

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 383 230**

51 Int. Cl.:

H04L 1/00 (2006.01)

H04L 12/18 (2006.01)

H04N 7/24 (2011.01)

H04N 7/66 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07701091 .6**

96 Fecha de presentación: **04.01.2007**

97 Número de publicación de la solicitud: **1969857**

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

54 Título: **Gestión de archivos contenedores de medios**

30 Prioridad:
05.01.2006 US 743095 P

45 Fecha de publicación de la mención BOPI:
19.06.2012

45 Fecha de la publicación del folleto de la patente:
19.06.2012

73 Titular/es:
**TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)
PATENT UNIT
164 83 STOCKHOLM, SE**

72 Inventor/es:
**LOHMAR, Thorsten;
WESTERLUND, Magnus y
FRÖJDH, Per**

74 Agente/Representante:
de Elzaburu Márquez, Alberto

ES 2 383 230 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 archivos contenedores de medios

Campo técnico

5 La presente invención se refiere de manera general a la 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 al cliente sobre diferentes redes ha aumentado tremendamente los últimos años. Hoy en día, Internet se emplea por numerosos usuarios para acceso y descarga de medios, por ejemplo en forma de secuencias o archivos de audio y vídeo, desde servidores de medios. Este suministro de medios también ha emergido en las redes de comunicaciones móviles basadas en radio. Actualmente hay un interés muy grande en usar redes móviles para contenido multimedia o de TV. Esto se conoce a menudo como TV Móvil en la técnica. Este suministro de medios en las redes móviles está hoy en día disponible 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 Multimedia de Radiodifusión/Multidifusión (MBMS) del 3GPP y la Radiodifusión de Vídeo Digital-De Mano (DVB-H) del Instituto Europeo de Estándares de Telecomunicaciones (ETSI).

15 En línea con esta demanda creciente para el 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 a clientes solicitantes. Hay una tendencia general hacia servidores de difusión en forma continua/de 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 a estas funciones se proporciona entonces 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.

20 Junto con el desarrollo de servidores de difusión en forma continua/de descarga flexibles, está teniendo lugar el desarrollo en el área de cómo se puede introducir la corrección de errores en las secuencias de medios. Las transmisiones multidifusión/radiodifusión son unidireccionales y dirigen un alto número de clientes de recepción simultáneamente. Los esquemas tradicionales de fiabilidad de unidifusión, tales como Petición de Repetición Automática (ARQ), no son escalables para servir el alto número de receptores de transmisiones multidifusión/radiodifusión.

25 De esta manera, hay una necesidad para la introducción de un esquema fiable en conexión con la transmisión de medios multidifusión/radiodifusión. La introducción de tal esquema de fiabilidad también debería estar en línea con la tendencia de soluciones flexibles de servidor de difusión en forma continua y de descarga.

35 La WO 2004/036760 especifica un archivo contenedor de medios para llevar a cabo la protección adaptativa y eficiente, que permita por ello a las aplicaciones conmutar dinámicamente entre diferentes estrategias de protección. La protección se logra proporcionando una pista de protección como una secuencia separada de la secuencia de datos de medios.

40 La US 2003/0204555 se refiere a usar pistas de índice en un archivo contenedor de medios. Una pista de índice indica cómo transmitir una secuencia relacionada con el tiempo de datos de medios de acuerdo con un protocolo de transmisión. La pista de índice incluye una secuencia de datos relacionada con el tiempo la cual está asociada con la secuencia de datos de medios relacionada con el tiempo. La pista de índice se puede utilizar por un sistema de procesamiento digital para transmitir la secuencia relacionada con el tiempo de datos de medios.

45 La US 2005/0182842 se refiere a no reparar sesiones para entrega de ficheros en un sistema capaz de una transmisión uno a muchos. Los bloques de datos se transfieren desde un remitente a al menos un receptor. Se identifica un bloque de datos que espera ser recibido pero no se recibe en absoluto o se recibe incorrectamente en el receptor. Se toman acciones para retransmitir el bloque de datos. La identificación del bloque de datos se conduce en base al número de bloques, un identificador de codificación y otra información de identificación.

50 VM de la Enmienda 2 para 14496-12:2005 Pistas de índice Flute en formato de archivos de medios basados en ISO, ISO/IEC JTC1/SC29/WG11 sugiere adiciones de cajas relacionadas con la pista de índice FLUTE en formato de archivos de medios basados en ISO.

Resumen

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

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

sesiones multimedia.

Es otro objeto de la invención proporcionar un archivo contenedor de medios que también se puede usar en procedimientos de reparación post sesión.

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

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

La generación de un archivo contenedor de medios implica proporcionar al menos un archivo fuente de medios que comprende datos de medios o multimedia a ser transmitidos a clientes solicitantes y reproducidos en los clientes. Este archivo contenedor se considera como que consta de uno o más bloques fuente de medios, dependiendo del tamaño del archivo fuente. Al menos uno de tal bloque fuente de medios se procesa de acuerdo con la presente invención para el propósito de calcular datos y símbolos de redundancia FEC basados en el bloque fuente. De esta manera, los datos de medios del bloque fuente se introducen a un algoritmo FEC para el cálculo de al menos un símbolo FEC. Este cálculo del símbolo FEC se realiza preferentemente para cada bloque fuente de medios de un archivo fuente. El al menos un bloque fuente de medios entonces se organiza en el archivo contenedor de medios. Por consiguiente, los datos FEC calculados también se organizan en el archivo contenedor de medios en una o más reservas FEC. Cada una de tales reservas FEC comprende los datos FEC calculados por un bloque fuente de medios particular. Los metadatos se proporcionan e incluyen en el archivo contenedor para proporcionar una asociación entre un bloque fuente de medios y su reserva FEC.

El archivo contenedor resultante se puede emplear por un servidor de medios durante una sesión de medios para compilar, usando metadatos en el archivo de contenedor, los paquetes de datos que comprenden datos de medios y datos FEC. El pre cálculo de datos FEC y la organización de datos de medios y FEC en el archivo contenedor de la invención permite al servidor de medios, de una manera simple barata computacionalmente, insertar datos de medios y datos FEC en paquetes de datos transmitidos a clientes solicitantes sin procesamiento de datos extensivo y cálculo FEC de demanda computacionalmente.

25 En una realización preferente, el archivo contenedor también comprende instrucciones de compilación que se usan y siguen por el servidor de medios cuando se compilan paquetes de datos que contienen datos de medios y FEC del archivo de contenedor. En tal caso, el archivo contenedor comprende todo el contenido de medios, datos e instrucciones de protección requeridos para ser capaces de reenviar con éxito los datos de medios de una manera fiable a los clientes.

30 Como ejemplo adicional útil para comprender la invención, el archivo contenedor también se puede usar en procedimientos de reparación post sesión, cuando, a pesar de la inclusión de protección de redundancia FEC en los paquetes de datos transmitidos, un cliente no era capaz de recibir con éxito todos los datos de medios durante una sesión de medios. En tal caso, una copia del mismo archivo contenedor que fue usado durante la sesión de medios previa se usa por un servidor de reparación. El servidor es capaz de recuperar datos de redundancia FEC desde una de las reservas FEC en el archivo contenedor en base a la petición desde el cliente. Estos datos FEC entonces se devuelven al cliente, permitiéndole reproducir con éxito todos los datos de medios.

Breve descripción de los dibujos

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

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

La Fig. 2 es un diagrama de flujo que ilustra pasos adicionales del método de generación de archivos de la Fig. 1;

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

La Fig. 5 es un diagrama de flujo que ilustra un método de gestión de sesiones de medios

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

50 La Fig. 7 es un diagrama de flujo que ilustra una realización de los pasos de compilación y transmisión de la Fig. 5 en más detalle;

La Fig. 8 es un diagrama de flujo que ilustra otro ejemplo de los pasos de compilación y transmisión de la Fig. 5

en más detalle;

La Fig. 9 es un diagrama de flujo que ilustra pasos adicionales del método de gestión de sesiones de la Fig. 5;

La Fig. 10 es un diagrama de flujo que ilustra un método de reparación post sesión

5 La Fig. 11 es una descripción esquemática de un archivo contenedor de medios de acuerdo aún con otro aspecto de la presente invención;

La Fig. 12 es una ilustración esquemática que muestra la compilación de secuencias de medios diferentes que emplean un archivo contenedor de medios de acuerdo con la presente invención;

La Fig. 13 es otra ilustración esquemática que muestra la compilación de secuencias de medios diferentes que emplean un archivo contenedor de medios

10 La Fig. 14 es una descripción de una red de comunicaciones que incluye servidores que manejan un archivo contenedor de medios

La Fig. 15 es un diagrama de bloques esquemático de un servidor de contenidos de acuerdo aún con otro aspecto de la presente invención;

15 La Fig. 16 es un diagrama de bloques esquemático que ilustra una realización del creador de archivos contenedores de la Fig. 15 en más detalle;

La Fig. 17 es un diagrama de bloques esquemático de un servidor de sesiones de medios

La Fig. 18 es un diagrama de bloques esquemático de un servidor de reparación

Descripción detallada

En todos los dibujos, los mismos caracteres de referencia se usarán para elementos correspondientes o similares.

20 La presente invención generalmente se refiere a la gestión de datos de medios o multimedia y en particular a la creación y utilización de archivos contenedores en conexión con servidores de medios, tales como servidores de difusión de forma continua o de descarga, en una red de comunicaciones basada en radio. Estos archivos contenedores de medios de la invención comprenden, además del contenido de medios a transmitir al(a los) cliente(s) solicitante(s) y las instrucciones usadas para realizar procesamiento de medios y transmisión en los
25 servidores de medios, datos que proporcionan protección de fiabilidad a la sesión de medios. Esta protección de fiabilidad es alcanzable debido a la presencia de datos de redundancia de corrección de error sin canal de retorno (FEC) pre generados incluidos en el archivo contenedor.

30 Como es conocido en la técnica, la FEC incluye añadir datos de redundancia a datos de carga útil transmitidos, lo cual permite a un receptor detectar y corregir errores sin la necesidad de pedir al remitente datos adicionales. La ventaja de la FEC es que la retransmisión de datos a menudo se puede evitar, aunque a costa de requisitos mayores de banda ancha en media. De esta manera, la FEC puede ser usada ventajosamente en conexión con la entrega de contenidos de medios basada en multidifusión/radiodifusión, en la que las retransmisiones serán difíciles de implementar.

35 La FEC se consume añadiendo redundancia a la información a ser transmitida usando un algoritmo o esquema predeterminado, típicamente denominado 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 no modificada en la salida se denomina códec sistemático. En otras palabras un códec FEC sistemático (N, K) conserva los K símbolos de carga útil o fuente y añade (N-K) símbolos FEC. Por consiguiente, un códec FEC no sistemático (N, K) crea N símbolos (FEC o fuente) a partir de K símbolos fuente sin conservar necesariamente todos ellos.

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

50 De acuerdo con la invención actual, los datos o contenidos de medios o multimedia se refieren a cualesquiera datos que se pueden proporcionar por un proveedor o servidor de contenidos a un cliente para 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 contenidos de audio o vídeo de tasa fija, precodificada o en forma de un contenido de audio o vídeo escalable: Otros ejemplos de medios incluyen aún imágenes (JPEG), gráficos de mapas de bits (GIF y PNG), gráficos de vector (SVG), y audio sintético (SP-MIDI) y texto (XHTML y SMIL).

5 En la Fig. 1 está un diagrama de flujo de un método de generación de 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 contenido de medios y formar datos de medios en paquetes de datos transmisibles. De esta manera, el archivo contenedor preferentemente comprende, 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
10 una sesión de medios.

El método comienza en el paso S 1 en que al menos un bloque fuente de medios se organiza y almacena en el archivo contenedor. Si más de dos bloques fuente de medios están presentes en el archivo contenedor, se pueden considerar como bloques de medios separados de un mismo archivo o secuencia de contenidos de medios, por ejemplo una secuencia de vídeo, y/o de diferentes archivos o secuencias de medios, por ejemplo unos pocos
15 bloques fuente de medios de una secuencia de vídeo y unos pocos bloques de medios de una secuencia de audio asociada correspondiente. El al menos un bloque fuente de medios comprende los datos o símbolos de medios que se pretenden que sean enviados a un cliente, en que se reproducen para presentar el contenido de medios a un usuario. Los bloques de medios pueden ser de un mismo tamaño, fijo o al menos una parte de los mismos, si más de uno, puede ser de diferentes tamaños de bit/símbolo.

20 El al menos un bloque de medios organizado en el archivo contenedor en el paso S1 preferentemente comprende colectivamente todos los datos de contenido de medios que van a ser transmitidos a un cliente durante una sesión de medios. En otras palabras, el archivo contenedor contiene datos de medios para una presentación multimedia entera. De esta manera, si el contenido de medios incluye un vídeo musical, el archivo contenedor preferentemente comprende bloques fuente de medios con los datos de vídeo y los bloques de medios con los datos de audio
25 correspondientes. No obstante se anticipa por la presente invención que uno y el mismo contenido de medios se puede proporcionar en múltiples versiones de medios potenciales. Por ejemplo, la parte de vídeo del vídeo musical se puede proporcionar en múltiples versiones de vídeo precodificadas, en que cada versión de vídeo se adapta para uso en conexión con un ancho de banda o nivel o intervalo de tasa de bit dado. Puede haber por lo tanto múltiples versiones de un contenido de medios dado en el archivo contenedor. En tal caso, cada dicha versión de medios se puede considerar como que consta de uno o más bloques fuente de medios. Aunque múltiples versiones de medios pueden estar disponibles en el archivo contenedor, típicamente solamente una de tales versiones se usa en un momento dado durante la sesión de medios, aunque puede haber un conmutador entre las versiones de medios durante la sesión basada en, por ejemplo, cambios en los niveles de ancho de banda disponibles.

35 Para proporcionar protección de fiabilidad al contenido de medios en el archivo contenedor, un siguiente paso S2 pre calcula los datos de redundancia FEC en base a al menos un bloque fuente de medios de los bloques de medios organizados en el archivo contenedor en el paso S1. En este paso de cálculo S2, un códec de bloques FEC, tal como el códec raptor de fuente digital, es empleado preferentemente que funciona en forma de bloque de medios. No obstante, un códec FEC convolucional 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 genera para el al menos un bloque fuente de medios. Este conjunto de símbolos FEC podría incluir uno pero preferentemente
40 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 por limitaciones en el códec FEC empleado, ser una función del número de símbolos fuente de medios en el bloque fuente de medios o limitado por algunos otros criterios, por ejemplo limitaciones de tamaño del archivo contenedor. La idea general de estos símbolos de redundancia FEC pre-calculados es proporcionar, en el archivo contenedor, una reserva de símbolos FEC que están disponibles y se pueden usar por un servidor de medios durante una sesión de medios para proporcionar protección de fiabilidad en la sesión de medios. El número de símbolos FEC por bloque fuente de medios es por lo tanto preferentemente determinado en base a este criterio, es decir que es capaz de proporcionar
45 protección de fiabilidad.

50 Además de ser útiles durante una sesión de medios para proporcionar protección de fiabilidad, los símbolos de redundancia FEC calculados en el paso S2 también se pueden usar en el procedimiento o sesión de reparación post sesión. Alternativamente, un subconjunto dedicado de los símbolos de redundancia FEC están disponibles durante la sesión de medios para la protección de fiabilidad y otro subconjunto de los símbolos de redundancia se dedica para reparación post sesión, la cual se describe en más detalle aquí dentro. Esto tiene la ventaja de que un mismo
55 archivo contenedor (o copia del mismo) se puede usar tanto por un servidor de medios durante una sesión de medios como por un servidor de reparación durante una sesión de reparación. Esto permite una alta flexibilidad en conexión con la gestión de medios. Para más información de los códec FEC, se hace referencia al Anexo B del documento [1].

60 Un siguiente paso S3 organiza los datos de redundancia FEC calculados en el paso S2 en el archivo contenedor junto con el bloque fuente de medios. Este paso S3 preferentemente almacena los símbolos de redundancia FEC del bloque fuente de medios como una reserva en el archivo contenedor. Si los símbolos de redundancia FEC se

han generado tanto para propósitos de protección de fiabilidad como de post reparación, estos símbolos se pueden proporcionar en la misma reserva FEC. Alternativamente, una primera reserva aloja los símbolos FEC para propósitos de fiabilidad y una segunda reserva contiene símbolos FEC de post reparación.

