificación de vídeo basada en bloques híbrida, en la que cada imagen es representada mediante el uso de macrobloques. Emplea el concepto de Imágenes, Marcos (frames) y Campos (fields). Una secuencia de vídeo consiste en una secuencia de imágenes decodificadas, donde una imagen decodificada puede representar un marco completo o un único campo en el caso de vídeo entrelazado (interlaced). Hace uso del espacio de colores YCbCr (Luminancia, Cromo Azul y Cromo rojo) y muestreo 4:2:0. Define el concepto de Slice(trozo) y grupos de Slices. Generalmente aunque también es posible realizar otros tipos de agrupaciones de macroblocking en slices, para aumentar la resistencia frente a errores de transmisión los Slices se componen como secuencias de macrobloques en el orden de barrido por escaneo (raster scanning).

Durante el proceso de codificación y decodificación de macrobloques, para la codificación de las transformadas, cada componente de color de la señal residual de predicción es subdividida en bloques de 4x4. Para proveer de gran eficiencia de codificación al codificar frames progresivos, el diseño de H264/AVC permite al codificador combinar de distinta forma los fields. Posee predicción Intra-Frame e Inter-Frame. H.264/AVC soporta dos métodos de codificación entrópica, el más simple y la Codificación de longitud adaptativa variable (Context Variable Length Coding - CAVLC).

H.264/AVC usa algunas técnicas de compresión de vídeo de codecs precedentes, e incluye mejoras para aumentar la habilidad de predecir los valores del contenido de una imagen a codificar. Soporta compensación de movimiento de tamaño de bloque variable para tamaños de bloque pequeños: flexibilidad en la selección de los tamaños de bloque para la compensación de movimiento y formas con un tamaño mínimo de bloque de 4x4. Compensación de movimiento con múltiples imágenes de referencia. Esta es una de las mejoras más relevantes. El estándar permite hasta dieciséis referencias a imágenes codificadas previamente. También permite desacoplar el orden de referencia del orden de reproducción, así como desacoplar los métodos de representación de las imágenes de la capacidad de referenciación de la misma. Las imágenes de tipo B pueden ser empleadas para compensación de movimiento.

La señal de predicción de compensación de movimiento puede ser balanceada en cantidades definidas por el codificador. Posee inferencia de movimiento mejorada y predicción espacial direccional para codificación intra.

Introduce el concepto de 'In-the-loop deblocking filtering' ayudando a prevenir elementos bloqueantes comunes a otras técnicas de compresión basadas en DCT resultando en una mejor apariencia visual y eficiencia de compresión.

Mejoras de codificación en otras partes del diseño:

- Transformada de tamaño de bloque pequeño. El diseño de H.264/AVC está basado principalmente en un tamaño de bloque para la transformada de 4x4.
- Transformada de bloque jerárquica: Selección adaptativa del codificador entre tamaños de bloque de 4x4 y 8x8.
- Transformada de longitud de palabra corta. El diseño de H.264/AVC requiere únicamente una aritmética de 16-bits.
- Transformada inversa ajustada con exactitud: H.264/AVC es el primer estándar en conseguir igualdad exacta del contenido de vídeo de todos los decodificadores.
- Codificación aritmética entrópica: El estándar incluye un método de codificación entrópica avanzado conocido como CABAC (context-adaptative binary arithmetic coding).

Otras mejoras que resaltan relacionadas con la robustez a errores/pérdidas de datos y flexibilidad de operación sobre gran variedad de entornos de red son: Estructura de sintaxis NALU. Tamaño de slice flexible. Orden de macrobloques flexible (FMO). Orden arbitrario de slices (ASO). PSlices redundantes, mejorando la robustez ante error/pérdida permitiendo al codificador enviar una representación de una región de imagen extra que pueda ser empleada si la representación primaria está corrupta o simplemente se perdió. El diseño del estándar separa elementos sintácticos más relevantes y menos relevantes

en diferentes paquetes de datos, permitiendo la aplicación de protección frente a errores. Introduce el concepto de Imágenes SP/SI de sincronización/intercambio para introducir saltos en medio del flujo de vídeo.

Network Abstraction Layer (NAL)

Su misión es hacer la conexión con la red del Video Coding Layer (VCL) más amigable. NAL da formato a la representación VCL del vídeo para permitir usar VCL en multitud de sistemas y redes como por ejemplo RTP/IP, formatos de ficheros o servicios de broadcasting MPEG-2. Algunos conceptos claves relacionados con NAL son:

- NAL units (NALU): Los datos del vídeo codificado están organizados en NAL units. Son paquetes que contienen un número entero de bytes/octetos donde el primero de ellos es una cabecera y los otros contienen carga útil.
- Uso de NALUs en formato de flujo de bytes: Algunos sistemas necesitan un formato de flujo de bytes para las NALUs. Para ser usado en dichos sistemas, la especificación de H.264/AVC define la forma de concatenar las NALUs en un único flujo.
- Uso de NALUs en sistemas de transporte de paquetes: Otros sistemas usan paquetes para transportar los datos codificados. NAL adapta la información VCL para estos casos, encapsulando un número entero de NALUs o un fragmento de una única NALU.
- VCL y no-VCL NALUs: Las VCL NALUs contienen los datos que representan valores de muestras en las imágenes de vídeo, y las no-VCL contienen otra información como conjuntos de parámetros (denominados parameter sets).
- Conjuntos de parámetros o Parameter Sets: Un conjunto de parámetros contiene información de control que probablemente no va a cambiar durante la decodificación de un número grande de VCL NALUs. Los Parameter Sets pueden ser Picture Parameter Set, relativos a cero o más imágenes que forman una secuencia, o Sequence Parameter Set, relativos a cero o más secuencias de vídeo.
- Unidades de Acceso o Access Units (AU). Las Access Units son un conjunto de NALUs relativas a una imagen.
- Secuencias de Vídeo Codificadas: Son Series de AUs que son secuenciales en el flujo de NALUs y usan únicamente un Sequence Parameter Set. La propiedad especial de estas series es que pueden ser decodificadas independientemente de cualquier otra secuencia de vídeo. También son denominados GOPs (Groups of Pictures).

H.264/SVC Introducción

En esta sección se va a hacer una breve descripción de los rasgos, en lo que a escalabilidad se refiere, del MPEG-4 Scalable Video Codec (SVC)[6][4], recientemente estandarizado como extensión del MPEG-4 Advanced Video Codec (AVC). El resultado es un codec de vídeo escalable basado en bloques híbrido con una capa base compatible con AVC.

Video Coding Layer

La jerarquía de imágenes B es una propiedad de escalabilidad importante puesto que permite la escalabilidad temporal. En la Figura1 las imágenes están divididas en capas en base a sus dependencias. De hecho, se podrían quitar todas las imágenes de la capa superior (T3) sin ninguna dependencia o problemas de decodificación. Esto reduciría el frame rate del flujo de datos, resultando un flujo de datos al que se le podría aplicar el mismo proceso de adaptación si se desea (ver Figura1). Se debe tener en cuenta que el número de capas temporales es configurable cambiando el tamaño del GoP (Group of pictures).

Junto con la dimensión temporal, el contenido SVC puede ser escalado también en la dimensión espacial. Resoluciones espaciales distintas pueden ser embebidas en el mismo flujo de datos, por ejemplo, CIF y 4CIF. Esto se consigue codificando imágenes como distintas NALUs VCL. En el ejemplo anterior, la primera unidad VCL contiene toda la información para decodificar la imagen en resolución CIF y la segunda unidad VCL contiene la información adicional necesaria para decodificar la imagen en formato 4CIF. Esto permite reducir fácilmente la resolución espacial simplemente desechando todas las unidades VCL correspondientes a la resolución 4CIF.

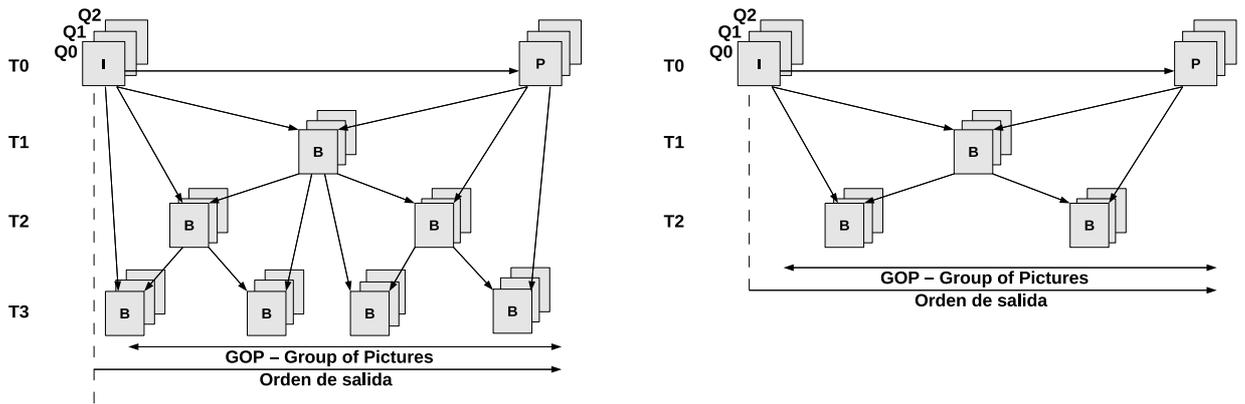


Figura1. Esquema de un grupo de imágenes SVC y de la omisión de una capa SVC

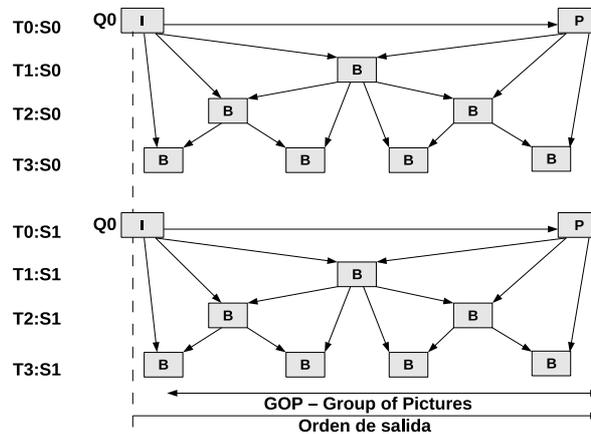


Figura2. Esquema de un grupo de imágenes SVC con CGS

El mismo mecanismo puede ser empleado para lograr escalabilidad en la dimensión de calidad. No obstante, en este caso la información adicional de la segunda unidad VCL no es usada para convertir la imagen a 4CIF, sino más bien para mejorar la calidad visual para la resolución CIF. Este tipo de escalabilidad se denomina Coarse Grained Scalability (CGS) o Escalabilidad de grano grueso (Figura2).

Estas tres dimensiones de escalabilidad tienen en común que entre cualesquiera dos imágenes clave, únicamente pueden ser desechadas capas completas. Las imágenes clave pueden ser I o P que están especialmente codificadas para permitir el intercambio entre niveles de escalabilidad. Para la escalabilidad en la dimensión de calidad se deseaba una forma más depurada (fina) de escalabilidad. Por tanto, la Medium Grained Scalability (MGS) o Escalabilidad de grano medio fue introducida. MGS se realiza de la misma forma que CGS, con la diferencia que con MGS se puede cambiar cada unidad de acceso en lugar de sólo en las imágenes clave. Es decir, las unidades VCL pertenecientes a una capa MGS pueden ser desechadas individualmente en cada unidad de acceso y por tanto permiten ajustar la tasa de transferencia (bitrate) en el contenido SVC con una granularidad más fina.

En cualquier caso, para decidir selectivamente qué unidades VCL pertenecen a qué tipo de capa, temporal, espacial o de calidad, esta información necesita ser incluida en la unidad VCL, de lo que se encarga la capa NAL.

Parameter Sets (PS) y Supplemental Enhancement Information (SEI)

No son unidades VCL. Un PS contiene información aplicable a un gran número de unidades VCL de una capa específica en casos en los que sería ineficiente codificar esta información para cada unidad VCL. La resolución espacial de un segmento de vídeo de una capa específica es un ejemplo de información que es incluida en un PS.

Los mensajes SEI proveen información suplementaria que no es necesaria para el proceso de decodificación, pero que podría ser útil para el procesamiento del flujo de datos. Esencialmente transportan información del límite de la capa que indica los valores máximos para el nivel temporal, de calidad e identificador de dependencia (Ver sección NAL) para todas las unidades VCL del flujo de datos. Adicionalmente contienen información de ancho de banda (bitrate) para cada capa en el flujo de datos escalable.

Network Abstraction Layer (NAL)

Para poder -entre otras cosas- decidir selectivamente qué unidades VCL pertenecen a qué capa, la denominada NALU Header se añade al principio de cada unidad VCL. Este paso de preceder las unidades VCL con NALU headers es el segundo tras la codificación de imágenes a la hora de componer un flujo SVC, y las unidades VCL resultantes son denominadas Network Abstraction Layer Units (NALUs). La NALU header se muestra a continuación:

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
F	NRI	Type	R	I	PRID	N	DID	QID	TID	U	D	0	RR																		

NALU Header

Seguidamente nos centraremos en esos campos de la cabecera que son importantes en relación a la adaptación. Estos son priority id (PRID), temporal id (TID), dependency id (DID), quality id (QID) y el discardable flag (D).

