

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 734 293**

51 Int. Cl.:

<b>H04L 29/06</b>	(2006.01) <b>H04N 21/44</b>	(2011.01)
<b>H04L 29/08</b>	(2006.01) <b>H04N 21/4402</b>	(2011.01)
<b>H04N 21/24</b>	(2011.01) <b>H04N 21/442</b>	(2011.01)
<b>H04N 21/231</b>	(2011.01) <b>H04N 21/84</b>	(2011.01)
<b>H04N 21/234</b>	(2011.01) <b>H04N 21/845</b>	(2011.01)
<b>H04N 21/2343</b>	(2011.01)	
<b>H04N 21/235</b>	(2011.01)	
<b>H04N 21/258</b>	(2011.01)	
<b>H04N 21/2662</b>	(2011.01)	
<b>H04N 21/435</b>	(2011.01)	

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **22.09.2010 PCT/US2010/049862**
- 87 Fecha y número de publicación internacional: **31.03.2011 WO11038028**
- 96 Fecha de presentación y número de la solicitud europea: **22.09.2010 E 10766396 (5)**
- 97 Fecha y número de publicación de la concesión europea: **10.04.2019 EP 2481194**

54 Título: **Transmisión de petición de bloques mejorada mediante partición de bloques o controles de petición para mejorar el manejo del lado del cliente**

30 Prioridad:

**22.09.2009 US 244767 P**  
**03.11.2009 US 257719 P**  
**04.11.2009 US 258088 P**  
**11.12.2009 US 285779 P**  
**20.01.2010 US 296725 P**  
**10.08.2010 US 372399 P**  
**21.09.2010 US 887483**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**05.12.2019**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**  
**International IP Administration**  
**5775 Morehouse Drive**  
**San Diego, CA 92121-1714, US**

72 Inventor/es:

**LUBY, MICHAEL G.;**  
**WATSON, MARK;**  
**VICISANO, LORENZO;**  
**PAKZAD, PAYAM y**  
**WANG, BIN**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 734 293 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

Transmisión de petición de bloques mejorada mediante partición de bloques o controles de petición para mejorar el manejo del lado del cliente

**CAMPO DE LA INVENCION**

**[0001]** La presente invención se refiere a sistemas y procedimientos de transmisión de medios mejorados, más particularmente a sistemas y procedimientos que se adaptan a las condiciones de red y de memoria intermedia para optimizar una presentación de medios transmitidos y permiten una entrega eficiente concurrente o distribuido temporalmente de datos de medios transmitidos.

**ANTECEDENTES DE LA INVENCION**

**[0002]** La entrega de medios de transmisión puede ser cada vez más importante a medida que se hace más común que el audio y el vídeo de alta calidad se entreguen a través de redes basadas en paquetes, como Internet, redes celulares e inalámbricas, redes eléctricas y otros tipos de redes. La calidad con la que se pueden presentar los medios de transmisión entregados puede depender de varios factores, incluida la resolución (u otros atributos) del contenido original, la calidad de codificación del contenido original, las capacidades de los dispositivos receptores para descodificar y presentar los medios, la puntualidad y la calidad de la señal recibida en los receptores, etc. Para crear una buena experiencia de transmisión de medios, el transporte y la puntualidad de la señal recibida en los receptores pueden ser especialmente importantes. Un buen transporte puede proporcionar fidelidad al flujo recibido en el receptor en relación con lo que envía un remitente, mientras que la puntualidad puede representar la rapidez con la que un receptor puede comenzar a reproducir el contenido después de una petición inicial de ese contenido.

**[0003]** Un sistema de entrega de medios se puede caracterizar como un sistema que tiene orígenes de medios, destinos de medios y canales (en tiempo y/o espacio) que separan las fuentes y los destinos. Típicamente, una fuente incluye un transmisor con acceso a medios en forma manejable electrónicamente, y un receptor con capacidad para controlar electrónicamente la recepción de los medios (o una aproximación de los mismos) y proporcionarlos a un consumidor de medios (por ejemplo, un usuario que tenga un dispositivo de visualización acoplado de alguna manera al receptor, a un dispositivo o elemento de almacenamiento, a otro canal, etc.).

**[0004]** Si bien son posibles muchas variaciones, en un ejemplo común, un sistema de entrega de medios tiene uno o más servidores que tienen acceso al contenido de medios en forma electrónica, y uno o más sistemas o dispositivos cliente realizan peticiones de medios a los servidores, y los servidores transmiten los medios que usan un transmisor como parte del servidor, transmitiendo a un receptor en el cliente para que el cliente pueda consumir los medios recibidos de alguna manera. En un ejemplo simple, hay un servidor y un cliente, para una petición y respuesta dadas, pero ese no es el caso necesariamente.

**[0005]** Tradicionalmente, los sistemas de entrega de medios pueden caracterizarse en un modelo de "descarga" o en un modelo de "transmisión". El modelo de "descarga" puede caracterizarse por la independencia de tiempo entre la entrega de los datos de medios y la reproducción de los medios al usuario o dispositivo receptor.

**[0006]** Como ejemplo, los medios se descargan con suficiente anticipación cuando se necesitan o se usarán y cuando se usan, ya que el destinatario ya tiene disponible todo lo necesario. La entrega en el contexto de descarga a menudo se realiza mediante un protocolo de transporte de archivos, como HTTP, FTP o entrega de archivos por transporte unidireccional (FLUTE) y la frecuencia de entrega puede estar determinada por un protocolo de control de flujo y/o congestión subyacente, como TCP/IP. La operación del protocolo de control de flujo o congestión puede ser independiente de la reproducción de los medios al usuario o dispositivo de destino, que puede tener lugar simultáneamente con la descarga o en otro momento.

**[0007]** El modo de "transmisión" se puede caracterizar por un acoplamiento estrecho entre la temporización de la entrega de los datos de medios y la reproducción de los medios al usuario o dispositivo receptor. La entrega en este contexto a menudo se realiza mediante un protocolo de transmisión, como el Protocolo de transmisión en tiempo real (RTSP) para el control y el Protocolo de transporte en tiempo real (RTP) para los datos de medios. La velocidad de entrega puede ser determinada por un servidor de transmisión, que a menudo coincide con la velocidad de reproducción de los datos.

**[0008]** Algunas desventajas del modelo de "descarga" pueden ser que, debido a la independencia de tiempo de entrega y reproducción, es posible que los datos de medios no estén disponibles cuando se necesiten para la reproducción (por ejemplo, debido a que el ancho de banda disponible es menor que la velocidad de datos de medios), lo cual hace que la reproducción se detenga momentáneamente ("detención"), lo cual da como resultado una experiencia de usuario deficiente, o es posible que se requiera que los datos de medios se descarguen mucho antes de la reproducción (por ejemplo, debido a que el ancho de banda disponible es mayor que la velocidad de datos de medios), que consume recursos de almacenamiento en el dispositivo receptor, que pueden ser escasos, y que

consume recursos de red valiosos para la entrega, que pueden desperdiciarse si el contenido, eventualmente, no se reproduce o se utiliza de otra manera.

5 **[0009]** Una ventaja del modelo de "descarga" puede ser que la tecnología necesaria para realizar dichas descargas, por ejemplo HTTP, es muy madura, ampliamente implementada y aplicable en una amplia gama de aplicaciones. Los servidores de descarga y las soluciones para la escalabilidad masiva de tales descargas de archivos (por ejemplo, los servidores de Internet HTTP y las Redes de Entrega de Contenido) pueden estar fácilmente disponibles, haciendo que la implementación de servicios basados en esta tecnología sea simple y de bajo coste.

10 **[0010]** Algunas desventajas del modelo de "transmisión" pueden ser que, en general, la velocidad de entrega de datos de medios no se adapta al ancho de banda disponible en la conexión del servidor al cliente y que se requieren servidores de transmisión especializados o una arquitectura de red más compleja que ofrezca garantías de ancho de banda y retardo. Aunque existen sistemas de transmisión que soportan la variación de la velocidad de datos de entrega de acuerdo con el ancho de banda disponible (por ejemplo, transmisión adaptable de Adobe Flash), estos no son en general tan eficientes como los protocolos de control de flujo de transporte de descarga, como TCP, que utilizan todo el ancho de banda disponible.

20 **[0011]** Recientemente, se han desarrollado e implementado nuevos sistemas de entrega de medios basados en una combinación de los modelos de "transmisión" y "descarga". Un ejemplo de un modelo de este tipo se conoce aquí como un modelo de "transmisión de petición de bloques", en el que un cliente de medios solicita bloques de datos de medios desde la infraestructura de servicio utilizando un protocolo de descarga, como HTTP. Un problema en tales sistemas puede ser la capacidad de comenzar a reproducir un flujo, por ejemplo, descodificar y renderizar los flujos de audio y vídeo recibidos usando un ordenador personal y visualizar el vídeo en la pantalla de un ordenador y reproducir el audio a través de altavoces integrados, o como otro ejemplo, descodificar y renderizar los flujos de audio y vídeo recibidas utilizando un descodificador y visualizar el vídeo en un dispositivo de visualización de televisión y reproducir el audio a través de un sistema estéreo.

30 **[0012]** Hay otros problemas, como la posibilidad de descodificar los bloques de origen lo suficientemente rápido para mantenerse al día con la velocidad de transmisión de origen, para minimizar la latencia de descodificación y para reducir el uso de los recursos de CPU disponibles. Otro problema es proporcionar una solución de entrega de transmisión sólida y escalable que permita que los componentes del sistema fallen sin afectar negativamente la calidad de las transmisiones entregadas a los receptores. Se pueden producir otros problemas basándose en la información que cambia rápidamente sobre una presentación, ya que se está distribuyendo. Por lo tanto, es deseable tener procesos y aparatos mejorados.

35 **[0013]** El documento (US 2009/0150557 A1) analiza un agente de descarga que se ejecuta en un dispositivo informático para seleccionar dinámicamente entre archivos de medios con diferente calidad de medios para la entrega de contenido de medios proporcionado por un proveedor de contenido de medios. El agente de descarga puede seleccionar entre diferentes archivos de medios con contenido similar pero con una calidad diferente basándose en la velocidad de reproducción del archivo de medios, la resolución del archivo de medios o el esquema de codificación del archivo de medios. El agente de descarga puede hacer una transición perfecta de un archivo de medios a otro archivo de medios en las tramas clave para evitar cualquier artefacto de movimiento y evitar que un usuario reinicie el archivo de medios.