5 El siguiente paso S4 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 S1 y los datos de redundancia FEC almacenados en el archivo en el paso S3. Esta asociación puede ser en forma de un puntero desde la ubicación de almacenamiento del bloque fuente de medios dentro del archivo y a la ubicación de almacenamiento de la reserva FEC, o viceversa. Estos metadatos por lo tanto permiten, dado el bloque fuente de medios particular o su ubicación dentro del archivo contenedor, la identificación de los datos de redundancia FEC asociados calculados en base al bloque fuente de medios o la ubicación de almacenamiento de estos datos de redundancia dentro del archivo. En lugar de emplear un puntero, los metadatos pueden incluir un identificador del bloque fuente de medios y/o la reserva FEC asociada, en el caso de que estos sean almacenados en ubicaciones "estándar", predefinidas en el archivo contenedor. Los metadatos se usan entonces para identificar uno del bloque fuente de medios y la reserva FEC en el archivo y en base a esta ubicación identificada se puede identificar el otro del bloque fuente de medios y la reserva FEC.

En el caso de al menos dos reservas FEC por bloque fuente de medios (tanto para uso en la sesión o post sesión), los metadatos preferentemente comprenden información que proporciona una asociación entre el bloque fuente de medios y ambas reservas FEC. Esto se puede realizar, por ejemplo, incluyendo metadatos en la sesión y metadatos post sesión, respectivamente, en el archivo contenedor para el bloque fuente de medios.

20 En una implementación típica de la invención, se organizan múltiples bloques fuente de medios en el archivo contenedor en el paso S1. En tal caso, los pasos S2 a S4 preferentemente se repiten para cada tal bloque fuente de medios 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 S4 se repiten preferentemente N veces, implicando también organizar al menos N reservas FEC y N versiones de metadatos en el archivo contenedor además de los bloques fuente.

Se anticipa por la invención que el archivo contenedor de medios generado puede contener todos los datos de medios y datos FEC requeridos para una sesión de medios completa. No obstante, los datos de medios y los datos FEC realmente pueden ser aplicables en múltiples sesiones diferentes. Por ejemplo, el archivo contenedor podría incluir datos de medios de un video musical, de un partido de futbol, etc. En tal caso, un servidor de medios no tiene necesariamente que transmitir todos los datos de medios contenidos en la reserva FEC sino solamente los datos particulares pedidos por los clientes. No obstante, otro servidor de medios podría haber recibido una copia del mismo archivo contenedor y en su lugar proporciona otros datos de medios en el archivo a sus clientes.

El método entonces finaliza.

35 La generación de archivos contenedores descrita anteriormente en conexión con la Fig. 1 es preferentemente conducida en un creador o servidor de contenidos que tiene acceso a fuentes de contenidos 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 señal de 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. Alternativamente, o en adición, el archivo contenedor se puede proporcionar a un servidor de reparación al que se puede acceder por clientes después de una sesión de medios con el servidor de medios para reparación post sesión del contenido de medios recibido, si es necesario.

45 A continuación, el término archivo contenedor de medios será usado en toda la revelación con un sentido que incluye tanto archivos de datos para almacenamiento en un medio de almacenamiento y señales para transferencia o distribución.

La Fig. 2 es un diagrama de flujo que ilustra pasos adicionales del método de generación de archivos contenedores 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 archivos contenedores en forma de archivos o secuencias fuente que contienen los datos de medios. En este paso S10, se puede proporcionar uno o más archivos fuente de medios para inclusión en un archivo contenedor. Por ejemplo, un primer archivo de medios puede contener datos de video, mientras que un segundo archivo asociado contiene datos de audio. Para permitir el cálculo eficiente de los datos de redundancia FEC y, más tarde, el uso de tales datos FEC, el(los) archivo(s) fuente de medios proporcionados en el paso S10 se divide(n) en un número de bloques de medios en un siguiente paso S11. El bloque fuente de medios entonces puede ser considerado como un segmento del archivo fuente de medios, al cual puede ser aplicado un código FEC. El tamaño de los bloques de medios en términos de símbolos fuente o 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 que se pretende sea empleado para el cálculo de los datos de redundancia FEC. De esta manera, el códec o algoritmo FEC real y/o la sobrecarga de protección FEC requerida podría afectar la división de archivos de medios y los tamaños de bloques de medios.

En el caso que el archivo fuente de medios de entrada tenga un tamaño de bit o símbolo que es más pequeño que el tamaño máximo que puede ser manejado efectivamente por un códec FEC, ninguna división de ese archivo fuente en bloques fuente de medios es por supuesto requerida y el paso S11 puede ser omitido. El archivo fuente de medios de entrada se considera entonces como un bloque fuente de medios de acuerdo con la invención.

5 Se tiene que señalar que incluso aunque hay un tamaño de bloque preferente, no todos los bloques fuente de medios generados desde un archivo fuente de medios necesita 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 igual dimensionados dado que la parte restante del archivo de medios no contiene suficientes datos de medios para alcanzar el tamaño de bloque preferente.

10 Este paso de división S11 no implica necesariamente que el archivo fuente de medios se divida físicamente en bloques fuente de medios separados que se almacenan en ubicaciones separadas en el archivo contenedor. En claro contraste, en implementaciones más prácticas, el archivo fuente de medios se almacena como una secuencia de datos continua en el archivo contenedor pero se considera como o virtualmente divide 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 y símbolo fuente $[N, 2N-1]$ pertenece al segundo bloque fuente.

15 El método entonces continúa al paso S1 de la Fig. 1, en que el(los) bloque(s) fuente de medios se organiza(n) en el archivo contenedor.

La Fig. 3 es un diagrama de flujo que ilustra pasos adicionales del método de generación de archivos contenedores de la Fig. 1. El método comienza en el paso S20 en que un bloque fuente de medios está particionado en símbolos fuente o los denominados pedazos. Estos símbolos generalmente constan de un número de cientos de bytes. Esta partición de bloques se realiza preferentemente al menos parcialmente en base a información del códec/algoritmo FEC a ser usado para calcular los símbolos FEC para el bloque fuente de medios actual. Como fue mencionado en lo que antecede, los códec FEC basados en Reed-Solomon requieren un cambio en la partición del bloque fuente en base al tamaño de sobrecarga de protección deseado, es decir el número de símbolo de redundancia FEC.

20 De esta manera, el códec o algoritmo FEC real y/o la sobrecarga de protección FEC requerida podría afectar la partición de fuente de medios y los tamaños de símbolo de medios. También otros parámetros tales como el tamaño de paquetes de datos, tales como los paquetes de Protocolo de Datagrama de Usuario (UDP), usados por un servidor de medios para transmitir el contenido de medios se pueden usar en esta partición de bloques fuente del paso S20. En tal caso, el tamaño de los símbolos fuente se podría limitar de manera que al menos un símbolo fuente completo se puede ajustar en un paquete UDP.

25 Este paso de partición S20 no implica necesariamente que el bloque fuente de medios se divide físicamente en símbolos fuente separados que se almacenan en ubicaciones 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, a su vez particionados virtualmente en símbolos fuente.

30 En un siguiente paso S21, se genera información de la partición. Esta información especifica básicamente qué partes de la secuencia de datos pertenece a cada 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 bytes de cada símbolo fuente. Entonces, conociendo la ubicación de inicio de un bloque fuente de medios en un archivo de medios, es posible determinar qué partes de datos pertenecen a los símbolos fuente diferentes.

35 El método entonces continúa al paso S2, en que el símbolo de partición se usa junto con el bloque fuente de medios para calcular los datos FEC. De esta manera, la información de partición permite la identificación de las partes de datos de un bloque fuente de medios que debería ser introducido al códec FEC para la generación de un símbolo FEC.

40 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 incluya protección de fiabilidad (datos FEC precalculados). Esto significa que la división de archivo a bloque fuente, la partición de bloque y el cálculo de protección FEC se hacen "fuera de línea" e independiente del proceso de transmisión de medios real en un servidor de medios. Este pre-procesamiento 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 en una secuencia de medios que se puede transmitir a los clientes solicitantes. El archivo contenedor además también comprende preferentemente la información e instrucciones requeridas por un servidor de reparación para identificar y componer datos FEC útiles en procedimientos de post reparación que siguen a la terminación de una sesión de medios. De esta manera, el archivo contenedor puede ser considerado por lo tanto como un paquete completo de datos, información e instrucción que puede ser usada por servidores transparentes y flexibles para compilación y transmisión de datos.

La Fig. 4 es un diagrama de flujo que ilustra pasos adicionales del método de generación de archivos contenedores de la Fig. 1. El método continúa desde el paso S4 de la Fig. 1. En un siguiente paso S30, se proporciona información del algoritmo FEC de esquema empleado para el cálculo de datos de redundancia FEC. Esta información puede estar en forma del nombre del algoritmo particular o alguna otra información descriptiva. En un planteamiento alternativo, cada algoritmo FEC disponible tiene un identificador predefinido. De esta manera, solamente este identificador FEC puede entonces ser proporcionado en el paso S30. En una implementación típica se ha empleado un mismo códec FEC para determinar datos de redundancia para todos los bloques fuente de medios de un archivo fuente de medios. No obstante, podría ser posible realmente usar códec FEC diferentes 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 podría especificar por lo tanto que todos los bloques en el archivo contenedor han sido procesados para generar los símbolos FEC con un códec FEC único o identificar qué bloque fuente o grupo de bloques fuente para el cual se han usado diferentes códec FEC.

En un siguiente paso S31, se proporciona información de la división de archivos fuente de medios particular. Esta información puede ser de relevancia para el servidor de medios o servidor de reparación, cuando van a proporcionarse secuencias de paquetes de datos de medios o datos FEC a los clientes solicitantes.

En un siguiente paso S32, se proporciona preferentemente una tabla de propiedades. Esta tabla de propiedades es útil en particular si más de un archivo/secuencia fuente de medios se incluye en el archivo contenedor pero ventajosamente se puede usar también 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) de los medios. De esta manera, esta información MIME podría especificar que los medios son medios de audio, medios de vídeo o algún otro tipo de medios, incluyendo el Lenguaje de Integración Multimedia Sincronizado (SMIL). Este tipo MIME proporciona información al servidor de medios de qué tipo de datos que está incluido realmente en el archivo contenedor. La tabla de propiedades también puede incluir información de cualquier esquema de codificación empleado para los datos de medios, incluyendo gzip. También la información del tamaño se puede incluir en la tabla de propiedades. Esta información de tamaño podría exponer el tamaño total de cada archivo fuente de medios en términos de número de bytes o símbolos, los respectivos tamaños de los bloques fuente de medios del(de los) archivo(s) fuente (básicamente correspondiente a la información de división proporcionada en el paso S31), el tamaño de la carga útil máximo u objetivo para los paquetes de datos a ser usados cuando se transmiten datos, el tamaño (en bytes) de un símbolo fuente de medios (básicamente correspondiente a la información de partición generada en el paso S21 de la Fig. 3) 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.

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 especifica la posición de inicio del primer bloque fuente de medios de ese archivo fuente y entonces los bloques de medios restantes se encuentran más tarde en esta posición en el archivo contenedor. Los metadatos generados en el paso S4 de la Fig. 1 y que proporcionan una asociación entre los bloques de fuente de medios y las reservas FEC en el archivo contenedor también se pueden incluir en la tabla de propiedades. Por consiguiente, la información de la partición de bloques empleada y el código FEC empleado para el cálculo de los datos de redundancia FEC se incluyen preferentemente en la tabla.

La tabla de propiedades del archivo contenedor podría ser usada por lo tanto como una fuente de información única para una fuente de medios para localizar un archivo/bloque fuente de medios relevante, proporcionar datos de redundancia FEC para ese bloque fuente y otra información útil en la compilación de paquetes de medios durante una sesión de medios.

Un siguiente paso S33 genera instrucciones de compilación para uso por un servidor de medios. Estas instrucciones se usan para definir la compilación, en base a los metadatos que proporcionan la asociación de bloques FEC, de datos de medios desde los bloques fuente de medios y los datos de redundancia FEC desde la(s) reserva(s) FEC asociada(s) para formar una secuencia de medios de paquetes de datos. De esta manera, estas instrucciones se podrían considerar como índices o metadatos que proporcionan instrucciones de cómo usar los datos (de medios y FEC) incluidos en el archivo contenedor para componer una secuencia de paquetes de medios transmisible que tiene protección de fiabilidad. Estas instrucciones se usan por lo tanto junto con los metadatos de asociación para compilar datos de medios y datos FEC juntos en paquetes adecuados para la transmisión a los clientes solicitantes durante una sesión de medios. Las instrucciones describirán por lo tanto la orden de transmisión del lado del servidor de datos fuente de medios y datos FEC. Señalar que aunque esas instrucciones típicamente no incluyen información de programación de tiempo, información de direcciones objetivo/fuente o puertos u otra información específica de la sesión. Esto significa que el archivo contenedor y las instrucciones de compilación allí dentro son transparentes a la sesión particular y pueden ser usados realmente por un servidor de medios durante múltiples sesiones diferentes con clientes de recepción diferentes pero también por diferentes servidores de medios.

Las instrucciones de compilación podrían aplicarse a un subconjunto de bloques fuente de medios y reservas FEC, implicando 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.

5 Se puede generar realmente más de un conjunto de instrucciones de compilación en el paso S33. En tal caso, se podrían proporcionar diferentes instrucciones alternativas, de manera que un servidor de medios tiene una opción de determinar qué conjunto de instrucciones particular emplear para una sesión de medios particular. Por ejemplo, se podría usar una primera instrucción de compilación para describir la orden de transmisión de los bloques fuente de medios y datos FEC cuando se emplea un canal de transmisión único 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 la orden de transmisión y compilación si están disponibles múltiples canales, implicando que los datos se pueden 10 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 del canal de transporte.

15 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 la 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 para 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 o símbolos de paridad como también son denominados en la técnica a una cantidad dada de símbolos fuente de medios.

20 Además de las instrucciones de compilación a ser usadas por un servidor de medios, las instrucciones de compilación aplicables a un servidor de reparación se pueden generar en el paso S33. Estas instrucciones dedicadas de reparación entonces se puede emplear principalmente por un servidor de reparación post sesión para permitir la identificación de datos de redundancia FEC, datos de redundancia FEC post sesión dedicados posibles, para transmitir a un cliente solicitante que no era capaz de recibir correctamente y descodificar todos los datos de 25 medios recibidos durante una sesión de medios previa.

Un siguiente paso S34 organiza la información, la tabla y las instrucciones proporcionadas y generadas en los pasos previos S30 a S33 y preferentemente el paso S21 de la Fig. 3 en el archivo contenedor. El archivo contenedor contendrá entonces el conjunto completo de datos de medios, datos FEC y metadatos "en bruto" requeridos por un servidor de medios o servidor de reparación para identificar y componer los datos para la transmisión al(a los) 30 cliente(s) solicitante(s). El método entonces finaliza.

La Fig. 11 es una descripción esquemática de un archivo contenedor de medios 1 de acuerdo con la presente invención. Como se ha descrito en lo anterior, el archivo contenedor 1 contiene datos de medios de una serie 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 10, 12, 14 se consideran como divididos en una serie de bloques fuente de medios 20, 22, 24. En la 35 figura Q_1 tales bloques 20, 22, 24 han sido ilustrados para 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 particionado en símbolos fuente.

Además de los bloques fuente de medios 20, 22, 24 con datos de medios, el archivo contenedor 1 comprende reservas FEC 30, 32, 34 que contienen datos de redundancia FEC pre-calculados a ser usados en conexión con los 40 datos de medios para proporcionar protección de fiabilidad. En una implementación preferente, cada bloque fuente de medios comprende al menos una reserva FEC dedicada 30, 32, 34. En tal caso, el número N de reservas FEC

30, 32, 34 en la figura M es $N \geq \sum_{i=1}^M Q_i$. En un planteamiento alternativo, cada archivo fuente de medios 10, 12, 14 tiene al menos una reserva FEC dedicada, es decir $N \geq M$. En tal caso, se podrían emplear departamentos o regiones respectivas de las reservas 30, 32, 34 por los diferentes bloques fuente de medios 20, 22, 24.

45 Los metadatos de asociación 40 de la invención que proporciona una asociación entre la reserva FEC 30, 32, 34 y el(los) bloque(s) fuente de medios 20, 22, 24 en base a los cuales los datos FEC en la reserva 30, 32, 34 se calculan también se proporciona en el archivo contenedor. La Fig. 11 ilustra múltiples ubicaciones 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 50 20, 22, 24 en el archivo también permite la identificación de los metadatos respectivos 40 de ese bloque fuente 20, 22, 24. Alternativamente, o además, los metadatos 40 se almacenan junto con las reservas FEC respectivas 30, 32, 34. De esta manera, cada reserva 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 la cual aplica la reserva particular 30, 32, 34. Si el archivo contenedor 1 comprende una tabla de propiedades de archivo preferentes 60, los metadatos 55 40 de asociación se podrían proporcionar allí dentro. En tal caso, un servidor de medios podría solamente investigar la tabla de propiedades de archivo 60 para identificar la ubicación de los datos de medios y datos FEC relevantes para usar durante una sesión de medios. En una implementación posible adicional, los metadatos 40 de asociación se almacenan en conexión con las diferentes instrucciones de compilación 50, 52, 54 del archivo contenedor 1, denominadas pistas de índice en la figura. En tal caso, cada pista de índice 50, 52, 54 solamente necesita contener

los metadatos 40 que se requieren para la sesión de medios implementable por las instrucciones en esa pista de índice 50, 52, 54. Una combinación de múltiples de estas ubicaciones de almacenamiento posibles también es posible y está dentro del alcance de la presente invención.