El PRID es el campo de 6 bits que especifica una propiedad de prioridad específica para la aplicación. Originalmente indicaba a menor valor menor prioridad, pero esta restricción fue eliminada recientemente.

El DID es el campo de 3 bits que especifica la dependencia entre capas para CGS y escalabilidad espacial. Las NALUs con un mayor DID pueden depender de NALUs con menor DID, pero nunca de manera inversa. Esto quiere decir que si las NALUs con DID x son desechadas, entonces cualquier NALU con DID x o mayor tienen que ser desechadas también entre cualesquiera dos imágenes clave. Con el fin de encontrar si una NALU provee CFS o mejora espacial, el Sequence Parameter Set (SPS) para el segmento actual del flujo de datos necesita ser investigado. Esta dependencia entre NALU header y SPS (entre otros) muestra que mientras la NALU header provee cierta información que puede ayudar a la adaptación, su mayor funcionalidad sigue siendo la de decodificar el flujo de datos, más que su adaptación.

El QID es el campo de 4 bits que especifica la jerarquía de niveles de calidad de una NALU MGS. De forma parecida a lo anterior, las MGS NALUs del nivel superior dependen de las MGS NALUs del nivel inferior, por tanto el máximo nivel puede ser desechado para escalabilidad de calidad.

El TID es un campo de 3 bits que especifica la jerarquía de nivel temporal para la NALU actual. Las mismas reglas que para QID son aplicables aquí.

El discardable flag es un bit que indica si la NALU actual se necesita para decodificar NALUs de la imagen actual. Adicionalmente, si se activa, esta NALU no es necesaria por ninguna otra NALU en imágenes sucesivas que tengan mayor DID que la actual, por tanto, esas NALUs pueden ser desechadas sin arriesgar la integridad de las capas con mayor DID. Cuando también se tiene en cuenta la definición de DID, esto significa que las NALUs que tienen este bit activo pueden ser siempre ignoradas sin restricción.

Las NALUs VCL AVC son mantenidas en los flujos de datos para compatibilidad hacia atrás con los reproductores AVC. Para estas NALUs VCL AVC se introduce una prefix NALU SVC, que consiste en un preámbulo y 4 octetos de cabecera SVC que indican las características de escalabilidad de la siguiente NALU.

Opcionalmente, cada NALU puede ser iniciada con un preámbulo inalterable de 3 octetos (0x000001). Esto es necesario para extraer las NALUs de fichero o de otros protocolos de transporte que no provean de sistema de paquetes.

2.2. Transmisión Multimedia

RTP Real-Time Transport Protocol RTP es el protocolo de transporte de datos con requisitos de tiempo real definido por el IETF. El reto para los diseñadores de RTP[7] fue construir un mecanismo para la entrega de contenido multimedia robusto y en tiempo real sobre una capa de transporte no confiable. Consiguieron su objetivo siguiendo los principios 'application-level framing' y 'end-to-end principle'.

Application-Level Framing → Este principio se basa en que sólo las aplicaciones tienen suficiente conocimiento de sus datos para tomar una decisión sobre cómo deberían ser transportados los mismos. Lo que implica que el protocolo de transporte debería aceptar y exponer los detalles de su entrega en tanto en cuanto sea posible de tal forma que la aplicación pueda tomar la decisión apropiada en caso de error. Estas técnicas dan a la aplicación gran flexibilidad para reaccionar a los problemas mejor que limitarlas mediante las directrices de la capa de transporte. Implica que las aplicaciones son conscientes de la red (network-aware) e inteligentes, capaces de reaccionar a los problemas.

End-to-End principle → Se basa en que la responsabilidad de los datos es de los extremos de la comunicación, asegurando la fiabilidad extremo a extremo aunque los saltos uno a uno no sean confiables. El resultado es un diseño que implica extremos inteligentes y conscientes de la red y una red best effort. Este diseño se ajusta perfectamente al modelo de Internet.

Elementos estándar de RTP

El estándar principal para transporte de audio y vídeo en las redes IP es RTP [8], junto con los perfiles y los formatos de carga útil asociados. RTP provee un marco para el transporte en tiempo real y necesita ser perfilado (profiled) para usos particulares antes de considerarse completo. Algunos de estos perfiles fueron definidos junto con RTP y otros muchos están siendo definidos. Junto con el perfil se necesita especificar un formato de carga útil para cada tipo de medio.

Especificación de RTP

Posee dos partes: El protocolo de transferencia de datos y su protocolo de control asociados. El primero administra la entrega de datos en tiempo real entre sistemas finales. Define un nivel adicional de encapsulado para carga útil multimedia, incorporando un número de secuencia para la detección de pérdidas, marcas de tiempo (timestamp) para permitir la recuperación de sincronización, tipo de carga útil, identificadores de fuente y una marca para eventos relevantes dentro del flujo multimedia.

El protocolo de control de RTP (RTCP) provee de feedback de la calidad de recepción, identificación de participantes, y sincronización entre flujos multimedia. RTCP funciona en combinación con RTP y provee notificación periódica de esta información. Si bien los paquetes de datos son enviados cada pocos milisegundos, las notificaciones son enviadas en términos de segundos. Esta información puede ser útil para adaptar la transmisión dependiendo de la información devuelta relacionada con calidad o para identificar participantes.

RTP soporta la noción de mezcladores (mixers) y traductores (translators), cajas intermedias que pueden operar sobre los datos propios mientras fluyen entre los extremos. Estos conceptos se extienden en el Apéndice A.

Es complicado situar a RTP dentro del modelo de referencia OSI puesto que hace funciones de transporte, si bien no es un protocolo de transporte completo. Por otra parte también realiza funciones de la capa de sesión y de la capa de presentación.

Perfiles RTP

Es importante tener en cuenta los límites de RTP puesto que fue definido de forma incompleta deliberadamente. El protocolo no define algoritmos para la reproducción de contenidos o regeneración de sincronización, ocultación de errores o control de congestión. Debido a la distinta naturaleza de las aplicaciones que emplean RTP no sería lógico definir todos estos algoritmos, si bien será necesario

El timestamp por su parte se inicializa con un valor aleatorio e identifica el instante correspondiente al primer octeto de los datos transportados en el paquete. Este valor debe estar basado en un reloj que se incremente de forma monótona, produciendo una única línea de tiempo. El ciclo de valores para el timestamp está contemplado por el estándar. Que tanto el sequence number como el timestamp tengan números continuos y que puedan inicializarse de forma aleatoria implica que un servidor de medios no puede tenerlos almacenados sino que debe generarlos al vuelo.

El SSRC se genera aleatoriamente por cada participante al inicio de la sesión y expira al final de esta. En caso de colisión, se debe regenerar tras indicárselo al otro participante con el que se generó la colisión.

Los CSRC son empleados en casos en los que la sesión pueda tener múltiples fuentes. Estos identificadores indican los participantes que han contribuido al paquete RTP pero sin contribuir a su coordinación o sincronización. Este tipo de paquete se genera en los mezcladores (mixers) y transcodificadores (transcoders).

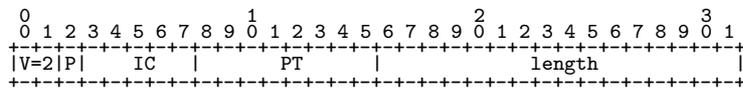
Cabeceras de extensión → Cuando el bit X lo indique, se pueden agregar cabeceras de extensión de longitud variable. Comienzan con una cabecera específica de 32 bits, 16 de tipo de cabecera de extensión y 16 de longitud de la cabecera, permitiendo a receptores incapaces de entender el significado de la cabecera ignorarla. No suelen ser empleadas fuera de ámbitos de investigación. Un diseño que necesite una cabecera mayor debería plantearse crear un nuevo perfil. En cualquier caso, su análisis para descarte debe ser soportado por toda implementación robusta.

Cabeceras de carga útil → La cabecera RTP proporciona información que es común a todos los formatos de carga útil. En muchos casos un formato de carga útil necesitará más información para operar óptimamente. Esta información forma una cabecera adicional que está situada justo tras todos los CSRC definidos en la cabecera RTP. Los valores a incluir en esta cabecera pueden ser estáticos para todas las sesiones, o dinámicos. En este último caso necesariamente se deberán especificar por sesión, por ejemplo mediante SDP.

RTP Control Protocol RTP[7][8] provee de notificación periódica de calidad, identificación de participantes y otra información de descripción del origen, notificación de cambios en los asistentes a una sesión y la información necesaria para sincronizar los flujos de medios.

Paquetes RTCP Cada participante debe enviar un conjunto de paquetes RTCP y en respuesta recibirá un conjunto de paquetes RTCP enviados por todos los participantes en la sesión. La filosofía peer-to-peer de RTP obliga a notificar a todos los participantes, ya sea directamente o mediante una entidad intermedia encargada de replicar.

Existen cinco tipos de paquetes RTCP: Receiver Report (RR), Sender Report (SR), Source Description (SDS), Membership Management (Bye) y Application-Defined (APP). Todos siguen la misma estructura si bien existen pequeñas diferencias dependiendo del tipo de paquete.



V, al igual que en RTP, hace referencia a la versión, actualmente 2. P indica si existirá relleno en el paquete. IC o Item Count: algunos paquetes llevan implícita una lista de elementos y con este campo se puede indicar el número de ellos. PT es el tipo de paquete y como se ha comentado hay 5 posibilidades. Length es la longitud del contenido del paquete que sigue a la cabecera común.

Los paquetes RTCP nunca son transportados individualmente. En lugar de eso son siempre agrupados juntos para transmisión, formando un grupo de paquetes. Cada grupo de paquetes es encapsulado en un único paquete de la capa inferior para ser transportado.

El resto de los paquetes se muestran para curiosidad del lector en el Apéndice C.

2.3. Señalización / Control de transmisión

SDP y RTSP SDP - Session Description Protocol

SDP es el protocolo de descripción de sesiones multimedia definido por el IETF. El logro de SDP[9][10] es expresar suficiente información para que un terminal receptor de la descripción de la sesión pueda unirse a la misma. En el caso multicast, la descripción sirve a su vez como descubrimiento de la misma.

Las sesiones SDP son codificadas en texto plano US-ASCII y UTF-8. Una descripción de sesión SDP consiste en múltiples líneas de texto separadas por caracteres CRLF (Retorno de carro más fin de línea), aunque se recomienda aceptar también LF únicamente. El formato de las líneas es:

$$\langle type \rangle = \langle value \rangle$$

donde $\langle type \rangle$ puede constar únicamente de un carácter y $\langle value \rangle$ constará de cadenas alfanuméricas separadas por espacios. Por tanto, no puede existir ningún espacio a los lados del carácter '='. Los campos SDP pueden ser clasificados en nivel de sesión o de medio. El nivel de sesión lo forman aquellos campos cuyos valores son importantes para toda la sesión y todos los flujos de medios. El nivel de medios se refiere a aquellos valores que sólo se aplican a un flujo de medios particular.

Por tanto, el mensaje de descripción de sesión consistirá en una sección de nivel de sesión seguida por cero o más sesiones de nivel de medios. Para delimitar las secciones se emplean los propios identificadores de tipo. Esto es así para simplificar los analizadores sintácticos SDP. El orden de los distintos atributos es estricto. La primera línea de tipo 'm' indica el inicio del nivel de medios. En [9] se definen los tipos presentados en el Apéndice D.

Agrupamiento de medios en SDP

En [11] se definen dos nuevos atributos que permiten describir las relaciones entre los medios definidos en una descripción SDP.

- mid-attribute → Se emplea para identificar flujos de medios dentro de una sesión SDP. Sigue el formato:

```
mid-attribute = "a=mid:" identification-tag
```

Donde la identification-tag o etiqueta de identificación debe ser única para toda la sesión.

- group-attribute → Se emplea para agrupar distintos flujos de medios definidos con un 'mid-attribute'. Sigue el formato:

```
group-attribute = "a=group:" semantics
                  *(space identification-tag)
semantics       = "LS" | "FID"
```

Donde LS significa Lip Synchronization (Sincronización de labios) y se emplea en sincronización, mientras que FID significa Flow Identificatino (Identificación de flujos) y se emplea para casos no abarcados por RTP y RTSP como pueden ser sesiones SIP de telefonía móvil. El * indica 0 o más veces.

Decoding Dependency in SDP

A día de hoy, si en una descripción SDP existe más de un atributo 'm' indicando el mismo tipo de medio, un receptor no puede identificar una relación específica entre ambos contenidos. Se define una Media partition (partición de medio) como un flujo de medios destinado a su transporte independiente. Así mismo, se define un MDC (Multiple Description Coding - Codificación con descripción múltiple) como un flujo en el que hacen falta N cualesquiera de M (con $N < M$) media partitions para que el flujo sea decodificable. Y definimos Layered Coding Dependency (Codificación con dependencia por capas) como aquel flujo en el que para que una de sus media partitions sea decodificable necesitamos haber recibido por completo las media partitions de las que depende. Para la codificación de MDC, las particiones de medios son complementarias, es decir, a más particiones recibamos mejor calidad podremos tener. En [12] se define una extensión a SDP para indicar de forma genérica dependencias en el proceso de decodificación.

