

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 392 461**

51 Int. Cl.:

H04N 21/2381 (2011.01)

H04N 21/262 (2011.01)

H04N 21/61 (2011.01)

H04N 21/6405 (2011.01)

H04N 21/643 (2011.01)

H04N 21/854 (2011.01)

H04N 7/66 (2006.01)

H04L 1/00 (2006.01)

H04L 29/06 (2006.01)

H04L 29/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07701090 .8**

96 Fecha de presentación: **04.01.2007**

97 Número de publicación de la solicitud: **1969856**

97 Fecha de publicación de la solicitud: **17.09.2008**

54 Título: **Gestión de archivo contenedor de medios**

30 Prioridad:

05.01.2006 US 743095 P

45 Fecha de publicación de la mención BOPI:

10.12.2012

45 Fecha de la publicación del folleto de la patente:

10.12.2012

73 Titular/es:

**TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)
(100.0%)
16483 Stockholm , SE**

72 Inventor/es:

**LOHMAR, THORSTEN;
WESTERLUND, MAGNUS y
FRÖJDH, PER**

74 Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

ES 2 392 461 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Gestión de archivo contenedor de medios.

Campo técnico

5 La presente invención se refiere de manera general a gestión de medios y multimedia, y en particular a la creación y uso de archivos contenedores de medios que contienen tal contenido de medios y multimedia.

Antecedentes

10 El suministro de medios y multimedia a un cliente sobre diferentes redes ha aumentado tremendamente los últimos años. Hoy en día, Internet se emplea por numerosos usuarios para acceder y descargar medios, por ejemplo, en forma de secuencias o archivos de vídeo y audio, a partir de servidores de medios. Este suministro de medios también ha emergido en redes de comunicaciones móviles basadas en radio. Hay actualmente un interés muy grande en usar redes móviles para contenido multimedia o de TV. Esto a menudo se conoce como TV Móvil en la técnica. Este suministro de medios en las redes móviles está disponible hoy en día principalmente a través de transporte unidifusión. No obstante, por el momento, los métodos de entrega de radiodifusión/multidifusión para TV Móvil están bajo desarrollo. Ejemplos de tales esfuerzos de estandarización son los Servicios de Radiodifusión/Multidifusión Multimedia del 3GPP (MBMS) y Radiodifusión de Vídeo Digital-De Mano (DVB-H) del Instituto de Estándares de Telecomunicaciones (ETSI).

20 En línea con esta demanda creciente de suministro de medios en diferentes redes de comunicaciones cableadas e inalámbricas, hay trabajo en curso en el desarrollo de servidores de difusión en forma continua y descarga disponibles en redes inalámbricas para proporcionar contenido de medios para clientes que lo requieran. Hay una tendencia general hacia servidores de difusión en forma continua/descarga transparente y flexible, que implica que los servidores deberían constar básicamente de una multitud de módulos o programas "estándar" que realizan diferentes funciones de gestión de medios. El contenido de medios de entrada para estas funciones entonces se proporciona junto con instrucciones de cómo los módulos/programas deberían procesar el contenido. Esto proporcionará un suministro de medios más flexible comparado con el uso de procesamiento de medios fijo, predefinido en los servidores.

25 De esta manera hay una necesidad de un formato de contenedor de medios que se puede usar en línea con la tendencia de tener servidores de difusión en forma continua/descarga transparente y flexible.

30 El documento "VM de la Enmienda 2 a la 14496-12:2005 de pistas de sugerencia de FLUTE de la ISO/IEC en el formato de archivo de medios de base ISO" propone la introducción de casillas relacionadas con las pistas de sugerencia de FLUTE a ser añadidas en el formato archivos de medios de base ISO como una enmienda. En tal caso, FLUTE se puede usar para la transmisión de partes estáticas de medios sobre conexiones multidifusión/radiodifusión, mientras que RTP se usa para la transmisión de medios en tiempo real.

35 El documento "FLUTE para descargar MBMS," borrador del 3GPP, S4-030772 de la Reunión de TSG-SA4#29 del 3GPP el 24-28 de noviembre de 2003, Tampere, Finlandia proporciona una introducción a FLUTE que es adecuada para el transporte de anuncios de servicio MBMS y descarga de medios.

El documento WO 2004/036760 A1 trata con el cálculo previo de datos FEC para soportar protección adaptativa de soporte que usa múltiples pistas de sugerencia.

Resumen

La presente invención supera estas y otras desventajas de las adaptaciones de la técnica anterior.

40 Es un objeto general de la presente invención proporcionar un archivo contenedor de medios que se puede usar en sesiones multimedia.

Este y otros objetos se cumplen por la invención como se define por las reivindicaciones de patente anexas.

Brevemente, la presente invención implica la generación y uso de un archivo contenedor de medios y a dispositivos para generar y usar tal archivo contenedor.

45 La generación de un archivo contenedor de medios implica proporcionar al menos un archivo fuente de medios que comprende datos de medios y multimedia a ser transmitidos a clientes que los requieren y presentados a los clientes. Este archivo contenedor se considera como que consta de uno o más bloques fuente de medios, dependiendo del tamaño del archivo contenedor. Al menos uno de tal bloque fuente de medios se procesa de acuerdo con la presente invención dividiendo el bloque fuente en símbolos fuente o partes que tienen un tamaño que las permiten ser procesadas por un códec FEC para el propósito de calcular datos o símbolos de redundancia FEC basados en el bloque fuente. La información de una partición de bloque seleccionado para usar el bloque fuente de medios se genera y almacena en el archivo contenedor junto con el contenido de medios real del bloque fuente. Los metadatos se proporcionan e incluyen en el archivo contenedor para proporcionar una asociación entre un bloque

fuentes de medios y su información de partición.

El archivo contenedor resultante se puede emplear por un servidor de medios durante una sesión de medios para calcular, usando la información de partición en el archivo contenedor, datos FEC y para compilar paquetes de datos que comprenden datos de medios a partir del archivo y los datos FEC calculados. El procesamiento previo del bloque fuente de medios en términos de dividirlos en base a diferentes algoritmos FEC y generar información FEC incluida en el archivo contenedor de la invención permite al servidor de medios, de una manera barata computacionalmente simple, calcular los datos FEC e insertar los datos de medios y los datos FEC en paquetes de datos transmitidos a clientes que los solicitan sin procesamiento de datos extensivo y particiones de bloques demandantes computacionalmente.

- 5
- 10 En una realización preferente, el archivo contenedor también comprende instrucciones que se usan y se siguen por el servidor de medios cuando se calculan los datos FEC y cuando se compilan paquetes de datos que contienen datos FEC y de medios a partir del archivo contenedor. En tal caso, el archivo contenedor comprende todo el contenido de medios e instrucciones requeridas para ser capaz con éxito de reenviar los datos de medios de una manera fiable a los clientes.

15 **Breve descripción de los dibujos**

La invención junto con otros objetos y ventajas de la misma, se pueden entender mejor haciendo referencia a la siguiente descripción tomada junto con los dibujos anexos, en los cuales:

La Fig. 1 es un diagrama de flujo que ilustra un método para generar un archivo contenedor de medios de acuerdo con un aspecto de la presente invención;

- 20 La Fig. 2 es un diagrama de flujo que ilustra el bloque de medios que proporciona el paso de la Fig. 1 en más detalle;

La Fig. 3 es un diagrama de flujo que ilustra pasos adicionales del método de generación de archivos de la Fig. 1;

La Fig. 4 es una vista general esquemática de un archivo contenedor de medios de acuerdo con otro aspecto de la presente invención;

- 25 La Fig. 5 es un diagrama de flujo que ilustra un método de gestión de sesiones de medios de acuerdo con un aspecto adicional de la presente invención;

La Fig. 6 es un diagrama de flujo que ilustra pasos adicionales del método de gestión de sesiones de la Fig. 5;

La Fig. 7 es un diagrama de flujo que ilustra pasos adicionales del método de gestión de sesiones de la Fig. 5;

- 30 La Fig. 8 es una ilustración esquemática que muestra la compilación de diferentes secuencias de medios que emplean un archivo contenedor de medios de acuerdo con la presente invención;

La Fig. 9 es una vista general de una red de comunicaciones y que incluye un servidor de medios que maneja un archivo contenedor de medios de acuerdo con la presente invención;

La Fig. 10 es un diagrama de bloques esquemático de un servidor de contenidos de acuerdo aún con otro aspecto de la presente invención;

- 35 La Fig. 11 es un diagrama de bloques esquemático que ilustra una realización del proveedor de bloques de medios de la Fig. 10 en más detalle;

La Fig. 12 es un diagrama de bloques esquemático que ilustra una realización del creador de archivos contenedores de la Fig. 10 en más detalle; y

- 40 La Fig. 13 es un diagrama de bloques esquemático de un servidor de sesiones de medios de acuerdo aún con otro aspecto de la presente invención.

Descripción detallada

En todos los dibujos, los mismos caracteres de referencia se usarán para elementos correspondientes o similares. La presente invención se refiere de manera general a una gestión de datos de medios y multimedia y en particular a la creación y utilización de archivos contenedores en conexión con servidores de medios, tales como servidores de difusión en forma continua o descarga, en una red de comunicaciones basada en radio. Estos archivos contenedores de medios de la invención comprenden el contenido de medios para transmitir al(a los) cliente(s) que lo solicitan y las instrucciones usadas para realizar el procesamiento de medios y la transmisión en los servidores de medios. De acuerdo con la invención, el contenido de medios se ha procesado previamente y está presente en una forma que permite la gestión eficiente por un servidor de medios durante una sesión de medios. En un aspecto particular de la invención, el contenido de medios original se ha procesado previamente para simplificar, en el

- 45
- 50

servidor de medios, el suministro de protección de fiabilidad a la sesión de medios. De esta manera, el contenido esta en un formato que permite el cálculo de los datos de protección de fiabilidad del contenido de medios que se pueden usar en la sesión de medios para aumentar las oportunidades de entrega con éxito a los clientes solicitantes.

5 De acuerdo con una realización preferente de la invención, el procesamiento previo del contenido de medios se realiza y adapta para el cálculo, en el servidor de medios, de los datos de redundancia de corrección de errores sin canal de retorno (FEC) a ser incluidos en la secuencia de medios transmitida.

10 Como se conoce en la técnica, la FEC implica añadir datos redundantes a los datos de carga útil transmitidos, lo cual permite a un receptor detectar y corregir errores sin la necesidad de solicitar al remitente datos adicionales. La ventaja de la FEC es que la retransmisión de datos a menudo puede ser ignorada, aunque a costa de mayores requisitos de ancho de banda de media. De esta manera, la FEC ventajosamente se puede usar en conexión con entrega de contenido de medios basado en multidifusión/radiodifusión, donde las retransmisiones serán difíciles de implementar.

15 La FEC se consume añadiendo redundancia a la información a ser transmitida usando un algoritmo o esquema seleccionado, típicamente designado códec FEC en la técnica. Cada tal bit redundante es invariablemente una función compleja de mucha información original o bits de carga útil. Un códec FEC que incluye la entrada sin modificar en la salida es designado como un códec sistemático. En otras palabras un códec FEC sistemático (N,K) preserva la fuente K o símbolos de carga útil y adjunta (N-K) símbolos FEC. De la misma manera, un código FEC (N,K) no sistemático crea N símbolos (FEC o fuente) a partir de K símbolos fuente sin preservar necesariamente todos ellos.

20 Hay dos categorías principales de códec FEC: códec bloque y códec convolucionales. Los códec bloque FEC trabajan en bloques de tamaño fijo (paquetes) de bits o símbolos de tamaño predeterminado, mientras los códec convolucionales trabajan en secuencias de bits o símbolos de longitud arbitraria. El código Raptor de Digital Fountains es un códec de bloque FEC que es capaz de crear un número arbitrario de símbolos de redundancia FEC de un único bloque fuente. Esta es una propiedad ventajosa de este códec FEC dado que esta significa que diferentes configuraciones de sobrecarga de protección no requieren un cambio en la construcción del bloque fuente. Reed-Solomon es otro códec de bloque FEC que, no obstante, requiere un cambio en la partición de bloque fuente para diferentes tamaños de sobrecarga de protección. Otros ejemplos de códec bloque FEC incluyen Golay, BCH (Bose, Ray-Chaudhuri, Hocquenghem) y Hamming. Un códec FEC preferente a usar en conexión con la presente invención es el código Raptor de Digital.

30 De acuerdo con la invención actual, datos o contenido de medios o multimedia se refiere a cualquier dato que puede ser proporcionado por un proveedor o servidor de contenidos a un cliente para la reproducción de los datos. Ejemplos típicos preferentes incluyen datos de vídeo y datos de audio. Los datos pueden estar en forma de una versión de contenido vídeo o audio de tasa fija, pre-codificado o en forma de un contenido audio o vídeo escalable. Otros ejemplos de medios incluyen imágenes fijas (JPEG), gráficos de asignación de bits (GIF y PNG), gráficos vectoriales (SVG), y audio sintético (SP-MIDI) y texto (XHTML y SMIL).

35 La Fig. 1 es un diagrama de flujo de un método para generar un archivo contenedor de medios de acuerdo con la presente invención. Este archivo contenedor de medios se puede considerar como un paquete de entrada completo que se usa por un servidor de medios durante una sesión de medios para proporcionar un contenido de medios y formar datos de medios en paquetes de datos transmisibles. De esta manera, el archivo contenedor comprende preferentemente, además del contenido de medios por sí mismo, información e instrucciones requeridas por el servidor de medios para realizar el procesamiento y permitir la transmisión del contenido de medios durante una sesión de medios.

40 El método empieza en el paso S1 donde se proporciona al menos un bloque fuente de medios. El al menos único bloque fuente de medios comprende los datos o símbolos de medios que se pretende que sean enviados a un cliente, donde son reproducidos para presentar contenido de medios a un usuario. Los bloques de medios pueden ser de un tamaño fijo, igual o al menos una parte del mismo, si hay más de uno, pueden ser de diferentes tamaños de bit/símbolo. Si se proporcionan más de dos bloques fuente de medios en el paso S1, se pueden considerar como bloques de medios separados de un mismo archivo o secuencia de contenido de medios, por ejemplo una secuencia de vídeo, y/o de diferentes archivos o secuencias de medios, por ejemplo unos pocos bloques fuente de medios de una secuencia de vídeo y unos pocos bloques de medios de una secuencia de audio asociada correspondiente.

45 En este paso S1, se puede proporcionar contenido de medios asociado. Por ejemplo, si el contenido de medios incluye un vídeo musical, al menos un bloque de medios proporcionado preferentemente comprende los datos de vídeo y al menos un bloque de medios contiene los datos de audio correspondientes. No obstante se anticipa mediante la presente invención que uno y el mismo contenido de medios se puede proporcionar en múltiples versiones de medios potenciales. Por ejemplo, la parte vídeo del vídeo musical se puede proporcionar en múltiples versiones de vídeo codificadas previamente, donde cada una de tal versión de vídeo se adapta para uso en conexión con un ancho de banda dado o nivel o intervalo de tasa de bit. Múltiples versiones de un contenido de medios dado se puede proporcionar por lo tanto en el paso S1. En tal caso, cada una de tal versión de medios se puede considerar como que consta de uno o más bloques de fuente de medios. Aunque múltiples versiones de

medios se pueden proporcionar y organizar en el archivo contenedor, típicamente solamente se usa una de tales versiones en un momento dado durante la sesión de medios, aunque puede haber una conmutación entre las versiones de medios durante la sesión en base a, por ejemplo, cambios en niveles de ancho de banda disponibles.

5 En un siguiente paso S2, el bloque fuente de medios proporcionado es parte en símbolos fuente o llamados trozos. Estos símbolos generalmente constan de un número de cientos de octetos. Esta partición de bloque se realiza al menos parcialmente en base a información del códec/algorithm FEC a ser usado, por un servidor de medios, para calcular símbolos FEC para el bloque fuente de medios actual. Como fue mencionado anteriormente, los códec FEC basados en Reed-Solomon requieren un cambio en la partición de bloque fuente en base al tamaño de sobrecarga de protección deseado, es decir el número de símbolo FEC de redundancia. De esta manera, el códec o algoritmo
 10 FEC real y/o la sobrecarga de protección FEC podría afectar a la partición de la fuente de medios y los tamaños de símbolo de medios. También otros parámetros tales como el tamaño de los paquetes de datos, tales como los paquetes de Protocolo de Datagramas de Usuario (UDP), usados por un servidor de medios para transmitir el contenido de medios se pueden usar en esta partición de bloque fuente del paso S2. En tal caso, el tamaño de los símbolos de fuente podría estar limitado de manera que al menos un símbolo de fuente completo se pueda ajustar
 15 dentro de un paquete UDP.