45 **[0014]** El documento (EP 1 298 931 A2) se refiere a flujos de medios transmitidos en una red, como Internet. La idea es cambiar la fuente del contenido de un flujo de medios cuando cambia la velocidad de transmisión disponible. Cada fuente contiene esencialmente la misma información (como vídeo y voz), pero la idoneidad de cada fuente para la transmisión se ha ajustado a una cierta velocidad.

50 **[0015]** Otro documento relacionado es el EP 1 471 707 A2.

### **BREVE RESUMEN DE LA INVENCION**

55 **[0016]** La presente invención está definida en las reivindicaciones independientes. Un sistema de transmisión de petición de bloques proporciona mejoras en la experiencia del usuario y en la eficiencia del ancho de banda de tales sistemas, típicamente utilizando un sistema de ingestión que genera datos en una forma para ser servida por un servidor de archivos convencional (HTTP, FTP o similar), en el que el sistema de ingestión ingiere contenido y lo prepara como archivos o elementos de datos que deben ser servidos por el servidor de archivos, que pueden o no incluir una memoria caché. Un dispositivo cliente puede adaptarse para aprovechar el proceso de ingestión, así como incluir mejoras que permitan una mejor presentación independiente del proceso de ingestión. El dispositivo cliente podría configurarse para optimizar el uso de los recursos, dada la información disponible para el mismo desde el sistema de ingestión. Esto puede incluir configuraciones para determinar la secuencia, la temporización y la construcción de peticiones de bloques basadas en técnicas avanzadas de administración de memorias intermedias basadas en la supervisión del tamaño de la memoria intermedia y la tasa de cambio del tamaño de la memoria intermedia.

[0017] En algunos modos de realización, se proporcionan mejoras novedosas a los procedimientos para realizar peticiones de tamaño variable basadas en condiciones tales como el estado de la memoria intermedia. En algunos modos de realización, se proporcionan mejoras novedosas a los procedimientos para la asignación de peticiones de bloques a las conexiones de transporte del protocolo de descarga de archivos subyacentes, incluida la canalización flexible de peticiones. En algunos modos de realización, se proporcionan procedimientos novedosos de selección de servidores, así como el uso de peticiones de archivos completos basadas en consideraciones estadísticas.

[0018] La siguiente descripción detallada junto con los dibujos adjuntos proporcionarán una mejor comprensión de la naturaleza y las ventajas de la presente invención.

## **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

### **[0019]**

La Fig. 1 representa elementos de un sistema de transmisión de petición de bloques de acuerdo modos de realización de la presente invención.

La Fig. 2 ilustra el sistema de transmisión de petición de bloques de la Fig. 1, mostrando mayor detalle en los elementos de un sistema cliente que está acoplado a una infraestructura de servicio de bloques ("BSI") para recibir datos que son procesados por un sistema de ingestión de contenido.

La Fig. 3 ilustra una implementación de hardware/software de un sistema de ingestión.

La Fig. 4 ilustra una implementación de hardware/software de un sistema cliente.

La Fig. 5 ilustra las posibles estructuras del almacén de contenido que se muestra en la Fig. 1, incluidos los segmentos y los archivos descriptores de presentación de medios ("MPD"), y un desglose de segmentos, temporización y otra estructura dentro de un archivo MPD.

La Fig. 6 ilustra detalles de un segmento de origen típico, como podría almacenarse en el almacén de contenido ilustrado en las Figs. 1 y 5.

Las Figs. 7a y 7b ilustran la indexación simple y jerárquica dentro de los archivos.

La Fig. 8(a) ilustra el tamaño del bloque variable con puntos de búsqueda alineados sobre una pluralidad de versiones de un flujo de medios.

La Fig. 8(b) ilustra el tamaño del bloque variable con puntos de búsqueda no alineados sobre una pluralidad de versiones de un flujo de medios.

La Fig. 9(a) ilustra una tabla de metadatos.

La Fig. 9(b) ilustra la transmisión de bloques y tabla de metadatos del servidor al cliente.

La Fig. 10 ilustra bloques que son independientes de los límites RAP.

La Fig. 11 ilustra la temporización continua y discontinua a través de segmentos.

La Fig. 12 es una figura que muestra un aspecto de bloques escalables.

La Fig. 13 muestra una representación gráfica de la evolución de ciertas variables dentro de un sistema de transmisión de petición de bloques a lo largo del tiempo.

La Fig. 14 muestra otra representación gráfica de la evolución de ciertas variables dentro de un sistema de transmisión de petición de bloques a lo largo del tiempo.

La Fig. 15 representa una cuadrícula de estados de una célula en función de los valores de umbral.

La Fig. 16 es un diagrama de flujo de un proceso que podría realizarse en un receptor que puede solicitar bloques únicos y múltiples bloques por petición.

La Fig. 17 es un diagrama de flujo de un proceso de yuxtaposición flexible.

La Fig. 18 ilustra un ejemplo de un conjunto de peticiones candidatas, sus prioridades y en qué conexiones se pueden emitir, en un momento determinado.

La Fig. 19 ilustra un ejemplo de un conjunto de peticiones candidatas, sus prioridades, y en qué conexiones se pueden emitir, que ha evolucionado de un momento a otro.

5 La Fig. 20 es un diagrama de flujo de una selección de proxy de servidor de almacenamiento en memoria caché coherente basada en un identificador de archivo.

La Fig. 21 ilustra una definición de sintaxis para un lenguaje de expresión adecuado.

10 La Fig. 22 ilustra un ejemplo de una función hash adecuada.

La Fig. 23 ilustra ejemplos de reglas de construcción de identificador de archivo.

Las Figs. 24 (a)-(e) ilustran las fluctuaciones de ancho de banda de las conexiones TCP.

15 La Fig. 25 ilustra múltiples peticiones HTTP para datos de origen y reparación.

La Fig. 26 ilustra el tiempo de zapping de ejemplo con y sin FEC.

20 La Fig. 27 ilustra los detalles de un generador de segmentos de reparación que, como parte del sistema de ingestión que se muestra en la Fig. 1, genera segmentos de reparación a partir de los segmentos de origen y los parámetros de control.

La Fig. 28 ilustra las relaciones entre los bloques de origen y los bloques de reparación.

25 La Fig. 29 ilustra un procedimiento para servicios en vivo en diferentes momentos en el cliente.

[0020] En las figuras, los elementos similares se referencian con números similares y los subíndices se proporcionan entre paréntesis para indicar múltiples instancias de elementos similares o idénticos. A menos que se indique lo contrario, el subíndice final (por ejemplo, "N" o "M") no pretende limitarse a ningún valor en particular y el número de instancias de un elemento puede diferir del número de instancias de otro elemento, incluso cuando se ilustra el mismo número y se reutiliza el subíndice.

### **DESCRIPCIÓN DETALLADA DE LA INVENCION**

35 [0021] Como se describe en el presente documento, un objetivo de un sistema de transmisión es mover los medios desde su ubicación de almacenamiento (o la ubicación donde se está generando) a una ubicación donde se está consumiendo, es decir, se está presentando a un usuario o, de lo contrario, "es usada" por un consumidor humano o electrónico. Idealmente, el sistema de transmisión puede proporcionar reproducción ininterrumpida (o más en general, "consumo" ininterrumpido) en un extremo receptor y puede comenzar a reproducir un flujo o una colección de flujos poco después de que un usuario haya solicitado el flujo o los flujos. Por razones de eficiencia, también es deseable que cada flujo se detenga una vez que el usuario indique que ya no es necesario, por ejemplo, cuando el usuario está cambiando de un flujo a otro u obedece a la presentación de un flujo, por ejemplo, flujo de "subtítulos". Si el componente de medios, como el vídeo, se continúa presentando, pero se selecciona un flujo diferente para presentar este componente de medios, a menudo se prefiere ocupar un ancho de banda limitado con el nuevo flujo y detener el flujo antiguo.

50 [0022] Un sistema de transmisión de petición de bloques de acuerdo con los modos de realización descritos en el presente documento proporciona muchos beneficios. Debe entenderse que un sistema viable no necesita incluir todas las características descritas en el presente documento, ya que algunas aplicaciones pueden proporcionar una experiencia satisfactoria con menos de todas las características descritas en el presente documento.

### **Transmisión HTTP**

55 [0023] La transmisión HTTP es un tipo específico de transmisión. Con la transmisión HTTP, las fuentes pueden ser servidores de Internet estándar y redes de distribución de contenido (CDN) y pueden usar la norma HTTP. Esta técnica puede involucrar la segmentación de flujos y el uso de múltiples flujos, todo dentro del contexto de peticiones HTTP estandarizadas. Los medios, como el vídeo, pueden codificarse en múltiples velocidades de transmisión de bits para formar diferentes versiones o representaciones. Los términos "versión" y "representación" se usan como sinónimos en este documento. Cada versión o representación se puede dividir en partes más pequeñas, quizá del orden de unos pocos segundos cada una, para formar segmentos. Cada segmento puede almacenarse en un servidor de Internet o CDN como un archivo separado.

65 [0024] En el lado del cliente, se pueden realizar peticiones, utilizando HTTP, para segmentos individuales que el cliente empalma a la perfección. El cliente puede cambiar a diferentes velocidades de datos basándose en el ancho de banda disponible. El cliente también puede solicitar múltiples representaciones, cada una presentando un componente de medios diferente, y puede presentar los medios en estas representaciones de manera conjunta y

síncrona. Los activadores para el cambio pueden incluir la ocupación de la memoria intermedia y las mediciones de red, por ejemplo. Cuando se opera en estado estable, el cliente puede pasar las peticiones al servidor para mantener una ocupación de la memoria intermedia objetivo.

5 **[0025]** Las ventajas de la transmisión HTTP pueden incluir la adaptación de la velocidad de transmisión de bits, la búsqueda y el inicio rápidos y una entrega mínima innecesaria. Estas ventajas provienen de controlar que la entrega esté solo un poco antes de la reproducción, aprovechar al máximo el ancho de banda disponible (a través de medios de velocidad de transmisión de bits variable) y optimizar la segmentación de flujos y los procedimientos de clientes inteligentes.