5 En un ejemplo útil para comprender la invención, el archivo contenedor 1 también comprende instrucciones de compilación 70 dedicadas para uso durante un procedimiento de reparación, denominado pista de índice de reparación 70 en la figura. Esta pista de índice 70 entonces preferentemente comprende metadatos de asociación 45 útiles para identificación de una reserva FEC 30, 32, 34 que comprende datos FEC post sesión que se pueden usar en el procedimiento de reparación, que se describe aquí dentro además.

10 De acuerdo con un ejemplo útil para comprender la invención, el archivo contenedor de medios 1 es una unidad intercalada, la cual está optimizada para descarga progresiva o difusión de forma continua. Por ello, se puede transmitir y descargar una presentación multimedia entera mediante la denominada difusión en forma continua o descarga progresiva a los clientes solicitantes.

15 El formato de archivos de medios basado en ISO [2, 3, 4] puede ser empleado ventajosamente como formato de archivos para el archivo contenedor de medios de la presente invención. Alternativamente los formatos de archivos contenedores incluyen, el formato de archivos MP4, el formato de archivos 3GP y el formato Quick Time.

20 La Codificación de Capas Asíncrona (ALC) es un protocolo de entrega de contenidos fiable masivamente escalable. Es un protocolo base para la entrega multidifusión fiable de objetos binarios arbitrarios y se ha adoptado como el protocolo obligatorio para entrega de archivos de radiodifusión/multidifusión en BCMCS (Servicio de Radiodifusión/Multidifusión) del 3GPP2 y el grupo de trabajo de Radiodifusión (BCAST) de Navegación y Contenido (BAC) de la Alianza Móvil Abierta (OMA).

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 se ha adoptado recientemente en MBMS del 3GPP y Selección de Datos IP (IPDC) DVB-H como el protocolo obligatorio para la entrega de archivos de radiodifusión/multidifusión. Tanto ALC como FLUTE se definen por el Grupo de Trabajo de Ingeniería de Internet (IETF).

25 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 metadatos de archivo. OMA BCAST define una Guía Electrónica de Servicios (ESG) que es entregada normalmente a clientes con suficiente antelación de la sesión ALC. Si los metadatos de archivo necesitan ser actualizados durante la sesión ALC, entonces se pueden
30 actualizar fragmentos de ESG usando los canales de entrega/actualización ESG.

Los archivos a ser entregados sobre ALC o FLUTE pueden ser almacenados como elementos en un archivo contenedor ISO. Las cajas Meta y las cajas hijas permiten el almacenamiento de una variedad de elementos de datos, tales como presentaciones SMIL y de medios (imágenes) estáticos, en un archivo de medios basado en ISO. También permiten asociar nombres de archivos y caminos a los elementos y la señalización de la estructura del directorio de archivos en el archivo de medios basado en ISO.
35

Generalmente, el primer paso antes de que los archivos se puedan enviar sobre ALC/FLUTE es partirlos en bloques fuente y símbolos fuente. Además, de acuerdo con la presente invención se calculan los símbolos de paridad codificados FEC. La partición puede depender del esquema FEC, el tamaño de paquete objetivo, y la sobrecarga FEC deseada. Para cada bloque fuente a ser codificado FEC, una reserva de símbolos de paridad se pre-calcula y almacena en el archivo de medios basado en ISO junto con la información sobre el esquema FEC y la partición del archivo fuente.
40

El siguiente paso para facilitar la transmisión de archivos es permitir al archivo de medios basado en ISO contener instrucciones para un servidor multidifusión/radiodifusión que describe las sesiones ALC/FLUTE (con Protocolo de Descripción de Sesiones) y cómo encapsular elementos en paquetes ALC o FLUTE.

45 La partición de archivos y reservas FEC, por un lado, y las pistas de índice para entrega de archivos, por el otro, se pueden usar independientemente una de otra. La primera ayuda al diseño de las pistas de índice y permite pistas de índice alternativas, con, por ejemplo diferentes sobrecargas FEC, para reutilizar los mismos símbolos FEC. También proporcionan medios para acceder a los símbolos fuente y símbolos FEC adicionales independientemente para reparación post entrega, la cual se puede realizar sobre ALC/FLUTE o fuera de banda a través de otro protocolo.
50 Para reducir la complejidad cuando un servidor sigue las instrucciones de pista de índice, no obstante, las pistas de índice se refieren directamente a las gamas de datos de elementos o datos copiados en muestras de índice.

A continuación se da un ejemplo de implementación más detallado de un archivo contenedor de acuerdo con la invención en la forma en el formato de archivo de medios basado en ISO y adaptado para transmisión sobre ALC/FLUTE. Esto debería ser visto, no obstante, meramente como un ejemplo ilustrativo de la presente invención y las modificaciones y cambios obvios a este ejemplo están dentro del alcance de la invención.
55

Almacenamiento de ficheros fuente y reservas FEC

5 Los ficheros previstos para transmisión sobre ALC/FLUTE se almacenan como elementos en una caja Meta de alto nivel ('meta') de un archivo de medios basado en ISO que actúa como un archivo contenedor. La caja de Ubicación de Elementos ('iloc') especifica la ubicación 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 fichero, tipo de contenido (tipo MIME), etc., de cada elemento se proporciona por la caja de información de elemento ('iinf').

De una manera similar, las reservas FEC pre-calculadas se almacenan como elementos adicionales en la caja Meta. Si un archivo fuente se divide en varios bloques fuente, las reservas FEC para cada bloque fuente se almacenan preferentemente como elementos separados. La relación entre las reservas FEC y los elementos fuente originales se graba en la caja de Información de Elementos de Entrega de Ficheros (FD) descrita en la siguiente sección.

10 Caja de Información de Elementos FD

Se proporcionan detalles en la partición de archivos fuente y reservas FEC en la caja de Información de Elementos FD ('fiin'). La caja es usada preferentemente para archivos que emplean pistas de índice FD y preferentemente exactamente uno se sitúa en la caja 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 caja de información de elementos FD proporciona detalles en una partición de archivos particular, codificación FEC, reservas FEC asociadas, y metadatos para un archivo fuente de medios particular. Es posible proporcionar múltiples entradas para un archivo fuente si se usan esquemas o particiones de codificación FEC alternativos en el archivo ISO. Todas las entradas de partición se pueden numerar implícitamente y la primera entrada típicamente tiene el número 1.

20 Entrada de Partición

Cada entrada de partición ('paen') de una fuente se define como sigue:

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

Puede contener dos cajas que juntas proporcionan todos los detalles sobre cómo un archivo fuente de medios se codifica FEC.

25 Caja de Partición de Archivos

La caja de Partición de Archivos ('fpart') 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:

5 **item_ID** indica el ítem_ID del archivo fuente. Es posible proporcionar particiones alternativas y/o codificaciones FEC de un archivo fuente usando el mismo ítem_ID en la caja de Partición de Archivos de más de una entrada de Información de Archivos.

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

10 **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” [5]. Un valor de uno preferentemente corresponde al “MBMS FEC” [1].

FEC_instance_ID proporciona una identificación más específica del codificador FEC que se usa para un esquema FEC Infra-Especificado. Este valor típicamente no se usa para esquemas FEC Especificados Completamente. Ver el documento [5] para detalles adicionales de esquemas FEC Infra-Especificados.

15 **max_source_block_lenght** da el número máximo de símbolos fuente por bloque fuente de medios.

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

20 **max_number_of_encoding_symbols** da el número máximo de símbolos de codificación que pueden ser generados para un bloque fuente para el ID de codificación FEC 129 definido en el documento [5].

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

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

block_count indica el número de bloques fuente consecutivos de tamaño **block_size** (en bytes). Un **block_size** que no es un múltiplo del tamaño del símbolo (proporcionado en la Caja de Información FEC) indica que el símbolo fuente incluye relleno no almacenado en el elemento de archivo.

30 **Caja de Reserva FEC**

La caja de Reserva FEC ('fecr') asocia un archivo fuente de medios con las reservas FEC almacenadas como elementos adicionales:

```
aligned(8) class FECReservoirBox extends FullBox('fecr', version = 0, 0)
{
    unsigned int(16) entry_count;
    for (i=1; i <= entry_count; i++)
    {
        unsigned int(16) item_ID;
        unsigned int(32) symbol_count;
    }
}
```

Semántica:

entry_count da el número de entradas en la lista de parejas (**item_ID**, **symbol_count**) que proporcionan el **item_ID** para cada reserva FEC y el número de símbolos fuente que contiene. La lista comienza con la reserva FEC asociada al primer bloque fuente del archivo fuente de medios y continúa secuencialmente a través del archivo.

Caja de información de Elemento

Para transmitir internamente medios discretos integrados usando el 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 se usa FLUTE como un protocolo de radiodifusión, y como parte de OMA BSCAST ESF, si ALC se usa en conjunto con OMA BCAST ESG.

Ya que 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 a FLUTE como a ALC se define como una segunda versión de la entrada de información de elemento. Esta versión de la entrada de información de elemento se usa en la caja 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 Marcas Extensible (XML) contenido en una caja 'xml') o bien el ID del elemento para el cual se define la siguiente información.

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

content_lenght da la longitud total (en bytes) del archivo (no codificado).

transfer_lenght da la longitud total (en bytes) 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).

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

content_type es una cadena terminada nula en 8 caracteres UTF con el tipo MIME del elemento. Si el elemento tiene el contenido codificado (ver más adelante), entonces el tipo MIME se refiere al elemento después de la

descodificación del contenido.

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

5 **content_encoding** es una cadena terminada nula en 8 caracteres UTF usada para indicar que el archivo binario se codifica y necesita ser descodificado antes de interpretado. Los valores son como se define para la Codificación de Contenidos para HTTP/1.1. Algunos valores posibles son “gzip”, “compress” y “deflate”. Una cadena vacía indica codificación sin contenido. Señalar que el elemento se almacena después de que se ha aplicado la codificación de contenido.

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

entry_content da el número de entradas en la siguiente lista.

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

15 Todos los campos son preferentemente empleados. No obstante, es posible que una cadena terminada nula solamente contenga un nulo para indicar que el valor correspondiente del campo no es proporcionado. Extensiones futuras de la caja pueden añadir campos adicionales al final.

Considerando la información proporcionada en la caja de Información de Archivo para cada elemento y la lista de elementos usados por una pista de índice, las entradas de archivo necesarias para una FDT o una ESG se pueden interpretar.

20 El **content_location** de los recursos de medios integrados se puede denominar usando las formas de la Localización Universal de Recursos (URL) definidas en la Sección 8.44.7 del formato de archivos de medios basado en ISO [2, 3].

Caja de Grupo de Sesión

25 Una sesión FD puede enviarse simultáneamente sobre varios canales FD, cada uno de los cuales se describe por una pista de índice FD. La caja de grupo de Sesión contiene una lista de sesiones así como todos los grupos de archivos de medios y pistas de índice que pertenecen a cada sesión. Si hay más de una pista de índice FD en el archivo contenedor, entonces una caja de grupo de sesión está preferentemente presente en la caja de información de elemento FD.

30 Solamente un grupo de sesión se debería procesar en cualquier momento. La primera pista de índice enumerada en un grupo de sesión especifica el canal base. Si el servidor de medios no tiene preferencia entre los grupos de sesión, la elección por defecto es típicamente el primer grupo de sesión. Los ID de grupo de todos los grupos de archivos que contienen los archivos referenciados por las pistas de índice se incluyen en la lista de grupos de archivos. Los ID de grupos de archivos puede a su vez ser traducidos en nombres de grupo de archivos (usando la caja ID del Grupo a Nombre) que pueden ser incluidos por el servidor en los 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:

35 **num_session_groups** especifica el número de grupos de sesión.

entry_count da el número de entradas en la siguiente lista que comprende todos los grupos de archivos con los

que el grupo de sesión cumple. El grupo de sesión contiene todos los archivos incluidos en los grupos de archivos enumerados como se especifica por la entrada de información de elementos de cada archivo fuente. La FDT para el grupo de sesión debería preferentemente contener solamente aquellos grupos que están enumerados en esta estructura.

5 **group_ID** indica un grupo de archivos con el que cumple el grupo de sesión.

num_channels_in_Session_groups especifica el número de canales en el grupo de sesiones. El valor de **num_channels_in_session_groups** es un entero positivo.

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

10 Caja de ID de Grupo a nombre

La caja de ID de Grupo A Nombre asocia nombres de grupo de archivos a los ID de grupo de archivos usados en las entradas de información de elementos.

```
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:

15 **entry_count** da el número de entradas en la siguiente lista.

group_ID indica un grupo de archivos.

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

Formato de Pista de Índice

20 La estructura de pista de índice se generaliza para soportar las muestras de índice en múltiples formatos de datos. La muestra de pista de índice contiene cualesquiera datos necesarios para construir la cabecera de paquete del tipo correcto, y también contiene un puntero al bloque fuente de medios de los datos que pertenecen en el paquete.

Formato de entrada de muestras

25 Las pistas de índice FD son pistas de índice (manipulador de medios 'índice') con un formato de entrada en la descripción de muestras de 'fdp', tipo para el Protocolo de Entrega de Archivos. La `FDHintSampleEntry` está contenida en la `SampleDescriptionBox` ('stsd') y tiene la siguiente sintaxis:

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

Semántica:

30 **partition_entry_ID** indica la entrada de partición en la caja de información de elemento FD. Un valor cero indica que no está asociada ninguna entrada de partición con esta entrada de muestras, por ejemplo para la FDT.

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

Los campos, "hinttrackversion" y "highestcompatibleversion" tienen la misma interpretación que en la

“RtpHintSampleEntry”, descrita en la sección 10.2 del formato de archivos de medios basado en ISO [2, 3]. Como datos adicionales una caja `time_scale_entry` se puede proporcionar. Si no se proporciona, no hay indicación dada en la temporización de paquetes.

- 5 Las entradas de archivos necesarias para una FDT o una ESG se pueden crear observando todas las entradas de muestras de una pista de índice y las cajas de información de Metadatos de Archivos correspondiente de los elementos referenciados por los anteriores `item_IDs`. No se deberían incluir entradas de muestras en la pista de índice si no están referenciados por ninguna muestra.

Se recomienda que el servidor de medios envíe un conjunto diferente de símbolos FEC para cada retransmisión del archivo.

10 Formato de muestras

- 15 Cada muestra FD en la pista de índice generará uno o más paquetes FD. Cada muestra contiene dos áreas: las instrucciones para componer los paquetes, y cualesquiera datos adicionales necesarios cuando se envían esos paquetes (por ejemplo símbolos de codificación que son copiados 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.

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

- 20 Los números de muestra de las muestras FD definen el orden en que serán procesadas por el servidor de medios. De manera similar, las cajas de Paquetes FD en cada muestra FD aparecen en el orden en que serán procesadas. Si la caja de Entrada de Escala de Tiempo está presente en la Entrada de Muestras de Índice FD, los momentos de muestras se definen y proporcionan momentos de envío relativos de paquetes para una tasa de bit por defecto. Dependiendo de la tasa de bit de transmisión real, un servidor puede aplicar ampliación de tiempo lineal. Los momentos de muestras pueden simplificar el proceso de programación, pero depende del servidor de medios enviar paquetes de una manera oportuna.

Formato de entrada de paquetes

- 25 Cada paquete en la muestra FD tiene la estructura siguiente [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: recuento de los siguientes constructores.

header_extension_constructors: estructuras que son usadas para construir las extensiones de cabecera LCT.

- 30 **entry_count2**: recuento de los siguientes constructores.

packet_constructors: estructuras que son usadas para construir el ID de 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;
}
```

- 5 La plantilla de cabecera LCT se puede usar por un servidor de medios para formar una cabecera LCT para un paquete. Señalar que algunas pares de la cabecera dependen de la política del servidor y no se incluyen 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 del TOI.

Formato de constructor de extensión de cabecera LCT

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

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

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

header_extension_content es el número de elementos igual a **header_extension_length**.