Este paso S2 de partición no implica necesariamente que el bloque fuente de medios esté físicamente dividido en símbolos fuente separados que se almacenan en ubicaciones separadas en el archivo contenedor. En claro contraste, en las implementaciones más prácticas, el archivo fuente multimedia se almacena como una secuencia de
 20 datos continua en el archivo contenedor pero se considera como o virtualmente divide en bloques fuente de medios, a su vez virtualmente particionados en símbolos fuente.

En un siguiente paso S3, se genera la información de la partición. Esta información básicamente especifica qué partes de la secuencia de datos pertenecen a qué símbolo fuente del bloque fuente de medios. La información de partición se puede organizar en una tabla que especifica que el bit X al bit Y del bloque fuente de medios Z constituye un símbolo fuente. Alternativamente, la información puede incluir el tamaño en octetos de cada símbolo
 25 fuente. Entonces, sabiendo la posición inicial de un bloque fuente de medios en un archivo de medios, es posible determinar qué partes de datos pertenecen a los diferentes símbolos fuente.

Se anticipa por la presente invención que un bloque fuente de medios dado se puede dividir de acuerdo a al menos dos operaciones de partición diferentes. Por ejemplo, un primer bloque de partición tal se podría adaptar a un primer algoritmo FEC, mientras una segunda partición se emplea adaptada a un segundo algoritmo FEC. Adicionalmente,
 30 la partición del bloque también puede ser dependiente de la sobrecarga FEC deseado real en el servidor de medios. En tal caso, diferentes particiones de bloques diseñadas para diferentes sobrecargas FEC se podrían determinar en la creación del archivo. La información de partición entonces comprende información de estas alternativas de partición aplicables a un bloque fuente de medios dado.

El bloque de medios divididos y la información de partición generada están organizados en el archivo contenedor de medios en el paso S4. De esta manera, el contenido de medios y la información se insertan como partes del archivo contenedor. Adicionalmente, el al menos un bloque de medios organizado en el archivo contenedor preferentemente comprende colectivamente todos los datos de contenido de medios que van a ser transmitidos a un cliente durante
 35 una sesión de medios. En otras palabras, el archivo contenedor contiene los datos de medios para una presentación multimedia entera.

Un siguiente paso S5 proporciona metadatos a ser incluidos en el archivo contenedor. Estos metadatos proporcionan una asociación entre el bloque fuente de medios añadido al archivo contenedor en el paso S4 y la partición almacenada en el archivo en el paso S4. Esta asociación puede ser en forma de un puntero desde la posición de almacenamiento del bloque fuente de medios dentro del archivo y a la posición de almacenamiento de la información de partición, o viceversa. Estos metadatos por lo tanto permiten, dado el bloque fuente de medios particular o su posición dentro del archivo contenedor, la identificación de la información de posicionamiento asociada o la posición de almacenamiento de esta información dentro del archivo. En lugar de emplear un puntero, los metadatos pueden incluir un identificador del bloque fuente de medios y/o la información asociada, en el caso de que éstos estén almacenados en posiciones predefinidas, "estándar" o "por defecto" en el archivo contenedor. Los metadatos entonces se usan para identificar uno del bloque fuente de medios y la partición en el archivo y se basan
 45 en esta posición identificada distinta del bloque fuente de medios y se puede identificar la información.
 50

En una implementación típica de la invención, se proporcionan bloques fuente de medios múltiples en el paso S1. En tal caso, los pasos S2 a S5 preferentemente se repiten para cada bloque fuente de medios tal o al menos para múltiples grupos de bloques fuente de medios, lo cual se ilustra esquemáticamente por la línea L1. De esta manera, si N bloques fuente de medios se organizan en el archivo contenedor en el paso S1, los pasos S2 a S5 preferentemente se repiten N veces, implicando también organizar N versiones de información de partición y metadatos en el archivo contenedor junto a los bloques fuente.
 55

El método entonces termina.

Como fue mencionado en lo anterior, un objetivo de la presente invención es proporcionar un archivo contenedor de

medios que, además de los datos de medios reales, también comprenda información de partición dependiente de la FEC a ser usada cuando se calculan datos FEC. Esto significa que la división archivo a bloque fuente y la partición de bloques se hacen “fuera de línea” e independiente del proceso de transmisión de medios real en un servidor de medios. Este procesamiento previo simplifica las tareas del servidor y reduce los requisitos de rendimiento y complejidad del servidor. Además, el archivo contenedor también comprende preferentemente información e instrucciones requeridas por un servidor de medios para identificar y componer datos de medios y datos FEC calculados en una secuencia de medios que se pueden transmitir a clientes que los requieran. De esta manera, el archivo contenedor se puede considerar por lo tanto como un paquete de datos completo, información e instrucción que se pueden usar por servidores transparentes y flexibles para compilación de datos y transmisión.

La generación del archivo contenedor descrita anteriormente en conexión con la Fig. 1 es preferentemente conducida a un creador o servidor de contenidos que tiene acceso a fuentes de contenido de medios internas o externas. El archivo contenedor generado entonces se puede representar en un medio de almacenamiento tal como una memoria de ordenador, o en una señal física tal como una señal eléctrica o una señal radio, por ejemplo, para transferir dentro de un sistema local o para transmisión sobre una red local o global. En una realización típica, el archivo contenedor se proporciona como una señal radio a un servidor de medios para uso en una sesión de medios con diferentes clientes.

A continuación, el término archivo contenedor de medios se usará en toda la revelación con un significado que incluye tanto archivos de datos para almacenamiento en un medio de almacenamiento como señales para transferencia o distribución.

La Fig. 2 es un diagrama de flujo que ilustra pasos adicionales del método de generación del archivo contenedor de la Fig.1. El método comienza en el paso S10 en que se proporciona al menos un archivo fuente de medios. En esta realización ilustrativa, el contenido de medios está disponible en el creador de archivo contenedor en forma de archivos o secuencias fuente que contienen los datos de medios. En este paso S10, uno o más archivos fuente de medios se pueden proporcionar para inclusión dentro de un archivo contenedor. Por ejemplo, un primer archivo de medios puede contener datos vídeo, mientras que un segundo archivo asociado contiene datos audio. Para permitir el cálculo eficiente de datos de redundancia FEC en el servidor de medios y, más tarde, el uso de tales datos FEC, el(los) archivo(s) fuente de medios proporcionado(s) en el paso S10 se divide(n) en un número de bloques de medios en un siguiente paso S11. Este bloque fuente de medios entonces se puede considerar como un segmento del archivo fuente de medios, al cual se puede aplicar u operar un código FEC. El tamaño de los bloques de medios en términos de símbolos fuente de bits de datos se puede predefinir o seleccionar en conexión con la división. En el primer caso, el tamaño se podría definir por el esquema o código FEC pretendido a ser empleado para el cálculo de datos de redundancia FEC. De esta manera, el códec o algoritmo FEC real y/o la sobrecarga de protección FEC requerido podría afectar la división de archivo de medios y los tamaños del bloque de medios.

En el caso de que el archivo fuente de medios de entrada tenga un tamaño de bit o símbolo que sea más pequeño que el tamaño máximo que se puede manejar eficazmente por un códec FEC, ninguna división de ese archivo fuente en bloques fuente de medios es motivo necesario y el paso S11 se puede omitir. El archivo fuente de medios de entrada entonces se considera como un bloque fuente de medios de acuerdo a la invención.

Se tiene que señalar que incluso aunque hay un tamaño de bloque preferente, no todos los bloques fuente de medios generados a partir de un archivo fuente de medios necesitan ser de ese tamaño preferente. Por ejemplo, el último bloque fuente de medios podría ser de un tamaño más pequeño comparado con los otros bloques equidimensionados dado que la parte restante del archivo de medios no contiene datos de medios suficientes para alcanzar el tamaño de bloque preferente.

El paso de división S11 no implica necesariamente que el archivo fuente de medios esté dividido físicamente en bloques fuente de medios separados que se almacenan en posiciones separadas en el archivo contenedor. En claro contraste, en la mayoría de las implementaciones prácticas, el archivo fuente de medios se almacena como una secuencia de datos continua en el archivo contenedor pero se considera como o divide virtualmente en bloques fuente de medios. Por ejemplo, un archivo fuente de medios que contiene $2N$ símbolos fuente se puede dividir de manera que el símbolo fuente $[0, N-1]$ pertenece al primer bloque fuente y el símbolo $[N, 2N-1]$ pertenece al segundo bloque fuente.

En un siguiente paso S12, se proporciona información del archivo fuente de medios particular. Esta información puede ser de relevancia para el servidor de medios, cuando va a proporcionar secuencias de paquetes de datos de medios y/o calcular datos FEC.

El método entonces continúa al paso S2 de la Fig. 1, donde el(los) bloque(s) fuente de medios está(n) divididos en bloques fuente de medios.

La Fig. 3 es un diagrama de flujo que ilustra pasos adicionales del método de generación del archivo contenedor de la Fig. 1. El método continúa desde el paso S5 de la Fig. 1. En un siguiente paso S20, se proporciona información del algoritmo FEC del esquema, en base a la cual fue calculada la partición de bloque. Esta información puede estar en forma del nombre del algoritmo particular o alguna otra información descriptiva. En un planteamiento alternativo,

5 cada algoritmo FEC disponible tiene un identificador predefinido. De esta manera, solamente este identificador FEC entonces se puede proporcionar en el paso S20. En una implementación típica, se ha empleado información de un códec FEC único para dividir todos los bloques fuente de medios de un archivo fuente de medios. No obstante, realmente podría ser posible usar diferente información de códec FEC cuando se dividen para diferentes bloques fuente de medios de un archivo fuente de medios dado o para bloques fuente de medios de diferentes archivos fuente. La información FEC por lo tanto podría especificar que todos los bloques en el archivo contenedor se han dividido para permitir el cálculo de símbolos FEC con un códec FEC único o identificar qué bloque fuente o grupo de bloques fuente para los cuales se han usado diferentes códec FEC. En este contexto, es posible realmente dividir un bloque fuente dado en base a diferentes códec FEC objetivo potenciales y/o sobrecargas FEC. En tal caso, la información de estos diferentes códec FEC se proporciona preferentemente en el paso S20.

10 En un siguiente paso S21, se proporciona preferentemente una tabla de propiedades. Esta tabla de propiedades es útil en particular si se incluye más de un/una archivo/secuencia fuente de medios en el archivo contenedor pero también se puede usar ventajosamente cuando solamente contiene un archivo fuente de medios único. La tabla de propiedades de archivo típicamente contiene información del tipo de medios de los archivos fuente de medios, preferentemente el tipo de Extensiones de Correo de Internet Multipropósito (MIME) del medio. De esta manera, esta información MIME podría especificar que los medios son medios de audio, medios de video o algún otro tipo de medios, incluyendo Lenguaje de Integración de Medios Sincronizado (SMIL). Este tipo MIME proporciona información al servidor de medios de qué tipo de datos están incluidos realmente en el archivo contenedor. La tabla de propiedades también puede incluir información de cualquier esquema de codificación empleado en los datos de medios, incluyendo gzip. También se puede incluir información de tamaño en la tabla de propiedades. Esta información de tamaño podría expresar el tamaño total de cada archivo fuente de medios en términos de número de octetos o símbolos, los respectivos tamaños de los bloques fuente de medios del(de los) archivo(s) fuente (básicamente correspondientes a la información de división proporcionada en el paso S12 de la Fig. 2), el tamaño de la carga útil máximo u objetivo para paquetes de datos a ser usado cuando se transmiten datos, el tamaño (en octetos) de un símbolo fuente de medios (básicamente correspondiente a la información de partición generada en el paso S3 de la Fig. 1) y/o símbolo FEC, etc. Un nombre de archivo o identificador de archivo se incluye preferentemente en la tabla de propiedades para cada archivo fuente de medios incluido en el archivo contenedor.

15 La información de la ubicación de almacenamiento real de cada archivo fuente de medios en el archivo contenedor se encuentra preferentemente en la tabla de propiedades. Esta información de ubicación podría especificar la posición de comienzo del primer bloque fuente de medios de ese archivo fuente y después los bloques de medios restantes se encuentran a continuación a esta posición en el archivo contenedor. Los metadatos generados en el paso S5 de la Fig. 1 y que proporcionan un asociado entre bloques fuente de medios e información de partición en el archivo contenedor también se puede incluir en la tabla de propiedades. De la misma manera, la información de la partición del bloque empleado y el códec FEC previsto para el cálculo de datos de redundancia se incluye preferentemente en la tabla.

20 La tabla de propiedades del archivo contenedor se podría usar por lo tanto como una fuente de información única para que una fuente de medios localice un archivo/bloque fuente de medios relevante y otra información útil en la compilación de paquetes de medios durante una sesión de medios.

25 Las instrucciones de cálculo FEC se generan en un paso siguiente S22. Estas instrucciones se usan por un servidor de medios para calcular los datos FEC para los bloques fuente de medios en base a los metadatos y los datos de partición de aquellos bloques fuente. De esta manera, estas instrucciones podrían ser consideradas como sugerencias o metadatos que proporcionan instrucciones de cómo símbolos fuente de medios de bloques fuente diferentes se introducen a un códec FEC, preferentemente definidas en base a la información FEC, para generar símbolos FEC a ser usados junto con el contenido de medios en una sesión de medios para proporcionar protección de fiabilidad. Las instrucciones se usan preferentemente junto con la información de partición y los metadatos (que permiten la identificación de la información de partición) para calcular un depósito de símbolos FEC para al menos uno de los bloques fuente de medios.

30 En una implementación preferente, la instrucción de cálculo FEC define el cálculo de un conjunto de símbolos de redundancia FEC para el al menos un bloque fuente de medios. Este conjunto de símbolos FEC podría incluir uno pero preferentemente múltiples símbolos FEC calculados en base a los símbolos fuente del bloque fuente de medios.

35 Las instrucciones de cálculo FEC podrían incluir múltiples instrucciones alternativas para un bloque fuente de medios dado. Por ejemplo, una primera instrucción FEC alternativa se emplea cuando un primer códec FEC va a ser usado para el cálculo de datos FEC como se determina en base a la información de algoritmo FEC. De la misma manera, una segunda instrucción FEC alternativa define el cálculo de datos FEC con un segundo códec FEC. Alternativamente, o además, las instrucciones FEC alternativas se podrían adaptar a diferentes sobrecargas FEC y por ello básicamente especificar el cálculo de diferente número de símbolos FEC dependiendo de la máxima (o mínima) sobrecarga FEC permitida.

40 Un siguiente paso S23 genera la compilación de instrucciones para uso por un servidor de medios. Estas instrucciones se usan para definir la compilación de datos de medios desde los bloques fuente de medios y los datos

de redundancia FEC calculados en base a la(s) instrucción(es) de cálculo FEC para formar una secuencia de medios de paquetes de datos. De esta manera, estas instrucciones se podrían considerar como sugerencias o metadatos que proporcionan instrucciones de cómo usar los datos incluidos en el archivo contenedor y calculados de allí para componer una secuencia de paquetes de medios transmisible que tiene protección de fiabilidad. Estas instrucciones se usan por lo tanto para compilar datos de medios y datos FEC juntos en paquetes adecuados para transmisión a clientes que los requieran durante una sesión de medios. Las instrucciones por lo tanto describirán el orden de transmisión en el lado del servidor de los datos fuente de medios y datos FEC.

Señalar sin embargo que las instrucciones típicamente no incluyen información de planificación de tiempo, información de direcciones objetivo/fuente o puertos u otra información de sesión específica. Esto significa que el archivo contenedor y las instrucciones de compilación allí dentro son transparentes a la sesión particular y realmente se pueden usar por un servidor de medios para sesiones diferentes múltiples con diferentes clientes de recepción pero también mediante servidores de medios diferentes.