10 **[0026]** Se puede proporcionar una descripción de presentación de medios a un cliente de transmisión HTTP, de modo que el cliente pueda usar una colección de archivos (por ejemplo, en formatos especificados por 3GPP, aquí llamados segmentos de 3 gp) para proporcionar un servicio de transmisión al usuario. Una descripción de presentación de medios, y posiblemente las actualizaciones de esta descripción de presentación de medios, describen una presentación de medios que es una colección estructurada de segmentos, cada uno de los cuales contiene componentes de medios de manera que el cliente puede presentar los medios incluidos de manera sincronizada y puede proporcionar características avanzadas, tales como como búsqueda, cambio de velocidades de transmisión de bits y presentación conjunta de componentes de medios en diferentes representaciones. El cliente puede usar la información de la descripción de presentación de medios de diferentes maneras para la provisión del servicio. En particular, a partir de la descripción de presentación de medios, el cliente de transmisión HTTP puede determinar a qué segmentos de la colección se puede acceder para que los datos sean útiles para la capacidad del cliente y para el usuario dentro del servicio de transmisión.

25 **[0027]** En algunos modos de realización, la descripción de presentación de los medios puede ser estática, aunque los segmentos pueden crearse dinámicamente. La descripción de presentación de los medios puede ser lo más compacta posible para minimizar el tiempo de acceso y descarga del servicio. Otra conectividad del servidor dedicado puede minimizarse, por ejemplo, la sincronización de tiempo regular o frecuente entre el cliente y el servidor.

30 **[0028]** La presentación de los medios puede construirse para permitir el acceso de los terminales con diferentes capacidades, como el acceso a diferentes tipos de redes de acceso, diferentes condiciones de red actuales, tamaños de pantalla, velocidades de bits de acceso y soporte de códec. El cliente puede a continuación extraer la información apropiada para proporcionar el servicio de transmisión al usuario.

35 **[0029]** La descripción de presentación de los medios también puede permitir compacidad y flexibilidad de despliegue de acuerdo con los requisitos.

40 **[0030]** En el caso más simple, cada Representación alternativa puede almacenarse en un solo archivo 3GP, es decir, un archivo conforme a lo definido en 3GPP TS26.244, o cualquier otro archivo que se ajuste al formato de archivo de medios de base ISO como se define en ISO/IEC 14496-12 o especificaciones derivadas (como el formato de archivo 3GP descrito en la especificación 26 244 de 3GPP). En el resto de este documento, al referirse a un archivo 3GP, debe entenderse que ISO/IEC 14496-12 y las especificaciones derivadas pueden asignar todas las características descritas al formato de archivo de medios de base ISO más general definido en ISO/IEC 14496-12 o cualesquiera especificaciones derivadas. A continuación, el cliente puede solicitar una parte inicial del archivo para conocer los metadatos de medios (que típicamente se almacenan en la ventana de cabecera de película, también denominada ventana "moov") junto con los tiempos de los fragmentos de película y las desviaciones de bytes. A continuación, el cliente puede emitir peticiones de obtención de HTTP parcial para obtener fragmentos de películas según sea necesario.

50 **[0031]** En algunos modos de realización, puede ser deseable dividir cada representación en varios segmentos, donde los segmentos. En caso de que el formato del segmento se base en el formato de archivo 3GP, entonces los segmentos contienen fragmentos de tiempo no superpuestos de los fragmentos de película, llamados "división temporal". Cada uno de estos segmentos puede contener múltiples fragmentos de películas y cada uno puede ser un archivo 3GP válido por derecho propio. En otro modo de realización, la representación se divide en un segmento inicial que contiene los metadatos (típicamente la ventana "moov" de la cabecera de película) y un conjunto de segmentos de medios, cada uno de los cuales contiene datos de medios y la concatenación del segmento inicial y cualquier segmento de medios constituye un archivo 3GP válido, así como la concatenación del segmento inicial y todos los segmentos de medios de una representación forman un archivo 3GP válido. La presentación completa se puede formar reproduciendo cada segmento de forma sucesiva, asignando las marcas de tiempo locales dentro del archivo al tiempo de presentación global de acuerdo con el tiempo de inicio de cada representación.

60 **[0032]** Se debe tener en cuenta que, a lo largo de esta descripción, debe entenderse que las referencias a un "segmento" incluyen cualquier objeto de datos que esté total o parcialmente construido o leído a partir de un medio de almacenamiento o que se obtenga de otra manera como resultado de una petición de protocolo de descarga de archivos, incluyendo por ejemplo una petición HTTP. Por ejemplo, en el caso de HTTP, los objetos de datos pueden almacenarse en archivos reales que residen en un disco u otro medio de almacenamiento conectado o que forma parte de un servidor HTTP, o los objetos de datos pueden construirse mediante un script CGI u otro programa

ejecutado dinámicamente, que se ejecuta en respuesta a la petición HTTP. Los términos "archivo" y "segmento" se usan como sinónimos en este documento a menos que se especifique lo contrario. En el caso de HTTP, el segmento puede considerarse como el cuerpo de la entidad de una respuesta de petición HTTP.

5 **[0033]** Los términos "presentación" y "elemento de contenido" se usan como sinónimos en este documento. En muchos ejemplos, la presentación es una presentación de audio, vídeo u otros medios que tiene un tiempo de "reproducción" definido, pero son posibles otras variaciones.

10 **[0034]** Los términos "bloque" y "fragmento" se usan como sinónimos en este documento, a menos que se especifique lo contrario, y en general se refieren a la agregación más pequeña de datos indexados. Basándose en la indexación disponible, un cliente puede solicitar diferentes porciones de un fragmento en diferentes peticiones HTTP, o puede solicitar uno o más fragmentos o porciones de fragmentos consecutivos en una petición HTTP. En el caso de que se utilicen segmentos basados en el formato de archivo de medios ISO o segmentos basados en el formato de archivo 3GP, un fragmento típicamente se refiere a un fragmento de película definido como la combinación de una ventana de cabecera de fragmento de película ("moof") y una ventana de datos de medios ("mdat").

15 **[0035]** En el presente documento, se supone que una red que transporta datos está basada en paquetes para simplificar las descripciones del presente documento, con el reconocimiento de que, después de leer esta divulgación, un experto en la técnica puede aplicar modos de realización de la presente invención descritos en el presente documento a otros tipos de redes de transmisión, tales como redes de flujos de bits continuos.

20 **[0036]** En el presente documento, se supone que los códigos FEC brindan protección contra tiempos de entrega de datos largos y variables, con el fin de simplificar las descripciones en el presente documento, con el reconocimiento de que, después de leer esta divulgación, un experto en la técnica puede aplicar modos de realización de la presente invención a otros tipos de problemas de transmisión de datos, como la corrupción de los datos con un cambio de bit. Por ejemplo, sin FEC, si la última parte de un fragmento solicitado llega mucho más tarde o tiene una varianza más alta en su tiempo de llegada que las partes anteriores del fragmento, entonces el tiempo de zapping de contenido puede ser grande y variable, mientras que utilizando FEC y peticiones paralelas, solo la mayoría de los datos solicitados para un fragmento deben llegar antes de poder recuperarse, reduciendo de este modo el tiempo de zapping de contenido y la variabilidad en el tiempo de zapping de contenido. En esta descripción, se puede suponer que los datos a codificar (es decir, los datos de origen) se han dividido en "símbolos" de igual longitud, que pueden ser de cualquier longitud (hasta un solo bit), pero los símbolos podrían ser de diferente longitudes para diferentes partes de los datos, por ejemplo, diferentes tamaños de símbolos podrían usarse para diferentes bloques de datos.

25 **[0037]** En esta descripción, para simplificar las descripciones en el presente documento, se supone que la FEC se aplica a un "bloque" o "fragmento" de datos a la vez, es decir, un "bloque" es un "bloque de origen" para la codificación FEC y con fines de decodificación. Un dispositivo cliente puede usar la indexación de segmentos descrita en el presente documento para ayudar a determinar la estructura de bloque de origen de un segmento. Un experto en la técnica puede aplicar modos de realización de la presente invención a otros tipos de estructuras de bloques de origen, por ejemplo, un bloque de origen puede ser una parte de un fragmento, o abarcar uno o más fragmentos o partes de fragmentos.

30 **[0038]** Los códigos FEC considerados para su uso con la transmisión de petición de bloques son típicamente códigos FEC sistemáticos, es decir, los símbolos de origen del bloque de origen pueden incluirse como parte de la codificación del bloque de origen y, por lo tanto, se transmiten los símbolos de origen. Como reconocerá un experto en la técnica los modos de realización descritos en el presente documento se aplican igualmente bien a los códigos FEC que no son sistemáticos. Un codificador FEC sistemático genera, a partir de un bloque de origen de símbolos de origen, un número de símbolos de reparación y la combinación de al menos algunos de los símbolos de origen y reparación son los símbolos codificados que se envían a través del canal que representa el bloque de origen. Algunos códigos FEC pueden ser útiles para generar de manera eficiente tantos símbolos de reparación como sean necesarios, tales como "códigos de aditivos de información" o "códigos fuente" y entre los ejemplos de estos códigos incluyen "códigos de reacción en secuencia" y "códigos de reacción en secuencia de múltiples etapas". Otros códigos FEC, como los códigos Reed-Solomon, pueden generar prácticamente solo un número limitado de símbolos de reparación para cada bloque de origen.

35 **[0039]** En muchos de estos ejemplos se supone que un cliente está acoplado a un servidor de medios o una pluralidad de servidores de medios y el cliente solicita la transmisión de medios a través de un canal o una pluralidad de canales desde el servidor de medios o la pluralidad de servidores de medios. Sin embargo, también son posibles disposiciones más involucradas.

40 **Ejemplos de beneficios**

45 **[0040]** Con la transmisión de peticiones de bloque, el cliente de medios mantiene un acoplamiento entre la temporización de estas peticiones de bloque y la temporización de reproducción de medios para el usuario. Este modelo puede conservar las ventajas del modelo de "descarga" descrito anteriormente, evitando al mismo tiempo algunas de las desventajas que se derivan del desacoplamiento habitual de la reproducción de medios de la entrega

de datos. El modelo de transmisión de petición de bloques utiliza los mecanismos de control de congestión y velocidad disponibles en los protocolos de transporte, como TCP, para garantizar que se utiliza el ancho de banda máximo disponible para los datos de medios. Además, la división de la presentación de medios en bloques permite que cada bloque de datos de medios codificados sea seleccionado de un conjunto de múltiples codificaciones disponibles.