Formato de Constructor de Paquetes

- 15 Hay varias formas del constructor. Cada constructor es de 16 bytes para hacer la iteración más fácil. El primer byte es un discriminador de unión. Esta estructura se basa en la sección 10.3.2 del formato de archivos de medios basado en ISO [2, 3]. Los constructores de paquetes se usan para incluir el ID de carga útil FEC así como los símbolos fuente en un paquete FD.

```
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) samplenumber;
    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;
}

```

Caja de Datos Adicionales

Cada muestra de una pista de índice FD puede incluir datos adicionales almacenados en una caja 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 sesión de medios. Esta gestión de sesión de medios se lleva a cabo en un servidor de medios, tal como 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 S40, en que se proporciona un archivo contenedor de medios. Este suministro de archivo se puede realizar trayendo el archivo contenedor desde una ubicación de memoria del servidor de medios, que implica que el servidor previamente ha recibido el archivo desde un proveedor o creador de contenido. 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.

10

En un siguiente paso S41, los paquetes de medios se compilan extrayendo datos de medios desde el(los) bloque(s) fuente de medios del archivo contenedor y extrayendo los datos de redundancia FEC desde el(las) reserva(s) FEC. Esta extracción de datos se realiza en base a los metadatos en el archivo contenedor que proporcionan una asociación entre el(los) bloque(s) fuente de medios y el(las) reserva(s) FEC. De esta manera, el servidor de medios preferentemente recibe un identificador de los datos de medios para transmitir durante la sesión de medios. Alternativamente, el archivo contenedor podría contener solamente datos de medios de un único archivo de datos de medios de manera que no es necesaria la selección de la fuente de medios. En 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 inicio del archivo de medios, es decir el primer bloque fuente de medios desde el cual se debería iniciar la transmisión. Adicionalmente, se podría usar información adicional incluida en el archivo contenedor como instrucciones de cómo los datos de medios y datos FEC deberían ser combinados e incluidos en los paquetes de datos adaptados para transmisión inalámbrica sobre un canal o múltiples canales basados en radio a diferentes clientes. Los metadatos de la invención permiten, dado el bloque fuente de medios desde el cual los datos de medios se extraen actualmente, la identificación de la reserva FEC asociada desde la cual los datos FEC deberían ser extraídos en el paso S41 de compilación de paquetes.

15

20

25

En un siguiente paso S42, los paquetes de datos de medios compilados con la protección de fiabilidad 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 se inicia típicamente una vez que un 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 se compilan e introducen en el almacenador temporal de transmisión, mientras que otros paquetes están siendo transmitidos, lo cual se ilustra esquemáticamente por la línea L2.

30

35

El archivo contenedor generado y la organización de datos de medios y el suministro de datos FEC pre-calculados allí dentro, reducen las necesidades de procesamiento del servidor de medios durante una sesión de medios. Esto por lo tanto conduce a reducir la complejidad del servidor y permite flexibilidad del servidor ya que el servidor no necesita la construcción del bloque fuente y codificación FEC sobre la marcha. En claro contraste, el servidor usa los metadatos e instrucciones en el archivo contenedor para extraer la fuente pre-calculada y símbolos FEC, añade información de cabecera y envía los paquetes de datos resultantes a los clientes.

40

El método entonces finaliza.

La Fig. 6 es un diagrama de flujo que ilustra pasos adicionales del método de gestión de sesión de la Fig. 4. El

método continúa desde el paso S40 de la Fig. 5. En un siguiente paso S50 se determina una capacidad de sobrecarga FEC que puede ser empleada actualmente 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, niveles de tasa de bit mínimo y máximo para la(las) portadora(s) de 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 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 de compilación 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 a añadir a los datos de medios cuando se compilan paquetes de datos de medios. Cuanto más grande sea la sobrecarga FEC aceptable, más datos FEC son añadidos. Teniendo diferentes instrucciones de compilación alternativas diferentes, el servidor de medios puede usar esas instrucciones que permiten una protección FEC permisible más alta dadas las limitaciones de sobrecarga reales y por ello aumenta las posibilidades de recepción y descodificación con éxito de los datos de medios en diferentes clientes comparado con el uso de un único conjunto de instrucciones de compilación.

El método entonces continúa al paso S41 de la Fig. 5, donde los paquetes de datos de medios se compilan a partir de los datos de contenido y los datos FEC asociados en base a las instrucciones de compilación seleccionadas en el paso S51.

La Fig. 12 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 preferentemente comprende múltiples bloques fuente de medios, tales como dos bloques fuente de medios. En esta realización, cada bloque fuente de medios del archivo fuente 10 tiene una reserva FEC asociada 30, 32 que comprende datos de redundancia FEC precalculados para el bloque fuente respectivo. El archivo contenedor 1 también comprende, en este ejemplo ilustrativo, tres pistas de índice 50, 52, 54 que contienen instrucciones de compilación para diferentes sobrecargas FEC. Por ejemplo, la primera pista de índice 50 podría ser usada cuando se desea una sobrecarga de redundancia del 10%, la segunda pista de índice 52 da una sobrecarga FEC de alrededor del 12% y la tercera pista de índice 54 da una sobrecarga FEC del 14%. En la figura, se ha empleado el algoritmo de construcción de bloques fuente sugerido en el Anexo B del documento [1]. Si la primera pista de índice 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 ha sido indicado en la figura). No obstante, si en su lugar la segunda pista de índice 52 se usa, se genera una segunda secuencia de paquetes de datos 91, 92, 93, 94. Comparada con la primera secuencia 80, la segunda secuencia 90 comprende bloques FEC más grandes, 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.

Como se ha descrito en lo anteriormente mencionado, el archivo contenedor de medios puede contener múltiples archivos fuente de medios o archivos que transportan diferente contenido de medios. La Fig. 7 es un diagrama de flujo que ilustra un ejemplo de los pasos de compilación y transmisión de la Fig. 5 con tal solución multiarchivo. El método continúa desde el paso S40 de la Fig. 5. En un siguiente paso S60, los paquetes de datos de medios de un primer archivo de contenido de medios y la primera reserva FEC son generados en base a los metadatos y las instrucciones de compilación dedicadas a este contenido de medios particular. El resultado de la generación de paquetes podría ser considerado como una secuencia de paquetes de datos que contienen contenido de medios del primer conjunto y datos FEC de la primera reserva FEC. Este primer conjunto de paquetes de datos es entonces transmitido en el paso S61 a los clientes solicitantes usando un canal de comunicación basado en radio, preferentemente un canal de radiodifusión o multidifusión.

No obstante, en la sesión de medios actual también el contenido de medios de un segundo archivo fuente de medios o conjunto en el archivo contenedor debería ser transmitido a los clientes. Como consecuencia, los paquetes de datos que contienen datos de contenido de medios desde el segundo archivo fuente de medios y datos de redundancia FEC desde una segunda reserva FEC asociada se generan en base a los metadatos y las instrucciones de compilación asociadas con este contenido en el paso S62. Los paquetes de datos resultantes son entonces transmitidos, en el paso S63, usando el mismo canal de comunicación basado en radio que fue empleado por el servidor en el paso previamente descrito S61. De esta manera, los dos conjuntos de paquetes de datos se enviarán como una secuencia continua de paquetes de datos.

La Fig. 8 es un diagrama de flujo que ilustra un ejemplo adicional de los pasos de compilación y transmisión de la Fig. 5 con una solución multiarchivo. El método continúa desde el paso S40 de la Fig. 5. En un siguiente paso S70, los paquetes de datos de medios se generan para contener datos de contenido de medios a partir de un primer archivo fuente de medios y datos FEC desde una primera reserva FEC. Este paso S70 básicamente corresponde al paso S60 de la Fig. 7 y no se describe además aquí dentro. Un siguiente paso S71 corresponde al paso S61 de la Fig. 6, en que los paquetes de datos que contienen el contenido de medios derivado de un segundo archivo fuente de medios y la segunda reserva FEC del archivo contenedor. En esta realización, el servidor de medios puede

gestionar dos grupos de clientes (grupos de multidifusión IP) simultáneamente. De esta manera, en un siguiente paso S72, el servidor de medios transmite el primer conjunto o secuencia de paquetes de datos generado en el paso S70 a los miembros clientes del primer grupo de medios usando un primer canal de comunicación basado en radio y simultáneamente o al menos en parte transmite simultáneamente la segunda secuencia de paquetes de datos de medios a miembros clientes de un segundo grupo de medios usando un segundo canal de comunicación basado en radio. Los dos conjuntos de paquetes de datos serán enviados, de esta manera, como secuencias de medios paralelas en esta realización. El método entonces finaliza.

Un ejemplo adicional útil para comprender la invención también abarca la situación en que los datos de medios de múltiples archivos de medios se compilan y transmiten colectivamente a un grupo de clientes. En tal caso, los datos de medios desde los múltiples archivos fuente se gestionan juntos por el servidor de medios durante la sesión de medios. Este es en particular el caso en que un archivo de medios contiene datos de vídeo y un segundo archivo de medios contiene datos de audio asociados. Los datos de medios de los dos archivos se pueden transmitir usando un mismo canal de comunicación basado en radio o diferente o diferentes tales canales.

Se anticipa que las enseñanzas descritas anteriormente en conexión con las Fig. 7 y 8 se pueden extender a manejar el caso con más de dos archivos fuente de medios y reservas FEC incluidas en el archivo contenedor.

La Fig. 13 es una representación esquemática de un archivo contenedor de medios 1 que tiene múltiples archivos fuente de medios 10, 12, 14 y múltiples reservas FEC 30, 32, 34. El archivo contenedor 1 también comprende dos pistas de índice 50, 52 con instrucciones de compilación y metadatos requeridos para extraer datos de contenido de medios desde los archivos fuente 10, 12, 14 y los datos FEC desde las reservas 30, 32, 34.

Una primera pista de índice 50 podría ser empleada por un servidor de medios cuando se compone una secuencia de paquetes de datos 81, 82, 91, 92 que contiene datos de medios y datos FEC de múltiples archivos fuente 10, 12, 14 y reservas FEC 30, 32, 34. La secuencia generada será transmitida entonces usando un canal basado en radio para clientes solicitantes (comparar con la Fig. 7). La segunda pista de índice 52 podría ser empleada cuando múltiples grupos de medios deberían ser gestionados simultáneamente. De esta manera, múltiples secuencias en paralelo de paquetes de datos 81, 82; 91, 92 se generan por el servidor de medios en base a la pista de índice 52, archivos fuente 10, 12, 14 y reservas FEC 30, 32, 34.

Si el archivo contenedor de medios también comprende información adicional, tal como información de división archivo a bloque, partición de bloques, 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, la información de la partición de bloques se puede usar por el servidor de medios junto con los metadatos para extraer datos de medios de los archivos y bloques fuente de medios en el archivo contenedor. La información de partición ayuda, en este sentido, al servidor a identificar correctamente la ubicación de almacenamiento correcta de los datos de medios deseados en los archivos contenedores. De la misma manera, la información de algoritmo FEC puede ser útil por el servidor de medios junto con los metadatos en la extracción de datos de medios y datos FEC desde el archivo contenedor. Diferentes algoritmos FEC requieren diferente partición de contenido de medios en símbolos y bloques fuente y además la partición podría depender de la sobrecarga de protección. De esta manera, la información del algoritmo FEC y otros parámetros FEC empleados cuando se precálculan los datos FEC puede ser de uso para el servidor de medios.

Los datos adicionales y preferentemente la información de tipo MIME, cualquier información de codificación, información de tamaño, etc., útil por 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 y transmisión de paquetes de datos.

El servidor de medios también podría ser empleado para procedimientos de post reparación. De esta manera, en tal caso, después que los datos de contenido de medios y datos FEC han sido transmitidos (multidifusión) a clientes, podría ser posible que algunos de los clientes no sean capaces, a pesar de la inclusión de datos de redundancia FEC en la secuencia de medios, de descodificar correctamente los símbolos de medios recibidos. La Fig. 9 es un diagrama de flujo que ilustra pasos adicionales que definen tal procedimiento post reparación. Este procedimiento es iniciado típicamente una vez que la transmisión de medios durante la sesión de medios ha sido finalizada o al menos a continuación de la transmisión de algunos datos de medios a los clientes. El método por lo tanto continúa desde el paso S42 de la Fig. 5. En el siguiente paso S80, el servidor de medios recibe una petición para un procedimiento de reparación post sesión que se origina desde un cliente, el cual previamente ha sido implicado en una sesión de medios con el servidor. La petición de reparación típicamente comprende un identificador de los datos de medios recibidos incorrectamente. Esta identificación puede ser el nombre del contenido de medios particular y posible alguna información que permite al servidor de contenidos identificar en más detalle la parte del contenido que no fue recibida con éxito por el cliente. En un siguiente paso S81, el servidor usa esta información para extraer los datos de redundancia FEC de una reserva FEC identificados en base a la información. En este paso de extracción S81, el servidor podría, por defecto, extraer un número predefinido de símbolos FEC que son transmitidos al cliente solicitante en el paso S82, preferentemente usando transmisión de datos basada en unidifusión. Si los símbolos FEC

proporcionados no son suficientes para la descodificación con éxito del contenido de medios, el cliente podría requerir más símbolos FEC desde el servidor. Alternativamente, el cliente dota al servidor con información que permite al servidor estimar al menos el número de símbolos FEC que serían requeridos para la descodificación con éxito del contenido de medios en el cliente. En tal caso, el servidor preferentemente usa esta información en el paso de extracción S81 y los símbolos FEC extraídos son transmitidos al cliente en el paso S82.

El archivo contenedor de medios también se puede emplear en un procedimiento de reparación post sesión dedicado siguiente a su uso en una sesión de medios. La Fig. 10 es un diagrama de flujo que ilustra tal método de reparación post sesión. El método comienza en el paso S90, en que un servidor de reparación recibe una petición de datos de redundancia FEC desde un cliente que previamente ha sido implicado en una sesión de medios y recibido datos de medios de un servidor de medios. La petición recibida en el paso S90 incluye un identificador que permite al servidor de reparación identificar un archivo contenedor de medios. De esta manera, el servidor proporciona en un siguiente paso S91, el archivo contenedor de medios relevante en base a la petición, típicamente el identificador en la petición. Este archivo contenedor típicamente es una copia del archivo contenedor usado por un servidor de medios en la sesión de medios previa y por lo tanto comprende bloque(s) fuente de medios y reserva(s) FEC que contienen de datos de medios y FEC, respectivamente, transmitidos pero no recibidos totalmente con éxito en el cliente solicitante. El archivo contenedor se puede ordenar o requerir desde un proveedor/creador de contenidos o realmente desde el servidor de medios que conduce la sesión de medios previa.

El servidor de reparación usa el identificador en la petición de reparación para extraer los datos de redundancia FEC desde el archivo contenedor de medios en un paso S92 en base a los metadatos. En una implementación preferente, el identificador podría ser un nombre del archivo fuente de medios que contiene el contenido de medios recibido incorrectamente o alguna información más detallada, que incluye la identificación del bloque fuente de medios particular que contenía los datos de medios omitidos. Los metadatos en el archivo contenedor entonces se usan junto con el identificador para identificar al menos una reserva FEC en el archivo contenedor, desde el cual deberían ser extraídos los datos FEC. Como se describió en conexión con el paso S81 de la Fig. 9, el servidor de reparación podría ser informado de una estimación de la cantidad de datos FEC que serían requeridos para descodificar con éxito los datos. Alternativamente, el servidor de reparación podría extraer y enviar un número por defecto de símbolos FEC y podría requerir al cliente requerir más datos FEC si este número por defecto de símbolos no fuese suficiente.

En una implementación preferente, el archivo contenedor comprende instrucciones de reparación post sesión 70, ver Fig. 11. Estas instrucciones entonces se usan por el servidor de reparación para compilar paquetes de datos que contienen datos FEC extraídos de una reserva FEC 30, 32, 34. Los metadatos 45 usados por el servidor en este procedimiento de extracción, podrían ser incluidos o constituir una parte de las instrucciones de compilación 70, como se ilustra en la figura. Alternativamente, los metadatos 45 podrían ser dotados en conexión con el archivo fuente de medios relevante 10, 12, 14, la reserva FEC 30, 32, 34 o en una tabla de propiedades de archivo 60 incluida en el archivo contenedor 1.

La Fig. 14 es una descripción esquemática de una red de comunicaciones que ilustra las partes que generan o que usan el archivo contenedor de medios 1 de la presente invención. Un servidor de contenidos 100 representa el proveedor o creador de contenidos que recibe o tiene acceso a 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 400, 410, 420 representados por terminales móviles en la figura. Un servidor de reparación 300 puede ser contactado por uno de los terminales móviles 400 a continuación de la sesión con el servidor de medios 200. En tal caso, el servidor de reparación 300 requiere una copia del archivo contenedor de medios 1 desde el servidor de contenidos 100 y la usa en un procedimiento post sesión.