Las instrucciones de compilación podrían aplicarse a un subconjunto de bloques fuente de medios, que implican que múltiples de tales instrucciones tienen que ser leídas y usadas por un servidor de medios durante una sesión. Alternativamente, una instrucción de compilación comprende toda la información requerida para un archivo fuente de medios único o verdaderamente para todos los archivos fuente de medios en el archivo contenedor.

Más de un conjunto de instrucciones de compilación se pueden generar realmente en el paso S23. En tal caso, diferentes instrucciones alternativas se podrían proporcionar, de manera que un servidor de medios tenga una opción para determinar qué conjunto de instrucciones particulares emplear para una sesión de medios particular. Por ejemplo, una primera instrucción de compilación se podría usar para describir el orden de transmisión de los bloques fuente de medios y datos FEC cuando se emplea un único canal de transmisión para la transmisión de datos. Una segunda instrucción se podría aplicar a los mismos bloques fuente de medios y datos FEC pero proporciona información de orden de compilación y transmisión si están disponibles múltiples canales, que implica que los datos se pueden transmitir en paralelo en lugar de secuencialmente. De esta manera, las diversas instrucciones de compilación se pueden usar para proporcionar sesiones de transmisión alternativas previstas para diferentes condiciones de canal de transporte.

De una manera similar, se pueden incluir instrucciones de compilación alternativas para diferentes sobrecargas de protección de fiabilidad. Por ejemplo, una primera instrucción de compilación se usa para describir el orden de compilación y transmisión de bloques fuente de medios y datos FEC para un primer nivel de sobrecarga de protección máximo, mientras que una segunda instrucción se usa por los mismos bloques fuente de medios pero con un segundo nivel de sobrecarga FEC diferente. Si este segundo nivel de sobrecarga FEC es mayor (menor) que el primer nivel, se pueden añadir más símbolos FEC o símbolos de paridad como también son denotados en la técnica a una cantidad dada de símbolos fuente de medios.

Un siguiente paso S24 organiza la información, tabla e instrucciones proporcionadas y generadas en los pasos previos S20 a S23 y preferentemente el paso S12 de la Fig. 2 en el archivo contenedor. El archivo contenedor entonces contendrá el conjunto completo de datos de medios, información, instrucciones y metadatos "brutos" requeridos por un servidor de medios para identificar, calcular y componer datos para transmisión a cliente(s) que los requieren. El método entonces termina.

La Fig. 4 es una vista general esquemática de un archivo contenedor de medios 1 de acuerdo con la presente invención. Como se ha descrito anteriormente, el archivo contenedor 1 contiene datos de medios de un número de archivos fuente de medios 10, 12, 14, M de tales archivos se ilustran en la figura, donde $M \geq 1$. Los datos de medios de cada archivo se consideran como divididos en un número de bloques fuente de medios 20, 22, 24. En la figura Q_1 de tales bloques 20, 22, 24 se han ilustrado por el primer archivo fuente de medios 10, donde $Q_1 \geq 1$. Cada uno de tales bloques fuente de medios 20, 22, 24 es a su vez considerado como que está dividido en símbolos fuente.

Además de los bloques fuente de medios 20, 22, 24 con datos de medios, el archivo contenedor 1 comprende conjuntos de información de partición 30, 32, 34 que contienen información generada previamente de las particiones de símbolos fuente que se han aplicado a los bloques fuente de medios 20, 22, 24 con el propósito de facilitar el cálculo de datos de redundancia FEC a ser usados en conexión con los datos de medios para proporcionar protección de fiabilidad. En una implementación preferente, cada bloque fuente de medios comprende un conjunto de información de partición dedicada 30, 32, 34. En tal caso, el número N de conjuntos de información 30, 32, 34 en

la figura M es:
$$N = \sum_{i=1}^M Q_i .$$

Los metadatos de asociación 40 de la invención que proporciona una asociación entre la información de partición 30, 32, 34 y el(los) bloque(s) fuente de medios 20, 22, 24 al cual la información de partición 30, 32, 34 es aplicable también se proporciona en el archivo contenedor 1. La Fig. 4 ilustra múltiples localizaciones posibles diferentes de estos metadatos 40 en el archivo 1. En una primera realización, los metadatos se almacenan en conexión con el(los) bloque(s) fuente de medios asociados 20, 22, 24. De esta manera, la identificación de un bloque fuente de medios 20, 22, 24 en el archivo también permite identificación de los metadatos 40 respectivos de ese bloque fuente 20, 22,

24. Alternativamente, o además, los metadatos 40 se almacenan junto con la información de partición respectiva 30, 32, 34. De esta manera, cada conjunto de información 30, 32, 34 tiene unos metadatos 40 de asociación conectados que permiten la identificación del(de los) bloque(s) fuente de medios relevantes 20, 22, 24 para el que aplica la información de partición particular 30, 32, 34. Si el archivo contenedor 1 comprende una tabla de propiedades de archivo preferente 60, los metadatos de asociación 40 se podrían proporcionar allí dentro. En tal caso, un servidor de medios podría sólo investigar la tabla de propiedades de archivo 60 para identificar la posición de los datos de medios relevantes y dividir datos para uso durante una sesión de medios. En una implementación posible adicional, los metadatos de asociación 40 se almacenan en conexión con las diferentes pistas de sugerencia 50, 52, 54 del archivo contenedor 1. Estas pistas de sugerencia comprenden el cálculo FEC y/o instrucciones de compilación para uso mediante un servidor de medios en conexión con una sesión de medios. En tal caso, cada pista de sugerencia 50, 52, 54 necesita sólo contener los metadatos 40 que se requieren para la sesión de medios implementable por las instrucciones en esa pista de sugerencia 50, 52, 54. Una combinación de múltiples de estas posibles localizaciones de almacenamiento también es posible y están dentro del alcance de la presente invención.

De acuerdo a una realización específica de la presente invención, el archivo contenedor de medios 1 es una unidad entrelazada, la cual está optimizada para descarga o difusión en forma continua progresiva. Por ello; una presentación multimedia entera se puede transmitir y descargar mediante las llamadas descarga o difusión en forma continua progresiva para clientes que lo solicitan.

El formato de archivo de medios de base ISO [1,2,3] se puede emplear ventajosamente como formato de archivo para el archivo contenedor de medios de la presente invención. Los formatos de archivo contenedor alternativos incluyen, el formato de archivo MP4, el formato de archivo 3GP y el formato QuickTime.

La Codificación de Capas Asíncrona (ALC) es un protocolo de entrega de contenido fiable escalable masivamente. Es un protocolo base para entrega multidifusión fiable de objetos binarios arbitrarios y ha sido adoptado como el protocolo obligatorio para entrega de archivos de radiodifusión/multidifusión en el grupo de trabajo de Radiodifusión (BCAST) de Navegador y Contenido (BAC) de la Alianza Móvil Abierta (OMA) y BCMCS (Servicio de Radiodifusión/Multidifusión) del 3GPP.

La FLUTE (Entrega de Archivos sobre Transporte Unidireccional) se construye en la parte superior de ALC y define un protocolo para entrega unidireccional de archivos y ha sido adoptado recientemente en el MBMS y Asignación de Datos DVB-H IP (IPDC) del 3GPP como el protocolo obligatorio para entrega de archivos de radiodifusión/multidifusión. Tanto FLUTE como ALC se definen por el Grupo de Trabajo de Ingeniería de Internet (IETF).

La FLUTE define una Tabla de Entrega de Archivos (FDT), la cual transporta metadatos asociados con los archivos entregados en la sesión ALC, y proporciona mecanismos para entrega en banda y actualizaciones de FDT. En contraste, ALC se basa en otros medios para entrega fuera de banda de los metadatos de archivo. OMA BCAST define una Guía de Servicio Electrónica (ESG) que normalmente se entrega a clientes con bastante antelación a la sesión ALC. Si los metadatos de archivo necesitan ser actualizados durante la sesión ALC, entonces los fragmentos de la ESG se pueden actualizar usando los canales entrega/actualización de la ESG.

Los archivos a ser entregados sobre ALC o FLUTE se pueden almacenar como elementos en un archivo contenedor ISO. La casilla Meta y sus casillas hijas permiten el almacenamiento de una variedad de elementos de datos, tales como medios estáticos (imágenes) y presentaciones SMIL, en un archivo de medios de base ISO. Ellas también permiten asociar nombres de archivo y trayectos a elementos y señalización de la estructura del directorio del archivo en el archivo de medios de base ISO.

Generalmente, el primer paso antes de que los archivos se puedan enviar sobre ALC/FLUTE es dividirlos en bloques fuente y símbolos fuente de acuerdo con la presente invención. La partición puede depender del esquema FEC, el tamaño del paquete objetivo, y la sobrecarga FEC deseada. Para cada bloque fuente de una codificación FEC, la información de partición se calculan previamente y se almacenan en el archivo de medios de base ISO junto con información sobre el esquema FEC y la información de división de archivos.

El siguiente paso para facilitar la transmisión de archivos es permitir que el archivo de medios de base ISO también contenga instrucciones que definen el cálculo de símbolos FEC a partir de los bloques fuente de medios usando la información de partición. Adicionalmente, el archivo de medios de base ISO preferentemente también comprende instrucciones para un servidor de multidifusión/radiodifusión que describe las sesiones ALC/FLUTE (con Protocolo de Descripción de Sesiones) y cómo encapsular elementos en paquetes ALC o FLUTE.

La partición de archivos, por una parte, y las pistas de sugerencia para entrega de archivos, por la otra, se pueden usar independientemente una de la otra. La primera ayuda al diseño de pistas de sugerencia y permite pistas de sugerencia alternativas, con, por ejemplo, diferentes sobrecargas FEC, para reutilizar los mismos símbolos FEC. Ellas también proporcionan medios para acceder a símbolos fuente. Para reducir la complejidad cuando un servidor sigue las instrucciones de pista de sugerencia, no obstante, las pistas de sugerencia se refieren directamente a gamas de datos de elementos o datos copiados en muestras de sugerencia.

A continuación se da un ejemplo de implementación de mayor detalle de un archivo contenedor en forma del formato

de archivo de medios de base ISO y adaptado para transmisión sobre ALC/FLUTE. Esto, no obstante, se debería ver meramente como un ejemplo ilustrativo útil para entender la presente invención.

Almacenamiento de archivos fuente

- 5 Los archivos previstos para transmisión sobre ALC/FLUTE se almacenan como elementos en una casilla Meta de nivel superior ('meta') de un archivo de medios de base ISO que actúa como un archivo contenedor. La casilla de Localización de Elementos ('iloc') especifica la localización de almacenamiento real de cada elemento (archivo fuente de medios) dentro del archivo contenedor así como el tamaño de archivo de cada elemento. El nombre de archivo, el tipo de contenido (tipo MIME), etc., de cada elemento se proporcionan por la casilla de Información de Elemento ('iinf').

10 Casilla de Información de Elementos FD

Los detalles en la partición de los archivos fuente se proporcionan en la casilla de Información de Elementos FD ('fiin'). La casilla se usa preferentemente para archivos que emplean pistas de sugerencia FD y preferentemente de manera exacta una se localiza en la casilla Meta ('meta'). Se define como sigue:

```
aligned(8) class FDItemInformationBox extends FullBox('fiin', version = 0, 0)
{
    unsigned int(16) entry_count;
    PartitionEntry[entry_count] partition_entries;
    SessionGroupBox session_info;
    GroupIdToNameBox group_id_to_name;
}
```

- 15 Cada PartitionEntry en la casilla de información de elementos FD proporciona detalles en una partición de archivos particular y metadatos para un archivo fuente de medios particular. Es posible proporcionar múltiples entradas para un archivo fuente si se usan particiones alternativas en el archivo ISO. Todas las entradas de partición se puede numerar implícitamente y la primera entrada tiene típicamente el número 1.

Entrada de partición

- 20 La Entrada de Partición ('paen') de una fuente se define como sigue:

```
aligned(8) class PartitionEntry extends Box('paen')
{
    FilePartitionBox blocks_and_symbols;
}
```

Puede contener una casilla que define la partición fuente de medios.

Casilla de Partición de Archivos

- 25 La casilla de Partición de Archivos ('fpar') identifica el archivo fuente y proporciona una partición de ese archivo en bloques y símbolos fuente. Definición:

```
aligned(8) class FilePartitionBox extends FullBox('fpar', version = 0, 0)
{
    unsigned int(16) item_ID;
    unsigned int(16) packet_payload_size;
    unsigned int(16) FEC_encoding_ID;
    unsigned int(16) FEC_instance_ID;
    unsigned int(16) max_source_block_length;
    unsigned int(16) encoding_symbol_length;
    unsigned int(16) max_number_of_encoding_symbols;
    string scheme_specific_info;
    unsigned int(16) entry_count;
    for (i=1; i <= entry_count; i++)
    {
        unsigned int(16) block_count;
        unsigned int(32) block_size;
    }
}
```

Semántica:

item_ID indica el item_ID del archivo fuente. Es posible proporcionar particiones alternativas de un archivo fuente usando el mismo item_ID en la casilla de Partición de Archivos de más de una entrada de Información de Archivos.

packet_payload_size da el tamaño de la carga útil de paquetes FLUTE o ALC objetivo del algoritmo de partición. Señalar que las cargas útiles de paquetes UDP son mayores, ya que también contienen cabeceras FLUTE o ALC.

5 **FEC_encoding_ID** identifica el esquema de codificación FEC. Un valor cero podría corresponder a un esquema por defecto, tal como el “esquema FEC sin Código Compacto” también conocido como “FEC Nulo” [4]. Un valor de uno corresponde preferentemente al “MBMS FEC” [5].

10 **FEC_instance_ID** proporciona una identificación más específica del codificador FEC que se usa para un esquema FEC Insuficientemente Especificado. Este valor típicamente no se usa para esquemas FEC Completamente Especificados. Ver el documento [4] para más detalles de los esquemas FEC Insuficientemente Especificados.

max_source_block_length da el número máximo de símbolos fuente por bloque fuente de medios.

encoding_symbol_length da el tamaño (en octetos) de un símbolo de codificación (símbolo fuente y símbolo de paridad FEC). Todos los símbolos de codificación de un elemento preferentemente tienen la misma longitud, excepto el último símbolo que puede ser más corto.

15 **max_number_of_encoding_symbols** da el número máximo de símbolos de codificación que se pueden generar para un bloque fuente para el ID de codificación FEC 129 definido en el documento [4].

scheme_specific_info es una cadena terminada nula codificada en base 64 de la información de transferencia de objeto de esquema específico (información de esquema específico FEC-OTI) en “FLUTEbis”. La definición de la información depende del ID de codificación FEC.

20 **entry_count** da el número de entradas en la lista de parejas (block_count, block_size) que proporciona una partición del archivo fuente. Comenzando desde el principio del archivo, cada entrada indica cómo el siguiente segmento del archivo se divide en bloques fuente y símbolos fuente.

25 **block_count** indica el número de bloques fuente consecutivo del tamaño de **block_size** (en octetos). Un block_size que no es un múltiplo del tamaño del símbolo (proporcionado en la Casilla de Información FEC) indica que el último símbolo fuente incluye relleno no almacenado en el elemento del archivo.

Casilla de Información de Elemento

30 Para transmitir internamente medios discretos integrados que usan protocolo de descarga de archivos de radiodifusión/multidifusión (ALC/FLUTE), es preferente para el servidor transmitir también algunos metadatos correspondientes a los medios discretos. Los metadatos se envían como parte de la FDT, si FLUTE se usa como un protocolo de radiodifusión, y como parte de OMA BCAS ES, si ALC se usa en conjunto con OMA BCAS ES.