**[0041]** Esta selección puede basarse en cualquier número de criterios, incluida la adaptación de la velocidad de datos de medios con el ancho de banda disponible, incluso cuando el ancho de banda disponible cambia con el tiempo, la adaptación de la resolución de los medios o la complejidad de decodificación con las capacidades o configuración del cliente, o la adaptación a las preferencias del usuario, como idiomas. La selección también puede incluir la descarga y presentación de componentes auxiliares, como componentes de accesibilidad, pies de página cerrados, subtítulos, vídeo en lenguaje de señas, etc. Algunos ejemplos de sistemas existentes que utilizan el modelo de transmisión de petición de bloques incluyen Move Networks™, Microsoft Smooth Streaming y el protocolo de transmisión de Apple iPhone™.

**[0042]** Comúnmente, cada bloque de datos de medios puede almacenarse en un servidor como un archivo individual y a continuación se usa un protocolo, como HTTP, junto con el software de servidor HTTP ejecutado en el servidor, para solicitar el archivo como una unidad. Típicamente, al cliente se le proporcionan archivos de metadatos, que pueden estar, por ejemplo, en formato de lenguaje de marcado extensible (XML) o en formato de texto de lista de reproducción o en formato binario, que describen características de la presentación de medios, como las codificaciones disponibles (por ejemplo, el ancho de banda requerido, las resoluciones, los parámetros de codificación, el tipo de medios, el idioma), que típicamente se denominan "representaciones" en este documento, y la manera en que las codificaciones se han dividido en bloques. Por ejemplo, los metadatos pueden incluir un localizador uniforme de recursos (URL) para cada bloque. Las direcciones URL en sí mismas pueden proporcionar un esquema como el que se añade con la secuencia "http://" para indicar que el protocolo que se utilizará para acceder al recurso documentado es HTTP. Otro ejemplo es "ftp://" para indicar que el protocolo que se va a usar es FTP.

**[0043]** En otros sistemas, por ejemplo, los bloques de medios pueden ser construidos "sobre la marcha" por el servidor en respuesta a una petición del cliente que indica la parte de la presentación de medios, a tiempo, que se solicita. Por ejemplo, en el caso de HTTP con el esquema "http://", la ejecución de la petición de esta URL proporciona una respuesta de petición que contiene algunos datos específicos en el cuerpo de la entidad de esta respuesta de petición. La implementación en la red sobre cómo generar esta respuesta de petición puede ser bastante diferente, dependiendo de la implementación del servidor que atiende dichas peticiones.

**[0044]** Típicamente, cada bloque puede ser decodificable independientemente. Por ejemplo, en el caso de los medios de vídeo, cada bloque puede comenzar con un "punto de búsqueda". En algunos esquemas de codificación, un punto de búsqueda se conoce como "Puntos de acceso aleatorio" o "RAP", aunque no todos los RAP se pueden designar como un punto de búsqueda. De manera similar, en otros esquemas de codificación, un punto de búsqueda comienza en una trama de "Actualización de datos independiente", o "IDR", en el caso de la codificación de vídeo H.264, aunque no todas las IDR pueden designarse como un punto de búsqueda. Un punto de búsqueda es una posición en el medio de vídeo (u otro) donde un decodificador puede comenzar a decodificar sin requerir ningún dato sobre tramas o datos o muestras anteriores, como podría ser el caso en el que una trama o muestra que se está decodificando no fuera codificada en un modo independiente, sino como, por ejemplo, la diferencia entre la trama actual y la trama anterior.

**[0045]** Un problema en tales sistemas puede ser la capacidad de comenzar a reproducir un flujo, por ejemplo, decodificar y renderizar los flujos de audio y vídeo recibidos usando un ordenador personal y visualizar el vídeo en la pantalla de un ordenador y reproducir el audio a través de altavoces integrados, o como otro ejemplo, decodificar y renderizar los flujos de audio y vídeo recibidas utilizando un decodificador y visualizar el vídeo en un dispositivo de visualización de televisión y reproducir el audio a través de un sistema estéreo. Un problema principal puede ser minimizar el retardo entre el momento en que un usuario decide ver un nuevo contenido entregado como un flujo y toma una acción que expresa esa decisión; por ejemplo, el usuario hace clic en un enlace dentro de una ventana del navegador o en el botón de reproducción de un dispositivo de control remoto, y cuando el contenido comienza a mostrarse en la pantalla del usuario, en lo sucesivo denominado el "tiempo de zapping de contenido". Cada uno de estos problemas puede abordarse mediante elementos del sistema mejorado descritos en el presente documento.

**[0046]** Un ejemplo de zapping de contenido es cuando un usuario está viendo un primer contenido entregado a través de un primer flujo y a continuación el usuario decide ver un segundo contenido enviado a través de un segundo flujo e inicia una acción para comenzar a ver el segundo contenido. La segunda transmisión puede enviarse desde el mismo conjunto o desde un conjunto diferente de servidores que la primera transmisión. Otro ejemplo del zapping de contenido es cuando un usuario visita un sitio web y decide comenzar a ver un primer contenido enviado a través de una primera transmisión haciendo clic en un enlace dentro de la ventana del navegador. De manera similar, un usuario puede decidir comenzar a reproducir el contenido no desde el principio, sino desde algún momento dentro del flujo. El usuario le indica a su dispositivo cliente que busque una posición de tiempo y el usuario puede esperar que el tiempo seleccionado se renderice instantáneamente. Minimizar el tiempo de zapping de contenido es importante para la visualización de vídeos, ya que permite a los usuarios una experiencia de navegación de contenido rápida de alta calidad al buscar y muestrear una amplia gama de contenidos disponibles.

- 5 **[0047]** Recientemente, se ha convertido en una práctica común considerar el uso de códigos de corrección de errores hacia adelante (FEC) para la protección de los medios de transmisión durante la transmisión. Cuando se envían a través de una red de paquetes, entre los que se incluyen Internet y redes inalámbricas como las estandarizadas por grupos como 3GPP, 3GPP2 y DVB, el flujo de origen se coloca en paquetes a medida que se genera o se pone a disposición, y por lo tanto los paquetes pueden utilizarse para transportar el flujo de contenido u origen en el orden en que se genera o se pone a disposición de los receptores.
- 10 **[0048]** En una aplicación típica de códigos FEC para estos tipos de escenarios, un codificador puede usar el código FEC en la creación de paquetes de reparación, que a continuación se envían además de los paquetes fuente originales que contienen el flujo de origen. Los paquetes de reparación tienen una propiedad que, cuando se produce la pérdida del paquete de origen, los paquetes de reparación recibidos pueden usarse para recuperar los datos contenidos en los paquetes de origen perdidos. Los paquetes de reparación se pueden usar para recuperar el contenido de los paquetes de origen perdidos que se pierden por completo, pero también se pueden usar para recuperarse de la pérdida parcial de paquetes, ya sea paquetes de reparación recibidos en su totalidad o incluso paquetes de reparación recibidos parcialmente. Por lo tanto, los paquetes de reparación recibidos total o parcialmente se pueden utilizar para recuperar paquetes de origen perdidos total o parcialmente.
- 15 **[0049]** En otros ejemplos más, se pueden producir otros tipos de corrupción en los datos enviados, por ejemplo, los valores de los bits pueden invertirse y, por lo tanto, los paquetes de reparación pueden usarse para corregir dicha corrupción y proporcionar la recuperación más precisa posible de los paquetes de origen. En otros ejemplos, el flujo de origen no se envía necesariamente en paquetes discretos, sino que se puede enviar, por ejemplo, como un flujo de bits continuo.
- 20 **[0050]** Hay muchos ejemplos de códigos FEC que se pueden usar para brindar protección a un flujo de origen. Los códigos Reed-Solomon son códigos bien conocidos para la corrección de errores y borrado en sistemas de comunicación. Para la corrección de borrado en, por ejemplo, las redes de paquetes de datos, una implementación eficiente y bien conocida de los códigos Reed-Solomon utiliza matrices de Cauchy o Vandermonde como se describe en L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", Computer Communication Review 27(2): 24-36 (April 1997) ["Códigos de borrado efectivos para protocolos de comunicación de ordenador fiables", Revisión de comunicación por ordenador 27 (2): 24-36 (abril de 1997)], (en lo sucesivo, "Rizzo") y Bloemer, et al., "An XOR- Based Erasure-Resilient Coding Scheme", Technical Report TR-95-48 ["Un esquema de codificación resistente al borrado basado en XOR", Informe técnico TR-95-48], International Computer Science Institute, Berkeley, California (1995) (en adelante "XOR-Reed-Solomon") o en cualquier otro lugar.
- 25 **[0051]** Entre otros ejemplos de códigos FEC se incluyen códigos LDPC, códigos de reacción en secuencia como los descritos en la patente de Estados Unidos n.º 6,307,487 y códigos de reacción en secuencia de múltiples etapas, como en la patente de Estados Unidos n.º 7,068,729.
- 30 **[0052]** Los ejemplos del proceso de descodificación FEC para variantes de códigos Reed-Solomon se describen en Rizzo y XOR-Reed-Solomon. En esos ejemplos, la descodificación puede aplicarse después de que se hayan recibido suficientes paquetes de datos de origen y reparación. El proceso de descodificación puede ser computacionalmente intensivo y, dependiendo de los recursos de CPU disponibles, esto puede tardar un tiempo considerable en completarse, en relación con el tiempo que abarca el medio en el bloque. El receptor puede tener en cuenta el tiempo requerido para la descodificación al calcular el retardo requerido entre el inicio de la recepción del flujo de medios y la reproducción de los medios. Este retardo debido a la descodificación es percibido por el usuario como un retardo entre su petición de un flujo de medios en particular y el inicio de la reproducción. Por lo tanto, es deseable minimizar este retardo.
- 35 **[0053]** En muchas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).
- 40 **[0054]** Algunos códigos FEC pueden estar basados en bloques, ya que las operaciones de codificación dependen de los símbolos que están en un bloque y pueden ser independientes de los símbolos que no están en ese bloque. Con la codificación basada en bloques, un codificador FEC puede generar símbolos de reparación para un bloque a partir de los símbolos de origen en ese bloque, a continuación pasar al siguiente bloque y no es necesario referirse a símbolos de origen distintos a los del bloque actual que se está codificando. En una transmisión, un bloque de origen que comprende símbolos de origen puede representarse mediante un bloque codificado que comprende símbolos codificados (que pueden ser algunos símbolos de origen, algunos símbolos de reparación o ambos). Con la presencia de símbolos de reparación, no se requieren todos los símbolos de origen en cada bloque codificado.
- 45 **[0055]** En algunas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).
- 50 **[0056]** En algunas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).
- 55 **[0057]** En algunas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).
- 60 **[0058]** En algunas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).
- 65 **[0059]** En algunas aplicaciones, los paquetes pueden subdividirse en símbolos en los que se aplica el proceso FEC. Un paquete puede contener uno o más símbolos (o menos de un símbolo, pero habitualmente los símbolos no se dividen en grupos de paquetes a menos que se sepa que las condiciones de error entre grupos de paquetes estén altamente correlacionadas). Un símbolo puede tener cualquier tamaño, pero a menudo el tamaño de un símbolo es como máximo igual al tamaño del paquete. Los símbolos de origen son aquellos símbolos que codifican los datos que deben transmitirse. Los símbolos de reparación son símbolos generados a partir de símbolos de origen, directa o indirectamente que se suman a los símbolos de origen (es decir, los datos a transmitir pueden recuperarse por completo si todos los símbolos de origen están disponibles y ninguno de los símbolos de reparación está disponible).