Nos centraremos en el caso de Layered Decoding Dependency. Una o más capas pueden ser transportadas en distintos flujos de medios cumpliendo con el estándar SDP[9]. Un caso de uso clásico es conocido como una recepción multicast multicapa dirigida por el receptor, en la cual un receptor selecciona una combinación de flujos de medios en respuesta a calidad o requisitos de ancho de banda. A mediados de los 90 tan solo se empleaba una capa de mejora. SDP contiene por tanto métodos rudimentarios para soportar ese caso de uso en particular, en el sentido que permite señalar un rango de direcciones de transporte en una descripción de medios específica. Por definición una dirección de transporte mayor implica una capa superior en la jerarquía unidimensional.

Los nuevos formatos multimedia permiten más de una dimensión pudiéndose implementar jerarquías de dependencia altamente complejas. En cualquier caso, SDP en su forma actual no permitiría señalar todas esas complejas relaciones. Este también es el caso de la codificación multi-vista (MVC Multi View Coding), que permite codificar las vistas con dependencias a otras vistas. Esto suele ocasionar la posibilidad de ser transportadas de forma independiente.

Existe, además, el concepto de Operation Point (Punto de operación), que se define como un conjunto de flujos de medios que incluyen todas las particiones de medios necesarias para reconstruir con una cierta calidad, resistencia a errores u otra propiedad.

Señalizando las dependencias

Principios de diseño → La señalización de dependencias es factible únicamente entre descripciones de medios especificados con un atributo 'm' y con una identificación de atributo de medio tal y como se define en [11]. Todas las descripciones de medios agrupadas de acuerdo a esa definición tienen que tener el mismo tipo de medio.

Significado → Decoding Dependency[12] (DDP o Dependencia de Decodificación) asocia un flujo de medios, identificado por su propio mid-attribute, con un grupo DDP. Cada flujo de medios tiene que componer un número entero de Particiones de Medios. En un grupo DDP, todos los flujos de medios deben tener el mismo tipo de dependencia de decodificación (dependency-type definido a continuación) y un punto de operación como mínimo. Cuando se empleen múltiples codecs, el flujo de medios tiene que tener la misma estructura de dependencia independientemente de qué tipo de carga útil RTP tenga asignada. Todos estos campos y alguno más se encuentran especificados en el Apéndice E.

RTSP - Real Time Streaming Protocol

Los servicios de streaming unidireccionales se caracterizan por ofrecer algún tipo de control VCR (Video Cassette Recorder) para seleccionar los contenidos multimedia y moverse adelante o atrás dentro del mismo. Este control puede transportarse de modo independiente al flujo de datos. Esta separación es además justificable debido a su capacidad de extensibilidad, ya que un protocolo de control diseñado con este objetivo en mente podrá emplearse con distintos y futuros medios de transporte. Adicionalmente estos mecanismos de control pueden proveer de mecanismos de descripción de sesión, adaptación a las preferencias de usuario o negociación de capacidades. Con estos objetivos en mente el IETF desarrolló el protocolo RTSP[13][10].

RTSP es un protocolo fuera de línea orientado al control de flujos con sincronización de tiempo (por ejemplo audio y vídeo de una película), si bien también es capaz de entrelazar datos junto con el control. RTSP puede emplear UDP o TCP indistintamente. Como complemento a RTSP se hace necesario un protocolo para descripción de sesión y contenidos, siendo habitualmente SDP, aunque RTSP es lo suficientemente genérico como para trabajar con otros protocolos de descripción.

Intencionadamente se diseñó RTSP con un formato de mensaje muy similar al de HTTP/1.1, debido a la experiencia satisfactoria en cuanto a extensibilidad lograda con HTTP. No obstante, existen algunas diferencias:

- RTSP define nuevos métodos y cabeceras.
- Los servidores RTSP mantienen estado de las sesiones.
- RTSP emplea preferiblemente codificación UTF-8, mientras que HTTP emplea ISO-8859-1.
- La URI contenida en un mensaje de petición (request) RTSP es absoluta. En HTTP sólo se indica el path o posición local al servidor.

- RTSP incluye métodos bidireccionales. Ambos, servidor y cliente, pueden enviar mensajes de tipo request.

Modo de operación

En un escenario prototípico la sesión se establece como sigue: El cliente recupera una descripción de la sesión de streaming. Puede recuperarla del servidor o por cualquier método fuera de línea. A continuación, el cliente comenzará a enviar comandos SETUP a los servidores de medios. Los servidores, al recibir estos mensajes, comenzarán a reservar recursos para la sesión y crearán las mismas. Cada servidor responderá a este mensaje con otro que contendrá el identificador de sesión. En ese momento el estado es el necesario para poder ejecutar la orden PLAY que disparará el envío y recepción de los datos. Finalmente, un cliente termina la conexión con cada uno de los servidores que la componen mediante un comando TEARDOWN, permitiendo a los últimos liberar los recursos.

En el Apéndice F podemos encontrar detallados los mensajes definidos para RTSP.

Metadatos

MPEG-21

El estándar MPEG-21[14] intenta definir un entorno abierto para la provisión y consumo de multimedia para ser usado por todos los nodos que deseen participar de una comunicación multimedia. Dicho estándar abre el mercado ofreciendo igualdad de oportunidades. Esto además beneficiará al consumidor puesto que podrá acceder a contenidos sin tener previamente en cuenta la interoperatividad de los dispositivos.

MPEG-21 está basado en dos conceptos esenciales: la definición de unidades fundamentales de distribución y transacción (Digital Items) y el concepto de usuarios interaccionando con los Digital Items. Los Digital Items pueden ser considerados como el 'qué' del framework multimedia y los usuarios pueden ser considerados como el 'quién'. MPEG-21 define la tecnología necesaria para dar soporte a los usuarios para intercambiar, acceder, consumir, comerciar y en cualquier caso manipular Digital Items de forma eficiente, transparente e interoperable. Para ello, hace uso del lenguaje de marcas XML.

MPEG-21 identifica y define los mecanismos y elementos necesarios para la distribución de multimedia así como las relaciones entre las operaciones soportadas. Los elementos dentro de MPEG-21 son elaborados definiendo la sintaxis y la semántica de sus características, como interfaces a los elementos.

El estándar posee una sección dedicada únicamente a los procesos de adaptación[15] (Digital Item Adaptation - DIA), en particular la sección 7. Dentro de la misma existe una subsección dedicada a la descripción del entorno (Usage Environment Description Tool) donde se indica cómo describir el entorno del cliente, por ejemplo, las capacidades del terminal. Dentro de las mismas se distinguen los siguientes grupos:

Capacidades de Terminal	Capacidades de Codec
Parámetros de Codec	Capacidades de entrada/salida
Capacidades de Pantalla	Capacidades de Audio
Soporte para la interacción del usuario	Capacidades de dispositivo
Características energéticas	Características de almacenamiento
Características de entrada/salida de datos	

Otra subsección interesante es la dedicada a las características de la red, con los siguientes grupos:

Características de red
Capacidad de la red
Condiciones de la red

Composite Capability/Preference Profiles (CC/PP)

Un perfil (profile) CC/PP[16] es una descripción de las capacidades de un dispositivo y preferencias de usuario. Esto normalmente se denomina *device's delivery context* y puede ser empleado para guiar la adaptación del contenido presentada a dicho dispositivo. Para crear dichos perfiles que describen las capacidades y preferencias del agente de usuario (user agent) se emplea RDF (Resource Description

Framework). Un perfil no se refiere (como ocurre por ejemplo en RTP) a un subconjunto de una especificación particular, sino que se refiere a los documentos intercambiados entre dispositivos para describir las capacidades de los mismos.

RDF fue diseñado por el W3C como un lenguaje de descripción de metadatos con propósito general, proveyendo a la arquitectura con herramientas básicas para extensibilidad e interoperatividad del vocabulario vía XML.

Un perfil CC/PP contiene un número de nombres de atributo y valores asociados que son empleados por un servidor para determinar el modo más apropiado de proporcionar un recurso a un cliente. Está estructurado para permitir a un cliente describir sus capacidades por referencia a un perfil estándar, accesible a un servidor origen o emisor de recursos, y un pequeño conjunto de cambios respecto al perfil estándar.

El uso de XML namespaces evita que las definiciones para distintas aplicaciones no interfieran. Sin embargo, para permitir que las mismas trabajen en conjunto será necesario algún vocabulario común o método para traducirlo.

CC/PP es compatible con 'IETF media feature sets'[17] en el sentido que todas sus etiquetas de características de medios y valores pueden ser expresados en CC/PP, no sucediendo lo mismo a la inversa.

2.4. Control de flujo/ Congestión

Control de congestión - TCP Friendly Rate Control

La mayor parte del tráfico actual de Internet se amolda perfectamente al mecanismo ofrecido por TCP. No obstante, para el tráfico best-effort, como pueden ser los flujos multimedia, sería deseable tener mecanismos para el control de congestión compatibles con TCP (TCP-Friendly), pero sin la necesidad de reducir su tasa de envío a la mitad debido a un único paquete perdido o truncado.

TFRC[18] es una propuesta que hace uso de la técnica de control de congestión basado en ecuaciones (Equation Based Congestion Control) como mecanismo viable para proveer de control de congestión relativamente lineal para el tráfico best-effort, a costa de una respuesta más moderada a cambios transitorios en la congestión.

Para el tráfico que compite con TCP, la ecuación de control apropiada será la función de respuesta TCP correspondiente a la tasa de envío en el estado steady-state, en función del RTT (Round-Trip time) y la tasa de eventos perdidos (loss-event rate).

En TFRC se tienen en cuenta conceptos claves en el diseño del control de congestión basado en ecuaciones como son: respuesta a congestión persistente, evasión de oscilaciones innecesarias, evitar introducir ruido innecesario y robustez frente a gran variedad de escalado de tiempos (timescale).

El algoritmo para calcular la tasa de eventos perdidos es la clave en el diseño de mecanismos de control de congestión basados en ecuaciones.

Fundamentos de Control de Congestión basado en ecuaciones

Una aplicación que emplease un ancho de banda significativamente mayor que TCP podría provocar inanición de este último si ambos tipos de tráfico estuvieran compitiendo en una cola FIFO en un momento de congestión.

Se define un *Flujo de datos compatible con TCP* como un flujo que encontrándose en steady-state no usa más ancho de banda que uno ejecutándose conforme a TCP en condiciones comparables.

En [19] se define la función de respuesta TCP como:

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)} \quad (1)$$

Esta función da un límite superior para la tasa de envío T en bytes/seg, donde:

- s es el tamaño de paquete (fijo o estimado).
- R es el RTT
- p es la tasa de eventos perdidos
- t_{RTO} es el timeout para retransmisión de TCP.

En el Apéndice G podemos encontrar más datos sobre cómo emplear TFRC.

3. Cuerpo del trabajo

3.1. MANE - Media Aware Network Element

Un MANE es un elemento de red, como un componente físico (middlebox) o un enrutador de capa de aplicación que es capaz de analizar ciertos aspectos de las cabeceras de carga útil RTP o la carga útil de RTP reaccionando a los contenidos. Un MANE se engloba por tanto en la categoría de los traductores que se describieron en la subsección dedicada a RTP. Debido a que RTP es muy sensible a cambios durante el transporte, un MANE no puede ser transparente en la conexión entre servidor y cliente, convirtiéndose normalmente en cliente para el primero y en servidor para el segundo. Es lógico, por tanto, introducir el MANE en un proxy, en particular, para sesiones de streaming el MANE quedará embebido en un proxy RTSP (ver Figura3). RTSP soporta el uso de proxy ampliamente[13] para el establecimiento y control de la sesión. Adicionalmente el MANE controlará los distintos flujos de datos entre el servidor y el cliente.

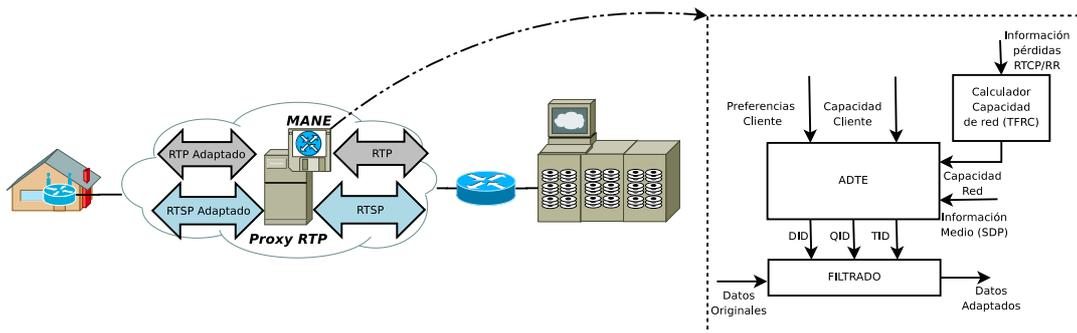


Figura3. Estructura de un MANE

La estructura a la que hacemos referencia tendrá tres partes fácilmente diferenciables: el proxy RTSP encargado de la sesión, el MANE encargado de analizar todos los flujos RTP configurados previamente por el proxy RTSP y un mecanismo o elemento capaz de sincronizar la información de sesión con la información de análisis de los flujos.