Como alguna de la información de Metadatos se podría crear sobre la marcha, una estructura de plantilla para la parte de los metadatos que es estática y común tanto para FLUTE como ALC se define como una segunda versión de la entrada de información de elementos. Esta versión de la entrada de información de elemento se usa en la casilla de información de elemento para elementos que tienen una partición de archivos fuente.

```
aligned(8) class ItemInfoEntry extends FullBox('infe', version = 1, 0)
{
    unsigned int(16) item_ID;
    unsigned int(16) item_protection_index;
    unsigned int(32) content_length;
    unsigned int(32) transfer_length;
    string item_name;
    string content_type;
    string content_location;
    string content_encoding;
    string content_MD5;
    unsigned int(8) entry_count;

    for (i=1; i <= entry_count; i++)
    {
        unsigned int(32) group_id;
    }
}
```

Semántica:

item_id contiene o bien 0 para el recurso primario (por ejemplo el Lenguaje de Marcado Extensible (XML) contenido en una casilla 'xml') o bien el ID del elemento para el cual se define la siguiente información.

5 **item_protection_index** contiene o bien 0 para un elemento no protegido, o bien el índice basado en uno en la casilla de protección del elemento que define la protección aplicada a este elemento (la primera casilla en la casilla de protección de elemento tiene el índice 1).

content_length da la longitud total (en octetos) del archivo (no codificado).

transfer_length da la longitud total (en octetos) del archivo (codificado). Señalar que la longitud de transferencia es igual a la longitud de contenido si no se aplica codificación de contenidos (ver más adelante).

10 **item_name** es una cadena terminada nula en caracteres UTF 8 que contienen un nombre simbólico del elemento, es decir el nombre de archivo del elemento (archivo fuente).

content_type es una cadena terminada nula en caracteres UTF 8 con el tipo MIME del elemento. Si el elemento tiene codificados los contenidos (ver más abajo), entonces el tipo MIME se refiere al elemento después de la decodificación de contenidos.

15 **content_location** es una cadena terminada nula en caracteres UTF 8 que contiene el URI del archivo como se define en HTTP/1.1 [6].

20 **content_encoding** es una cadena terminada nula en caracteres UTF 8 usada para indicar que el archivo binario está codificado y necesita ser decodificado antes de interpretado. Los valores son como se define para Codificación de Contenidos para HTTP/1.1. Algunos valores posibles son "gzip", "comprimir" y "desinflar". Una cadena vacía indica ninguna codificación de contenidos. Señalar que el elemento se almacena después de que la codificación de contenidos se ha aplicado.

content_MD5 es una cadena terminada nula en caracteres UTF 8 que contiene un digesto MD5 del archivo [6, 7].

entry_count da el número de entradas en la lista siguiente.

group_ID indica un grupo de archivos al cual pertenece el elemento de archivo.

25 Todos los campos se emplean preferentemente. No obstante, es posible que una cadena terminada nula solamente contenga un nulo para indicar que el valor correspondiente del campo no se proporciona. Extensiones futuras a la casilla pueden añadir campos adicionales al final.

Considerando la información proporcionada en la casilla de Información de Archivos para cada elemento de la lista de elementos usados por una pista de sugerencias, se pueden construir las entradas de archivo necesarias para una FDT o una ESG.

30 La content_location de los recursos de medios integrados se pueden referir usando formas de Localización Universal de Recursos (URL) definidas en la Sección 8.44.7 del formato de archivos de medios de base ISO [1, 2]

Casilla de Grupo de Sesiones

35 Una sesión FD puede enviar simultáneamente sobre varios canales FD, cada uno de los cuales se describe por una pista de sugerencia FD. La casilla de grupo de Sesiones contiene una lista de sesiones así como todos los grupos de archivos de medios y pistas de sugerencia que pertenecen a cada sesión. Si hay más de una pista de sugerencia FD en el archivo contenedor, entonces una casilla de grupo de sesiones se presenta preferentemente en la casilla de información de elemento FD.

40 Solamente se debería procesar un grupo de sesiones en cualquier momento. La primera pista de sugerencia enumerada en un grupo de sesiones especifica el canal base. Si el servidor de medios no tiene preferencia entre los grupos de sesiones, la elección por defecto es típicamente el primer grupo de sesiones. Los ID de grupo de todos los grupos de archivos que contienen los archivos referenciados por las pistas de sugerencia se incluyen en la lista de grupos de archivos. Los ID de grupo de archivos a su vez se puede traducir en nombres de grupos de archivos (usando el ID del Grupo a la casilla de Nombre) que se puede incluir por el servidor en las FDT.

```
aligned(8) class SessionGroupBox extends Box('segr')
{
    unsigned int(16) num_session_groups;
    for(i=0; i < num_session_groups; i++)
    {
        unsigned int(8) entry_count;
        for (j=0; j < entry_count; j++)
        {
            unsigned int(32) group_ID;
        }
        unsigned int(16) num_channels_in_session_group;
        for(k=0; k < num_channels_in_session_group; k++)
        {
            unsigned int(32) hint_track_id;
        }
    }
}
```

Semántica:

num_session_groups especifica el número de grupos de sesión.

- 5 **entry_count** da el número de entradas en la lista siguiente que comprende todos los grupos de archivos con los que cumple el grupo de sesión. El grupo de sesión contiene todos los archivos incluidos en los grupos de archivos enumerados como se especifica por la entrada de información de elemento de cada archivo fuente. La FDT para el grupo de sesión preferentemente debería contener solamente aquellos grupos que se enumeran en esta estructura.

group_ID indica un grupo de archivos con el cual cumple el grupo de sesión.

- 10 **num_channels_in_session_groups** especifica el número de canales en el grupo de sesión. El valor de `num_channels_in_session_groups` es un número entero positivo.

hint_track_ID especifica el ID de pista de la pista de sugerencia FD que pertenece a un grupo de sesión particular. Una pista de sugerencia FD corresponde a un canal de Transporte de Codificación en Capas (LCT).

ID de Grupo a casilla de nombre

- 15 El ID de Grupo A casilla de Nombre asocia los nombres de grupos de archivos a ID de grupos de archivos usados en las entradas de información de elemento,

```
aligned(8) class GroupIdToNameBox extends FullBox('gitn', version = 0, 0)
{
    unsigned int(32) entry_count;
    for (i=1; i<=entry_count; i++)
    {
        unsigned int(32) group_ID;
        string group_name;
    }
}
```

Semántica:

entry_count da el número de entradas en la lista siguiente.

- 20 **group_ID** indica un grupo de archivos

group_name es una cadena terminada nula en caracteres UTF 8 que contiene el nombre del grupo de archivos correspondiente.

Formato de Pista de Sugerencia

- 25 La estructura de pista de sugerencia está generalizada para soportar muestras de sugerencias en múltiples formatos de datos. La muestra de pista de sugerencia contiene cualquier dato necesario para construir la cabecera del paquete del tipo correcto, y también contiene un puntero al bloque fuente de medios de los datos a los que pertenece en el paquete.

Formato de entrada de muestras

Las pistas de sugerencia FD son pistas de sugerencia ('sugerencia' de manejador de medios con un formato de entrada en la descripción de la muestra de 'fdp', corto para el Protocolo de Entrega de Archivos. El FDHintSampleEntry está contenido en la SampleDescriptionBox ('stds') y tiene la siguiente sintaxis:

```
class FDHintSampleEntry() extends SampleEntry ('fdp')
{
    uint(16) hintrackversion = 1;
    uint(16) highestcompatibleversion = 1;
    uint(16) partition_entry_ID;
    uint(16) FEC_overhead;
    box additionaldata[];
}
```

5

Semántica:

partition_entry_ID indica la entrada de partición en la casilla de información de elemento FD. Un valor cero indica que ninguna entrada de partición está asociada con esta entrada de muestra, por ejemplo, para la FDT.

10 **FEC_overhead** es un valor 8.8 fijo que indica el porcentaje de sobrecarga de protección usado por la(s) muestra(s) de sugerencia. La intención de proporcionar FEC_overhead es proporcionar características para ayudar a un servidor de medios a seleccionar un grupo de sesión (y las pistas de sugerencia FD correspondientes).

15 Los campos, "hintrackversion" y "highestcompatibleversion" tienen la misma interpretación como en la "RtpHintSampleEntry", descrita en la sección 10.2 del formato de archivos de medios de base ISO [1, 2]. Como datos adicionales se puede proporcionar una casilla time_scale_entry. Si no se proporciona, no hay indicación dada sobre temporización de paquetes.

Las entradas de archivos necesarias para una FDT o una ESG se pueden crear observando todas las entradas de muestra de una pista de sugerencia y las casillas de información de Metadatos de Archivos correspondientes de los elementos referenciados por los item_IDs anteriores. No se incluirán entradas de muestra en la pista de sugerencia si no están referenciadas por ninguna muestra.

20 Formato de muestra

Cada muestra FD en la pista de sugerencia generará uno o más paquetes FD. Cada muestra contiene dos áreas: las instrucciones para componer los paquetes, y cualquier dato adicional necesario cuando se envían esos paquetes (por ejemplo los símbolos de codificación que se copian en la muestra en lugar de residir en elementos para archivos fuente o FEC). Señalar que el tamaño de la muestra es conocido a partir de la tabla de tamaño de muestras.

25

```
aligned(8) class FDsample extends Box('fdsa')
{
    FDPacketBox packetbox[]
    ExtraDataBox extradata;
}
```

30

Los números de muestra de muestras FD definen el orden en que serán procesados por el servidor de medios. Del mismo modo, las casillas de Paquete FD en cada muestra FD aparecen en el orden que serán procesadas. Si la casilla de Entrada de Escala de Tiempo está presente en la Entrada de Muestra de Sugerencia FD, los tiempos de muestra se definen y proporcionan tiempos de envío relativos de paquetes para una tasa de bits por defecto. Dependiendo de la tasa de bit de transmisión real, un servidor puede aplicar escalado de tiempo lineal. Los tiempos de muestra pueden simplificar el proceso de planificación, pero es hasta que el servidor de medios envía paquetes de una manera oportuna.

Formato de entrada de paquete

35 Cada paquete en la muestra FD tiene la siguiente estructura [8-10]:

```
aligned(8) class FDpacketBox extends Box('fdpa')
{
    header_template LCT_header_info;
    unsigned int(16) entrycount1;
    dataentry header_extension_constructors[entrycount1];
    unsigned int(16) entrycount2;
    dataentry packet_constructors[entrycount2];
}
```

LCT_header_info contiene plantillas de cabecera LCT para el paquete FD actual.

entry_count1: cuenta los siguientes constructores.

header extension_constructors: estructuras que se usan para construir las extensiones de cabecera LCT.

5 **entry_count2**: cuenta los siguientes constructores.

packet_constructors: estructuras que se usan para construir el ID de la carga útil FEC y los símbolos fuente en un paquete FD.

Formato de plantilla de cabecera LCT

```
class header_template
{
    unsigned int(1) sender_current_time_present;
    unsigned int(1) expected_residual_time_present;
    unsigned int(1) session_close_bit;
    unsigned int(1) object_close_bit;
    unsigned int(4) reserved;
    unsigned int(16) transport_object_identifier;
}
```

10 La plantilla de cabecera LCT se puede usar por un servidor de medios para formar una cabecera LCT para un paquete. Señalar que algunas partes de la cabecera dependen de la política del servidor y no están incluidas en la plantilla. Algunas longitudes de campo también dependen de los bits de cabecera LCT asignados por el servidor. El servidor también puede necesitar cambiar el valor de la TOI.

Formato de constructor de extensión de cabecera LCT

15 Señalar que un servidor de medios puede identificar paquetes que incluyen la FDT observando si está presente la EXT_FDT.

```
aligned(8) class LCTheaderextension
{
    unsigned int(8) header_extension_type;
    unsigned int(8) header_extension_length;
    unsigned int(8) header_extension_content[];
}
```

header_extension_length se expresa en múltiplos de palabras de 32 bits. Un valor cero significa que la cabecera se genera por el servidor.

20 **header-extension_content** es el número de elementos igual a la **header_extension_length**.

Formato de Constructor de Paquete

Hay varias formas del constructor. Cada constructor es de 16 octetos para hacer más fácil la iteración. El primer octeto es un discriminador de unión. Esta estructura está basada en la sección 10.3.2 del formato de archivos de medios de base ISO [1, 2].

```

aligned(8) class FDconstructor(type)
{
    unsigned int(8) constructor_type = type;
}
aligned(8) class FDnoopconstructor extends FDconstructor(0)
{
    unsigned int(8) pad[15];
}
aligned(8) class FDimmediateconstructor extends FDconstructor(1)
{
    unsigned int(8) count;
    unsigned int(8) data[count];
    unsigned int(8) pad[14 - count];
}

aligned(8) class FDsampleconstructor extends FDconstructor(2)
{
    signed int(8) trackrefindex;
    unsigned int(16) length;
    unsigned int(32) samplenum;
    unsigned int(32) sampleoffset;
    unsigned int(16) bytesperblock = 1;
    unsigned int(16) samplesperblock = 1;
}
aligned(8) class FDitemconstructor extends FDconstructor(3)
{
    unsigned int(16) item_ID;
    unsigned int(16) extent_index;
    unsigned int(64) data_offset;
    unsigned int(24) data_length;
}
aligned(8) class FDxmlboxconstructor extends FDconstructor(4)
{
    unsigned int(64) data_offset;
    unsigned int(32) data_length;
    unsigned int(24) reserved;
}

```

Casilla de Datos Adicionales

Cada muestra de una pista de sugerencia FD puede incluir datos adicionales en una casilla de Datos Adicionales:

```

aligned(8) class ExtraDataBox extends Box('extr')
{
    bit(8) extradata[];
}

```

5

La Fig. 5 es un diagrama de flujo que ilustra un método de gestión de sesiones de medios de acuerdo con la presente invención. Esta gestión de sesiones de medios se dirige en un servidor de medios, tal como un servidor de difusión en forma continua o de descarga, y usa el archivo contenedor de medios de la presente invención. El método comienza en el paso S30, donde se proporciona un archivo contenedor de medios. Este suministro de archivos se puede realizar trayendo el archivo contenedor desde una ubicación de memoria del servidor de medios, que implica que el servidor ha recibido previamente el archivo desde un proveedor o creador de contenidos. Alternativamente, el servidor de medios puede, en conexión con una petición de datos de medios, ordenar o recibir el archivo contenedor desde un proveedor de contenidos.

En un siguiente paso S41, los datos de redundancia FEC se calculan usando al menos un bloque fuente de medios y en base a los metadatos y la información de partición asociada con el al menos un bloque fuente. En este paso de cálculo S41, un códec de bloque FEC, tal como un códec raptor de Digital Fountain, se emplea preferentemente para operar en forma de bloque de medios. No obstante, un códec FEC de convolución también se podría emplear y está dentro del alcance de la presente invención. En una implementación preferente, un conjunto de símbolos de redundancia FEC se generan para el al menos un bloque fuente de medios. Este conjunto de símbolos FEC podría incluir uno pero preferentemente múltiples símbolos FEC calculados en base a los símbolos fuente del bloque fuente de medios. El número de símbolos FEC a calcular para un bloque fuente de medios se podría definir mediante

20

limitaciones en el códec FEC empleado, ser una función del número de símbolos fuente de medios en el bloque fuente de medios o limitar por algunos otros criterios, por ejemplo sobrecarga FEC. Adicionalmente, la información incluida en el archivo contenedor de medios podría especificar la cantidad de datos de redundancia FEC a calcular.

5 En una implementación preferente, el archivo contenedor de medios también comprende instrucciones de cálculo FEC usada en este paso S31. Estas instrucciones se emplean entonces junto con la información de partición (y la meta información usada para identificar la información de partición) cuando se seleccionan símbolos fuente a ser introducidos al códec FEC para generación de símbolos FEC.

10 Si el archivo contenedor también comprende información de la división del(de los) archivo(s) fuente de medios en los bloques fuente de medios, esta información de división se puede usar para identificar símbolos de fuente correctos a ser usados para el cálculo de datos FEC.

15 El número de símbolos FEC calculado en el paso S31 se podría determinar por el servidor de medios en base a algún criterio local, tal como la sobrecarga FEC máxima/mínima actualmente permisible durante la sesión de medios. Alternativamente, como se mencionó anteriormente, la información proporcionada en el archivo contenedor, tal como las instrucciones de cálculo FEC, especifica la cantidad de datos de redundancia FEC a ser calculados para un bloque fuente de medios dado.