5 [0055] Para algunos códigos FEC, especialmente los códigos Reed-Solomon, el tiempo de codificación y  
 10 descodificación puede volverse impráctico a medida que aumenta el número de símbolos codificados por bloque de  
 origen. Por lo tanto, en la práctica, a menudo hay un límite superior práctico (255 es un límite práctico aproximado  
 para algunas aplicaciones) en el número total de símbolos codificados que pueden generarse por bloque de origen,  
 especialmente en un caso típico en el que el proceso de codificación o descodificación Reed-Solomon se realiza  
 mediante hardware personalizado; por ejemplo, los procesos MPE-FEC que usan códigos Reed-Solomon incluidos  
 como parte de la norma DVB-H para proteger flujos contra la pérdida de paquetes se implementan en hardware  
 especializado dentro de un teléfono celular que está limitado a 255 símbolos totales codificados Reed-Solomon por  
 15 bloque de origen. Dado que a menudo se requiere que los símbolos se coloquen en cargas útiles de paquetes  
 separados, esto coloca un límite superior práctico en la longitud máxima del bloque de origen que se está codificando.  
 Por ejemplo, si una carga útil del paquete está limitada a 1024 bytes o menos y cada paquete lleva un símbolo  
 codificado, entonces un bloque de origen codificado puede tener un máximo de 255 kilobytes, y esto también es, por  
 supuesto, un límite superior en el tamaño del bloque de origen en sí.

20 [0056] Otros problemas, como la posibilidad de descodificar los bloques de origen lo suficientemente rápido como  
 para mantenerse al día con la velocidad de transmisión de origen, para minimizar la latencia de descodificación  
 introducida por la descodificación FEC, y usar solo una pequeña fracción de la CPU disponible en el dispositivo  
 receptor en cualquier momento durante la descodificación FEC se abordan mediante los elementos descritos en el  
 presente documento, así como el tratamiento de los mismos.

[0057] La necesidad de proporcionar una solución de entrega de transmisión sólida y escalable que permita que los  
 componentes del sistema fallen sin afectar negativamente la calidad de las transmisiones entregadas a los receptores.

25 [0058] Un sistema de transmisión de peticiones de bloque debe soportar cambios en la estructura o los metadatos  
 de la presentación, por ejemplo, cambios en la cantidad de codificaciones de medios disponibles o cambios en los  
 parámetros de las codificaciones de medios como la velocidad de transmisión de bits, la resolución, la relación de  
 aspecto, códecs de audio o vídeo o parámetros de códec de cambios en otros metadatos, como las URL asociadas  
 con los archivos de contenido. Dichos cambios pueden ser necesarios por varios motivos, entre ellos, la edición  
 30 conjunta de contenido de diferentes fuentes, como publicidad o diferentes segmentos de una presentación más  
 grande, modificación de URL u otros parámetros que se vuelven necesarios como resultado de cambios en la  
 infraestructura de servicio, por ejemplo debido a cambios de configuración, fallos de equipos o recuperación de fallos  
 de equipos u otras razones.

35 [0059] Existen procedimientos en los que una presentación puede ser controlada por un archivo de lista de  
 reproducción continuamente actualizado. Dado que este archivo se actualiza continuamente, al menos algunos de los  
 cambios descritos anteriormente se pueden realizar dentro de estas actualizaciones. Una desventaja de un  
 procedimiento convencional es que los dispositivos cliente deben recuperar continuamente, también conocido como  
 "sondear", el archivo de lista de reproducción, cargando la infraestructura de servicio y que este archivo no se puede  
 40 almacenar en memoria caché durante más tiempo que el intervalo de actualización, lo cual hace que la tarea sea  
 mucho más difícil para la infraestructura de servicio. Esto se aborda en los elementos del presente documento para  
 que se proporcionen actualizaciones del tipo descrito anteriormente sin la necesidad de que los clientes realicen un  
 sondeo continuo para el archivo de metadatos.

45 [0060] Otro problema, especialmente en los servicios en vivo, típicamente conocido por la distribución de  
 retransmisión, es la falta de capacidad para que el usuario visualice el contenido que se ha retransmitido antes del  
 momento en que el usuario se unió al programa. Típicamente, la grabación personal local consume almacenamiento  
 local innecesario o no es posible ya que el cliente no estaba sintonizado con el programa o está prohibido por las  
 reglas de protección de contenido. Se prefiere la grabación de la red y la visualización de turnos, pero requiere  
 50 conexiones individuales del usuario al servidor y un protocolo e infraestructura de entrega independientes a los  
 servicios en vivo, lo cual da como resultado una infraestructura duplicada y costes significativos para el servidor. Esto  
 también es abordado por los elementos descritos en el presente documento.

55 **Resumen del sistema**

[0061] Un modo de realización de la invención se describe con referencia a la Fig. 1, que muestra un diagrama  
 simplificado de un sistema de transmisión de petición de bloques que incorpora la invención.

60 [0062] En la Fig. 1, se ilustra un sistema de transmisión de bloques 100, que comprende la infraestructura de servicio  
 de bloques ("BSI") 101, que a su vez comprende un sistema de ingestión 103 para ingerir contenido 102, preparar ese  
 contenido y empaquetarlo para su servicio mediante un servidor de transmisión HTTP 104 almacenándolo en un  
 almacén de contenido 110 que sea accesible tanto para el sistema de ingestión 103 como para el servidor de  
 transmisión HTTP 104. Como se muestra, el sistema 100 también puede incluir una memoria caché HTTP 106. En  
 funcionamiento, un cliente 108, como un cliente de transmisión HTTP, envía peticiones 112 al servidor de transmisión  
 65 HTTP 104 y recibe respuestas 114 del servidor de transmisión HTTP 104 o la memoria caché HTTP 106. En cada

caso, los elementos mostrados en la Fig. 1 podrían implementarse, al menos en parte, en un software, que comprende un código de programa que se ejecuta en un procesador u otro sistema electrónico.

**[0063]** El contenido puede incluir películas, audio, vídeo planar 2D, vídeo 3D, otros tipos de vídeo, imágenes, texto temporizado, metadatos temporizados o similares. Algunos contenidos pueden incluir datos que se presentarán o consumirán de manera temporizada, como datos para presentar información auxiliar (identificación de la estación, publicidad, cotizaciones de acciones, secuencias de Flash™, etc.) junto con otros medios que se están reproduciendo. También se pueden usar otras presentaciones híbridas que combinan otros medios y/o van más allá de simplemente audio y vídeo.

**[0064]** Como se ilustra en la Fig. 2, los bloques de medios pueden almacenarse dentro de una infraestructura de servicio de bloque 101(1), que podría ser, por ejemplo, un servidor HTTP, un dispositivo de red de entrega de contenido, un proxy HTTP, un proxy o servidor FTP, o algún otro servidor o sistema de medios. La infraestructura de servicio de bloque 101(1) está conectada a una red 122, que podría ser, por ejemplo, una red de protocolo de Internet ("IP") como Internet. Se muestra un cliente de sistema de transmisión de petición de bloques que tiene seis componentes funcionales, a saber, un selector de bloque 123, provisto con los metadatos descritos anteriormente y que realiza una función de selección de bloques o bloques parciales a solicitar entre la pluralidad de bloques disponibles indicados por los metadatos, un solicitante de bloque 124, que recibe instrucciones de petición del selector de bloque 123 y realiza las operaciones necesarias para enviar una petición para el bloque especificado, partes de un bloque o bloques múltiples, para bloquear la infraestructura de servicio 101(1) a través de la red 122 y para recibir los datos que comprenden el bloque a cambio, así como una memoria intermedia de bloques 125, un monitor de memoria intermedia 126, un descodificador de medios 127 y uno o más transductores de medios 128 que facilitan el consumo de medios.

**[0065]** Los datos de bloques recibidos por el solicitante de bloque 124 se pasan al almacenamiento temporal para bloquear la memoria intermedia 125, que almacena los datos de medios. De forma alternativa, los datos de bloques recibidos pueden almacenarse directamente en la memoria intermedia de bloques 125 como se ilustra en la Fig. 1. El descodificador de medios 127 está provisto de datos de medios por la memoria intermedia de bloques 125 y realiza las transformaciones en estos datos que son necesarias para proporcionar una entrada adecuada a los transductores de medios 128, que renderizan los medios en una forma adecuada para el usuario u otro consumo. Entre los ejemplos de transductores de medios se incluyen dispositivos de visualización visual como los que se encuentran en teléfonos móviles, sistemas informáticos o televisores, y también pueden incluir dispositivos de reproducción de audio, como altavoces o auriculares.

**[0066]** Un ejemplo de un descodificador de medios sería una función que transforme los datos en el formato descrito en la norma de codificación de vídeo H.264 en representaciones analógicas o digitales de tramas de vídeo, como un mapa de píxeles en formato YUV con marcas de tiempo de presentación asociadas para cada trama o muestra.