Un MANE es realmente interesante para codecs escalables capaces de ser decodificadas a menor calidad habiendo recibido una parte de la información enviada. En particular, en el caso de SVC que define dependencias entre múltiples capas, el MANE puede realizar funciones de control de congestión u optimización de ancho de banda con relativa facilidad. Para AVC, el MANE tiene menos posibilidades puesto que la adaptación del ancho de banda suele requerir realizar una transcodificación.

Carga útil RTP para AVC y SVC

Tanto en [20] como en [21] se hacen múltiples recomendaciones al que debería ser el funcionamiento y/o funcionalidades de un MANE para los formatos de vídeo de AVC y SVC respectivamente.

En [20], correspondiente a la definición de carga útil para RTP de H.264/AVC, se recomienda, entre otras cosas, el uso del *Forbidden bit (F)* de la cabecera de la NALU para indicar al cliente que durante la recepción del paquete hubo algún error. El cliente, por tanto, puede tratar el paquete de forma especial,

normalmente desechándolo. De hecho, el propio MANE puede tomar la decisión de desechar este, por ejemplo, para ahorrar ancho de banda y ya de paso desechar todos los dependientes del mismo. También resultaría interesante el *análisis del campo NRI*. El MANE debe tener en cuenta el valor de este campo a la hora de desechar paquetes. A menor valor de NRI menor relevancia tiene el paquete para el flujo de datos. De hecho un MANE no debería desechar nunca NALUs con valor NRI 11 (el máximo posible). Hay que tener en cuenta que un MANE no debe *nunca duplicar NALUs VCL*, únicamente se le permite duplicar los Parameter Sets y sin afectar a los ya existentes. Esta duplicación se debe hacer a nivel de aplicación y nunca a nivel de red duplicando paquetes debido a la necesidad de actualizar el número de secuencia RTP.

Un MANE puede, por ejemplo, detectar (mediante buffering) los casos en los que se pierda una o varias fragmentation units. Dado ese caso, se puede hacer uso del forbidden bit para indicárselo al cliente. También se indica que un MANE podría juntar múltiples Single NALUs en un Aggregation Packet o a la inversa, e incluso mezclar ambos conceptos. Por ser un traductor RTP, además de tener soporte para RTCP, deberá tener en cuenta los siguientes parámetros: tamaño MTU máximo del camino completo, mecanismos de corrección de errores hacia adelante (FEC - Forward Error Correction), la latencia máxima admisible por el sistema y el tamaño del buffer del cliente.

En entornos de seguridad y privacidad, un MANE deberá ser una entidad de confianza en la arquitectura para poder llevar a cabo su cometido. En el caso de privacidad, el MANE será incapaz ni tan siquiera de desechar paquetes mediante el análisis de los contenidos si no forma parte del entorno de privacidad.

En la correspondiente definición de carga útil de RTP para SVC[21], actualmente en proceso de estandarización, se hace mayor énfasis en las posibilidades que ofrece un MANE a la arquitectura. Una de las principales características que debe tener un MANE SVC es la capacidad de filtrar toda información SVC para producir un flujo de datos AVC compatible, es decir, dejar únicamente la denominada capa base para que cualquier dispositivo compatible H.264/AVC pueda reproducir sin problemas a la calidad mínima del flujo. Para ello, el parámetro *sprop-avc-ready* durante la negociación RTSP debe encontrarse en la descripción SDP, indicando que es compatible con las definiciones del rfc3984bis[20]. Se especifica[21] que un MANE será capaz de agregar múltiples flujos RTP, posiblemente de distintas sesiones RTP.

Gracias a las PACSI (Payload Content Scalability Information) NALUs definidas en [21], un MANE será capaz de tomar decisiones sobre una aggregation unit sin necesidad de inspeccionar la misma. Esto facilita el análisis del flujo. En un aggregation packet es posible detectar el inicio de una nueva representación de capa detectando cambios en la terna DID,QID,TID, a excepción de en la primera y última NALU del paquete. Para ello, se definen los bits S y E de las NALUs PACSI, que permitirán al MANE detectar pérdidas de slices permitiendo llevar a cabo la acción más indicada como podría ser la retransmisión. El bit M de RTP, tal y como se define en el perfil para AVC, sigue indicando el final de una access unit, no el final de la representación de capa. Los campos TL0PICIDX e IDRPICID de la PACSI NALU deben contener los valores de *t10_dep_rep_idx*[6] y *effective_idr_pic_id*[6] del SEI respectivamente. Estos campos permiten al MANE la detección de pérdidas de representaciones de capas en la capa temporal más relevante.

Los MANEs deben intentar preservar en la medida de lo posible las NALUs que indiquen con bit I en la cabecera de extensión que son IDR, y en caso de ser Prefix NALUs la siguiente VCL en orden de decodificación. Con preservar podemos referirnos tanto a no desechar como a proteger mediante un FEC o cualquier otro mecanismo posible. Consecuentemente, no será necesario enviar ninguna NALU con DID superior al de la NALU con DID más alto y bit I activado puesto que no sería útil para el proceso de decodificación y sobrecargaría la funcionalidad de red. También se puede inspeccionar el campo PRID para decidir el nivel de protección a aplicar, a menor valor de PRID mayor relevancia tiene la NALU para el proceso de decodificación. La terna DID, QID y TID indica, al igual que el campo PRID, a menor valor mayor relevancia de la NALU. El bit U (*use_ref_base_pic_flag*) de la cabecera de extensión SVC indica cuándo se emplearán imágenes de referencia base para la decodificación inter. Este parámetro también resulta interesante para un MANE puesto que podrá proteger aquellas que se indique que serán empleadas para ese menester. El bit D (*discardable_flag*) indica que dicha NALU no se empleará para la decodificación de NALUs con DID mayor. En tal caso, la NALU puede ser desechada sin hacer peligrar el resto de la decodificación.

Control de Congestión

El control de congestión para flujos AVC[20] o SVC[21] debe ejercerse de acuerdo con lo definido en RTP[8] y con cualquier perfil aplicable al caso. En el caso del servicio best-effort (mayormente UDP) existe un requisito extra: se debe monitorizar la tasa de pérdidas de paquetes para asegurar que se mantiene dentro de parámetros aceptables. En general se dice que es aceptable si una conexión que estuviese empleando el mismo camino que una conexión orientada a la conexión (mayormente TCP) con las mismas condiciones en la red obtuviese aproximadamente la misma tasa de transferencia no menor a la que posee el flujo RTP.

Existen dos casos diferenciados para ejercer el control de la congestión, cuando es la fuente que produce el servicio en tiempo real la encargada de hacerlo, en cuyo caso se pueden cambiar los parámetros en la codificación al vuelo para adaptarse a los requisitos. O cuando el origen está previamente precodificado, en cuyo caso dependerá de las distintas capacidades almacenadas previamente.

Cuando los parámetros necesarios para cumplir con las necesidades no pueden ser provistos por el perfil o sesión actuales, se hace necesario reiniciar la conexión para reconfigurarla.

Para el caso específico de SVC, es posible realizar esta adaptación de distintas maneras proveyendo distintas experiencias de usuario. Dentro de la capa con DID mayor es posible emplear el campo TID o QID para desechar NALUs que afectarán en menor medida a la percepción humana.

3.2. Información de control para la adaptación

Mensajes SEI (Supplemental Enhanced Information)

Los mensajes SEI ayudan durante los procesos relacionados con la decodificación. En ningún caso son estos necesarios para la decodificación de la crominancia o luminancia, ni siquiera es obligatorio su procesamiento, no obstante, son necesarios para comprobar la validez del flujo de datos y sincronización de la salida del decodificador.

Los Tipos de mensaje SEI más importantes para el proceso de adaptación SVC son:

- *Sub-Sequence information* → Se emplean para indicar la posición de una imagen en la jerarquía de dependencia de datos que consiste en capas de subsecuencia y subsecuencias. Una capa de subsecuencias consiste en un subconjunto de imágenes decodificadas en una secuencia. Una subsecuencia es un subconjunto de imágenes codificadas dentro de una capa de subsecuencia.
- *Scalability_info* → Contienen información de escalabilidad para toda una capa del flujo, por tanto, para una terna QID, DID, TID específica. El subconjunto del flujo necesario para decodificar una capa se denomina la 'representación' de la capa. Una capa tiene asociado un identificador creciente con respecto a la dependencia de la capa; una capa *podría* ser dependiente de una o varias de las capas con identificador menor que el suyo, nunca de una con identificador mayor. Una capa requiere de otra para la decodificación si depende directa o indirectamente de la misma. Cuando este mensaje aparece en una NALU de tipo SEI, siempre tiene que encontrarse en una access unit cuyas VCL NALUs sean todas IDR (AVC o SVC). A excepción de la dependencia entre capas, que puede ser señalizada con un 'layer_dependency_change', se mantienen vigentes hasta la próxima SEI con mensaje scalability_info. El mensaje incluye el identificador, TID, QID, DID, discardable_flag, ... de cada capa. También se especifican[6] el bitrate necesario (avg_bitrate), en media y en pico máximo (max_bitrate_layer), el bitrate para la representación de la capa(max_bitrate_layer_representation) o el número de capas de las que es directamente dependiente(num_directly_dependent_layers)
- *Layer_not_present* → Indica el número de capas que no seguirán existiendo en el flujo a partir de la Access Unit actual y las subsecuentes. Se indican los parámetros num_layers y layer_id[i], los cuales indican el número de capas omitidas y los identificadores de cada una respectivamente.
- *Layer_dependency_change* → Modifica las relaciones de dependencia entre capas indicadas en el scalability_info anterior. En este mensaje se indican qué capas(layer_id) cambian su dependencia con qué otras(directly_dependent_layer_id_delta_minus1).

Parameter Set

En H.264/AVC[20] se ha desacoplado información relativa a más de un slice de un flujo de medios.

Esta capa de metainformación de alto nivel puede ser transferida asincrónicamente al resto de datos y con antelación a los mismos. A la combinación de parámetros de alto nivel se le denomina 'Parameter Set'. Para permitir la modificación de los parámetros sin la necesidad de sincronizar datos y meta datos, los codificadores y decodificadores deben tener la capacidad de tener una lista de Parameter Set. Cada cabecera de slice señala cuál será empleado mediante un campo específico.

Existen dos tipos: *Picture Parameter Set* y *Sequence Parameter Set*. Un *Picture Parameter Set* es una estructura que contiene elementos de sintaxis aplicables a 0 o más imágenes codificadas completas, es decir, una secuencia. Un *Sequence Parameter Set* es una estructura que contiene elementos de sintaxis que se aplican a 0 o más secuencias completas de vídeo determinadas por un *Picture Parameter Set*.

SVC[21] mantiene el concepto de *Parameter Set* y añade un nuevo tipo denominado *Subset Sequence Parameter Set*, que serán los únicos empleados por las NALUs VCL pertenecientes a la extensión SVC. Los *Subset Sequence Parameter Set* emplean un espacio de identificadores independiente del de los *Picture* y *Sequence Parameter Set*. Tanto los *Sequence* como los *Subset Sequence Parameter Set* deben permanecer inalterados entre imágenes IDR. Los *Picture Parameter Set*, por otro lado, pueden cambiar entre una imagen y otra de cualquier tipo aunque siempre al inicio de las mismas. Puede existir un *Picture Parameter Set* por cada una de las capas.

Características del receptor

El receptor debe proporcionar al emisor, y por tanto, de forma indirecta al MANE, las características técnicas de que dispone. Algunas de estas características se conocen al inicio de la sesión y pueden ser consideradas estáticas. En tal caso, el cliente puede emplear MPEG-21 o CC/PP para expresar los metadatos referentes a sus características y proporcionárselas a la fuente. No obstante estas características iniciales deberán evolucionar conforme la sesión avance, para lo cual el receptor deberá notificar al emisor de forma directa o indirecta.

MPEG-21

Dentro de los grupos de características y capacidades enumerados en la sección anterior se definen atributos que resultan interesantes a la hora de tomar decisiones de adaptación.

Entre las 'Capacidades de Pantalla' definidas por el estándar se describen 'Resolution' y 'refreshRate'. El primero nos permitirá decidir si una capa de escalabilidad espacial debe ser desechada o no. El refreshRate, por otra parte puede servir también para casos hipotéticos y aislados en los que la fuente ofreciese un frame-rate mayor a la tasa de refresco en Hz del visor. Sirva como ejemplo un streaming de una partida de un videojuego; en ese caso se puede aprovechar la escalabilidad temporal para evitar enviar más información de la que podrá ser presentada.

Las 'Capacidades del codec' nos indicarán de forma indirecta y con la restricción de tener que conocer el codec en profundidad, ciertos parámetros a la hora de adaptar. En particular la parte de decodificación ('Decoding') que será empleada por los clientes, puede indicar un perfil y/o nivel particular. En la recomendación H.264[6] podemos observar sendas tablas donde se relaciona perfil y nivel con restricciones importantes como el bitrate soportado. De forma más directa este mismo parámetro puede ser obtenido de los 'Parámetros del codec' con el parámetro 'BitRate' que ofrece valores medios y máximos.

También podrían resultar interesantes dentro de las 'Capacidades Energéticas' los atributos 'battery-CapacityRemaining' y 'batteryTimeRemaining', pudiéndose por ejemplo establecer mínimos para estos valores y una vez superados enviar únicamente la capa base. No obstante, este caso es más complejo puesto que las preferencias del usuario pueden entrar en juego y un punto intermedio de la red puede tomar la decisión no deseada. Sin embargo, para entornos más acotados, este comportamiento podría ser más que interesante: pongamos como ejemplo un sistema de videovigilancia remota, permitiendo al vigilante seguir observando mientras se dirige a por otro aparato con carga.