El cálculo de datos FEC en el paso S31 se puede realizar sobre la marcha durante la sesión de medios en curso. Alternativamente, el servidor de medios podría generar un depósito de datos FEC antes de la sesión real y entonces almacenar el depósito FEC en el archivo contenedor o en una memoria. Los datos FEC calculados previamente entonces se pueden usar con el contenido de medios en el archivo contenedor en una sesión de medios siguiente.

20 En un siguiente paso S32, los paquetes de datos de medios se compilan extrayendo datos multimedia desde el(los) bloque(s) fuente de medios del archivo contenedor y proporcionar datos de redundancia FEC como se calculó en el paso S31. El servidor de medios preferentemente recibe un identificador de los datos de medios a transmitir durante la sesión de medios. Alternativamente, el archivo contenedor solamente podría contener datos de medios de un único archivo de datos de medios de manera que no es necesaria ninguna selección de fuente de medios. En
 25 cualquiera de los dos casos, la información descrita previamente incluida en el archivo contenedor, tal como en la tabla de propiedades de archivo, se puede usar para identificar el comienzo del archivo de medios, es decir el primer bloque fuente de medios desde el cual la transmisión se debería iniciar. Adicionalmente, se podría usar información adicional incluida en el archivo contenedor como instrucciones de cómo los datos de medios y los datos FEC se deberían combinar e incluir en paquetes de datos adoptados para la transmisión inalámbrica sobre un canal o
 30 múltiples canales basados en radio a diferentes clientes.

35 En un siguiente paso S33, los paquetes de datos de medios compilados con protección de seguridad FEC se transmiten, preferentemente a través de técnicas de radiodifusión o multidifusión, a clientes, donde los datos de medios se pueden reproducir. La transmisión de paquetes típicamente se inicia una vez que el almacenador temporal de transmisión en el servidor de medios ha alcanzado un nivel dado. No obstante, durante la sesión de medios, nuevos paquetes de datos son compilados e introducidos en el almacenador temporal de transmisión, mientras que otros paquetes están siendo transmitidos, lo cual se ilustra esquemáticamente por la línea L2.

40 El archivo contenedor generado y la organización de datos de medios y el suministro de datos de partición calculados previamente allí dentro, reducen las necesidades de procesamiento del servidor de medios durante una sesión de medios. Esto por lo tanto conduce a una complejidad del servidor reducida y permite flexibilidad del servidor ya que el servidor no necesita realizar la construcción de bloque fuente y la partición sobre la marcha.

El método entonces termina.

45 La Fig. 6 es un diagrama de flujo ilustrando un paso adicional del método de gestión de sesiones de medios de la Fig. 5. El método continúa desde el paso S30 de la Fig. 5. En un siguiente paso S40, el servidor de medios selecciona un algoritmo o códec FEC a emplear para el cálculo de datos FEC. Este algoritmo FEC es preferentemente el algoritmo FEC usado cuando se divide un bloque fuente de medios, en base a qué dato FEC va a ser calculado. El archivo contenedor de medios por lo tanto comprende preferentemente información de este algoritmo/códec FEC, cuya información se emplea por el servidor multimedia en el paso de selección S40.

50 Si la información FEC comprende identificadores de múltiples algoritmos FEC disponibles y un bloque de fuente de medios dado está disponible en múltiples particiones diferentes alternativas, el servidor de medios preferentemente usa otra información de entrada, que incluye la capacidad de sobrecarga FEC y/o las instrucciones de cálculo FEC, para seleccionar qué algoritmo FEC y qué información de partición de bloques usar cuando se calculan los datos FEC.

El método entonces pasa al paso S31 de la Fig. 5, donde el algoritmo FEC seleccionado se usa para calcular los datos FEC.

55 La Fig. 7 es un diagrama de flujo que ilustra pasos adicionales del método de gestión de sesiones de la Fig. 4. El método continúa desde el paso S31 de la Fig. 5. En un siguiente paso S50 se determina una capacidad de

sobrecarga FEC que se puede emplear realmente para la transmisión de datos en la sesión de medios. Esta capacidad se puede determinar o al menos estimar en base a los niveles de ancho de banda asignables al servidor para la transmisión de medios, los niveles mínimo y máximo de tasa de bits para la(las) portadora(s) radio empleada(s) para esta transmisión de medios, etc. Realmente cualquier técnica para determinar tal capacidad de sobrecarga en conexión con la transmisión de datos conocida en la técnica se puede emplear en este paso S50.

Una vez que la capacidad de sobrecarga FEC se ha determinado, un siguiente paso S51 selecciona un conjunto de instrucciones de compilación en base a la capacidad de sobrecarga determinada. De esta manera, el archivo contenedor de medios entonces contiene múltiples conjuntos alternativos de instrucciones compiladoras que se pueden usar para un contenido de medios dado pero proporciona diferentes niveles de sobrecarga FEC. En otras palabras, estas instrucciones de compilación alternativas básicamente definen la cantidad de datos de redundancia FEC para añadir a los datos de medios cuando se compilan paquetes de datos de medios. Cuanto mayor sea la sobrecarga FEC aceptable, más datos FEC son añadidos. Teniendo diferentes instrucciones de compilación alternativas, el servidor de medios puede usar aquellas instrucciones que permiten una protección FEC permisible más alta dadas las limitaciones de sobrecarga actuales y por ello aumenta las oportunidades de recepción exitosa y la decodificación de los datos de medios en diferentes clientes comparado con el uso de un único conjunto de instrucciones de compilación.

El método entonces pasa al paso S32 de la Fig. 5, donde los paquetes de datos de medios se compilan desde los datos de contenido de medios y los datos FEC asociados en base a las instrucciones de compilación seleccionadas en el paso S51.

Si el archivo contenedor multimedia también comprende información adicional, tal como información de la división archivo a bloque, información de un algoritmo FEC y/o tabla de propiedades de archivo, el servidor de medios puede usar esta información adicional en la generación y transmisión de paquetes de datos.

Por ejemplo, los datos adicionales y preferentemente la información de tipo MIME, cualquier información de codificación, información de tamaño, etc. útil para el servidor de medios se puede incluir o al menos anunciar en la tabla de propiedades de archivo. En una implementación preferente, esta tabla de propiedades constituye una información única o fuente de búsqueda a la que se puede acceder por el servidor de medios para obtener información requerida o ventajosa en conexión con la extracción de medios, compilación de paquetes de datos y transmisión.

La Fig. 8 es una ilustración esquemática de un archivo contenedor de medios 1 de acuerdo con la invención que se usa para mostrar el uso de instrucciones de compilación alternativas de acuerdo con una realización de la invención. El archivo contenedor 1 comprende un archivo fuente de medios 10 que comprende preferentemente múltiples bloques fuente de medios. En esta realización, cada bloque fuente de medios del archivo fuente 10 ha asociado información de partición 30. El archivo contenedor 1 también comprende, en este ejemplo ilustrativo, tres pistas de sugerencia 50, 52, 54 que contienen instrucciones de compilación para diferentes sobrecargas FEC. Por ejemplo, la primera pista de sugerencia 50 se podría usar cuando se desea un 10% de sobrecarga de redundancia, la segunda pista de sugerencia 52 da una sobrecarga FEC de alrededor del 12% y la tercera pista de sugerencia 54 da un 14% de sobrecarga FEC. En la figura, se ha empleado el algoritmo de construcción de bloque fuente sugerido en el Anexo B del documento [5]. La información de partición 10 se emplea por un códec FEC para calcular los datos FEC 70 en base a los bloques fuente de medios en el archivo fuente 10. Si el archivo fuente 10 comprende múltiples bloques fuente de medios, se calculan múltiples conjuntos de datos FEC o depósitos 70, es decir preferentemente un conjunto de datos FEC por bloque fuente de medios.

Si la primera pista de sugerencia 50 se selecciona, se genera una primera secuencia de paquetes de datos 81, 82, 83, 84 (solamente un paquete de datos por bloque fuente de medios y bloque FEC se ha indicado en la figura). No obstante, si en su lugar se usa la segunda pista de sugerencia 52, se genera una segunda secuencia de paquetes de datos 91, 92, 93, 94. Comparada a la primera secuencia 80, la segunda secuencia 90 comprende bloques FEC mayores, es decir más datos de redundancia FEC, por bloque fuente de medios. No obstante, el bloque fuente respectivo contiene la misma cantidad de datos de medios en las dos secuencias 80, 90.

La Fig. 9 es una vista general esquemática de una red de comunicaciones que ilustra las partes que generan o usan el archivo de contenedor de medios 1 de la presente invención. Un servidor de contenido 100 representa el proveedor o creador de contenidos que recibe o tiene acceso a los datos fuente de medios y construye un archivo contenedor de medios 1. Una copia de este archivo contenedor 1 se envía a un servidor de medios 200 que usa el archivo contenedor 1 en una sesión de medios para compilar paquetes de datos que contienen datos de medios y FEC que se transmiten (multidifusión) a diferentes clientes 300, 310, 320 representados por terminales móviles en la figura.

La Fig. 10 es un diagrama de bloques esquemático de un servidor de contenidos de medios 100 de acuerdo con la presente invención. El servidor de contenidos 100 comprende una unidad de entrada y salida general (I/O) 110 dispuesta y que comprende una funcionalidad (transmisor/receptor, modulador/demodulador, codificador/descodificador) para comunicarse con unidades externas. Esta unidad I/O 110 en particular se dispone para recibir contenido de medios de entrada y para recibir peticiones de archivos contenedores de medios. La

unidad I/O 110 también se emplea por el servidor 100 cuando se transmiten tales archivos contenedores a otros servidores en la red de comunicaciones.

El servidor de contenidos 100 también comprende un creador de archivo contenedor 160 dispuesto para crear archivos contenedores de medios de la invención. El servidor 100 también comprende un proveedor de bloques de medios 130 dispuestos para proporcionar al menos un bloque fuente de medios a ser introducido en el archivo contenedor de medios por un gestor de bloques de medios 161 del creador de archivo 160. El proveedor de bloques de medios 130 proporciona al menos un bloque fuente de entrada desde un almacenamiento de datos interno 120 o desde la unidad I/O 110 que recibe el contenido de medios desde fuentes de medios externas 400, 410.

El(los) bloque(s) fuente de medios entonces se reenvía(n) a un divisor de bloques 140 del servidor de contenidos 100. Este divisor 140 divide el(los) bloque(s) fuente de medios en un número de símbolos fuente, típicamente múltiples de tales símbolos fuente, en base al menos parcialmente a la información de un algoritmo FEC a ser aplicado al bloque fuente para el propósito de cálculo de datos FEC. La operación de partición no tiene necesariamente que dividir físicamente el bloque fuente en símbolos fuente. En claro contraste, la partición podría ser una división virtual asignando diferentes partes del bloque fuente a diferentes símbolos fuente.

El algoritmo FEC a ser aplicado al bloque fuente y el cual podría afectar a la partición de bloque, podría ser un algoritmo FEC estándar predefinido que se emplea consistentemente para bloques fuente de medios. Alternativamente, el divisor de bloque 140 o alguna otra unidad del servidor de contenidos 100 selecciona, a partir de múltiples de tales algoritmos disponibles, un algoritmo FEC particular para usar. En esta selección, diferentes datos de entrada se podrían usar, tales como la sobrecarga FEC máximo esperada máxima. En una realización adicional, el bloque divisor 140 realiza múltiples particiones alternativas de un bloque fuente de medios dado en base a la información de diferentes algoritmos FEC disponibles alternativos. Por ejemplo, si el servidor de contenido s100 tiene conocimiento de los posibles algoritmos FEC disponibles en el servidor de medios, el divisor de bloque 140 puede realizar una partición de bloque separada para cada uno de los algoritmos FEC.

Además de realizar el partición de bloques en base al algoritmo o esquema FEC particular a ser aplicado al bloque fuente de medios, el divisor de bloque 140 podría también operar para realizar la partición adaptada para encajar los símbolos fuente en paquetes de datos que serán empleados mediante un servidor de medios durante una sesión de medios. De esta manera, información del tamaño de paquete, tal como el tamaño de paquete UDP, se podría emplear por el divisor 140.

La información de la partición de bloque fuente particular realizada por el divisor 140 se genera por el generador de información 150. Esta información puede, como ha sido previamente descrito, especificar qué bits de los bloques fuente de medios pertenecen a qué símbolos fuente o definir un tamaño de símbolo que se aplica a todos los símbolos fuente de un bloque fuente posiblemente con la excepción del último símbolo fuente que podría ser de un tamaño más pequeño.

En el caso, el divisor de bloque 140 realiza múltiples particiones alternativas de un bloque fuente de medios, la información generada por el generador 150 incluye datos que conciernen a estas particiones de bloques alternativas.

La información de partición se reenvía desde el generador a un gestor de información 162 del creador de archivo contenedor 160 que inserta la información en el archivo contenedor de medios.

Un gestor de metadatos 163 del creador de archivo 160 proporciona metadatos dentro del archivo contenedor. Estos metadatos proporcionan una asociación entre los bloques fuente de medios organizados mediante el gestor de bloques 161 y la información de partición organizada por el gestor de información de partición 162.

El archivo contenedor de medios resultante entonces se puede almacenar, por lo menos temporalmente, en el almacenamiento de datos 120 o ser transmitido por la unidad I/O 110 a un servidor de medios.

Las unidades 110, 130, 140, 150, 160, 161, 162 y 163 del servidor de contenidos 100 se pueden implementar o proporcionar como programas de ordenador, componentes físicos o una combinación de los mismos. Las unidades 110 a 163 se pueden implementar todas en el servidor de contenidos 100 en un nodo de red único en un sistema de comunicaciones. Alternativamente, una implementación distribuida también es posible y dentro del alcance de la invención. En tal caso, las diferentes unidades 110 a 163 del servidor de contenidos 100 se pueden disponer en diferentes nodos de red pero a pesar de ello realizarán sus operaciones previstas como se describió anteriormente.

La Fig. 11 es un diagrama de bloques esquemático que ilustra una realización del proveedor de bloque de medios 130 de la Fig. 10 en más detalle. En una implementación preferente, el contenido de medios de entrada está en forma de un archivo fuente de medios que se proporciona por un divisor de archivos de medios 132, por ejemplo del almacenamiento de datos o unidad I/O del servidor de contenidos. El archivo fuente de medios se reenvía por el proveedor de archivos 132 a un divisor de archivos multimedia 134. Este divisor 134 divide el archivo fuente en uno o más bloques fuente. El divisor 134 podría basar esta división de archivo en base a información o parámetros diferentes. Por ejemplo, la división de archivo se podría determinar al menos parcialmente en base al algoritmo FEC a ser aplicado para el cálculo de datos de redundancia FEC. En tal caso, la división de archivo 134 preferentemente tiene acceso a información de tal algoritmo FEC. El divisor 134 entonces podría dividir un archivo fuente de medios

en N-1 bloques fuente de medios dimensionados igualmente y un bloque fuente de medios que podría tener un tamaño más pequeño que los otros N-1 bloques.

Un generador de información de división 136 está dispuesto conectado al divisor de archivos 134. El generador 136 genera información de la división de archivo determinada y posiblemente realizada por el divisor de archivo 134. En una primera implementación, la información generada podría especificar qué bits de un archivo fuente de medios que pertenece a diferentes bloques de fuente de medios. En una segunda implementación, la información especifica el tamaño (de bit o de símbolo) de bloques fuente de medios, posiblemente con excepción de un último bloque fuente que podría tener un tamaño más pequeño. En tal caso, conociendo la localización inicial del archivo fuente de medios, los diferentes bloques fuente de medios se pueden identificar usando esta información de división (de tamaño).

Un gestor de información de división 138 se implementa en el proveedor de bloques 130 para organizar la información de división del generador de información 136 en el archivo contenedor de medios.

Las unidades 132 a 138 del proveedor de bloque de medios 130 se pueden implementar o proporcionar como programa de ordenador, componentes físicos o una combinación de los mismos. Las unidades 132 a 138 se pueden implementar todas en el proveedor de bloque de medios 130. Alternativamente, una implementación distribuida también es posible y está dentro del alcance de la invención. En tal caso, diferentes unidades 132 a 138 del proveedor de bloques de medios 130 se pueden disponer en otra parte en el servidor de contenidos.