La Fig. 15 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 (I/O) general 110 dispuesta para y que comprende la funcionalidad (transmisora/receptora, moduladora/demoduladora, codificadora/descodificadora) para comunicar con unidades externas. Esta unidad de I/O 110 se dispone en particular para recibir contenido de medios de entrada y para recibir peticiones de archivos contenedores de medios. La unidad de 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 archivos contenedores 120 dispuesto para crear archivos contenedores de medios de la invención. El creador 120 comprende un gestor de bloques de medios 130 dispuesto para introducir y organizar al menos un bloque fuente de medios en un archivo contenedor de medios. Este al menos un bloque fuente de entrada se proporciona preferentemente desde un divisor de archivos 190 del servidor de contenidos 100. Alternativamente, el al menos un bloque fuente podría ser recuperado desde un almacenamiento de datos interno 115 y ser proporcionado, usando la unidad de I/O 110, desde fuentes de medios externas 500, 510.

Un códec FEC 160 del servidor de contenidos 100 calcula los datos de redundancia FEC para los bloques fuente de medios introducidos en el archivo contenedor por el gestor de bloques 130. Este códec FEC 160 podría usar

cualquiera de los algoritmos FEC previamente mencionados en este procedimiento de cálculo. El códec 160 se puede ajustar para usar un algoritmo FEC dado para todos sus cálculos de datos de redundancia. Alternativamente, el códec 160 puede tener acceso a múltiples algoritmos diferentes y por lo tanto tiene una opción de algoritmo real. La selección del algoritmo podría ser realizada en base a diferentes parámetros, tales como la división de bloques particular hecha por el divisor de archivos 190, el tipo de fecha en los bloques fuente de medios o algún otro parámetro. Los datos FEC calculados por el códec 160 podrían ser empleados tanto como datos de redundancia en una sesión de medios como datos de reparación en un procedimiento de reparación post sesión.

Los datos (símbolos) FEC calculados resultantes se introducen a un gestor de datos FEC 140 del creador de archivos 120. Este gestor FEC 140 organiza los datos FEC de entrada en al menos una reserva FEC en el archivo contenedor de medios. En una implementación preferente, el códec FEC 160 y el gestor FEC 140 genera símbolos FEC para cada bloque fuente organizado en el archivo por el gestor de bloques 130 y proporciona al menos una reserva FEC por bloque fuente en el archivo contenedor.

Un gestor de metadatos 150 del gestor de archivos 120 proporciona los metadatos en el archivo contenedor. Estos metadatos proporcionan una asociación entre los bloques fuente de medios organizado por el gestor de bloques 130 y las reservas FEC organizadas por el gestor FEC 140.

El archivo contenedor de medios resultante entonces puede ser almacenado, al menos temporalmente, en el almacenamiento de datos 115 o ser transmitido por la unidad de I/O 110 a un servidor de medios o servidor de reparación.

En una implementación preferente, el contenido de medios de entrada está en forma de un archivo fuente de medios que se proporciona al divisor de archivos 190. Este divisor 190 divide el archivo fuente en uno o más bloques fuente. El divisor 190 podría basar esta división de archivo en base a diferente información o parámetros. Por ejemplo, la división de archivos podría ser determinada al menos parcialmente en base al algoritmo FEC empleado por el códec FEC 160. En tal caso, la división de archivos 190 preferentemente tiene acceso a la información de tal algoritmo FEC. El divisor 190 podría dividir entonces un archivo fuente de medios en $N-1$ bloques fuente de medios dimensionados de igual manera y un bloque fuente de medios que podría tener un tamaño más pequeño que los otros $N-1$ bloques.

El archivo de medios también se introduce a un particionador de bloques 195 operable sobre los datos de medios preferentemente siguientes al procesamiento de datos por el divisor de archivos. Este particionador 195 funciona sobre un bloque fuente de medios en tiempo y parte el bloque en símbolos fuente de un tamaño definido. Esta partición de bloques se basa preferentemente en el algoritmo o esquema FEC particular a ser empleado por el códec FEC 160. El particionador de bloques 195 también podría funcionar para realizar la partición adaptada a ajustar los símbolos fuente en paquetes de datos que serán empleados por un servidor de medios durante una sesión de medios. De esta manera, la información del tamaño del paquete, tal como el tamaño del paquete UDP, podría ser empleada por el particionador 195.

El particionador 195 también genera información de la partición de bloques resultante. Esta información entonces se reenvía al códec FEC 160 para uso cuando se generan los datos de redundancia FEC.

Las unidades 110, 120, 130, 140, 150, 160, 190 y 195 del servidor de contenidos 100 pueden ser implementadas o proporcionadas como soporte lógico, componentes físicos o una combinación de los mismos. Las unidades 110 a 195 pueden ser implementadas 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 está dentro del alcance de la invención. En tal caso, se pueden disponer diferentes unidades 110 a 195 del servidor de contenidos 100 en nodos de red diferentes pero a pesar de esto realizarán sus operaciones previstas como se describió en lo anteriormente mencionado.

La Fig. 16 es un diagrama de bloques esquemático que ilustra una realización del creador de archivos contenedores 120 de la Fig. 15 en más detalle. Este creador de archivos 120 también comprende un gestor de información 170 para incluir y organizar datos de información diferentes en el archivo contenedor. El gestor 170 preferentemente proporciona información del algoritmo FEC usado por el códec FEC 160, la división de archivos usada por el divisor de archivos 190 y/o la partición de bloques del particionador de bloques 195 en el archivo. Otra información podrían ser los datos descriptivos de los datos de medios, tales como el tipo de medios, información de tamaño (de bloque y/o símbolo), nombre o identificadores de contenido, información de ubicación, información de codificación opcional, etc. En una realización preferente, esta información se organiza, posiblemente junto con los metadatos desde el gestor meta 150, en una tabla de propiedades de archivo incluida en el archivo contenedor.

Un gestor de instrucciones 180 del creador de archivos 120 genera e inserta instrucciones de compilación en el archivo contenedor. Estas instrucciones incluyen información usada por los servidores de medios para compilación, en base a metadatos desde el gestor de metadatos 150, datos de medios de los bloques fuente de medios y datos FEC de las reservas FEC. El gestor 180 podría generar una instrucción única o conjunto de instrucciones por contenido de medios en el archivo. Alternativamente, tales instrucciones diferentes adaptadas para sobrecargas FEC diferentes, diferentes tipos de datos FEC y/o diferente número de canales de comunicaciones basados en radio

empleados en la sesión de medios se podrían proporcionar por el gestor 180 y organizar en el archivo contenedor.

5 Las unidades 130 a 180 del creador de archivos contenedores 120 pueden ser implementadas o proporcionadas como soporte lógico, componentes físicos o una combinación de los mismos. Las unidades 130 a 180 todos pueden ser implementadas en el creador de archivos contenedores 120. Alternativamente, también es posible una implementación distribuida y está dentro del alcance de la invención. En tal caso, se pueden disponer unidades diferentes 120 a 180 del creador de archivos contenedores 120 en otra parte en el servidor de contenidos 100.

10 La Fig. 17 es un diagrama de bloques esquemático de un servidor de sesiones de medios 200. Este servidor de medios 200 comprende una unidad de I/O 210 para llevar a cabo la comunicación con unidades externas. Esta unidad de I/O 210 se dispone en particular para requerir y recibir contenedores de archivos de medios desde un servidor de contenidos. La unidad de I/O 210 también recibe la petición de contenido de medios que se origina desde clientes usuarios diferentes, o al menos información de a qué clientes se debería transmitir el contenido de medios. Los paquetes de datos compilados por el servidor de medios 200 también se transmiten por la unidad de I/O 210 a estos clientes.

15 El servidor 200 comprende un proveedor de archivos de medios 220 que proporciona un archivo de contenidos de medios a usar en la sesión actual. Este proveedor de archivos 220 puede generar una petición de un archivo contenedor particular que es transmitido a un creador de contenidos por la unidad de I/O 210. Alternativamente, el proveedor 220 trae un archivo contenedor previamente recibido desde un almacenamiento de datos 260 proporcionado en el servidor de medios 200.

20 Un compilador de paquetes de datos 230 usa los metadatos y preferentemente las instrucciones de compilación incluidas en el archivo contenedor del proveedor 220 para extraer datos de medios y datos FEC desde el archivo y genera paquetes de datos que contienen estos datos extraídos. Los paquetes de datos así generados entonces se transmiten por (difundidos en forma continua o descargados desde) la unidad de I/O 210.

25 El compilador de paquetes de datos 230 preferentemente usa también otra información incluida en el archivo (información de algoritmo FEC, información de división/partición, información de tamaño, nombre de contenido, información de almacenamiento de contenidos) en este proceso de compilación. De esta manera, realmente todas las instrucciones y datos requeridos para generar paquetes de datos con datos de medios y datos FEC se proporcionan en el archivo de contenidos, el cual permite una gestión de sesiones de medios flexible y eficiente.

30 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 deberían ser transmitidas determina las instrucciones de compilación reales a usar por el compilador 230. En el último caso, un estimador de capacidad FEC 240 es incluido preferentemente en el servidor 200 para estimar una cantidad máxima de sobrecarga FEC que podría ser empleada durante la sesión. La estimación de la sobrecarga realizada por el estimador 240 se actualiza preferentemente de manera dinámica durante la sesión, ya que la capacidad de sobrecarga podría ser cambiada a través de la sesión. Un conjunto selector 250 usa las estimaciones de capacidad del estimador 240 para seleccionar qué instrucción o conjunto de instrucciones de compilación particulares de aquellas disponibles en el archivo usar. El compilador de paquetes 230 entonces usa esta instrucción (conjunto) para compilar los metadatos y datos FEC en paquetes de datos.

40 El servidor de medios 200 puede comprender opcionalmente un gestor FEC 270 que se puede usar por el servidor de medios 200 en un procedimiento de reparación post sesión. En tal caso, la unidad de I/O 210 recibe una petición de reparación que se origina a partir de un cliente, al cual el servidor 200 previamente ha transmitido datos de medios y FEC. Esta petición se reenvía desde la unidad de I/O 210 al gestor FEC 270. El gestor 270 usa la petición para identificar y extraer los datos (símbolos) de redundancia FEC desde el archivo contenedor de medios, por ejemplo como se almacenaron en el almacenamiento de datos 260. En esta extracción FEC post sesión, el gestor FEC 270 podría emplear instrucciones de reparación post sesión incluidas en el archivo contenedor para identificar los datos FEC correctos y/o para compilar los datos FEC extraídos en paquetes de datos. Los paquetes de datos resultantes entonces se transmiten al cliente solicitante por la unidad de I/O 210.

50 Las unidades 210, 220, 230, 240, 250, y 270 del servidor de medios 200 se pueden implementar o proporcionar como soporte lógico, componentes físicos o una combinación de los mismos. Las unidades 210 a 270 todas pueden ser implementadas, 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, se pueden disponer unidades diferentes 210 a 270 del servidor de medios 200 en nodos de red diferentes pero a pesar de esto realizarán sus operaciones previstas como se describe en lo anteriormente mencionado.

55 La Fig. 18 es un diagrama de bloques esquemático de un servidor de reparación 300. El servidor 300 incluye una unidad de I/O 310 dispuesta para presentar la comunicación con unidades externas. Esta unidad de I/O 310 se proporciona en particular para recibir peticiones de reparación post sesión de clientes y para requerir archivos contenedores de medios desde un creador de contenidos o servidor de sesiones de medios. La transmisión de paquetes de datos que comprende datos de redundancia FEC también se realiza por la unidad de I/O 310.

5 El servidor de reparación 300 comprende un proveedor de archivos de medios 320 que proporciona un archivo contenedor en respuesta a la recepción de una petición de reparación desde un cliente. Este proveedor 320 podría componer una petición de contenedor que se transmite por la unidad de I/O 310 a un creador de contenidos para proporcionar el archivo. Alternativamente, el servidor de reparación 300 ha recibido previamente el archivo contenedor de manera que el proveedor de archivos 320 lo trae de un almacenamiento de datos 340.

10 Un extractor de datos FEC 330 se dispone en el servidor 300 para extraer los datos de redundancia FEC desde el archivo contenedor proporcionado por el proveedor de archivos 320 en base a un identificador incluido en la petición de reparación. Este identificador podría ser un nombre u otro identificador del contenido de medios que no fue recibido con éxito por el cliente. Otros identificadores, tales como el identificador del servidor de sesiones de medios y/o tiempo de entrega del contenido de medios se podrían usar que permiten al servidor de reparación 300, posiblemente por medio de una petición al servidor de medios, identificar el contenido de medios correcto. Información más detallada, tal como la parte particular del contenido de medios (permite la identificación del bloque fuente de medios recibido erróneamente) podría ser empleado alternativamente o en adición.

15 La cantidad de datos FEC extraídos por el extractor de datos 330 se podría predefinir. Si se requieren más símbolos FEC que este nivel predefinido por defecto, el cliente debe notificar expresamente al servidor 300 en consecuencia o el cliente necesita enviar una petición de datos FEC adicionales. Alternativamente, el extractor 330 recibe una estimación desde el cliente de cuántos símbolos FEC se podrían necesitar para la descodificación con éxito de todo el contenido de medios. El extractor 330 entonces usaría esta estimación en la extracción de datos FEC.

20 En una implementación preferente, el archivo contenedor comprende instrucciones de reparación post sesión que definen la compilación de datos FEC desde una reserva FEC en el archivo contenedor en paquetes de datos que se pueden enviar al cliente. En tal caso, el extractor usa estas instrucciones en el proceso de extracción y compilación.

25 Las unidades 310 a 330 del servidor de reparación 300 pueden ser implementadas o proporcionadas como soporte lógico, componentes físicos o una combinación de los mismos. Las unidades 310 a 340 todas pueden ser implementadas en el servidor de reparación 300 en un nodo de red único en un sistema de comunicaciones. Alternativamente, también es posible una implementación distribuida. En tal caso, unidades diferentes 310 y 340 del servidor de reparación 300 se pueden disponer en nodos de red diferentes pero a pesar de esto realizarán sus operaciones previstas como se describió en lo anteriormente mencionado.

Se entenderá por una persona experta en la técnica que se pueden hacer varias modificaciones y cambios a la presente invención sin salirse del alcance de la misma, el cual se define por las reivindicaciones anexas.