En 'Capacidades de Almacenamiento' está el parámetro 'inputTransferRate', que limita la velocidad a la que el dispositivo puede almacenar en disco la información. Este parámetro también podría ser utilizado para limitar el ancho de banda a emplear por el MANE. También este caso es un poco subjetivo, puesto que muchos (la inmensa mayoría) clientes de streaming emplean buffering en memoria volátil, que recordemos tiene una cierta tasa de transferencia distinta a cualquier dispositivo de almacenamiento no volátil; por lo tanto, este parámetro podría resultar inservible. En tal caso, en los 'Parámetros del

codec' se indica el 'MemoryBandwidth', que indica los requisitos en cuanto a tasa de transferencia en bits/segundo del codec a la memoria. Sabiendo esto podemos suponer que como mínimo la memoria funcionará a dicha velocidad.

CC/PP

Cualquier aplicación o entorno operacional que emplee CC/PP podría definir su propio vocabulario, pero se obtiene una interoperatividad mejorada si los vocabularios se definen de forma que puedan ser más ampliamente empleados. De acuerdo con esto, CC/PP define un pequeño vocabulario núcleo/principal/básico de atributos que son aplicados ampliamente por agentes de impresión y visionado cuyo uso, cuando sea apropiado, se aconseja energicamente. Este vocabulario está basado en el 'Media Features for Display, Print and Fax'[22] donde se describen algunas características básicas aplicables a dichos entornos de forma general, destacando para nuestro cometido pic-x y pic-y, que hacen referencia a la resolución x*y en pixels de la pantalla.

Características del enlace

Serán tres los puntos en los que vamos a centrar nuestra atención. Como ya hemos comentado, los parámetros iniciales deben evolucionar con la sesión. En el caso de las características de la red el método más común para la actualización es el RR de RTCP, dentro del cual se envía al emisor un feedback de la visión del receptor. Con esta información, que debe contrastar con su visión, el emisor (en este caso el MANE) podrá modificar sus parámetros para adaptar el/los flujos a las características instantáneas.

Tasa de Pérdidas

En MPEG-21 se define 'packetLossRate' como la tasa de pérdidas del modelo de red proporcionado por el receptor. Este valor puede ser contrastado y/o actualizado con la recepción de mensajes RR de RTCP del receptor, en los que el campo 'cumulative number of packets lost' indica el número de paquetes perdidos para toda la sesión hasta el momento. Si el emisor almacena el último valor del mismo, se puede averiguar con una simple resta el número exacto de paquetes perdidos desde la última recepción RTCP. De manera adicional, el campo 'fraction lost' informa al emisor de la proporción de paquetes perdidos sobre el número de paquetes esperados para el intervalo comprendido entre la notificación actual y la anterior. Un aumento en cualquiera de estos dos campos indica problemas en la red, ya sean por congestión o por deficiencias en el enlace.

Tiempo de ida y vuelta

MPEG-21 define el parámetro 'Delay' que modela los tiempos de propagación en toda la conexión sin tener en cuenta intervalo alguno. Este valor puede ser empleado como valor de inicialización en el MANE. No obstante y como ya se ha comentado será necesario actualizar su valor, para lo cual se podrán emplear los mensajes RR de RTCP, que poseen campos mediante los que el emisor conoce la evolución del tiempo de propagación de la red, que puede considerarse un indicativo de la carga de los nodos intermedios y por tanto, de la congestión.

En [8] el interarrival_jitter se define como una estimación de la varianza estadística del tiempo de llegada de los paquetes RTP, medido en unidades de marca de tiempo y expresado como un entero sin signo. El interarrival_jitter J se define como la desviación media de la diferencia D en que el intervalo de recepción de un par de paquetes consecutivos en el receptor, comparado con el intervalo de emisión para el mismo par de paquetes en el emisor.

$$D(p1, p2) = |(R_{p1} - R_{p2}) - (E_{p1} - E_{p2})| \quad (2)$$

Donde p1 hace referencia al paquete 1, p2 al paquete2, E a la marca de tiempo de emisión y R a la marca de tiempo de recepción.

Por tanto el valor del interarrival_jitter J se debe calcular para cada nuevo paquete que llega y teniendo en cuenta el valor acumulado, mediante la media móvil:

$$J(p_x) = J(p_{x-1}) + (D(p_{x-1}, p_x) - J(p_{x-1}))/16 \quad (3)$$

Ancho de banda disponible

En MPEG-21 dentro de las Capacidades de la red, los atributos 'maxCapacity' y 'minGuaranteed', re-

sultan a su vez interesantes a la hora de adaptar los flujos. Los parámetros de condición de red para un 'interval' específico 'AvailableBandwidth', 'minimum', 'maximum' y 'average' resultan también interesantes. No obstante estos parámetros, como ya hemos comentado anteriormente, deben ser adaptados durante la sesión.

Un problema recurrente en los servicios de streaming es el cálculo de la capacidad disponible para la emisión de datos óptima. En ese aspecto ya introdujimos en la sección anterior el mecanismo TFRC[18] en el que se explica cómo obtener un flujo de datos compatible con TCP. Si observamos la fórmula en la que se define T (1), y en particular sus argumentos, podremos ver que los argumentos necesarios son conocidos siempre que se tenga mecanismo de feedback RTCP.

3.3. Alternativas de transmisión SVC

Independientemente del método de transporte empleado, SVC permite dos métodos para la subscripción a los contenidos: una única sesión por flujo (SST) o múltiples sesiones por transmisión (MST). El caso MST se introduce por primera vez con SVC[21] y se recomienda para casos en los que, empleando multicast, más de un cliente pueda solicitar capas distintas del mismo flujo. Como mínimo, al emplear MST, se deberían crear tantas sesiones como puntos de operación puedan ser solicitados por los clientes. Los distintos modos MST se crean, ante todo, debido a la sincronización intersesión necesaria al emplear distintas sesiones.

Single-Session Transmission (SST)

Es la única opción posible para H.264/AVC[20] definiendo tres modos de operación, también denominados 'session packetization modes'[21].

- Single NAL Unit Mode: se emplea en sistemas conversacionales que tienen que cumplir con la recomendación del ITU-T H.241. El orden de decodificación lo establece el número de secuencia RTP.
- Non-Interleaved Mode: se emplea en sistemas conversacionales que podrían no cumplir con la recomendación del ITU-T H.241. Las NALUs se transmiten en el orden de decodificación. El orden de decodificación se infiere primero del orden implícito dentro de las NALUs de tipo STAP (Single Time Aggregation Packet) y finalmente del número de secuencia RTP para el resto. SVC provee de mecanismos de Agregación (Aggregation Units) para aprovechar mejor los paquetes en casos en los que las NALUs sean pequeñas y Fragmentación (Fragmentation Units) para casos en que las NALUs excedan el tamaño máximo de paquete. En particular las NALUs de tipo STAP pertenecen al primero de ellos.
- Interleaved Mode: se emplea en sistemas en los que la latencia no es un requisito indispensable. Este modo permite a las NALUs ser transmitidas en un orden distinto al necesario para la decodificación. En este último caso se introduce el concepto de 'Decoding Order Number' o DON, que se ve reflejado como un campo en la estructura de carga útil de la NALU que indica el orden de decodificación. Aumenta la resistencia frente a ráfagas de pérdidas.

Multiple-Session Transmission (MST)

SVC define[21] cuatro tipos de paquetización multisesión, que se diferencian entre sí en dos aspectos fundamentales, si las NALUs deben ser transmitidas en orden de decodificación y los mecanismos provistos para recuperar qué tipo de orden existe en cada sesión.

- Non-Interleaved timestamp based mode (NI-T): No permite interleaving. En este caso se emplean marcas de tiempo (timestamps) para recuperar el orden de decodificación.
- Non Interleaved cross-session decoding order number (CS-DON) based mode (NI-C): No permite interleaving. Emplea el mecanismo 'Cross-Session Decoding DON' (CS-DON). El CS-DON es un número que indica el orden de decodificación de todas las NALUs a lo largo de todas las sesiones RTP involucradas en MST. Al contrario que para el caso de DON, CS-DON se emplea también en entornos sin interleaving.

- Non-Interleaved combined timestamp and CS-DON mode (NI-TC): No permite interleaving. Emplea un modo mixto para el orden de decodificación, CS-DON junto con marcas de tiempo.
- Interleaved CS-DON (I-C) mode: Este es el único caso de MST en el que está permitido emplear interleaving. Al igual que NI-C emplea CS-DON para obtener el orden de decodificación.

El modo MST a emplear debe ser señalado mediante el parámetro opcional 'mst-mode'[21] para SDP[9]. En ausencia del mismo, por defecto, se indica que se empleará SST. Además, se debe emplear el parámetro packetization-mode[20][21], el cual originalmente mediante los valores 0,1 y 2 indicaba los 3 modos de transmisión SST posibles, Single, Non-Interleaved e Interleaved respectivamente. Para MST se limita su significado pudiendo tomar únicamente el valor 2 para el caso I-C.

Unicast y Multicast

RTP soporta la entrega a direcciones multicast, si la capa de nivel inferior lo soporta. Para el caso de codecs escalables con independencia del transporte, como es SVC, cada uno de los flujos puede transmitirse en su propio flujo RTP y en su propia sesión empleando IP multicast, indicando la dirección y el puerto como único punto de demultiplexión. Los receptores o clientes deberán suscribirse a las distintas capas, normalmente con IGMP (Internet Group Management Protocol), correspondientes a la IP y puerto designada. A este método se le denomina control de congestión dirigida por el receptor (receiver-driven).

Aunque la transmisión multicast ofrezca un gran ahorro de ancho de banda, sobretodo para los primeros saltos desde el origen de los contenidos, para un número elevado de usuarios, tiene también la desventaja de hacer necesaria la apertura de puertos en los routers y NATs. Además podría no disponer de canal de vuelta hacia el origen, lo que implica deshabilitar el mecanismo de feedback, como podría ser el mensaje RR de RTCP. No obstante, la transmisión multicast de codecs multicapa recibe un gran interés por parte de entornos muy acotados, como pueden ser servicios de televisión digital.

Entre los modos para identificar la dirección de destino de red soportados por RTSP existen dos relacionados con multicast, uno en el que el servidor selecciona la dirección y otro en la que la elige el cliente. Este último caso se emplea para servidores que se agregan a sesiones ya existentes. Además RTSP define el campo de cabecera 'Transport'[13], en el se definen atributos específicos para multicast como el campo 'TTL' o el campo 'layers' que indica el número de capas relacionadas con la sesión. Todas estas capas serán enviadas a direcciones consecutivas a partir de la indicada por el campo 'destination'.

En el caso de SVC, los receptores deben inspeccionar la información de cabecera RTP para reconstruir la información de sincronización producida por la fuente para medir las pérdidas de paquetes y los timeouts. En unicast, el cliente notifica con esta información a la fuente, pero en el caso de multicast la fuente se podría ver desbordada, por lo que normalmente se evita el reporte. Este efecto puede ser mitigado mediante el uso de varios niveles de MANEs, que hacen a su vez de fuente y cliente para cada uno de los lados de la red. Así, un cliente notificará al MANE inmediato en su camino hacia la fuente, este al siguiente haciendo una única notificación para todos los clientes asociados al mismo y así sucesivamente, creándose un árbol que tendrá como nodo raíz la fuente. La fuente recibirá, por tanto, tantas notificaciones como MANEs estén exactamente a un salto en términos del codec, no de tramos de red.

Para el caso de multicast, los receptores intentan inferir el estado de la red, usando señales de congestión, y controlar el flujo entrante de datos, mediante el mecanismo de 'pruning' de encaminamiento IP multicast. Desgraciadamente, todavía tenemos que lidiar con algunos problemas adicionales:

- El abandono de un receptor de una suscripción no tiene efecto sobre la congestión hasta que sus vecinos no hagan lo propio y se pueda realizar un prune.
- Si un receptor causa congestión, puede que un vecino la considere suya e intente reducirla eliminándose de la lista de suscriptores mientras que el causante de la misma continua sin detectar lo sucedido.
- Si dos receptores vecinos tienen distintos niveles de suscripción, el ancho de banda reservado no se aprovecha por completo.

Mecanismos para el control de congestión en multicast:

Synchronization Points (SPs)[23]. Son paquetes especialmente marcados en el flujo de datos. Un receptor

puede intentar unirse únicamente tras la recepción de un SP. Para la toma de decisiones tendrá en cuenta únicamente los eventos sucedidos desde la recepción del último SP. Los SPs deberían encontrarse lo suficientemente distanciados para que a la red le de tiempo a estabilizarse. Se reduce el efecto de la propagación y la repercusión a largo plazo de los eventos.

Server-initiated Probes[23]. Los receptores pueden realizar intentos de consulta del ancho de banda disponible para elevar su suscripción. Para ello, generan pequeñas ráfagas seguidas de periodos de silencio que simulan el efecto de una suscripción. Dichas ráfagas no deben ser interpretadas como una señal por los vecinos para reducir su nivel de suscripción, pero sí como indicación de no aumentar el mismo.