La Fig. 12 es un diagrama de bloques esquemático que ilustra una realización del creador de archivo contenedor 160 de la Fig. 10 en más detalle. Este creador de archivo 160 comprende, además del gestor de bloque de medios 161 descrito previamente, gestor de información de partición 162 y gestor de metadatos 163, un gestor de información FEC 164. Este gestor de información FEC 164 genera y organiza, en el archivo contenedor, información del(de los) algoritmo(s) FEC a ser aplicados al(los) bloque(s) fuente de medios del archivo contenedor. La partición de bloque realizada por el divisor de bloque del servidor de contenido también usó tal información de algoritmo FEC cuando se realiza la partición de bloque. De esta manera, la información de(de los) algoritmo(s) FEC, en base a la cual se ha realizado la partición de bloque, se genera preferentemente por el gestor 164 e incluye en el archivo contenedor.

Un gestor de tablas 165 se puede incluir en el creador de archivos 160 para generar e incluir una tabla de propiedades dentro del archivo contenedor. Esta tabla de propiedades entonces podría incluir información del(de los) archivo(s) fuente de medios en el archivo contenedor, tal como el tipo de archivo, tamaño de archivo, localización de almacenamiento de archivo, encriptado de archivo, identificadores/nombre de archivo, etc. También la información de partición, la información de división de archivo y los metadatos de la invención se podrían incluir en la tabla de propiedades generada por el gestor de tablas 165.

Un gestor de instrucciones FEC 166 se incluye en el creador de archivo 160 para generar instrucciones FEC a ser seguidas por un servidor de medios cuando se generan datos FEC en base a los bloques fuente de medios del archivo contenedor. Estas instrucciones entonces especificarán, junto con la información de partición asociada con el bloque fuente de medios particular, qué símbolos fuente que podrían ser introducidos a un códec FEC para cálculo de símbolos FEC. Se podrían proporcionar diferentes instrucciones FEC alternativas por el gestor 166 para un bloque de fuente dado, donde las instrucciones FEC se adaptan para uso en conexión con diferentes códec FEC y/o sobrecargas FEC.

Un gestor de instrucción de compilación 167 del creador de archivo 160 genera e inserta instrucciones de compilación en el archivo contenedor. Estas instrucciones incluyen información usada por servidores de medios para compilar datos multimedia desde los bloques fuente de medios y datos FEC calculados usando los bloques fuente y la información de partición. El gestor 167 podría generar una instrucción única o conjunto de instrucciones por contenido de medios en el archivo. Alternativamente, tales instrucciones diferentes adaptadas para diferentes sobrecargas FEC, diferentes tipos de datos FEC y/o diferente número de canal de comunicaciones basado en radio empleado en la sesión de medios se podrían proporcionar por el gestor 167 y organizar en el archivo contenedor.

Las unidades 161 a 167 del creador de archivo contenedor 160 se pueden implementar o proporcionar como un programa de ordenador, componentes físicos o una combinación de los mismos. Las unidades 161 a 167 se pueden implementar todas en el creador de archivo contenedor 160. Alternativamente, una implementación distribuida también es posible y está dentro del alcance de la invención. En tal caso, las diferentes unidades 161 a 167 del creador de archivo contenedor 160 se pueden disponer en otra parte en el servidor de contenidos.

La Fig. 13 es un diagrama de bloques esquemático de un servidor de sesión de medios 200 de acuerdo con la presente invención. Este servidor de medios 200 comprende una unidad de I/O 210 para conducir la comunicación con unidades externas. Esta unidad I/O 210 se dispone en particular para solicitar y recibir contenedores de archivo de medios desde un servidor de contenidos. La unidad de I/O 210 también recibe petición de contenido de medios que se originan desde diferentes clientes de usuario o al menos información a la que se debería transmitir el contenido de medios de los clientes. Los paquetes de datos compilados por el servidor de medios 200 también se transmiten por la unidad de I/O 210 a estos clientes.

El servidor 200 comprende un proveedor de archivos de medios 220 que proporciona un archivo de contenido de medios para usar en la sesión actual. El proveedor de archivos 220 puede generar una petición para un archivo contenedor particular que se transmite a un creador de contenidos por la unidad I/O 210. Alternativamente, el proveedor 220 trae un archivo contenedor recibido previamente desde un almacenamiento de datos 250 proporcionado en el servidor de medios 200.

Una calculadora o códec FEC 240 del servidor de medios 200 calcula datos de redundancia FEC para al menos un bloque fuente de medios almacenado en el archivo contenedor. Esta calculadora FEC 240 podría usar cualquiera de los algoritmos FEC previamente mencionados en este procedimiento de cálculo. En una implementación preferente, el archivo contenedor da instrucciones a la calculadora 240 de qué algoritmo FEC emplear si hay una elección. La calculadora FEC 240 primero identifica la información de partición relevante para un bloque de fuente previamente mencionados de entrada. En este procedimiento de identificación, los metadatos asociados con el bloque fuente y proporcionados en el archivo contenedor se usan por la calculadora 240. La calculadora 240 entonces usa la información de partición para identificar aquellas partes del bloque fuente de medios, a las cuales el algoritmo FEC (seleccionado en base a la información FEC en el archivo contenedor) se debería aplicar para generar datos FEC.

En una realización preferente, el archivo contenedor de medios también comprende instrucciones de cálculo FEC que son usadas por la calculadora 240 además de la información de partición cuando se identifican símbolos fuente de entrada correctos y que realizan el cálculo de datos FEC real. Información de entrada adicional podría ser la previamente descrita información de algoritmo FEC (permite la selección del algoritmo FEC correcto a usar) y la información de división de archivo (permite identificación de bloque fuente de medios correcto).

Un compilador de paquete de datos 230 preferentemente usa instrucciones de compilación incluidas en el archivo contenedor desde el proveedor 220 para extraer datos de medios desde el archivo y datos FEC desde la calculadora FEC para generar paquetes de datos que contienen estos datos extraídos. Los paquetes de datos así generados entonces se transmiten por (secuenciados o descargados de) la unidad de I/O 210.

Diferentes instrucciones de compilación se podrían incluir en el archivo para un contenido de medios dado. Por ejemplo, las instrucciones podrían ser dependientes del canal o dependientes de la capacidad. En el primer caso, el número de canales radio disponibles y el número de secuencias de medios paralelas que se deberían transmitir determina las instrucciones de compilación reales a usar por el compilador 230. En este último caso, un estimador de capacidad FEC 260 se incluye preferentemente en el servidor 200 para estimar una cantidad máxima de sobrecarga FEC que se podría emplear durante la sesión. La estimación de cabecera realizada por el estimador 260 preferentemente se actualiza dinámicamente durante la sesión, ya que la capacidad de sobrecarga se podría cambiar a través de la sesión. Un selector de conjunto 270 usa las estimaciones de capacidad del estimador 260 para seleccionar qué instrucción o conjunto de instrucciones de compilación particular de aquellas disponibles en el archivo usar. El paquete compilador 230 entonces usa esta instrucción (conjunto) para compilar los datos de medios y los datos FEC en paquetes de datos.

Las unidades 210, 220, 230, 240, 260, y 270 del servidor de medios 200 se pueden implementar o proporcionar como programa de ordenador, componentes físicos o una combinación de los mismos. Las unidades 210 a 270 se pueden implementar todas en el servidor de medios 200 en un nodo de red único en un sistema de comunicaciones. Alternativamente, una implementación distribuida también es posible y está dentro del alcance de la invención. En tal caso, las unidades diferentes 210 a 270 del servidor de medios 200 se pueden disponer en diferentes nodos de red pero a pesar de ello realizarán sus operaciones previstas como se describió anteriormente.

Se entenderá por una persona experta en la técnica que se puede realizar diversas modificaciones y cambios a la presente invención sin salirse del alcance de la misma, el cual se define por las reivindicaciones adjuntas.

REFERENCIAS

[1] ISO/IEC 14496-12:2005: "Formato de archivo de medios de base ISO"

[2] ISO/IEC 15444-12:2005: "Formato de archivo de medios de base ISO"

[3] Solicitud internacional WO 2005/039131

[4] RFC 3695; Esquemas de Corrección de Errores sin canal de retorno (FEC) Compactos, febrero de 2004

[5] TS 26.346 V7.0.0 del 3GPP, Proyecto de Cooperación de 3ª Generación; Servicios y Sistema de Grupo de Especificación Técnica y Aspectos del Sistema; Servicio de Radiodifusión/Multidifusión Multimedia (MBMS); Protocolos y códec, junio de 2006.

[6] RFC 2616; Protocolo de Transferencia Hipertexto - HTTP/1.1, junio de 1999

[7] RFC 1864; El Campo de Cabecera de Contenido-MDS, octubre de 1995

[8] RFC 3926; FLUTE – Entrega de Archivos sobre Transporte Unidireccional, octubre de 2004

[9] RFC 3450; Instanciación del Protocolo de Codificación de Capas Asíncrono (ALC), diciembre de 2002.

[10] RFC 3451; Bloque de Construcción de Transporte de Codificación de Capas (LCT), diciembre de 2002

REIVINDICACIONES

1. Un método para generar un archivo contenedor de medios (1), dicho método comprende los pasos de:
 - proporcionar al menos un bloque fuente de medios (20; 22; 24) que comprende datos de medios;
 - organizar dicho al menos un bloque fuente de medios (20; 22; 24) en dicho archivo contenedor de medios (1);
- 5 organizar la información de partición calculada previamente (30; 32; 34) en dicho archivo contenedor de medios (1), en el que la información de partición es descriptiva de la partición calculada previamente para cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) en un número de símbolos fuente en base a la información de un algoritmo de Corrección de Errores sin canal de retorno, FEC, a ser aplicado al bloque fuente respectivo (20; 22; 24) y
- 10 organizar, en dicho archivo contenedor de medios (1), los metadatos (40) que proporcionan una asociación entre dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34), **caracterizado por:**
 - organizar instrucciones de cálculo FEC calculadas previamente (50; 52; 54) en dicho archivo contenedor de medios (1), en el que las instrucciones de cálculo FEC calculadas previamente definen el cálculo de datos de redundancia FEC (70) en base a dichos metadatos (40), dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34) para el bloque fuente de medios respectivo (20; 22; 24);
 - generar a una primera pista de sugerencia un primer conjunto de instrucciones de compilación (50) que definen la compilación de datos de medios de cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) y dichos datos de redundancia FEC (70) para formar una primera secuencia de medios (80) de paquetes de datos (81, 82, 83, 84) que tienen un primer nivel de sobrecarga de redundancia FEC;
 - generar a una segunda pista de sugerencia la cual es diferente de la primera pista de sugerencia un segundo conjunto de instrucciones de compilación (52), el cual es diferente del primer conjunto de instrucciones de compilación (50) que define la compilación de datos de medios de cada uno de dicho al menos un bloque fuente de medios y datos de redundancia FEC (70) para formar una segunda secuencia de medios (90) de paquetes de datos (91, 92, 93, 94) que tiene un segundo nivel de sobrecarga de redundancia FEC, la cual es diferente del primer nivel de sobrecarga de redundancia FEC; y
 - organizar dicha primera pista de sugerencia y segunda pista de sugerencia en dicho archivo contenedor de medios (1).
2. El método de acuerdo con la reivindicación 1, en el que dicho paso de proporcionar dicho al menos un bloque fuente de medios (20; 22; 24) comprende los pasos de:
 - proporcionar un archivo fuente de medios (10);
 - dividir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24);
 - generar información de división descriptiva de dicha división; y
 - organizar dicha información de división en dicho archivo contenedor de medios (1).
3. El método de acuerdo con la reivindicación 2, en el que dicho paso de división comprende dividir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24) en base a la información de dicho algoritmo FEC.
4. El método de acuerdo con cualquiera de las reivindicaciones 1 a 3, que además comprende proporcionar, en dicho archivo contenedor de medios (1), información de dicho algoritmo FEC.
5. El método de acuerdo con cualquiera de las reivindicaciones 1 a 4, que además comprende proporcionar, en dicho archivo contenedor de medios (1), una tabla de propiedades (60) que comprende información de localización de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).
6. El método de acuerdo con cualquiera de las reivindicaciones 1 a 5, en el que dicho primer conjunto de instrucciones de compilación (50) define la compilación de dichos datos de medios desde dicho al menos un bloque fuente de medios (20; 22; 24) y dichos datos de redundancia FEC (70) para formar una primera secuencia de medios (80) de paquetes de datos FLUTE (81, 82, 83, 84) que tienen dicho primer nivel de sobrecarga de redundancia FEC y dicho segundo conjunto de instrucciones de compilación (52) define la compilación de dichos datos de medios desde dicho al menos un bloque fuente de medios (20; 22; 24) y dichos datos de redundancia FEC (70) para formar una segunda secuencia de medios (90) de paquetes de datos FLUTE (91, 92, 93, 94) que tienen dicho segundo nivel de sobrecarga de redundancia FEC.

7. Un servidor de contenidos de medios (100) que comprende:

un proveedor de bloques de medios (130) dispuesto para proporcionar al menos un bloque fuente de medios (20; 22; 24) que comprende datos de medios;

5 un gestor de bloque de medios (161) dispuesto para organizar dicho al menos un bloque fuente de medios (20; 22; 24) en dicho archivo contenedor de medios (1);

10 un gestor de información de partición (612) dispuesto para organizar información de partición calculada previamente (30; 32; 34) en dicho archivo contenedor de medios (1), en el que la información de partición es descriptiva de la partición calculada previamente para cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) en un número de símbolos fuente en base a la información de un algoritmo de corrección de errores sin canal de retorno, FEC, a ser aplicado al bloque fuente respectivo (20; 22; 24); y

un gestor de metadatos (163) dispuesto para organizar, en dicho archivo contenedor de medios (1), los metadatos (40) que proporcionan una asociación entre dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34), **caracterizado por**:

15 un gestor de instrucciones FEC (166) dispuesto para organizar las instrucciones de cálculo FEC calculadas previamente (50; 52; 54) en dicho archivo contenedor de medios (1), en el que las instrucciones de cálculo FEC calculadas previamente definen el cálculo de los datos de redundancia FEC (70) en base a dichos metadatos (40), dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34) para el bloque fuente de medios respectivo (20; 22; 24); y

20 un gestor de instrucciones de compilación (167) dispuesto para i) generar a una primera pista de sugerencia un primer conjunto de instrucciones de compilación (50) que definen la compilación de datos de medios desde cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) y ii) generar a una segunda pista de sugerencia la cuál es diferente de la primera pista de sugerencia un segundo conjunto de instrucciones de compilación (52), el cual es diferente del primer conjunto de instrucciones de compilación (50) que define la compilación de datos de medios para cada uno de dicho al menos un bloque fuente de medios y dichos datos de redundancia FEC (70) para formar una segunda secuencia de medios (90) de paquetes de datos (91; 92; 93; 94) que tiene un segundo nivel de sobrecarga de redundancia FEC, el cual es diferente del primer nivel de sobrecarga de redundancia FEC y iii) organizar dicha primera pista de sugerencias y segunda pista de sugerencia en dicho archivo contenedor de medios (1).

30 8. El servidor de contenidos de medios de acuerdo con la reivindicación 7, en el que dicho proveedor de bloque de medios (130) comprende:

un proveedor de archivo de medios (132) dispuesto para proporcionar un archivo fuente de medios (10);

un divisor de archivo de medios (134) dispuesto para dividir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24);

35 un generador de información de división (136) dispuesto para generar información de división descriptiva de dicha división;

un gestor de información de división (138) dispuesto para organizar dicha información de división en dicho archivo contenedor de medios (1).

40 9. El servidor de contenidos de medios de acuerdo con la reivindicación 8, en el que dicho divisor de archivo de medios (134) está dispuesto para dividir, dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24) en base a la información de dicho algoritmo FEC.

10. El servidor de contenidos de medios de acuerdo con cualquiera de las reivindicaciones 7 a 9, que además comprende un gestor de información FEC (164) dispuesto para proporcionar, en dicho archivo contenedor de medios (1), información de dicho algoritmo FEC.

45 11. El servidor de contenidos de medios de acuerdo con cualquiera de las reivindicaciones 7 a 10, que además comprende un gestor de tabla (165) dispuesto para proporcionar, en dicho archivo contenedor de medios (1), una tabla de propiedades (60) que comprende información de localización de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).