30 **Referencias**

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

[2] ISO/IEC 14496-12:2005: "Formato de archivos de medios basados en ISO"

35 [3] ISO/IEC 15444-12:2005: "Formato de archivos de medios basados en ISO"

[4] Solicitud internacional WO 2005/039131

[5] RFC 3695: Esquemas de Corrección de Errores sin Canal de Retorno (FEC) Compactos, febrero de 2004

[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

40 [8] RFC 3926: FLUTE – Entrega de Archivos sobre Transporte Unidireccional, octubre

[9] RFC 3450: Particularización del Protocolo de Codificación de Capas Asíncronas (ALC), diciembre de 2002

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

REIVINDICACIONES

1. Un método de generación de un archivo contenedor de medios (1), dicho método que comprende:
 - organizar al menos un bloque fuente de medios (20; 22; 24) en dicho archivo contenedor de medios (1); y
 - pre calcular, los datos de redundancia de corrección de errores sin canal de retorno, FEC, en base a dicho al menos un bloque fuente de medios (20; 22; 24);
 - organizar dichos datos de redundancia FEC en al menos una reserva FEC (30; 32; 34) en dicho archivo contenedor de medios (1); y
 - proporcionar, en dicho archivo contenedor de medios (1), metadatos (40; 45) que proporcionan una asociación entre cada bloque fuente de medios (20) de dicho al menos un bloque fuente de medios (20; 22; 24) y una reserva FEC respectiva (30) de dicha al menos una reserva FEC (30; 32; 34), **caracterizado por:**
 - generar para una primera pista de índice un primer conjunto de instrucciones de compilación (50) que definen la compilación, en base a dichos metadatos (40), de datos de medios de cada uno de dichos al menos un bloque fuente de datos (20; 22; 24) y datos de redundancia FEC desde cada una de sus reserva FEC asociada para formar una primera secuencia de medios única (80) de paquetes de datos de Entrega de Archivos sobre Transporte Unidireccional, FLUTE, (81, 82, 83, 84) que tiene un primer nivel de sobrecarga de redundancia FEC;
 - generar para una segunda pista de índice que es diferente de la primera pista de índice un segundo conjunto de instrucciones de compilación (52), el cual es diferente del primer conjunto de instrucciones de compilación (50), definiendo la compilación, en base a dichos metadatos (40), de dichos metadatos de medios desde cada uno de dichos al menos un bloque fuente de medios (20; 22; 24) y los datos de redundancia FEC desde cada una de su reserva FEC asociada, reutilizando los mismos símbolos FEC de los datos de redundancia FEC de su reserva FEC asociada usados para formar la primera secuencia de medios única (80), para formar una segunda secuencia de medios única (90) de paquetes de datos FLUTE (91, 92, 93, 94) que tienen un segundo nivel de sobrecarga de redundancia FEC el cual es diferente del primer nivel de sobrecarga de redundancia FEC; y
 - organizar dichas primera pista de índice y segunda pista de índice en dicho archivo contenedor de medios (1).
2. El método de acuerdo con la reivindicación 1, **caracterizado por:**
 - proporcionar un archivo fuente de medios (10); y
 - dividir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24).
3. El método de acuerdo con la reivindicación 2, **caracterizado porque** 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 un algoritmo FEC usado para pre calcular dichos datos de redundancia FEC.
4. El método de acuerdo con cualquiera de las reivindicaciones 1 a 3, **caracterizado porque** dicho paso de cálculo comprende los pasos de:
 - partir dicho al menos un bloque fuente de medios (20; 22; 24) en múltiples símbolos de medios en base a un algoritmo FEC a usar para pre calcular dichos datos de redundancia FEC;
 - generar información de partición de dicha partición del bloque fuente de medios que define qué partes de dicho al menos un bloque fuente de medios (20; 22; 24) que pertenece a qué símbolo de medios de dichos múltiples símbolos de medios; y
 - calcular dichos datos de redundancia FEC en base a al menos un bloque fuente de medios (20; 22; 24) y dicha información de partición.
5. El método de acuerdo con la reivindicación 4, **caracterizado por** proporcionar, en dicho archivo contenedor de medios (1), información de dicho algoritmo FEC y dicha información de partición.
6. El método de acuerdo con cualquiera de las reivindicaciones 1 a 5, **caracterizado por** proporcionar, en dicho archivo contenedor de medios (1), una tabla de propiedades (60) que comprende información de ubicación de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).
7. El método de acuerdo con cualquiera de las reivindicaciones 1 a 6, **caracterizado porque** dichos datos de redundancia FEC son datos de redundancia FEC a ser usados durante una sesión de medios, y dicho método que

además comprende los pasos de:

calcular, en base a dicho al menos un bloque fuente de medios (20; 22; 24), los datos de redundancia FEC post sesión útiles en un procedimiento de reparación post sesión después del final de dicha sesión de medios;

5 organizar dichos datos de redundancia FEC post sesión en al menos una reserva FEC (30; 32; 34) en dicho archivo contenedor de medios (1); y

proporcionar, en dicho archivo contenedor de medios (1), metadatos (45) que permiten la identificación de dicha al menos una reserva FEC (32; 34; 36) en base a un identificador asociado con dicho al menos un bloque fuente de medios (20; 22; 24).

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

10 un gestor de bloques de medios (130) dispuesto para organizar al menos un bloque fuente de medios (20; 22; 24) en un archivo contenedor de medios (1), y

un códec de corrección de errores sin canal de retorno, FEC, (160) dispuesto para pre calcular los datos de redundancia de FEC en base a dicho al menos un bloque fuente de medios (20; 22; 24);

15 un gestor de datos de FEC (140) dispuesto conectado a dicho códec FEC (160) para organizar dichos datos de redundancia de FEC en al menos una reserva FEC (30; 32; 34) en dicho archivo contenedor de medios (1); y

un gestor de metadatos (150) dispuesto para proporcionar, en dicho archivo contenedor de medios (1), metadatos (40; 45) que proporcionan una asociación entre cada bloque fuente de medios (20) del citado al menos un bloque fuente de medios (20; 22; 24) y una reserva FEC respectiva (30) de la citada al menos una reserva FEC (30; 32; 34), **caracterizado por**:

20 un gestor de instrucciones (180) dispuesto para i) generar para una primera pista de índice un primer conjunto de instrucciones de compilación (50) que definen la compilación, en base a dichos metadatos (40) proporcionados por dicho gestor de metadatos (150), de datos de medios de cada uno de dicho al menos un bloque fuente de datos (20; 22; 24) y los datos de redundancia FEC desde cada una de su reserva FEC asociada para formar una primera secuencia de medios única (80) de paquetes de datos de Entrega de Archivos sobre Transporte Unidireccional, FLUTE, (81, 82, 83, 84) que tienen un primer nivel de sobrecarga de redundancia de FEC, ii) generar para una segunda pista de índice la cual es diferente de la primera pista de índice un segundo conjunto de instrucciones de compilación (52) el cual es diferente del primer conjunto de instrucciones de compilación (50), definiendo la compilación, en base a dichos metadatos (40) proporcionados por dicho gestor de metadatos (150), de dichos metadatos de cada uno de dichos al menos un bloque fuente de medios (20; 22; 24) y los datos de redundancia FEC desde cada una de su reserva FEC asociada, reutilizando los mismos símbolos de FEC de los datos de redundancia de FEC de su reserva FEC asociada usada para formar la primera secuencia de medios única (80), para formar una segunda secuencia de medios única (90) de paquetes de datos FLUTE (91, 92, 93, 94) que tienen un segundo nivel de sobrecarga de redundancia FEC, la cual es diferente del primer nivel de sobrecarga de redundancia FEC y iii) organizar dichas primera pista de índice y segunda pista de índice (54) en dicho archivo contenedor de medios (1).

9. El servidor de contenidos de medios de acuerdo con la reivindicación 8, **caracterizado por**:

una fuente de medios (110; 195; 500; 510) dispuesta para proporcionar un archivo fuente de medios (10); y

40 un divisor de archivos (195) dispuesto conectado a dicho servidor de medios (110; 195; 500; 510) para dividir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24).

10. El servidor de contenidos de medios de acuerdo con la reivindicación 9, **caracterizado porque** dicho divisor de archivos (195) está dispuesto para partir dicho archivo fuente de medios (10) en dicho al menos un bloque fuente de medios (20; 22; 24) en base a un algoritmo FEC usado por dicho códec FEC (160) para pre calcular dichos datos de redundancia FEC.

45 11. El servidor de contenidos de medios de acuerdo con cualquiera de las reivindicaciones 8 a 10, **caracterizado por** un particionador de bloques (190) para partir dicho al menos un bloque fuente de medios (20; 22; 24) en múltiples símbolos de medios en base a un algoritmo FEC a ser usado por dicho códec FEC (160) y para generar información de partición de dicho bloque fuente de medios que parte definiendo qué partes de dicho al menos un bloque fuente de medios (20; 22; 24) pertenecen a cada símbolo de medios de dichos múltiples símbolos de medios, en el que dicho códec FEC (160) está dispuesto para pre calcular datos de redundancia FEC en base a dicho al menos un bloque fuente de medios (20; 22 24) y dicha información de partición.

50 12. El servidor de contenidos de medios de acuerdo con la reivindicación 11, **caracterizado por** un gestor de información (170) dispuesto para proporcionar, en dicho archivo contenedor de medios (1), información de dicho algoritmo FEC usado por dicho códec FEC (160) y dicha información de partición.

13. El servidor de contenidos de medios de acuerdo con cualquiera de las reivindicaciones 8 a 12, **caracterizado por** un gestor de información (170) dispuesto para proporcionar, en dicho archivo contenedor de medios (1), una tabla de propiedades (60) que comprende información de ubicación de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).
- 5 14. El servidor de contenidos de medios de acuerdo con cualquiera de las reivindicaciones 8 a 13, **caracterizado porque** dichos datos de redundancia FEC generados por el códec FEC (160) son datos de redundancia FEC que pretenden ser usados durante una sesión de medios, dicho códec FEC (160) está dispuesto para calcular, en base a dicho al menos un bloque fuente de medios (20; 22; 24), los datos de redundancia FEC post sesión útiles en un procedimiento de reparación post sesión después del final de dicha sesión de medios, dicho gestor de datos FEC (140) está dispuesto para organizar dichos datos de redundancia FEC post sesión calculados por dicho códec FEC (160) en al menos una reserva FEC (30; 32; 34) en dicho archivo contenedor de medios (1), y dicho gestor de metadatos (150) está dispuesto para proporcionar, en dicho archivo contenedor de medios (1), metadatos (45) que permiten la identificación de dicha al menos una reserva FEC (30; 32; 34) en base a un identificador asociado con dicho al menos un bloque fuente de medios (20; 22; 24).
- 10 15. Un archivo contenedor de medios (1) que comprende:
- 15 al menos un bloque fuente de medios (20; 22; 24);
- al menos una reserva de corrección de errores sin canal de retorno, FEC, (30; 32; 34) que comprende datos de redundancia FEC pre generados en base a dicho al menos un bloque fuente de medios (20; 22; 24); y
- 20 metadatos (40; 45) que proporcionan una asociación entre cada bloque fuente de medios (20) de dicho al menos un bloque fuente de medios (20; 22; 24) y una reserva FEC respectiva (30) de dicha al menos una reserva FEC (30, 32, 34), **caracterizado por:**
- 25 una primera pista de índice que comprende un primer conjunto de instrucciones de compilación (50) que definen la compilación en base a dichos metadatos (40), de datos de medios de cada uno de dichos al menos un bloque fuente de datos (20; 22; 24) y los datos de redundancia FEC desde cada una de su reserva FEC asociada para formar una primera secuencia de medios (80) de paquetes de datos de Entrega de Archivos sobre Transporte Unidireccional, FLUTE, (81, 82, 83, 84) que tienen un primer nivel de sobrecarga de redundancia FEC; y
- 30 una segunda pista de índice que es diferente de la primera pista de índice, en la que la segunda pista de índice comprende un segundo conjunto de instrucciones de compilación (52), el cual es diferente del primer conjunto de instrucciones de compilación (50), que definen la compilación, en base a dichos metadatos (40), de dichos datos de medios de cada uno de dichos al menos un bloque fuente de medios (20; 22; 24) y los datos de redundancia FEC desde cada una de su reserva FEC asociada, reutilizando los mismos símbolos FEC de los datos de redundancia FEC de su reserva FEC asociada usada para formar la primera secuencia de medios única (80), para formar una segunda secuencia de medios única (90) de paquetes de datos FLUTE (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.
- 35 16. El archivo contenedor de medios de acuerdo con la reivindicación 15, **caracterizado por:**
- 40 la información de partición de una partición de bloque fuente usada para pre calcular, dichos datos de redundancia FEC y que define qué partes de dicho al menos un bloque fuente de medios (20; 22; 24) pertenece a cada símbolo de medios de dichos múltiples símbolos de medios; y
- la información de un algoritmo FEC usado para calcular dichos datos de redundancia FEC.
17. El archivo contenedor de medios de acuerdo con la reivindicación 15 o 16, **caracterizado por** una tabla de propiedades (60) que comprende información de ubicación de almacenamiento de dicho al menos un bloque fuente de medios (20; 22; 24) dentro de dicho archivo contenedor de medios (1).
- 45 18. El archivo contenedor de medios de acuerdo con cualquiera de las reivindicaciones 15 a 17, **caracterizado porque** dichos datos de redundancia FEC son datos de redundancia FEC que se pretenden usar durante una sesión de medios, y dicho archivo contenedor de medios (1) que además comprende:
- 50 al menos una reserva FEC (30; 32; 34) que comprende datos de redundancia FEC post sesión útiles en un procedimiento de reparación post sesión; y
- metadatos (45) que permiten la identificación de dicha al menos una reserva FEC (30; 32; 34) en base a un identificador asociado con dicho al menos un bloque fuente de medios (20; 22; 24).