Deaf Period (periodo de sordera t_D)[23]. Tras una pérdida y la consiguiente reducción del nivel de congestión, un receptor no vuelve a reaccionar a más pérdidas durante un periodo t_D para evitar verse afectado por los posibles retrasos de un desregistro anterior.

3.4. Señalización al receptor de las decisiones de adaptación

Cuando un MANE tome una decisión para desechar una parte del contenido, debe ser capaz de indicárselo al receptor para que éste no traduzca la ausencia de la información desecheda como errores en la transmisión. Cómo y cuándo notificar al receptor es lo que intentaremos desglosar ahora.

Al inicio de la sesión multimedia

Si la decisión se realiza durante el establecimiento de la sesión, el MANE puede informar de forma transparente al receptor mediante la adaptación de la respuesta de la fuente original o generando una respuesta nueva.

Para ello, en la respuesta SDP se obviarán las definiciones de los flujos que no concuerden con la decisión del MANE, de tal forma que el receptor no podrá realizar un 'SETUP' de los mismos.

Este mecanismo tiene la ventaja de evitar al MANE el control de las suscripciones a esos flujos no proporcionables, bien por limitaciones del cliente, del MANE o simplemente por motivos tan diversos como la tarificación. El problema, no obstante, es que iniciada la sesión no se puede mejorar por encima de lo ofrecido, obligando a renegociar la sesión si hubiese mejoras en las condiciones de cualquiera de los puntos anteriores.

Durante la sesión multimedia

Sería deseable que el MANE ofreciese toda la información posible respecto a la fuente y que estas decisiones de adaptación fuesen notificadas dinámicamente al receptor tal y como son tomadas, ya sea por reducción o aumento del nivel de suscripción obligado.

SVC[6] permite notificar los cambios en las dependencias de las capas o incluso la estructura de las mismas mediante el uso de mensajes SEI, como son 'Scalability_info', 'Layer_not_present' o 'Layer_dependency_change', los cuales son normalmente emitidos por la fuente ante cambios en el codificador o el flujo almacenado, pero que pueden ser aprovechadas por el MANE para informar al receptor.

3.5. Algoritmo de Adaptación

Nos vamos a centrar en la adaptación dinámica de flujos simples con feedback por parte del cliente. Como se ha comentado en la subsección 3.1, la terna DID,TID,QID puede ser empleada para identificar a cada NALU con la capa a la que corresponde. De igual forma, mediante el mensaje SEI denominado 'scalability_info', el MANE recibe de forma dinámica la información sobre todas las capas disponibles para la fuente de streaming. Por tanto, tras analizar ese atributo tendremos una relación entre el identificador de la capa, su DQT y el bitrate medio necesario para su transmisión.

También se comentó en la subsección 3.2 cómo se podía obtener la capacidad disponible en la red para una transmisión óptima mediante el empleo de TFRC, en parte gracias al empleo del feedback de los mensajes RTCP. Con esto calcularemos la capacidad de la red y por tanto el bitrate máximo a emplear por el flujo.

Sabiendo los bitrates medios de las capas y el bitrate disponible, podemos fácilmente seleccionar una capa que cumpla con los requisitos, si es que esta existe. No obstante, una adaptación de este tipo no tendría en cuenta las preferencias del cliente; nótese que hablamos de cliente y no de usuario puesto que un usuario probablemente no sabrá, ni tiene la necesidad de ello, qué es una capa de escalabilidad o un bitrate. Además, un MANE puede ejercer de cliente de otro MANE sin necesidad de que un usuario intervenga. Por preferencias de cliente nos referimos básicamente al orden en que el cliente desea que los atributos DQT se vean reducidos y qué valores mínimos desea preservar para cada uno de ellos. Esto permite al cliente, por ejemplo, sacrificar FPS frente a tamaño de vídeo o a la inversa. Como ejemplos aplicados podemos imaginar un guarda de seguridad, al que realmente lo que le interesa es que la imagen se vea grande en el monitor aunque esta vaya a 10 FPS, frente a una persona viendo un programa deportivo, en el que lo realmente importante es ver fluida la acción, aunque sea más pequeño.

Teniendo en cuenta esto, la solución propuesta reduce el nivel de escalabilidad de forma 'desordenada' empleando el orden DQT y los mínimos DQT. Para ello, se inicia el DQT seleccionado al máximo valor aceptado por el flujo. A continuación, se toma el parámetro con menos preferencia y se reduce en uno siempre y cuando no se haya llegado al mínimo establecido por el usuario para el mismo y no se haya llegado a 0. Si cualquiera de las dos anteriores ocurriese, se reestablecería el valor del parámetro al máximo y se realizaría la misma acción con el siguiente parámetro en orden de preferencia. Por cada reducción, tendremos que comprobar si el bitrate seleccionado correspondiente a la capa de ese DQT, se adapta al bitrate disponible y calculado. En caso de que llegemos a los valores mínimos de cliente y no hayamos llegado a una capa válida, se seleccionará la capa con bitrate medio inmediatamente por debajo del bitrate disponible. Si ninguna cumpliera con esta condición, se enviaría la capa base o se cancelaría la comunicación. Esta decisión debe ser tomada con cautela, un aborto de la emisión puede ser no deseable en casos en que haya contratos o similares, y el envío de la capa base aun sin tener bitrate disponible podría agravar el problema de congestión detectado.

En el Apéndice H podemos observar una especificación en pseudocódigo de lo aquí descrito y que ha sido implementado para la evaluación de esta tesis.

3.6. Simulación

Para comprobar la efectividad del método expuesto, se han realizado una serie de simulaciones sobre NS-2. La simulación ha consistido en realizar una transmisión de un vídeo codificado con SVC e introducir una situación de congestión que obligase al punto intermedio a realizar adaptación. Para simular la congestión, hemos optado por reducir la capacidad del buffer de paquetes de salida del router en el camino del MANE al cliente. Además, ha sido necesario indicar las preferencias de cliente tal y como se expuso anteriormente.

Para implementar el entorno de simulación se ha tomado como base la implementación RTP de NS-2. Ha sido necesario ampliar la misma para dotarla de cierta funcionalidad omitida por los anteriores implementadores, como por ejemplo la inicialización de los campos de RTCP que resultan vitales para esta simulación. Además, ha sido necesario implementar un sistema de control de flujo que se adaptase al cálculo de la tasa TFRC calculada. Para ello se ha empleado una aproximación del timeout entre paquete basada en el Frame Rate y el número de paquetes por $Frame.T_o = \frac{1/FR}{Paquetes \times Frame}$

Para calcular la tasa TFRC empleando la fórmula expuesta (1) se ha empleado, entre otra información, el campo FractionLost de RTCP. Para ello, y siguiendo las recomendaciones de [18], se ha empleado un sistema de pesos que suaviza el efecto inmediato de una detección de pérdida que podría ser debida a una ráfaga y se enfoca más a pérdidas más prolongadas normalmente debidas a congestión. Para esta simulación se tendrán en cuenta en cada cálculo 10 muestras que perderán peso paulatinamente.

El vídeo empleado se transmite en aproximadamente 20 segundos. Se han establecido dos líneas de 1,8Mb, una unidireccional y otra bidireccional, que es la empleada para la transmisión de RTCP (evitándose la pérdida de paquetes). El vídeo posee un bitrate medio de 1,9Mb en su capa más alta. Inicialmente el router de la línea unidireccional se ha establecido con un tamaño de cola de 1500 paquetes, lo que a razón de 1400 bytes por paquete de media significa 2MB de buffer para esta conexión. Hay que recordar que los buffers de los routers son compartidos por todas las conexiones, en este caso como únicamente tenemos nuestra conexión, lo que haremos es reducir la capacidad global del router obligando

al mismo a desechar paquetes. En el segundo 7 de la simulación reducimos el buffer a 30 paquetes. Esta reducción produce la pérdida de paquetes que origina notificaciones RR de RTCP con valores positivos de Fraction Lost y que originan la entrada en acción de nuestra solución.

El vídeo empleado es un vídeo 4CIF a 30 FPS con 28 niveles de escalabilidad, situándose el más alto de todos en un DQT igual a 2,1,4 (268 en decimal). Se han realizado simulaciones de todas las posibles órdenes de preferencia de cliente con valores mínimos de DQT seleccionados por el cliente de 000, 004 y 102 que representan respectivamente: ceder todo el control al sistema de control de congestión indicándole únicamente la preferencia pero no el límite inferior de los parámetros, mantener siempre la calidad al máximo y una opción mixta que intenta mantener calidad y tamaño con un mínimo. Como era de esperar, a la selección (004) no le afecta el resto de parámetros, mientras que con las otras dos opciones conseguimos resultados diferentes dependiendo de las preferencias de orden de reducción de DQT del cliente, pero siempre adaptándonos a lo establecido por TFRC.

A continuación se mostrarán 8 gráficas correspondientes al estudio realizado. En estas gráficas podemos observar 4 valores representados:

SBR → Es el Selected BitRate, es decir, es el bitrate que indica la capa con el DQT seleccionado. Corresponde a la línea de color rojo y su valor original está dividido por 10^5 para una mejor visualización.

TFRC → Es el Bitrate teóricamente disponible y calculado tal y como se indica en [18]. Corresponde a la línea de color verde y su valor original está dividido por 10^5 para una mejor visualización.

FL → Es el FractionLost de RTCP, indicador de la visión que tiene el emisor de las pérdidas. Corresponde a la línea de color rosa.

PL → Es el Packet Lost, indica el número de paquetes perdidos durante el intervalo. Corresponde a la línea de color azul.

Enfrentados a 3 valores correspondiente a la D (rojo), Q (cyan), T (verde) seleccionada por el Algoritmo de adaptación.

Podemos observar en todas las gráficas cómo la línea roja, correspondiente al SBR, siempre se sitúa por debajo de la línea verde, correspondiente al bitrate TFRC. La distancia entre estas dos líneas es el bitrate excedente y no aprovechado debido a la selección del DQT (en rojo), por otra parte, esto implica una mayor descongestión de la red.

La Figura 4 muestra cómo, debido a las preferencias de usuario, los campos D y T oscilan ampliamente para adaptar el flujo a la capacidad de la red. En la Figura 5 podemos observar cómo el campo Q oscila entre su máximo valor 1 y su mínimo 0. En la Figura 6 es el campo T el que oscila de manera desmesurada mientras Q y D se mantienen casi inalterados. Si observamos con detención las Figuras 4 y 6, podremos observar cómo gracias a las preferencias del cliente se han reducido los picos correspondientes al Fraction Lost, lo que se podría traducir como una reducción en la congestión del router intermedio. No obstante, una mala selección de las preferencias de cliente pueden provocar caídas desmesuradas en la calidad del vídeo, puesto que la capa seleccionada es la base (0,0,0), tal y como se puede observar en la Figura 7. Este comportamiento podría ser intencionado si se antepone la solución a la congestión a la percepción del servicio del usuario. Viendo las Figuras 6 y 7, junto con otras no incluidas en esta memoria, podemos inferir en general que al ajustar primero el DID dándole la menor preferencia del cliente, el impacto en el bitrate es más notable, consiguiéndose una reducción mayor en la tasa de pérdidas.

4. Conclusiones y Vías Futuras

Todavía queda mucho por hacer respecto a la transmisión responsable de contenidos multimedia. La extensión de escalabilidad para H.264/AVC o H.264/SVC goza de las propiedades necesarias para el futuro de la transmisión multimedia. No obstante, en la actualidad, las implementaciones existentes se encuentran todavía en un estado muy precoz, lo que ha dificultado la conclusión de esta tesis. Hubiese sido conveniente incluir un análisis PSNR posterior a la transmisión de cada una de las pruebas, pero no ha sido posible debido a que el decodificador que proporciona el ITU-T no contempla la posibilidad de reducir el nivel de escalabilidad dinámicamente.