12. Un archivo contenedor de medios (1) que comprende:

al menos un bloque fuente de medios (20; 22; 24);

50 información de partición calculada previamente (30; 32; 34), en el que la información de partición es descriptiva de la partición calculada previamente para cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) en un número de símbolos fuente en base a la información de un algoritmo de corrección de errores sin canal de retorno,

FEC, a ser aplicado al bloque fuente respectivo (20; 22; 24); y

metadatos (40) que proporcionan una asociación entre dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34), **caracterizado por:**

5 instrucciones de cálculo FEC calculadas previamente (50; 52; 54), en el que las instrucciones de cálculo FEC calculadas previamente definen el cálculo de los datos de redundancia FEC (70) en base a dichos metadatos (40), dicho al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición (30; 32; 34) para el bloque fuente de medios respectivo (20; 22; 24); e

10 instrucciones de compilación (50; 52; 54) que comprenden un primer conjunto de instrucciones de compilación (50) en una primera pista de sugerencias que definen la compilación de datos de medios de cada uno de dicho al menos un bloque fuente de medios (20; 22; 24) y dichos datos de redundancia FEC (70) para formar una primera secuencia de medios (80) de paquetes de datos (81, 82, 83, 84) que tienen un primer nivel de sobrecarga de redundancia FEC; y un segundo conjunto de instrucciones de compilación (52) en una segunda pista de sugerencias la cual es diferente de la primera pista de sugerencias que define la compilación de datos de medios de cada uno de dicho al menos un bloque fuente de medios y dichos datos de redundancia FEC (70) para formar
15 una segunda secuencia de medios (90) de paquetes de datos (91, 92, 93, 94) que tienen un segundo nivel de sobrecarga de redundancia FEC, que es diferente del primer nivel de sobrecarga de redundancia FEC.

13. El archivo contenedor de medios de acuerdo con la reivindicación 12, que además comprende información de división descriptiva de una división de un archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24).

20 **14.** El archivo contenedor de medios de acuerdo con la reivindicación 12 o 13, que además comprende información de dicho algoritmo FEC.

15. El archivo contenedor de medios de acuerdo con cualquiera de las reivindicaciones 12 o 14, que además comprende una tabla de propiedades (60) que comprende información de localización de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).