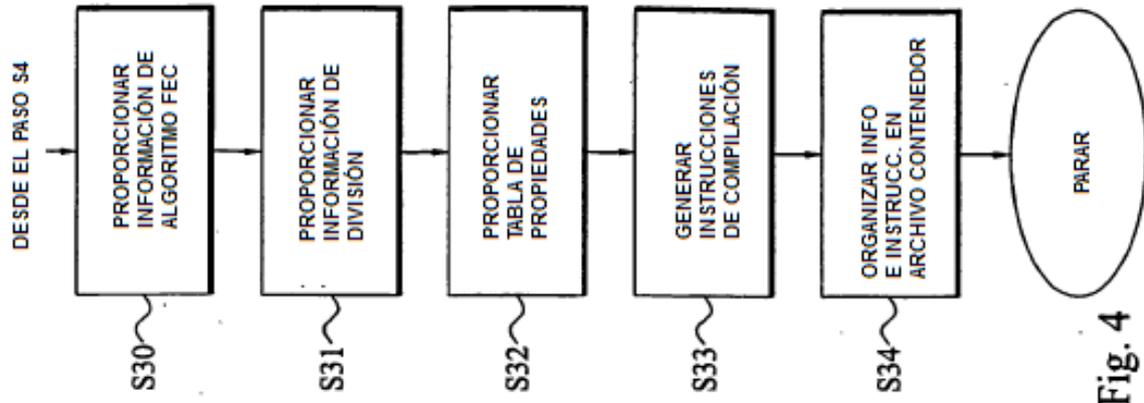


Fig. 4

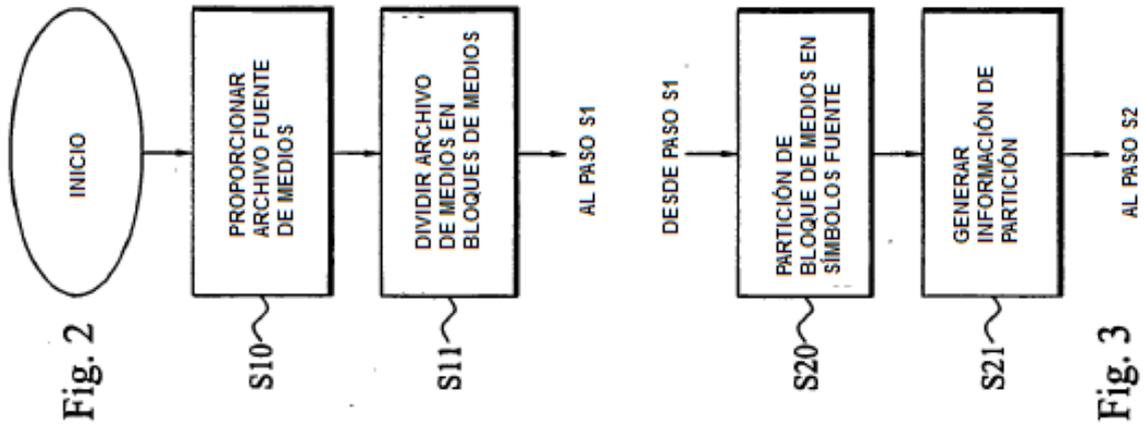


Fig. 2

Fig. 3

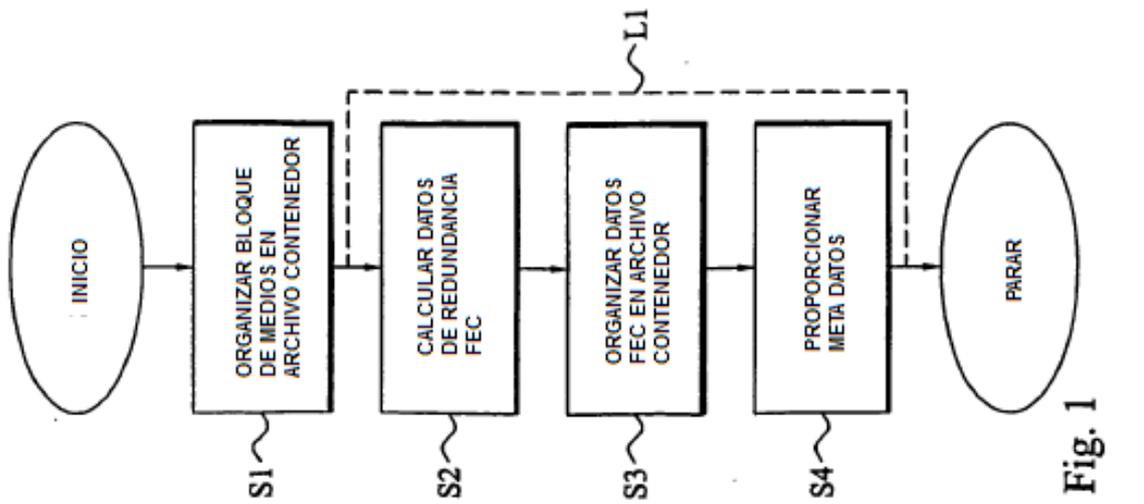


Fig. 1

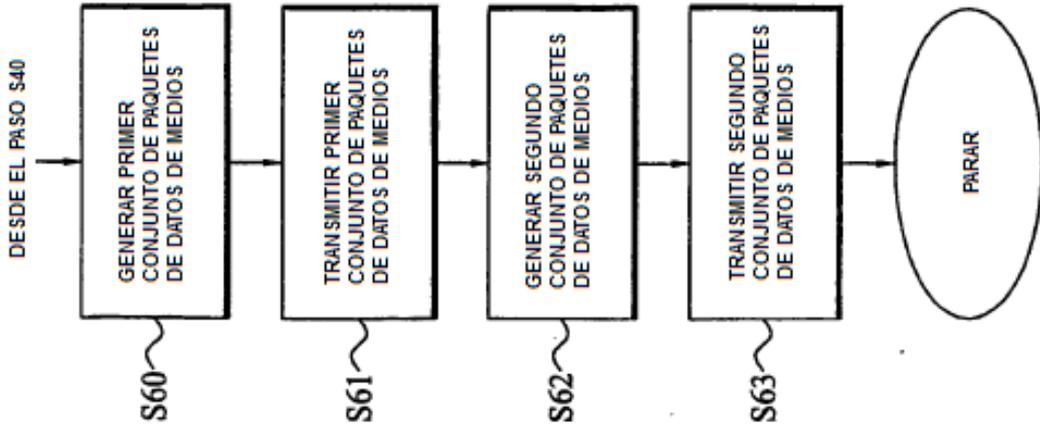


Fig. 7

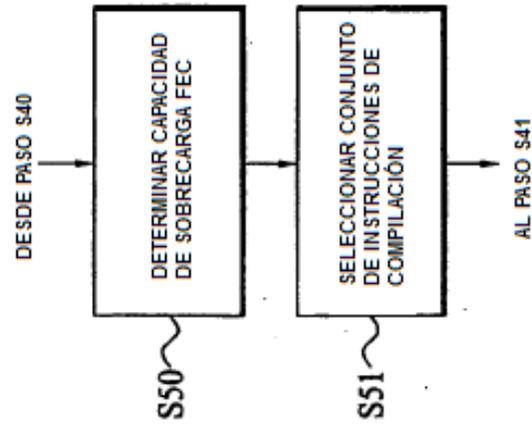


Fig. 6

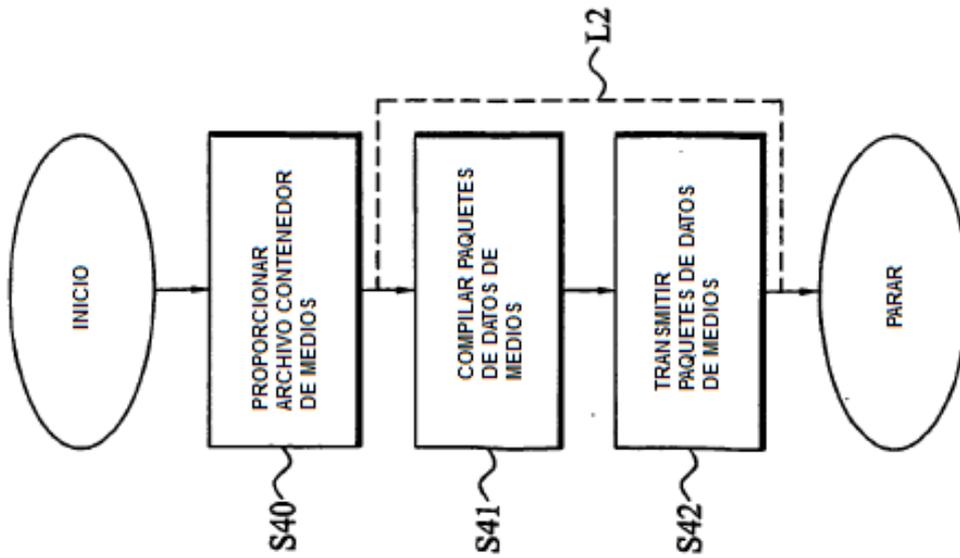


Fig. 5

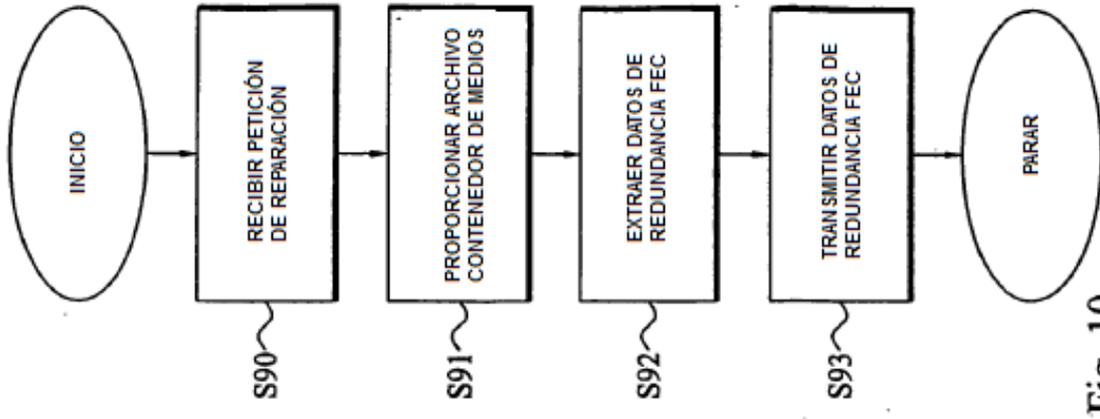


Fig. 10

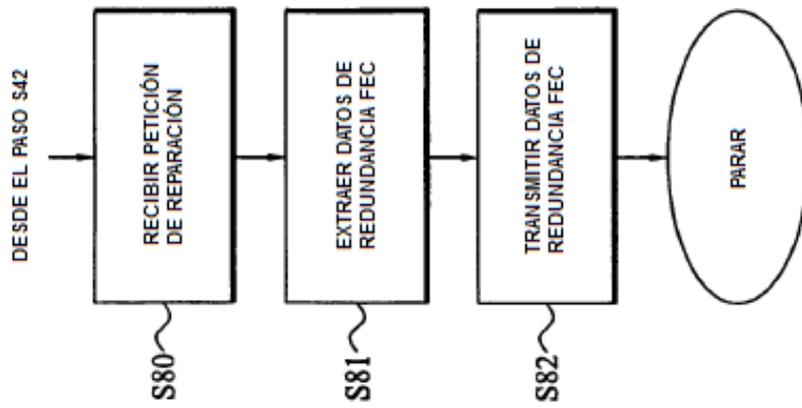


Fig. 9

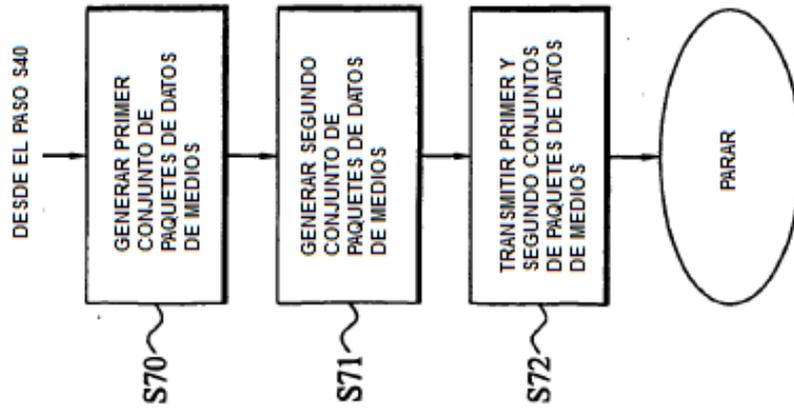