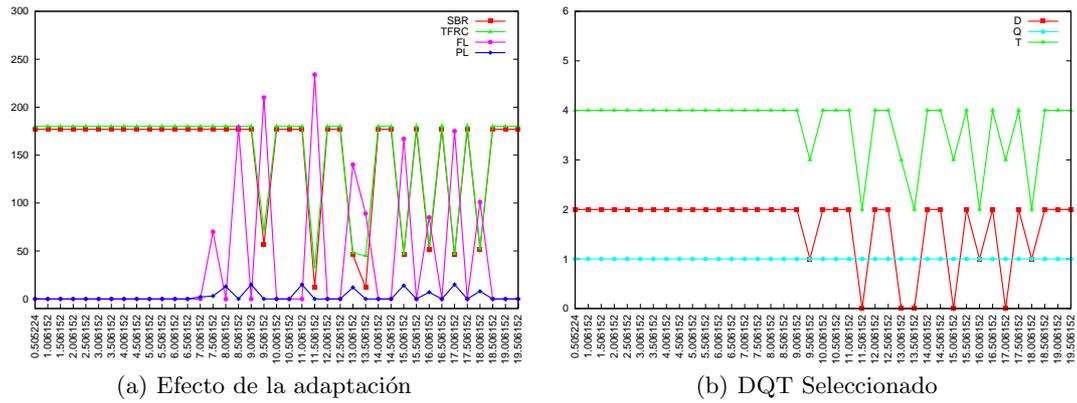


Figura4. Orden de Preferencia DQT del cliente 2,1,3 mínimos 1,0,2

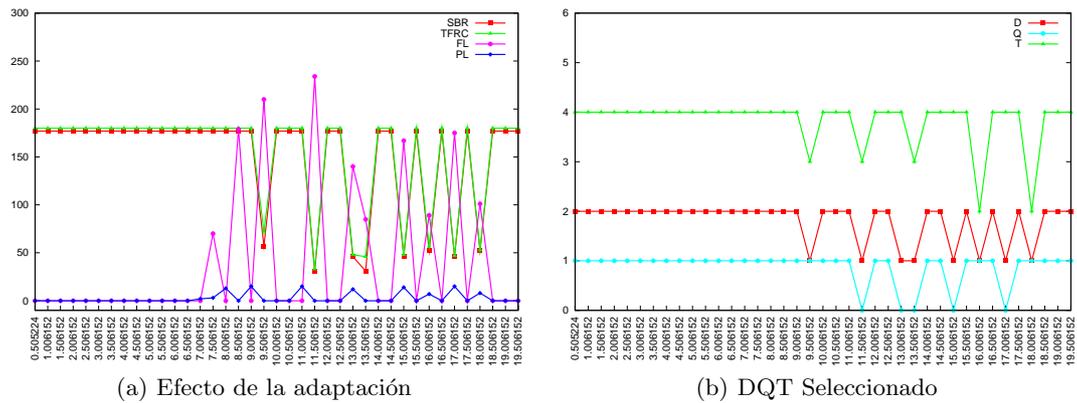


Figura5. Orden de Preferencia DQT del cliente 1,2,3 mínimos 1,0,2

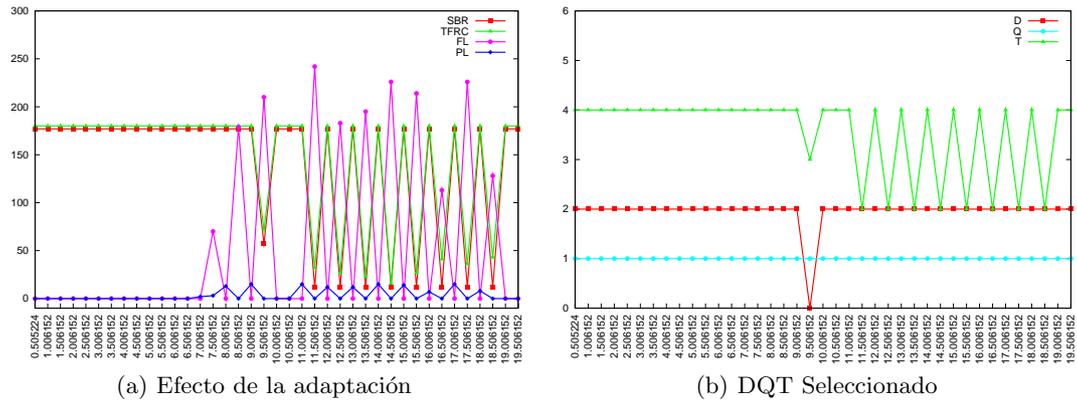


Figura6. Orden de Preferencia DQT del cliente 3,1,2 mínimos 0,0,0

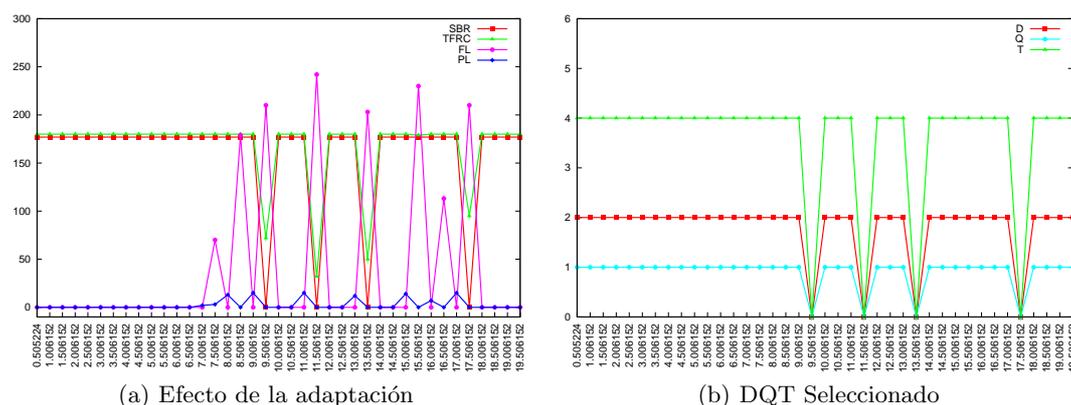


Figura 7. Orden de Preferencia DQT del cliente 3,1,2 mínimos 0,0,4

Por otra parte, en las simulaciones hemos aplicado inmediatamente el DQT calculado por el algoritmo, esto podría producir comportamientos no deseados en el cliente y la percepción del flujo por parte del usuario se vería mermada. En cualquier caso, el objetivo era ver el efecto sobre la congestión. En una implementación para un despliegue sería interesante suavizar esta reacción. Aplicar los cambios al principio de los GoPs (Imágenes I) sería una buena opción que además haría independiente el método de si se emplea MGS o no.

Sería interesante estudiar cómo el cliente podría transmitir sus preferencias al MANE. En este aspecto una ampliación del MPEG-21 sería necesaria, ya que parece la mejor opción para este objetivo. El proyecto Celtic SCALNET[4], del cual esta tesis forma parte como proceso de divulgación, propondrá próximamente una ampliación al respecto. Sería interesante un análisis ampliado de cómo realizar la adaptación cuando se emplea agregación, tanto unitemporal como multitemporal, y fragmentación.

Aprovechando la capacidad de transmisión en múltiples flujos de cada fuente SVC, sería interesante investigar la posibilidad de cómo distribuir la carga por distintas conexiones y/o MANEs. En el caso de los MANEs, sería necesario algún sistema de distribuir la información necesaria para la adaptación.

También resultaría interesante realizar un estudio en profundidad para diferenciar las situaciones de congestión frente a situaciones de pérdidas de paquetes, normalmente debidas a problemas en la red. En este último caso se pueden aplicar codificación FEC (Forward Error Correction) para mitigar el problema en la medida de lo posible, algo que ya se está aplicando en proyectos como SCALNET[4]. En cuanto al control de la congestión que ha sido introducido en esta tesis, sería interesante estudiar, implementar y evaluar la solución propuesta en[24] para vídeo escalable, en la que se procura, entre otras cosas, tener en cuenta la historia de la congestión de forma más avanzada a como se hizo en esta tesis por el sistema de pesos.

La plataforma SVC ofrece grandes posibilidades para los sistemas VoD. Una propiedad que sería deseable en tal caso sería la de poder integrar el control de acceso. Se podría, por ejemplo, cifrar la capa base permitiendo únicamente a los usuarios legítimos acceder al vídeo, mientras que el resto de capas circularían en claro. Cifrar un vídeo para un número elevado de usuarios puede suponer un problema de escalabilidad. Una posible propuesta sería situar el MANE en el Autenticador en una sesión EAP (Extensible Authentication Protocol). En tal caso, y suponiendo que el método EAP empleado sea generador de información criptográfica, el MANE y el cliente poseerían una clave simétrica compartida, lo que permitiría cifrar el contenido de tal forma que; únicamente ese usuario pudiese acceder al mismo. No obstante, habría que realizar un estudio exhaustivo de cuanta información del flujo sería necesario cifrar, puesto que generalmente, los autenticadores suelen encontrarse en dispositivos de capacidad de cálculo reducida y sería necesario optimizarlo. Quizá únicamente con el cifrado en la capa base de las cabeceras de las NALUs VLCs y las NALUs no-VCL al completo sería suficiente para lograr un control de acceso efectivo.

Referencias

1. Robert Kuschnig Hermann Hellwagner Ingo Kofler, Martin Prangl. An h.264/svc-based adaptation proxy on a wifi router. Technical report, Klagenfurt University, 2008.
2. Muchael Ransburg Hermann Hellwagner Robert Kuschnig, Ingo Kofler. Design options and comparison of in-network h.264/svc adaptation. Technical report, Klagenfurt University, 2008.
3. Ist danae project. <http://danae.rd.francetelecom.com/index.php>.
4. Celtic scalnet project. <http://www.celtic-initiative.org/Projects/SCALNET/default.asp>.
5. Iain E. G. Richardson. *H.264 and MPEG-4 video compression*. 2003.
6. Advanced video coding for generic audiovisual services. Itu-t recomendation, ITU-T, November 2007. ITU-T Recommendation H.264.
7. Colin Perkins. *RTP - Audio and Video for the Internet*. 2003.
8. R. Frederick V. Jacobson H. Schulzrinne, S.Casner. Rtp: A transport protocol for real-time applications. RFC 3550, IETF, July 2003.
9. C. Perkins M. Handley, V. Jacobson. Sdp: Session description protocol. RFC 4566, IETF, July 2006.
10. Nikos Passas Apostolis K. Salkintzis. *Emerging Wireless Multimedia Services and Technologies*. 2005.
11. J. Holler G. Camarillo, G. Ericsson. Grouping of media lines in the session description protocol (sdp). RFC 3388, IETF, December 2002.
12. S. Wenger T. Schierl. Signaling media decoding dependency in session description protocol. internet-draft, IETF, November 2008. draft-ietf-mmusic-decoding-dependency-08.txt.
13. R. Lanphier H. Schulzrinne, A. Rao. Real time streaming protocol (rtsp). RFC 2326, IETF, April 1998.
14. Mpeg-21. <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm>.
15. Information technology – multimedia framework (mpeg-21) – part 7: Digital item adaptation. Technical report, ISO/IEC, 2004.
16. Cc/pp. <http://www.w3.org/TR/CCPP-struct-vocab/>.
17. G. Klyne. A syntax for describing media feature sets. RFC 2533, IETF, March 1999.
18. Jitendra Padhye Jorg Widmer Sally Floyd, Mark Handley. Equation-based congestion control for unicast applications. Technical report, September 2000.
19. D. Towsley J. Kurose J. Padhye, V. Firoiu. Modeling tcp throughput: A simple model and its empirical validation. Technical report, SIGCOMM, August 1998.
20. T. Kristensen Y.-K. Wang, R.Even. Rtp payload format for h.264 video. internet-draft, IETF, April 2009. draft-ietf-avt-rtp-rfc3984bis-06.txt.
21. T. Schierl A. Eleftheriadis S. Wenger, Y.-K. Wang. Rtp payload format for svc video. internet-draft, IETF, November 2008. draft-ietf-avt-rtp-svc-18.txt.
22. A. Mutz K. Holtman L. Masinter, D. Wing. Media features for diplay, print and fax. RFC 2534, IETF, March 1999.
23. L. Rizzo L. Vicisano, J. Crowcroft. Tcp-like congestion control for layered multicast data transfer. Technical report, *UniversityCollegeLondon, UniversitàdiPisa*, 1998.
24. Martin May Bernhard Plattner Jinyao Yan, Kostas Katrinis. Media- and tcp-friendly congestion control for scalable video streams. Technical report, IEEE, April 2006.
25. P. Overell D. Crocker. Augmented bnf for syntax specifications: Abnf. RFC 4234, IETF, October 2005.

- o Propietario/creador e identificación de sesión.
- s Nombre de sesión.
- i*¹ Información de la sesión.
- u* URI de la descripción.
- e* Dirección de email.
- p* Número de teléfono.
- c* Información de la conexión (no indispensable si está incluida posteriormente en cada uno de los medios).
- b* Información de ancho de banda.
 - Una o más *descripciones temporales*.
- z* Ajustes de zona horaria.
- k* Clave de encriptación.
- a* Cero o más líneas de atributo de sesión.
 - Cero o más *descripciones de medio*.
- Descripción temporal.
 - t Tiempo en que la sesión está activa.
 - r* Cero o más instantes repetidos.
- Descripción de medios.
 - m Nombre del medio y dirección a nivel de transporte.
 - i* Título del medio.
 - c* Información de conexión (opcional si se incluyó en la descripción de sesión).
 - b* Información de ancho de banda.
 - k* Clave de encriptación.
 - a* Cero o más líneas de atributo de medio.

El atributo 'o' sigue el siguiente formato:

o=<nombre de usuario> <identificador de sesión> <versión> <tipo de red> <tipo de dirección> <dirección>
 p.e. o=Jordi 362348795 362348795 IN IP4 192.168.1.1

El identificador de sesión debe ser único. Se recomienda hacer uso de una marca de tiempo NTP. Adicionalmente el campo versión se emplea para diferenciar la antigüedad de las sesiones, aunque se puede emplear linealmente; se aconseja emplear también una marca de tiempo NTP. Otro factor importante en la descripción de la sesión es la descripción temporal, donde básicamente se informa sobre el tiempo de inicio y finalización. Pueden existir tantos atributos 't' como reinicios de la sesión, sin embargo en casos de repeticiones periódicas se aconseja emplear 'r'.