**[0067]** El monitor de memoria intermedia 126 recibe información sobre el contenido de la memoria intermedia de bloques 125 y, basándose en esta información y posiblemente en otra información, proporciona información al selector de bloque 123, que se utiliza para determinar la selección de bloques a solicitar, como se describe en el presente documento.

**[0068]** En la terminología utilizada en el presente documento, cada bloque tiene un "tiempo de reproducción" o "duración" que representa la cantidad de tiempo que tardaría el receptor en reproducir los medios incluidos en ese bloque a la velocidad normal. En algunos casos, la reproducción de los medios dentro de un bloque puede depender de que ya hayan recibido datos de bloques anteriores. En casos raros, la reproducción de algunos de los medios en un bloque puede depender de un bloque posterior, en cuyo caso el tiempo de reproducción para el bloque se define con respecto a los medios que se pueden reproducir dentro del bloque sin referencia al bloque subsiguiente, y el tiempo de reproducción para el bloque subsiguiente se incrementa el tiempo de reproducción de los medios dentro de este bloque que solo puede reproducir después de haber recibido el bloque subsiguiente. Dado que incluir medios en un bloque que depende de bloques subsiguientes es un caso raro, en el resto de esta divulgación suponemos que los medios en un bloque no dependen de bloques subsiguientes, pero tenga en cuenta que los expertos en la técnica reconocerán que esta variante se puede agregar fácilmente a los modos de realización descritos a continuación.

**[0069]** El receptor puede tener controles tales como "pausa", "avance rápido", "retroceso", etc. que pueden dar como resultado que el bloque se consuma mediante la reproducción a una velocidad diferente, pero si el receptor puede obtener y descodificar cada secuencia de bloques consecutiva en un tiempo agregado igual o menor que su tiempo de reproducción agregado excluyendo el último bloque en la secuencia, entonces el receptor puede presentar los medios al usuario sin detenerse. En algunas descripciones del presente documento, una posición particular en el flujo de medios se conoce como un "tiempo" particular en los medios, correspondiente al tiempo que hubiera transcurrido entre el inicio de la reproducción de los medios y el momento en que se alcanza la posición particular en el flujo de vídeo. El tiempo o la posición en un flujo de medios es un concepto convencional. Por ejemplo, cuando el flujo de vídeo comprende 24 tramas por segundo, se podría decir que la primera trama tiene una posición o tiempo de  $t=0,0$  segundos y la trama 241 puede decirse que tiene una posición o tiempo de  $t=10,0$  segundos. Naturalmente, en un flujo de vídeo basado en tramas, la posición o el tiempo no necesitan ser continuos, ya que cada uno de los bits en el

flujo desde el primer bit de la trama 241 hasta justo antes del primer bit de la trama 242 puede tener el mismo valor de tiempo.

**[0070]** Adoptando la terminología anterior, un sistema de transmisión de petición de bloques (BRSS) comprende uno o más clientes que realizan peticiones a uno o más servidores de contenido (por ejemplo, servidores HTTP, servidores FTP, etc.). Un sistema de ingestión comprende uno o más procesadores de ingestión, en el que un procesador de ingestión recibe contenido (en tiempo real o no), procesa el contenido para su uso por el BRSS y lo almacena en un almacenamiento accesible para los servidores de contenido, posiblemente también junto con los metadatos generados por el procesador de ingestión.

**[0071]** El BRSS también puede contener memorias caché de contenido que se coordinan con los servidores de contenido. Los servidores de contenido y las memorias caché de contenido pueden ser servidores HTTP convencionales y memorias caché de HTTP que reciben peticiones de archivos o segmentos en forma de peticiones HTTP que incluyen una URL, y también pueden incluir un rango de bytes, para solicitar menos de todo el archivo. o segmento indicado por la URL. Los clientes pueden incluir un cliente HTTP convencional que realiza peticiones de servidores HTTP y maneja las respuestas a esas peticiones, donde el cliente HTTP es impulsado por un nuevo sistema cliente que formula peticiones, las pasa al cliente HTTP, obtiene respuestas del cliente HTTP y las procesa (o almacenamiento, transformación, etc.) para proporcionarlas a un reproductor de presentaciones para que las expida un dispositivo cliente. En general, el sistema cliente no sabe de antemano qué medios se van a necesitar (ya que las necesidades pueden depender de la entrada del usuario, los cambios en la entrada del usuario, etc.), por lo que se dice que es un sistema de "transmisión" en que los medios se "consumen" tan pronto como se reciben, o poco después. Como resultado, los retardos en la respuesta y las restricciones de ancho de banda pueden ocasionar retardos en una presentación, como ocasionar una pausa en una presentación a medida que el flujo alcanza hasta donde el usuario está consumiendo la presentación.

**[0072]** Para proporcionar una presentación que se considere de buena calidad, se pueden implementar una serie de detalles en el BRSS, ya sea en el extremo del cliente, al final de la ingestión, o ambos. En algunos casos, los detalles que se implementan se realizan teniendo en cuenta, y para tratar, la interfaz cliente-servidor en la red. En algunos modos de realización, tanto el sistema cliente como el sistema de ingestión son conscientes de la mejora, mientras que en otros modos de realización. solo un lado es consciente de la mejora. En tales casos, todo el sistema se beneficia de la mejora, aunque un lado no lo sepa, mientras que en otros, el beneficio solo se acumula si ambos lados lo saben, pero cuando un lado no lo sabe, todavía funciona sin fallar.

**[0073]** Como se ilustra en la Fig. 3, el sistema de ingestión puede implementarse como una combinación de componentes de hardware y software, de acuerdo con varios modos de realización. El sistema de ingestión puede comprender un conjunto de instrucciones que pueden ejecutarse para hacer que el sistema realice una o más de las metodologías que se describen en el presente documento. El sistema puede realizarse como una máquina específica en forma de ordenador. El sistema puede ser un ordenador servidor, un ordenador personal (PC) o cualquier sistema capaz de ejecutar un conjunto de instrucciones (secuenciales o de otro tipo) que especifiquen las acciones que debe realizar ese sistema. Además, aunque solo se ilustra un solo sistema, el término "sistema" también debe considerarse como un conjunto de sistemas que ejecutan individual o conjuntamente un conjunto (o conjuntos múltiples) de instrucciones para realizar una o más de las metodologías que se describen en el presente documento.

**[0074]** El sistema de ingestión puede incluir el procesador de ingestión 302 (por ejemplo, una unidad central de procesamiento (CPU)), una memoria 304 que puede almacenar el código del programa durante la ejecución y el almacenamiento en disco 306, todos los cuales se comunican entre sí a través de un bus 300. El sistema puede incluir además una unidad de visualización de vídeo 308 (por ejemplo, una pantalla de cristal líquido (LCD) o un tubo de rayos catódicos (CRT)). El sistema también puede incluir un dispositivo de entrada alfanumérico 310 (por ejemplo, un teclado) y un dispositivo de interfaz de red 312 para recibir la fuente de contenido y entregar el almacén de contenido.

**[0075]** La unidad de almacenamiento de disco 306 puede incluir un medio legible por máquina en el que puede almacenarse uno o más conjuntos de instrucciones (por ejemplo, software) que incorporan una o más de las metodologías o funciones descritas en el presente documento. Las instrucciones también pueden residir, total o al menos parcialmente, dentro de la memoria 304 y/o dentro del procesador de ingestión 302 durante la ejecución del mismo por el sistema, y la memoria 304 y el procesador de ingestión 302 también constituyen medios legibles por máquina.

**[0076]** Como se ilustra en la Fig. 4, el sistema cliente puede implementarse como una combinación de componentes de hardware y software, de acuerdo con varios modos de realización. El sistema cliente puede comprender un conjunto de instrucciones que se pueden ejecutar para hacer que el sistema realice una o más de las metodologías que se analizan en el presente documento. El sistema puede realizarse como una máquina específica en forma de ordenador. El sistema puede ser un ordenador servidor, un ordenador personal (PC) o cualquier sistema capaz de ejecutar un conjunto de instrucciones (secuenciales o de otro tipo) que especifiquen las acciones que debe realizar ese sistema. Además, aunque solo se ilustra un solo sistema, el término "sistema" también debe considerarse como un conjunto de sistemas que ejecutan individual o conjuntamente un conjunto (o conjuntos múltiples) de instrucciones para realizar una o más de las metodologías que se describen en el presente documento.

[0077] El sistema cliente puede incluir el procesador cliente 402 (por ejemplo, una unidad central de procesamiento (CPU)), una memoria 404 que puede almacenar el código del programa durante la ejecución, y el almacenamiento en disco 406, todos los cuales se comunican entre sí a través de un bus 400. El sistema puede incluir además una unidad de visualización de vídeo 408 (por ejemplo, una pantalla de cristal líquido (LCD) o un tubo de rayos catódicos (CRT)). El sistema también puede incluir un dispositivo de entrada alfanumérico 410 (por ejemplo, un teclado) y un dispositivo de interfaz de red 412 para enviar peticiones y recibir respuestas.

[0078] La unidad de almacenamiento de disco 406 puede incluir un medio legible por máquina en el que puede almacenarse uno o más conjuntos de instrucciones (por ejemplo, software) que incorporan una o más de las metodologías o funciones descritas en el presente documento. Las instrucciones también pueden residir, total o al menos parcialmente, dentro de la memoria 404 y/o dentro del procesador cliente 402 durante su ejecución mediante el sistema, y la memoria 404 y el procesador cliente 402 también constituyen medios legibles por la máquina.

### Uso del formato de archivo 3GPP

[0079] El formato de archivo 3GPP o cualquier otro archivo basado en el formato de archivo de medios de base ISO, como el formato de archivo MP4 o el formato de archivo 3GPP2, se puede usar como el formato de contenedor para la transmisión HTTP con las siguientes características. Se puede incluir un índice de segmento en cada segmento para indicar las desviaciones de tiempo y los rangos de bytes, de modo que el cliente pueda descargar los fragmentos de archivos o segmentos de medios apropiados según sea necesario. La temporización de la presentación global de toda la presentación de los medios y la temporización local dentro de cada archivo 3Gp o segmento de medios puede alinearse con precisión. Las pistas dentro de un archivo 3GP o segmento de medios pueden alinearse con precisión. Las pistas a través de las representaciones también pueden alinearse asignando cada una de ellas a la línea de tiempo global, de modo que el cambio a través de la representación puede ser perfecto y la presentación conjunta de los componentes de los medios en diferentes representaciones puede ser sincrónica.