Fig. 8

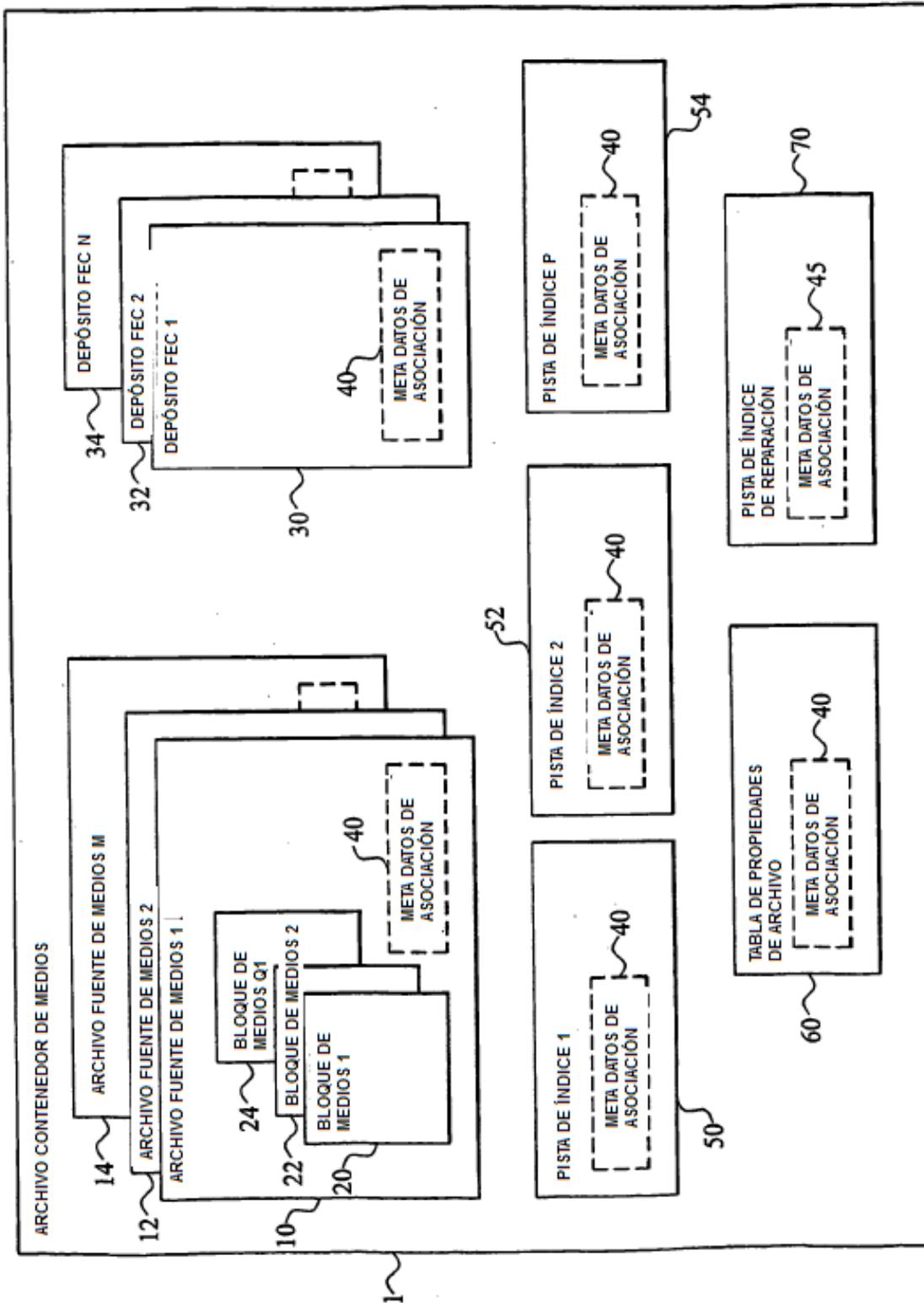


Fig. 11

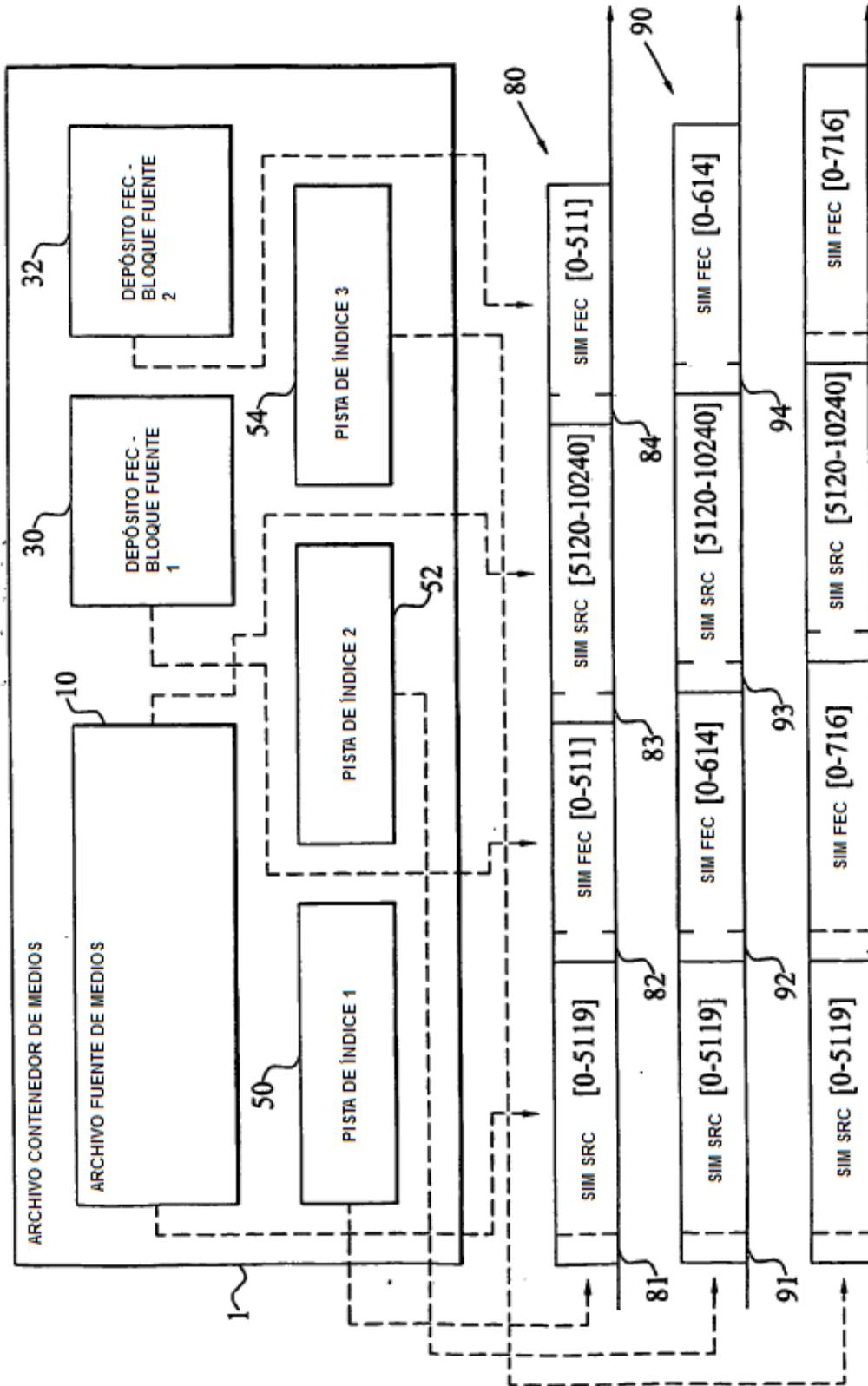


Fig. 12

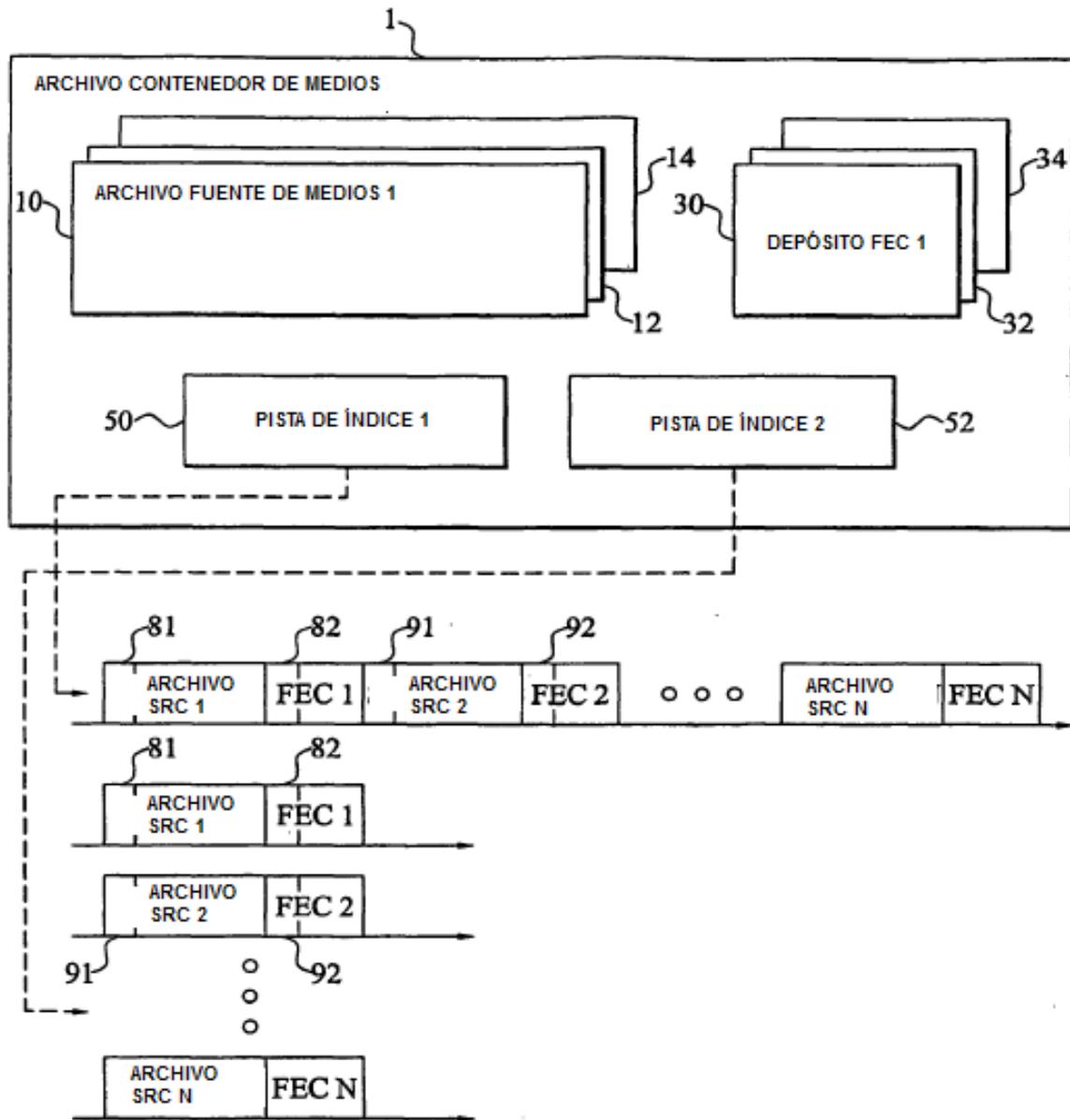


Fig. 13

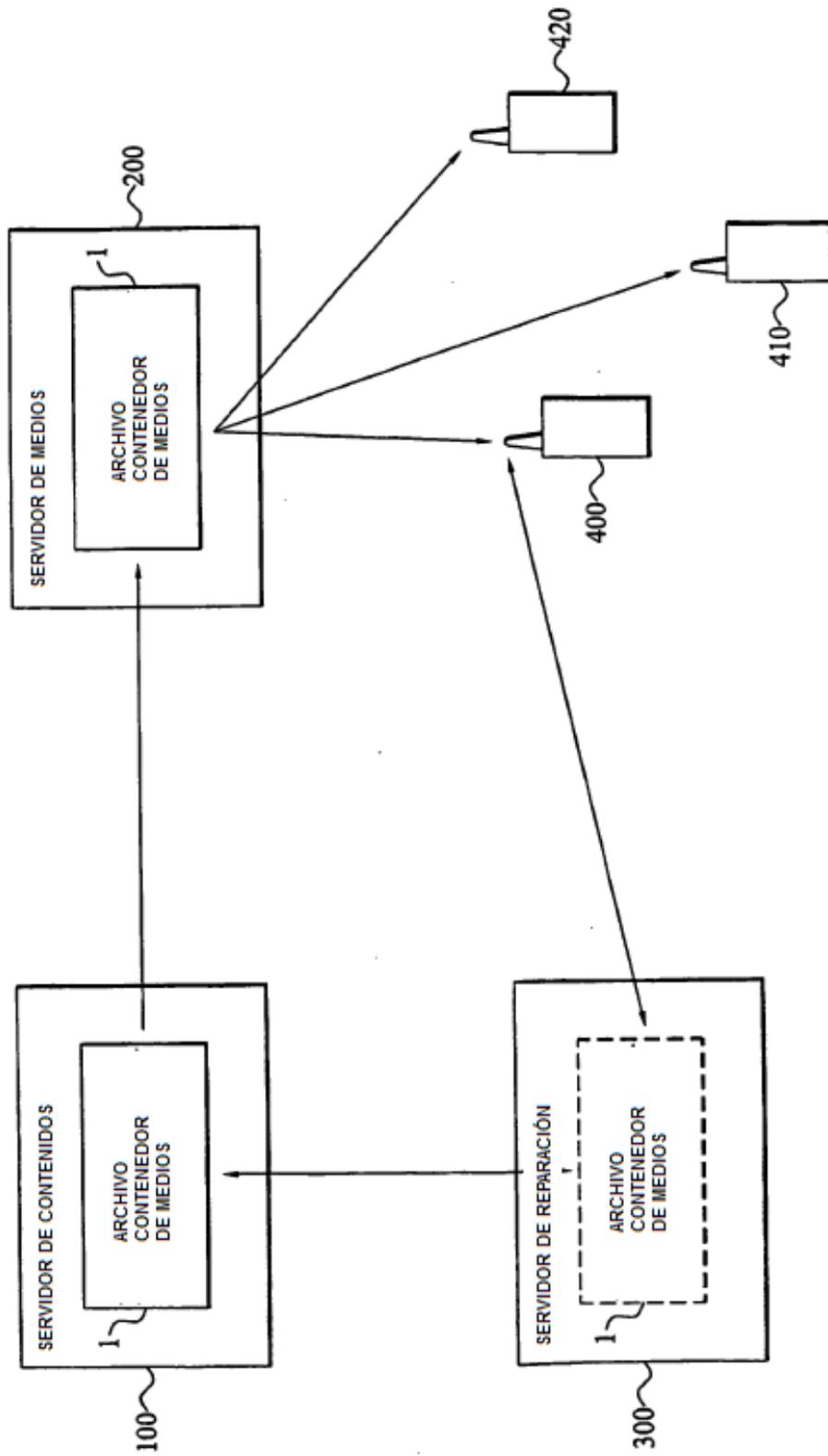


Fig. 14

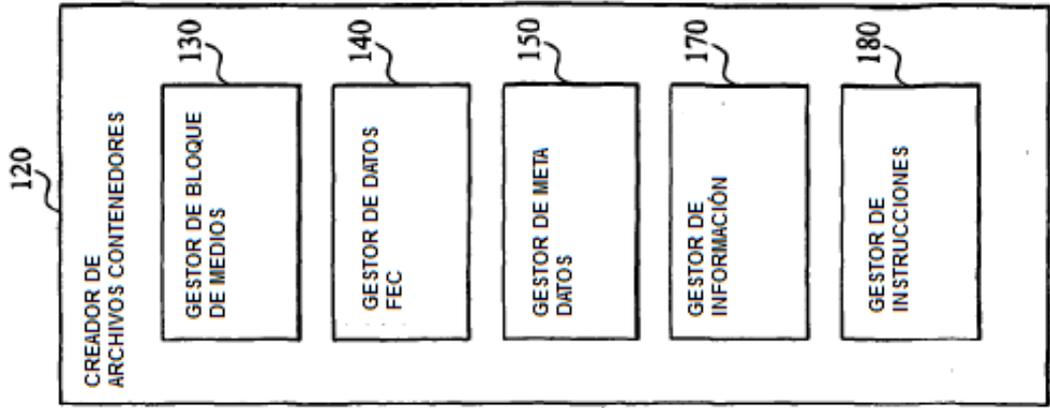


Fig. 16

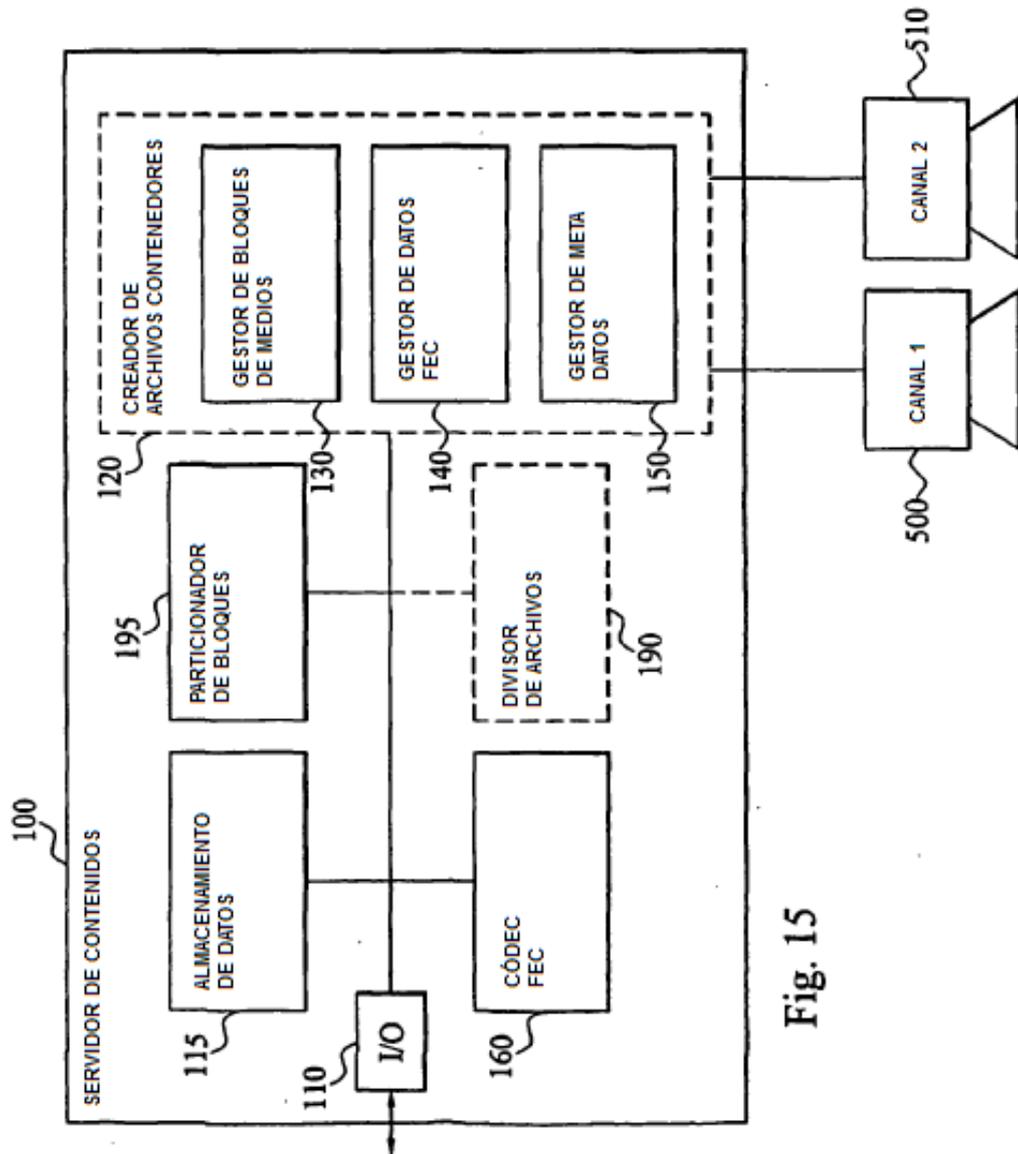


Fig. 15

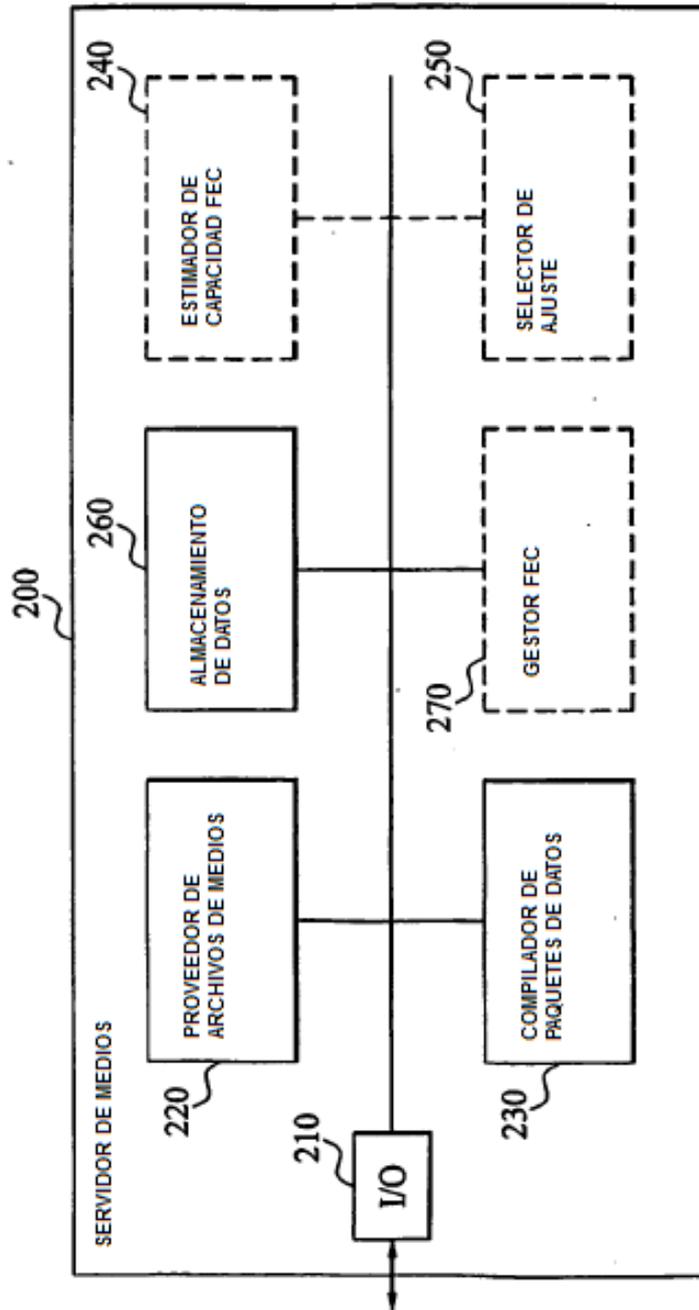


Fig. 17

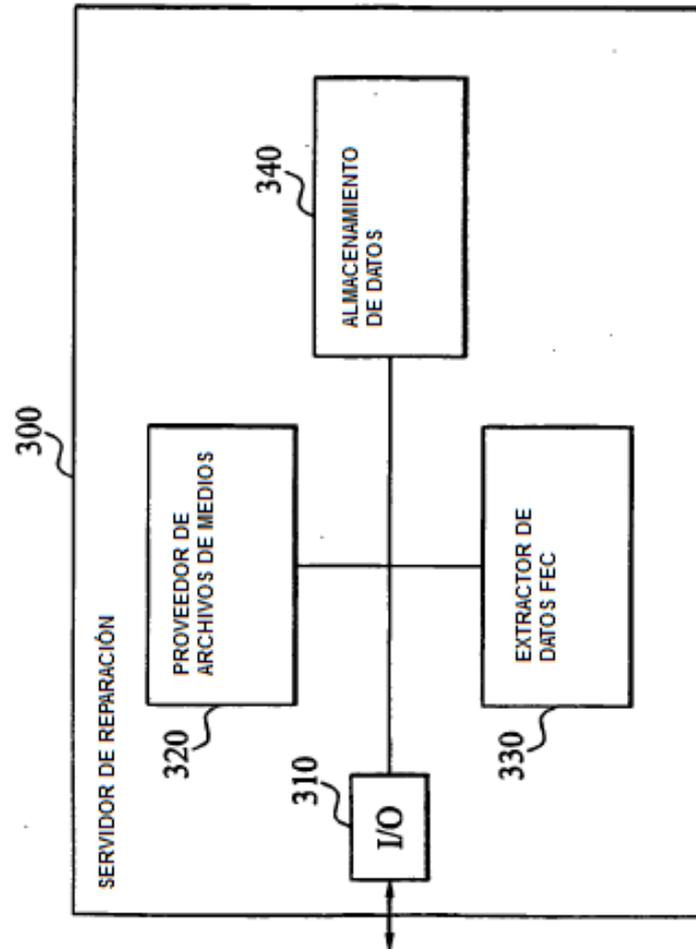


Fig. 18