Finalmente la parte más importante de toda descripción SDP es la referente a los distintos medios. El atributo 'm' sigue el siguiente formato:

m=<medio> <puerto> <transporte> <fmt list>

Los tipos definidos para m son *audio, video, application, data, control, text y message*; data se refiere a modo crudo, si bien data y control están obsoletos. El significado del puerto depende del parámetro 'c'. El transporte también depende de c, si bien el IANA define como mínimo los valores 'RTP/AVP' y 'udp', en el caso particular de RTP/XYZ, XYZ hace referencia al perfil de RTP empleado. El 'fmt list' será una lista de tipos de carga útil tal y como se define en el perfil de RTP. En caso de no seleccionar RTP como transporte se indicará para este último parámetro un tipo de contenido MIME como se describe en el Apéndice B de [9]. Para RTP se permite además emplear un valor entre 96 y 127 para establecer un tipo dinámico, en cuyo caso será necesaria información adicional denominada 'rtpmap' que es agregada en forma de atributo de medio 'a' (uno por cada uno de los valores en la fmt list).

a=rtpmap:<tipo de carga útil> <codificación> / <reloj>[<parámetros de codificación>]

```
m=audio 49230 RTP/AVP 96 97 98
a=rtpmap:96 L8/8000
a=rtpmap:97 L16/8000
a=rtpmap:98 L16/11025/2
```

E. Atributos empleados en DDP

Atributo 'depend'

```
depend-attribute =
    "a=depend:" dependent-fmt SP dependency-tag
    *(";" SP dependent-fmt SP dependency-tag) CRLF
dependency-tag =
    dependency-type *1( SP identification-tag ":"
    fmt-dependency *("," fmt-dependency ))
dependency-type = "lay"
                  / "mdc"
                  / token
dependent-fmt = fmt
fmt-dependency = fmt
```

Los valores SP y CRLF se definen en [25] donde está definida toda la notación ABNF. La relación dependent-fmt dependency-tag es uno a uno. Recordar que fmt se refiere al tipo de carga útil RTP o un tipo MIME. Cuando dependency-type="lay" todos los flujos de medios necesarios para la decodificación del punto de operación tiene que ser identificados por identification-tag y fmt-dependency tras la cadena "lay". Por ejemplo:

```
a=mid:L3
a=depend:100 lay L1:96,97; 101 lay L1:97 L2:99
```

Donde los valores 100 y 101 son dependent-fmt haciendo referencia al propio flujo que está definiéndose. L1,L2,L3 son identification tags, L3 corresponde al flujo actualmente bajo definición y L1, L2 deben ser definidos con anterioridad. 96,97,99 Son flujos de los que depende el flujo bajo definición.

F. Mensajes RTSP

Existe un tipo de cabecera que es común a ambos tipos de mensaje Request y Response, las denominadas 'general-header':

```
general-header = Cache-Control;
                 | Connection;
                 | Date;
                 | Via;
```

- El mensaje *Request* sigue el siguiente formato

```
Request = Request-Line;
        *( general-header; |request-header; | entity-header );
        CRLF
        [ message-body ];
```

donde:

```
Request-Line = Method SP Request-URI SP RTSP-Version CRLF
```

```
Method = "DESCRIBE";
         | "ANNOUNCE";
         | "GET_PARAMETER";
         | "OPTIONS";
         | "PAUSE";
         | "PLAY";
         | "RECORD";
         | "REDIRECT";
         | "SETUP";
         | "SET_PARAMETER";
         | "TEARDOWN";
         | extension-method
```

```
extension-method = token
Request-URI = "*" | absolute_URI
```

```
RTSP-Version = "RTSP" "/" 1*DIGIT "." 1*DIGIT
```

```
request-header = Accept;
```

```

| Accept-Encoding;
| Accept-Language;
| Authorization;
| From;
| If-Modified-Since;
| Range;
| Referer;
| User-Agent;

```

Actualmente RTSP-Version es "RTSP/1.0". Una Request-URI con valor '*' implica que se desea aplicar el comando a todo el servidor.

- El mensaje *Response* sigue el siguiente formato

```

Response = Status-Line;
          *( general-header;
            | response-header;
            | entity-header ); CRLF [ message-body ]
Status-Line = RTSP-Version SP Status-Code SP Reason-Phrase CRLF

```

En general, siguiendo el modelo de HTTP, se definen los siguientes grupos de estados:

```

1xx: Informational - Request received, continuing process
2xx: Success - The action was successfully received, understood, and accepted
3xx: Redirection - Further action must be taken in order to complete the request
4xx: Client Error - The request contains bad syntax or cannot be fulfilled
5xx: Server Error - The server failed to fulfill an apparently valid request

```

G. Protocolo TFRC

G.1. TFRC Protocol

Principios de diseño

- No buscar agresivamente ancho de banda excedente, esto es, incrementar la tasa de envío lentamente en respuesta a un decremento en la tasa de eventos perdidos.
- No reducir la tasa de envío a la mitad en respuesta a un evento de pérdida único. En cualquier caso, reducir la tasa de envío a la mitad en caso de múltiples eventos de pérdida sucesivos.
- El receptor debe notificar al emisor como mínimo una vez por RTT si no recibió ningún paquete en ese intervalo.
- Si el emisor no ha recibido notificaciones tras múltiples RTTs, entonces el emisor debería reducir su tasa de envío y finalmente detener totalmente la emisión.

Requisitos

- Los parámetros R y p son determinados. La tasa de eventos perdidos (p) tiene que ser calculada en el receptor, mientras que RTT (R) puede ser medida bien en el emisor o en el receptor.
- El receptor envía bien el parámetro p o bien el valor calculado T.
- El emisor incrementa o decrementa su tasa de transferencia basado en su cálculo de T.

Funcionalidad del emisor

El emisor suaviza el RTT medido usando una media móvil exponencialmente balanceada. Este balanceo determina la sensibilidad de la tasa de transferencia a los cambios en el RTT.

El emisor puede derivar el t_{RTO} como: $t_{RTO} = SRTT + 4 * RTT_{var}$, siendo RTT_{var} la varianza y SRTT la estimación. No obstante en la práctica sólo afecta críticamente a la tasa de envío permitida cuando la tasa de pérdida de paquetes es muy alta.

Si $T_{actual} > T$, el emisor tiene que decrementar su tasa de envío. Algunas opciones son:

- Decrementar exponencialmente.
- Decrementar en dirección hacia T.
- Decrementar a T. Esta es la opción seleccionada por los autores de [18].

Funcionalidad del receptor

El cálculo de la tasa de eventos perdidos (p) es una de las partes más críticas de TFRC.

Algunas recomendaciones para la estimación de la tasa de eventos perdidos:

- Debería mantenerse relativamente lineal en un entorno con una tasa de eventos perdidos en steady-state.
- Debería medir la tasa de eventos perdidos mejor que la tasa de paquetes perdidos, donde un evento perdido puede consistir en múltiples paquetes perdidos en un RTT.
- Debería responder enérgicamente a eventos perdidos en múltiples RTT sucesivos.
- Debería incrementarse únicamente como respuesta a un evento perdido.
- Debería permitir definir un intervalo de pérdidas como el número de paquetes entre eventos perdidos. La tasa de eventos perdidos estimada debería decrecer sólo en respuesta a un nuevo intervalo mayor que la media previamente calculada o un intervalo lo suficientemente largo desde la última pérdida.

H. Algoritmo de adaptación

DID_MaxValue \leftarrow Valor máximo que puede tomar el DID
 QID_MaxValue \leftarrow Valor máximo que puede tomar el QID
 TID_MaxValue \leftarrow Valor máximo que puede tomar el TID
 DQT_MinBitRate[] \leftarrow Relación de bitrates mínimos para las combinaciones de DQT.
 DQT_MaxBitRate[] \leftarrow Relación de bitrates mínimos para las combinaciones de DQT.
 NetMaxBitrate \leftarrow Bitrate máximo soportado por la red, calculado por el TFRC.
 SelectedDQT \leftarrow Combinación DQT seleccionada
 primero, segundo y tercero \leftarrow Punteros a los valores temporales DIDtemp, QIDtemp o TIDtemp, asignados según las preferencias del usuario
 m1, m2 y m3 \leftarrow Multiplicadores de los punteros.

User:

MinUserPrimero \leftarrow mínimo seleccionado por el usuario para el de mayor preferencia [0,1]
 MinUserSegundo \leftarrow mínimo seleccionado por el usuario para el segundo de mayor preferencia [0,1]
 MinUserTercero \leftarrow mínimo seleccionado por el usuario para el de menos preferencia [0,1]
 RDQT \leftarrow Relación de preferencia de DID, TID, QID podría representarse como una cadena cuya suma debiera ser 6, del estilo 3,2,1 donde a DID le damos el valor de preferencia 3 y así sucesivamente para TID y QID, aunque sería más interesante que fuese un entero 123, 132, 213, 231, 312, 321. Esta última es la solución seleccionada por simplicidad

```

MECANISMO PARA LA SELECCIÓN DEL DQT MÁXIMO
Para cada paquete RTP de la sesión{
  //Inicialización del valor al máximo permitido por el stream
  SelectedDQT = DID_MaxValue * 128 + QID_MaxValue * 8 + TID_MaxValue
  // En caso que sea posible mandar el máximo, mandarlo
  Si(DQT_MaxBitRate[SelectedDQT] <= NetMaxBitrate) return SelectedDQT
  Sino //Necesitamos ajustar el ancho de banda
  {
    Según Sea(RDQT) //Inicialización de variables
    {
      caso 123:
        primero = DIDtemp; //Asignación de punteros
        primero_maxvalue = DID_MaxValue;
        segundo = QIDtemp;
        segundo_maxvalue = QID_MaxValue;
        tercero = TIDtemp;
        tercero_maxvalue = TID_MaxValue;
        m1=128; m2=8; m3=1;
      caso 132:
        primero = DIDtemp; //Asignación de punteros
        primero_maxvalue = DID_MaxValue;
        segundo = TIDtemp;
        segundo_maxvalue = TID_MaxValue;
        tercero = QIDtemp;
        tercero_maxvalue = QID_MaxValue;
        m1=128; m2=1; m3=8;
      caso 213:
        primero = QIDtemp; //Asignación de punteros
        primero_maxvalue = QID_MaxValue;
    }
  }
}
  
```

```

segundo = DIDtemp;
segundo_maxvalue = DID_MaxValue;
tercero = TIDtemp;
tercero_maxvalue = TID_MaxValue;
m1=8; m2=128; m3=1;
caso 231:
primero = TIDtemp; //Asignacion de punteros
primero_maxvalue = TID_MaxValue;
segundo = DIDtemp;
segundo_maxvalue = DID_MaxValue;
tercero = QIDtemp;
tercero_maxvalue = QID_MaxValue;
m1=1; m2=128; m3=8;
caso 312:
primero = QIDtemp; //Asignacion de punteros
primero_maxvalue = QID_MaxValue;
segundo = TIDtemp;
segundo_maxvalue = TID_MaxValue;
tercero = DIDtemp;
tercero_maxvalue = DID_MaxValue;
m1=8; m2=1; m3=128;
caso 321:
primero = TIDtemp; //Asignacion de punteros
primero_maxvalue = TID_MaxValue;
segundo = QIDtemp;
segundo_maxvalue = QID_MaxValue;
tercero = DIDtemp;
tercero_maxvalue = DID_MaxValue;
m1=1; m2=8; m3=128;
}
//Mientras que no nos ajustemos al ancho de banda seleccionado por TFRC
Mientras(DQT_MaxBitRate[SelectedDQT] > NetMaxBitrate)
{
Si(tercero > MinUserTercero)
{
//Todavía podemos reducir el de menos preferencia
tercero --;
}
Sino si (tercero == MinUserTercero y segundo > MinUserSegundo)
{
//Reducimos el segundo de preferencia
tercero = tercero_maxvalue;
segundo --;
}
Sino si (tercero == MinUserTercero y segundo == MinUserSegundo y primero > MinUserPrimero)
{
//Reducimos el de más preferencia
segundo = segundo_maxvalue;
primero --;
}
Sino //No nos queda más preferencia que reducir, tomar directamente valor que se ajuste
{
//Intentaremos minimizar este valor para ajustarnos lo máximo posible al bitrate calculado
bitratesobrante = NetMaxBitRate;
ParaTodo(DQT_MinBitRate[i])
{
Si((DQT_MinBitRate[i] - NetMaxBitrate > 0) y (DQT_MinBitRate[i] - NetMaxBitrate < bitratesobrante))
{
bitratesobrante = DQT_MinBitRate[i] - NetMaxBitrate;
SelectedDQT = i;
FinMientras; //Ya tenemos el valor seleccionado
}
}
SelectedDQT = 0; //Tomamos el flujo de la capa base H.264/AVC Compatible
}
} //Fin mientras
//En este punto tenemos una seleccion de primero,
// segundo y tercero que nos permite adaptar el flujo,
// ahora tenemos que deshacer la asignación y calcular el selectedDQT
selectedDQT = primero * m1 + segundo * m2 + tercero * m3;
//Podemos adaptar a SelectedDQT
} //Fin sino
} //Fin para
/*Ahora debemos comprobar que el SelectedDQT existe.
Podría darse el caso de que no existiese para un valor de QID máximo.
En tal caso, tomamos el valor de la capa anterior a la no existente, es decir,
si el QID máximo es 4, puede que un DQT 0,0,4 no exista, en tal caso, tomaríamos el 0,0,3.*/
} //Fin para

```