25

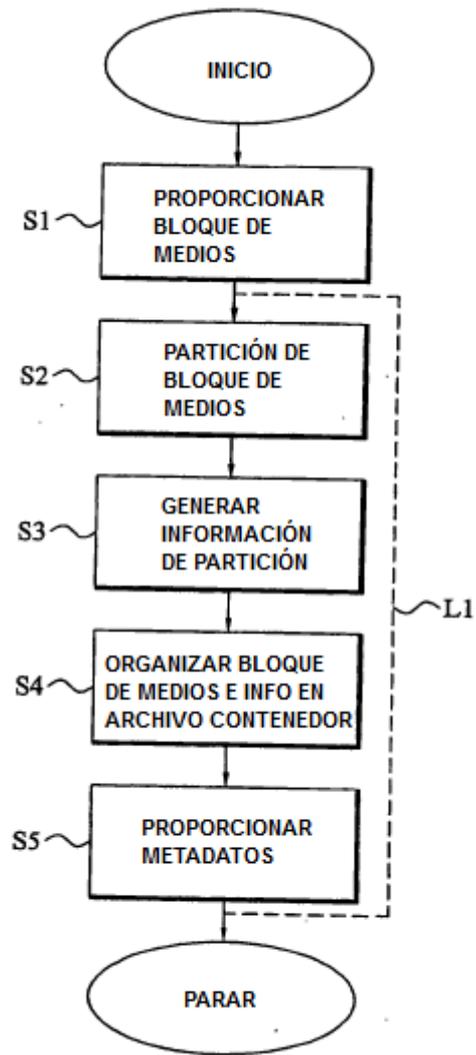


Fig. 1

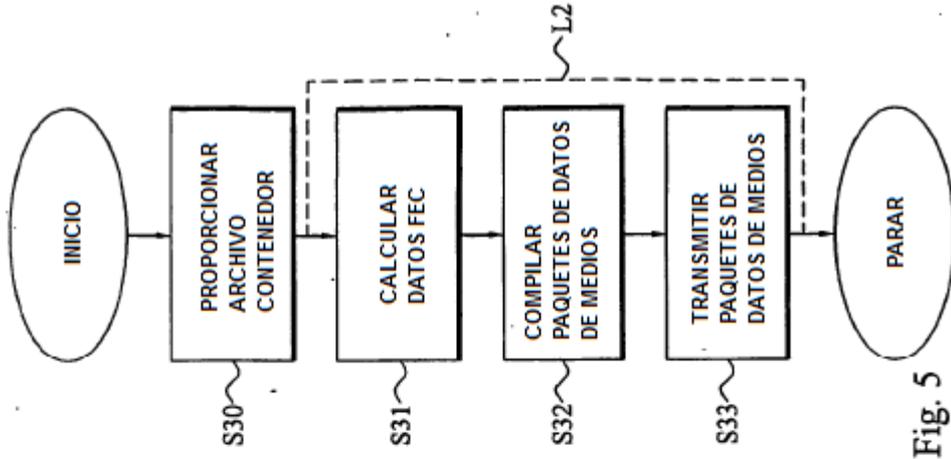


Fig. 5

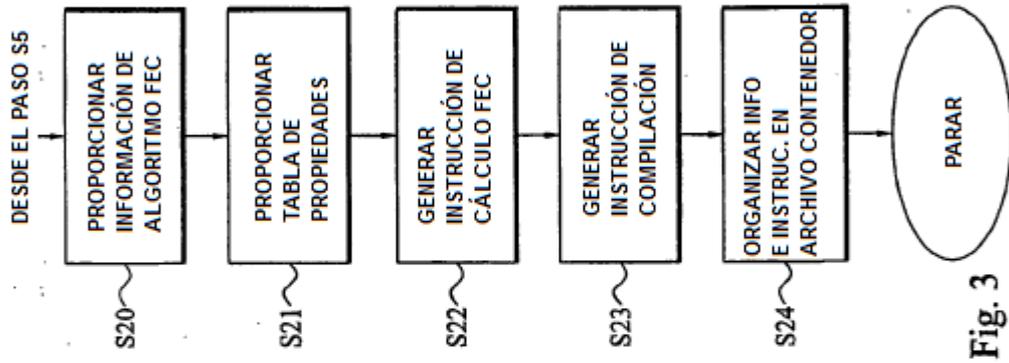


Fig. 3

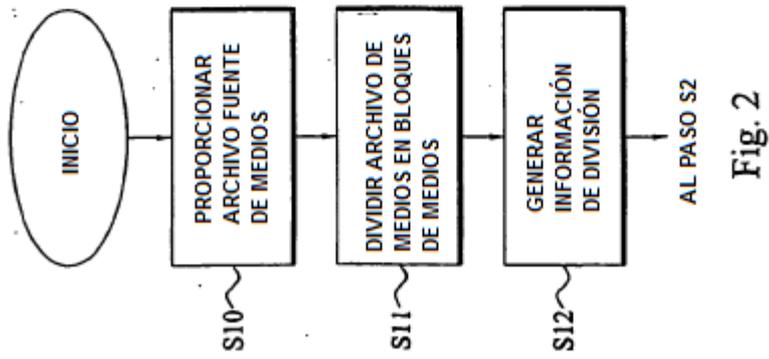


Fig. 2

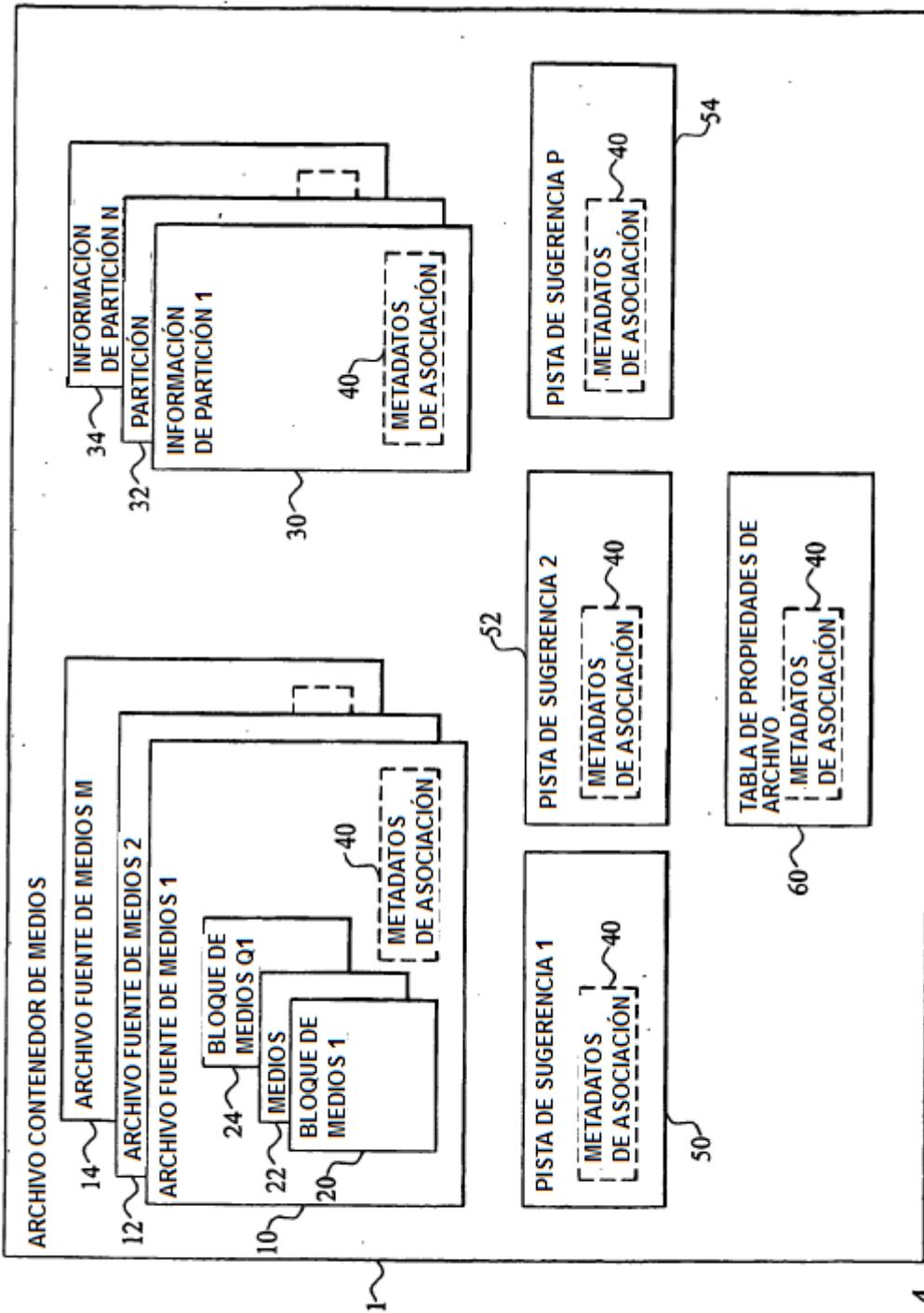


Fig. 4

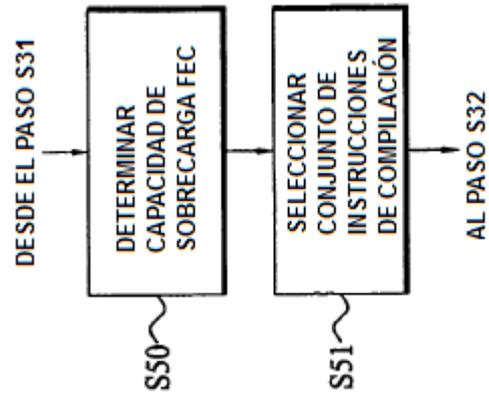


Fig. 7

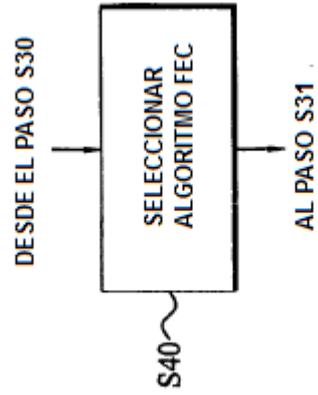


Fig. 6

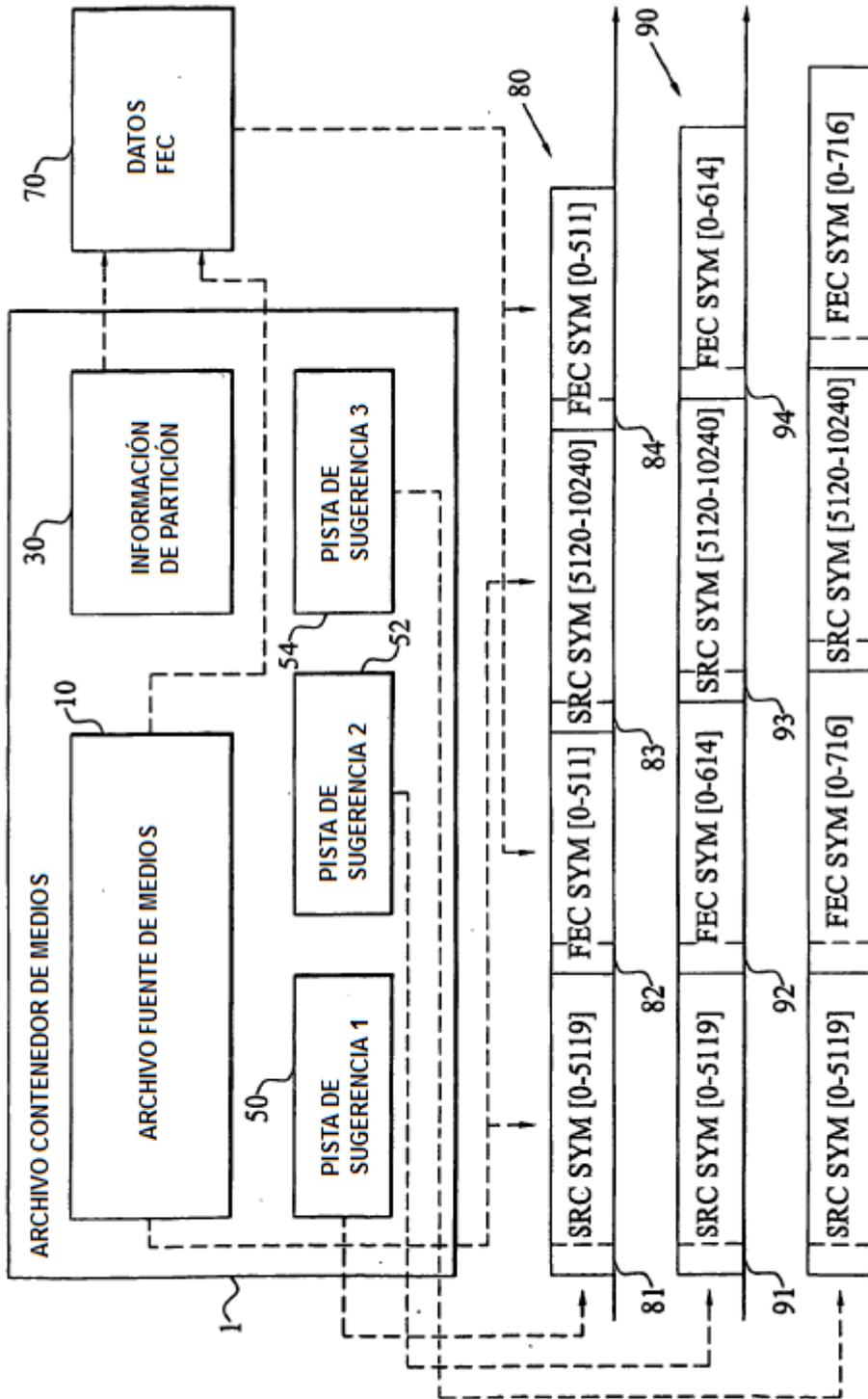


Fig. 8

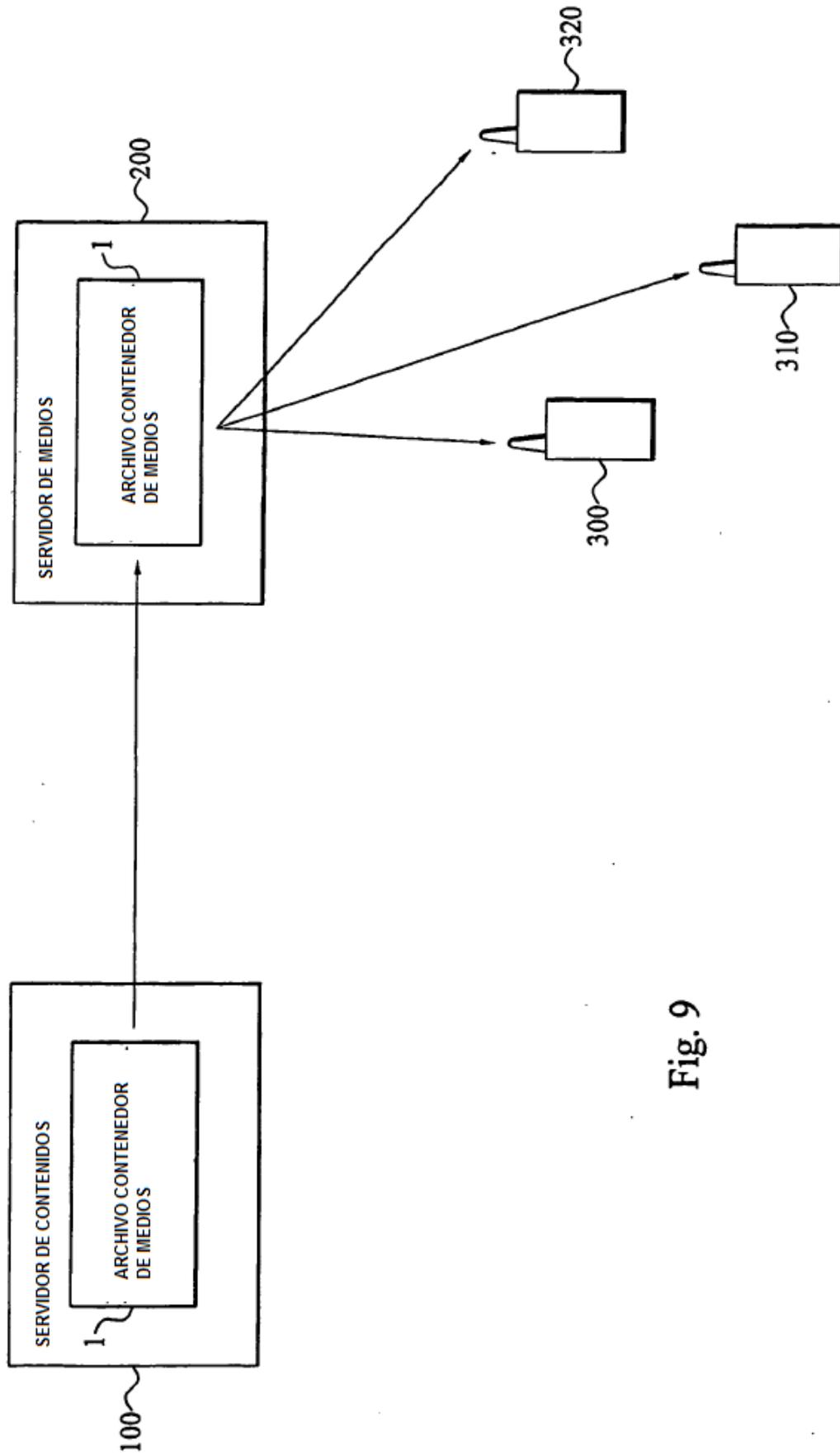


Fig. 9

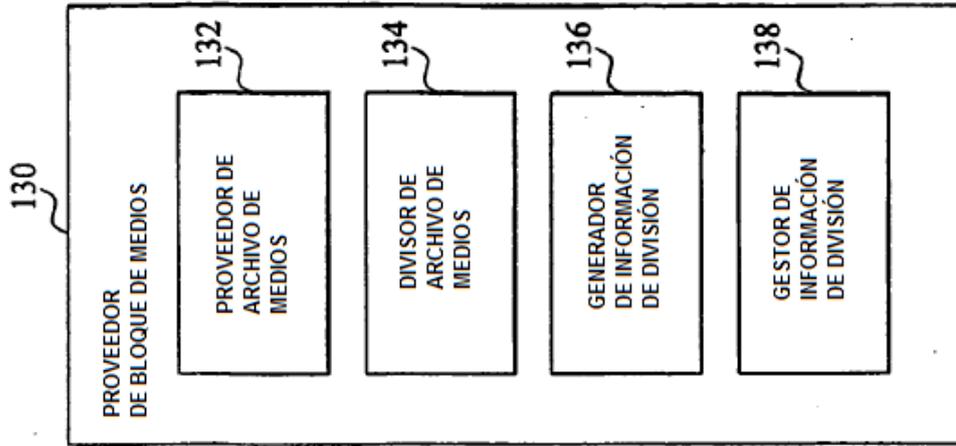


Fig. 11

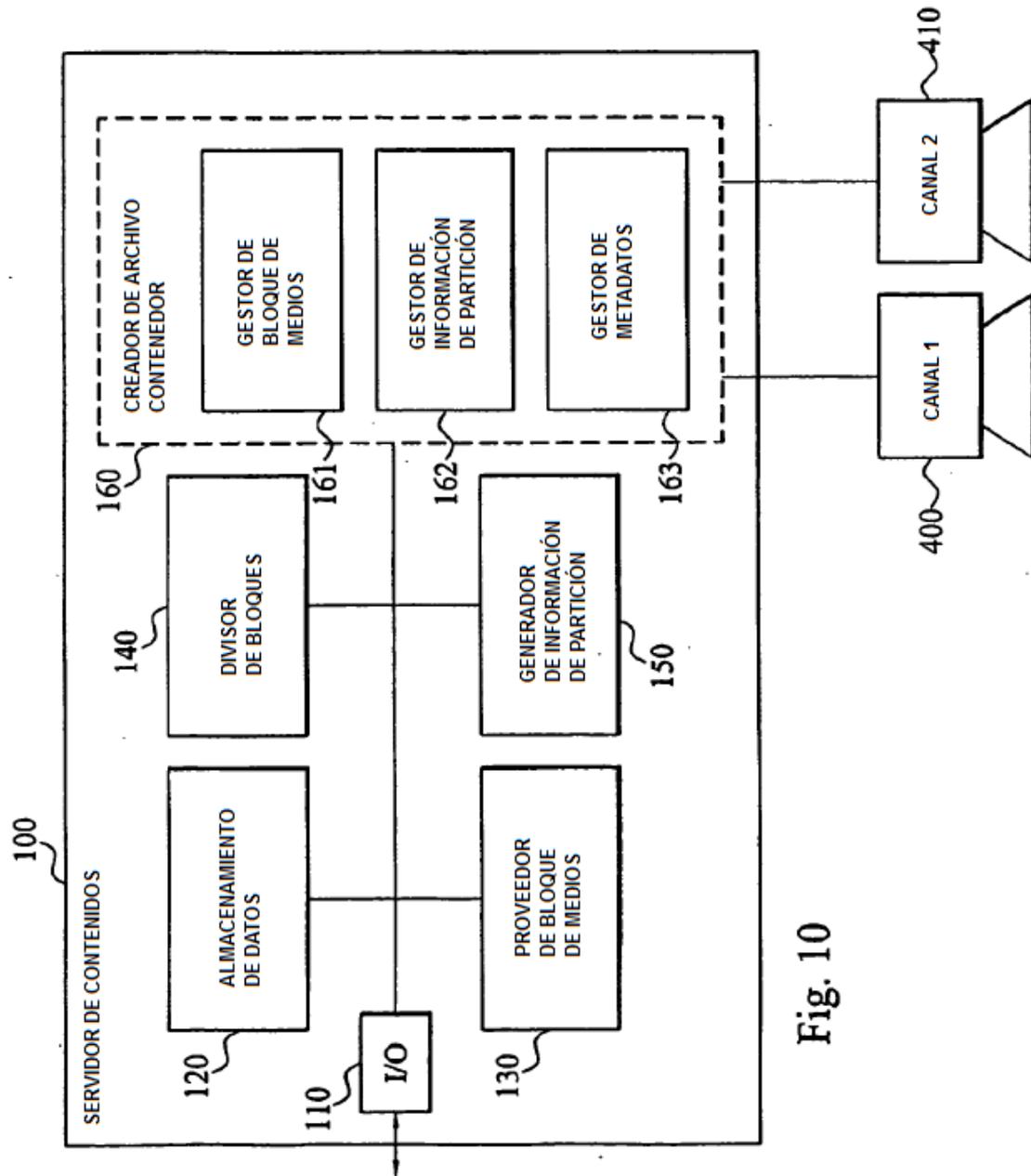


Fig. 10

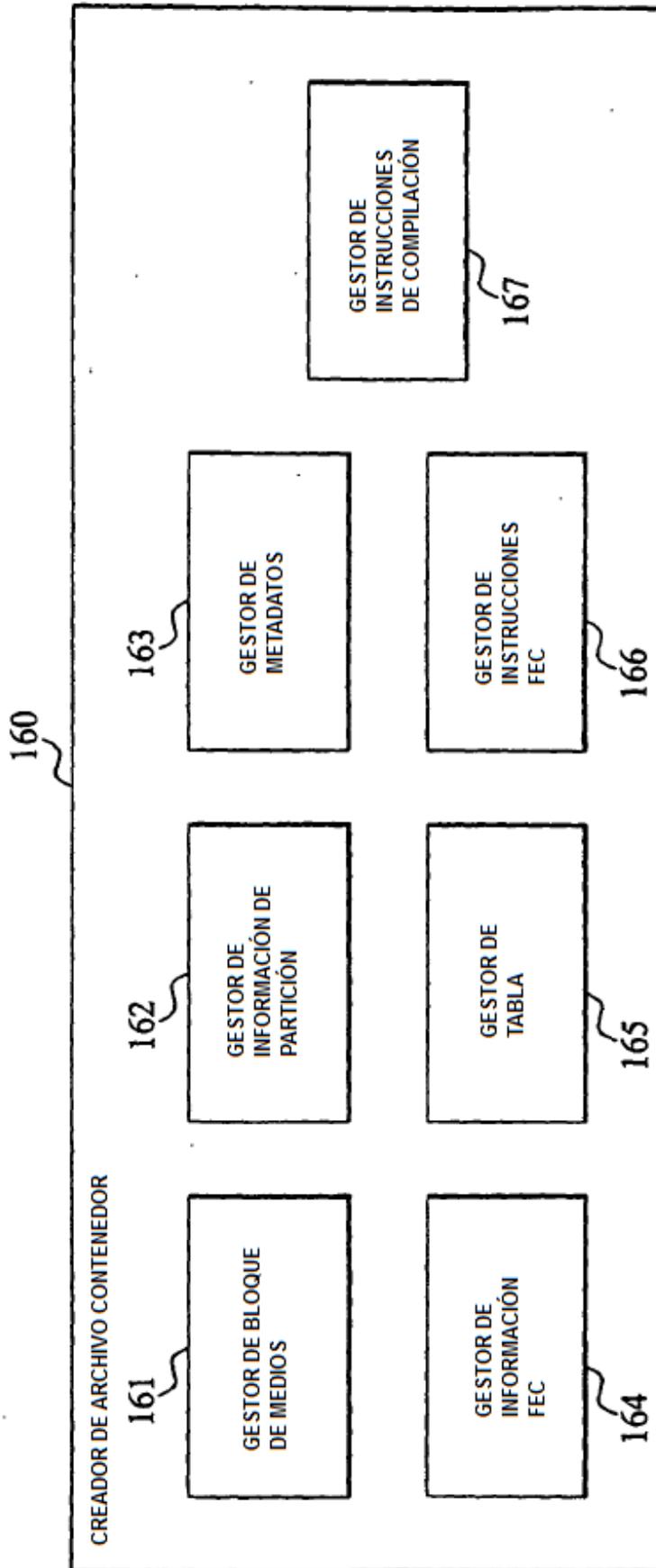


Fig. 12

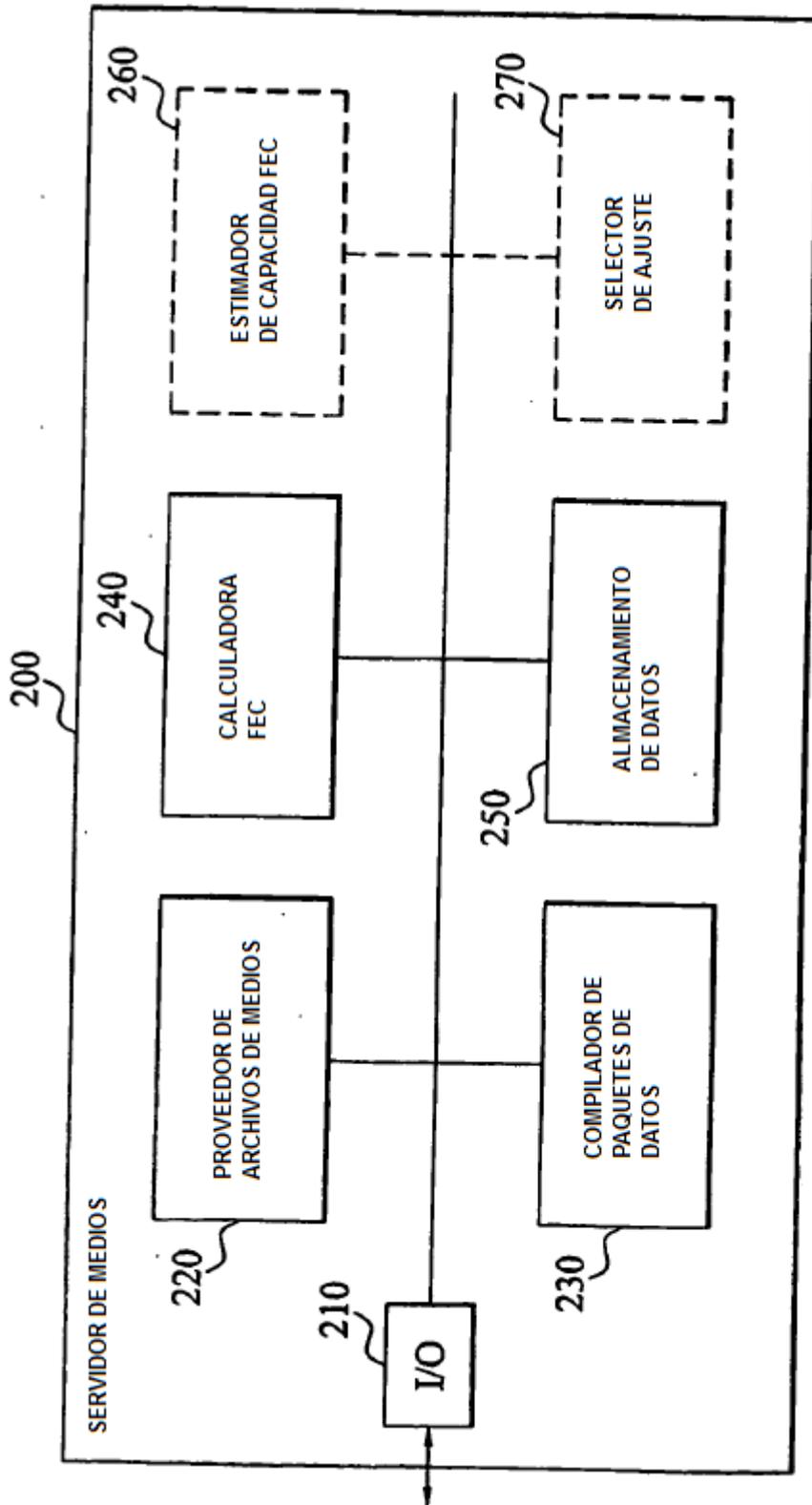


Fig. 13