[0080] El formato del archivo puede contener un perfil para la transmisión adaptable con las siguientes propiedades. Todos los datos de la película pueden estar contenidos en fragmentos de películas; es posible que la ventana "moov" no contenga ninguna información de muestra. Los datos de muestra de audio y vídeo se pueden intercalar, con requisitos similares a los del perfil de descarga progresiva especificado en TS26.244. La ventana "moov" se puede colocar al inicio del archivo, seguida de los datos de desviación de fragmento, también denominados índice de segmento, que contienen información de desviación en el tiempo y rangos de bytes para cada fragmento o al menos un subconjunto de fragmentos en el segmento que los contiene.

[0081] También puede ser posible que la descripción de presentación de medios haga referencia a los archivos que siguen el perfil de descarga progresiva existente. En este caso, el cliente puede usar la descripción de presentación de medios simplemente para seleccionar la versión alternativa apropiada entre las múltiples versiones disponibles. Los clientes también pueden usar peticiones de obtención parcial HTTP con archivos compatibles con el perfil de descarga progresiva para solicitar subconjuntos de cada versión alternativa y, por lo tanto, implementar una forma menos eficiente de transmisión adaptable. En este caso, las diferentes representaciones que contienen los medios en el perfil de descarga progresiva aún pueden adherirse a una línea de tiempo global común para permitir el cambio sin problemas entre las representaciones.

### Resumen de procedimientos avanzados

[0082] En las siguientes secciones, se describen los procedimientos para mejorar los sistemas de transmisión de petición de bloques. Debe entenderse que algunas de estas mejoras se pueden utilizar con o sin otras de estas mejoras, dependiendo de las necesidades de la aplicación. En el funcionamiento general, un receptor realiza peticiones de un servidor u otro transmisor para bloques específicos o partes de bloques de datos. Los archivos, también llamados segmentos, pueden contener varios bloques y están asociados con una representación de una presentación de medios.

[0083] Preferentemente, se genera información de indexación, también llamada "indexación de segmentos" o "mapa de segmentos", que proporciona una asignación desde tiempos de reproducción o decodificación a desviaciones de bytes de bloques o fragmentos correspondientes dentro de un segmento. Esta indexación de segmentos puede incluirse dentro del segmento, típicamente al principio del segmento (al menos parte del mapa de segmentos está al principio) y con frecuencia es pequeña. El índice de segmento también se puede proporcionar en un segmento o archivo de índice separado. Especialmente en los casos en que el índice de segmento está contenido en el segmento, el receptor puede descargar parte o la totalidad de este mapa de segmentos rápidamente y posteriormente usarlo para determinar la asignación entre las desviaciones de tiempo y las posiciones de bytes correspondientes de los fragmentos asociados con esas desviaciones de tiempo dentro del archivo.

[0084] Un receptor puede usar la desviación de bytes para solicitar datos de los fragmentos asociados con desviaciones de tiempo particulares, sin tener que descargar todos los datos asociados con otros fragmentos no asociados con las desviaciones de tiempo de interés. De esta manera, el mapa de segmentos o la indexación de

segmentos puede mejorar en gran medida la capacidad de un receptor para acceder directamente a las partes del segmento que son relevantes para las desviaciones de tiempo actuales de interés, con beneficios que incluyen mejores tiempos de zapping de contenido, la capacidad de cambiar rápidamente de una representación a otra según varían las condiciones de la red y menores desperdicios de recursos de red que descargan medios que no se reproducen en un receptor.

**[0085]** En el caso de que se considere cambiar de una representación (denominada aquí la representación de "cambio de") a otra representación (denominada aquí la representación de "cambio de a"), el índice de segmento también se puede usar para identificar el tiempo de inicio de un punto de acceso aleatorio en la representación de cambio a para identificar la cantidad de datos que se solicitarán en la representación de cambio de para garantizar que el cambio sin interrupciones esté habilitado en un sentido que los medios en la representación de cambio desde se descarguen hasta un tiempo de presentación tal que la reproducción de la representación de cambio a pueda comenzar sin problemas desde el punto de acceso aleatorio.

**[0086]** Esos bloques representan segmentos de medios de vídeo u otros medios que el receptor solicitante necesita para generar la salida para el usuario del receptor. El receptor de los medios puede ser un dispositivo cliente, como cuando el receptor recibe contenido de un servidor que transmite el contenido. Los ejemplos incluyen descodificadores, ordenadores, consolas de juegos, televisores especialmente equipados, dispositivos de mano, teléfonos móviles especialmente equipados u otros receptores clientes.

**[0087]** Muchos procedimientos avanzados de gestión de memoria intermedia se describen en el presente documento. Por ejemplo, un procedimiento de administración de memoria intermedia permite a los clientes solicitar bloques de la mejor calidad de medios que pueden recibirse a tiempo para reproducirse con continuidad. Una característica de tamaño de bloque variable mejora la eficiencia de compresión. La capacidad de tener múltiples conexiones para transmitir bloques al dispositivo solicitante mientras limita la frecuencia de las peticiones proporciona un mejor rendimiento de transmisión. Se pueden usar bloques de datos parcialmente recibidos para continuar con la presentación de los medios. Una conexión se puede reutilizar para varios bloques sin tener que confirmar la conexión al inicio de un conjunto particular de bloques. La coherencia en la selección de servidores entre múltiples servidores posibles por múltiples clientes se mejora, lo cual reduce la frecuencia de contenido duplicado en servidores cercanos y mejora la probabilidad de que un servidor contenga un archivo completo. Los clientes pueden solicitar bloques de medios basándose en los metadatos (como las codificaciones de medios disponibles) que están integrados en las URL de los archivos que contienen los bloques de medios. Un sistema puede proporcionar el cálculo y la minimización de la cantidad de tiempo de memoria intermedia requerido antes de que pueda comenzar la reproducción del contenido sin incurrir en pausas posteriores en la reproducción de los medios. El ancho de banda disponible se puede compartir entre múltiples bloques de medios, ajustar a medida que se aproxima el tiempo de reproducción de cada bloque, de modo que, si es necesario, se pueda asignar una mayor proporción del ancho de banda disponible hacia el bloque con el tiempo de reproducción más cercano.

**[0088]** La transmisión HTTP puede emplear metadatos. Los metadatos de nivel de presentación incluyen, por ejemplo, duración de la transmisión, codificaciones disponibles (velocidades de transmisión de bits, códecs, resoluciones espaciales, frecuencias de tramas, idioma, tipos de medios), punteros para transmitir los metadatos para cada codificación y protección de contenido (información de gestión de derechos digitales (DRM)). Los metadatos de flujo pueden ser URLs de los archivos de segmento.

**[0089]** Los metadatos del segmento pueden incluir información de rango de bytes en función del tiempo para peticiones dentro de un segmento e identificación de puntos de acceso aleatorio (RAP) u otros puntos de búsqueda, donde parte o toda esta información puede ser parte de un índice de segmento o mapa de segmentos.

**[0090]** Los flujos pueden comprender múltiples codificaciones del mismo contenido. A continuación, cada codificación se puede dividir en segmentos donde cada segmento corresponde a un archivo o una unidad de almacenamiento. En el caso de HTTP, un segmento suele ser un recurso al que se puede hacer referencia mediante una URL y la petición de dicha URL da como resultado el retorno del segmento como el cuerpo de la entidad del mensaje de respuesta de la petición. Los segmentos pueden comprender múltiples grupos de imágenes (GoPs). Cada GoP puede comprender además múltiples fragmentos en los que la indexación de segmentos proporciona información de desviación de tiempo/byte para cada fragmento, es decir, la unidad de indexación es un fragmento.

**[0091]** Se pueden solicitar fragmentos o porciones de fragmentos a través de conexiones TCP paralelas para aumentar el rendimiento. Esto puede reducir los problemas que surgen al compartir conexiones en un enlace de cuello de botella o cuando las conexiones se pierden debido a la congestión, lo cual aumenta la velocidad general y la fiabilidad de la entrega, lo cual puede mejorar sustancialmente la velocidad y la fiabilidad del tiempo de zapping de contenido. El ancho de banda puede intercambiarse por latencia mediante una petición excesiva, pero se debe tener cuidado para evitar realizar peticiones demasiado lejanas en el futuro que puedan aumentar el riesgo de escasez.

**[0092]** Se pueden yuxtaponer múltiples peticiones de segmentos en el mismo servidor (haciendo la siguiente petición antes de que se complete la petición actual) para evitar retardos de inicio de TCP repetitivos. Las peticiones de fragmentos consecutivos se pueden agregar en una sola petición.

5 [0093] Algunos CDN prefieren archivos grandes y pueden activar búsquedas en segundo plano de un archivo completo desde un servidor de origen cuando se ve por primera vez una petición de rango. Sin embargo, la mayoría de los CDN atenderán las peticiones de rango desde la memoria caché si los datos están disponibles. Por lo tanto, puede ser ventajoso tener parte de las peticiones del cliente para un archivo de segmento completo. Estas peticiones pueden ser canceladas más tarde si es necesario.

10 [0094] Los puntos de cambio válidos pueden ser puntos de búsqueda, específicamente RAP, por ejemplo, en el flujo de destino. Son posibles diferentes implementaciones, como estructuras de GoP fijas o alineación de RAP a través de flujos (basándose en el inicio de los medios o basándose en los GoP).

15 [0095] En un modo de realización, los segmentos y GoPs pueden alinearse a través de diferentes flujos de velocidad. En este modo de realización, los GoP pueden ser de tamaño variable y pueden contener múltiples fragmentos, pero los fragmentos no están alineados entre los diferentes flujos de velocidad.

20 [0096] En algunos modos de realización, la redundancia de archivos se puede emplear ventajosamente. En estos modos de realización, se aplica un código de borrado a cada fragmento para generar versiones redundantes de los datos. Preferentemente, el formato de origen no se modifica debido al uso de FEC, y se generan segmentos de reparación adicionales, por ejemplo, como representación dependiente de la representación original, que contienen datos de reparación FEC y se ponen a disposición como un paso adicional en el sistema de ingestión. El cliente, que es capaz de reconstruir un fragmento utilizando solo datos de origen para ese fragmento, solo puede solicitar datos de origen para el fragmento dentro del segmento de los servidores. Si los servidores no están disponibles o la conexión a los servidores es lenta, lo cual puede determinarse antes o después de la petición de datos de origen, se pueden solicitar datos de reparación adicionales para el fragmento del segmento de reparación, lo cual disminuye el tiempo para entregar de manera fiable suficientes datos para recuperar el fragmento, posiblemente utilizando la decodificación FEC para usar una combinación de datos de origen y reparación recibidos para recuperar los datos de origen del fragmento. Además, se pueden solicitar datos de reparación adicionales para permitir la recuperación del fragmento si un fragmento se vuelve urgente, es decir, su tiempo de reproducción se vuelve inminente, lo cual aumenta el intercambio de datos para ese fragmento en un enlace pero es más eficiente que cerrar otras conexiones en el enlace para liberar ancho de banda. Esto también puede reducir el riesgo de escasez por el uso de conexiones paralelas.

35 [0097] El formato del fragmento puede ser un flujo almacenado de paquetes de protocolo de transporte en tiempo real (RTP) con sincronización de audio/vídeo lograda a través del protocolo de control de transporte en tiempo real RTCP.

[0098] El formato de segmento también puede ser un flujo almacenado de paquetes MPEG-2 TS con sincronización de audio/vídeo que se logra con la temporización interna MPEG-2 TS.

40 **Uso de señalización y/o creación de bloques para hacer que la transmisión sea más eficiente**

45 [0099] Se puede usar una serie de características, en un sistema de transmisión de petición de bloques, para proporcionar un mejor rendimiento. El rendimiento puede estar relacionado con la capacidad de reproducir una presentación sin detenerse, obtener datos de medios dentro de las restricciones de ancho de banda y/o hacerlo dentro de los recursos limitados del procesador en un cliente, servidor y/o sistema de ingestión. Ahora se describirán algunas de estas características.

**Indexación dentro de segmentos**

50 [0100] Con el fin de formular peticiones GET parciales para los fragmentos de película, se puede informar al cliente sobre la desviación de bytes y el tiempo de inicio en la decodificación o el tiempo de presentación de todos los componentes de medios contenidos en los fragmentos dentro del archivo o segmento y también qué fragmentos comienzan o contienen unos puntos de acceso aleatorio (y, por lo tanto, son adecuados para ser utilizados como puntos de cambio entre representaciones alternativas), en el que esta información suele denominarse indexación de segmentos o mapa de segmentos. el tiempo de inicio en el tiempo de decodificación o presentación puede expresarse directamente o puede expresarse como deltas en relación con un tiempo de referencia.

60 [0101] Esta información de indexación de desviación de tiempo y bytes puede requerir al menos 8 bytes de datos por fragmento de película. Como ejemplo, para una película de dos horas contenida en un solo archivo, con fragmentos de película de 500 ms, esto sería un total de aproximadamente 112 kilobytes de datos. La descarga de todos estos datos al iniciar una presentación puede dar como resultado un retardo de inicio adicional significativo. Sin embargo, los datos de desviación de tiempo y bytes se pueden codificar jerárquicamente, de modo que el cliente puede encontrar rápidamente una pequeña porción de datos de desviación y tiempo relevantes para el punto de la presentación en el que desea comenzar. La información también puede distribuirse dentro de un segmento de tal manera que un cierto refinamiento del índice de segmento pueda ubicarse intercalado con datos de medios.

**[0102]** Tenga en cuenta que si la representación se segmenta en forma de tiempo en varios segmentos, el uso de esta codificación jerárquica tal vez no sea necesario, ya que los datos completos de desviación y tiempo para cada segmento pueden ser bastante pequeños. Por ejemplo, si los segmentos son de un minuto en lugar de dos horas en el ejemplo anterior, la información de indexación de desviación de bytes de tiempo es de alrededor de 1 kilobyte de datos, que típicamente puede caber dentro de un solo paquete TCP/IP.

**[0103]** Diferentes opciones son posibles para añadir datos de desviación de bytes y tiempo de fragmento a un archivo 3GPP:

Primero, se puede usar la ventana de acceso aleatorio de fragmentos de película ("MFRA") para este propósito. El MFRA proporciona una tabla, que puede ayudar a los lectores a encontrar puntos de acceso aleatorios en un archivo utilizando fragmentos de películas. En soporte de esta función, el MFRA contiene incidentalmente las desviaciones de bytes de las ventanas MFRA que contienen puntos de acceso aleatorio. El MFRA puede colocarse en el o cerca del final del archivo, pero este no es necesariamente el caso. Al escanear desde el final del archivo una ventana de desviación de acceso aleatorio de fragmentos de película y usar la información de tamaño que contiene, es posible que se pueda ubicar el comienzo de un ventana de acceso aleatorio de fragmentos de película. Sin embargo, colocar el MFRA al final de la transmisión HTTP requiere en general al menos 3-4 peticiones HTTP para acceder a los datos deseados: al menos una para solicitar el MFRA desde el final del archivo, una para obtener el MFRA y finalmente una para obtener el fragmento deseado en el archivo. Por lo tanto, la ubicación al principio puede ser deseable, ya que a continuación el MFRA se puede descargar junto con los primeros datos de medios en una sola petición. Además, el uso de MFRA para la transmisión HTTP puede ser ineficaz, ya que no se necesita nada de la información en "MFRA" aparte del tiempo y la desviación de Moof y la especificación de desviaciones en lugar de longitudes puede requerir más bits.

En segundo lugar, se puede utilizar la ventana de ubicación del artículo ("ILOC"). El "ILOC" proporciona un directorio de recursos de metadatos en este u otros archivos, al ubicar el archivo que lo contiene, su desviación dentro de ese archivo y su longitud. Por ejemplo, un sistema podría integrar todos los recursos de metadatos referenciados externamente en un archivo, reajustando las desviaciones de archivos y las referencias de archivos en consecuencia. Sin embargo, el "ILOC" está destinado a proporcionar la ubicación de los metadatos, por lo que puede ser difícil que esto coexista con los metadatos reales.

Por último, y quizá lo más adecuado, está la especificación de una nueva ventana, denominada ventana de índice de tiempo ("TIDX", por sus siglas en inglés), dedicada específicamente al propósito de proporcionar tiempos o duraciones exactos de fragmentos y desviación de bytes de manera eficiente. Esto se describe con más detalle en la siguiente sección. Una ventana alternativa con las mismas funcionalidades puede ser la ventana de índice de segmento ("SIDX"). En el presente documento, a menos que se indique lo contrario, estas dos pueden ser intercambiables, ya que ambas ventanas ofrecen la capacidad de proporcionar tiempos o duraciones exactos de fragmentos y desviación de bytes de una manera eficiente. La diferencia entre el TIDX y el SIDX se proporciona a continuación. Debe ser evidente cómo intercambiar las ventanas TIDX y SIDX, ya que ambas ventanas implementan un índice de segmento.

### **Indexación de segmentos**

**[0104]** Un segmento tiene un tiempo de inicio identificada y un número de bytes identificado. Se pueden concatenar múltiples fragmentos en un solo segmento y los clientes pueden emitir peticiones que identifican el rango de bytes específico dentro del segmento que corresponde al fragmento o subconjunto requerido del fragmento. Por ejemplo, cuando se usa HTTP como protocolo de petición, entonces se puede usar la cabecera de rango HTTP para este propósito. Este enfoque requiere que el cliente tenga acceso a un "índice de segmento" del segmento que especifica la posición dentro del segmento de los diferentes fragmentos. Este "índice de segmento" puede proporcionarse como parte de los metadatos. Este enfoque tiene el resultado de que es necesario crear y administrar muchos menos archivos en comparación con el enfoque donde cada bloque se guarda en un archivo separado. La gestión de la creación, la transferencia y el almacenamiento de una gran cantidad de archivos (que podrían extenderse a muchos miles para una presentación de 1 hora, por ejemplo) puede ser compleja y propensa a errores, por lo que la reducción del número de archivos representa una ventaja.

**[0105]** Si el cliente solo conoce el tiempo de inicio deseada de una parte más pequeña de un segmento, puede solicitar todo el archivo y a continuación leer el archivo para determinar la ubicación de inicio de reproducción adecuada. Para mejorar el uso del ancho de banda, los segmentos pueden incluir un archivo de índice como metadatos, donde el archivo de índice asigna los rangos de bytes de bloques individuales con los rangos de tiempo a los que corresponden los bloques, lo cual se denomina indexación de segmentos o mapa de segmentos. Estos metadatos pueden formatearse como datos XML o pueden ser binarios, por ejemplo, siguiendo la estructura atómica y de ventana del formato de archivo 3GPP. La indexación puede ser simple, en el que los rangos de tiempo y bytes de cada bloque son absolutos con respecto al inicio del archivo, o pueden ser jerárquicos, en el que algunos bloques se agrupan en bloques principales (y aquellos en bloques superiores, etc.) y el intervalo de tiempo y bytes para un bloque dado se expresa en relación con el intervalo de tiempo y/o bytes del bloque principal del bloque.

**Ejemplo de estructura de mapa de indexación**

**[0106]** En un modo de realización, los datos de origen originales para una representación de un flujo de medios pueden estar contenidos en uno o más archivos de medios aquí llamados "segmento de medios", en el que cada segmento de medios contiene los datos de medios utilizados para reproducir un segmento de tiempo continuo de los medios, por ejemplo, 5 minutos de reproducción de medios.

**[0107]** La Fig. 6 muestra un ejemplo de la estructura general de un segmento de medios. Dentro de cada segmento, ya sea al principio o extendido a lo largo del segmento de origen, también puede haber información de indexación, que comprende un mapa de segmentos de tiempo/desviación de bytes. El mapa de segmentos de desviación de tiempo/bytes en un modo de realización puede ser una lista de pares de desviación de tiempo/bytes  $(T(0), B(0))$ ,  $(T(1), B(1))$ , ...,  $(T(i), B(i))$ , ...,  $(T(n), B(n))$ , en el que  $T(i-1)$  representa un tiempo de inicio dentro del segmento para la reproducción del  $i$ -ésimo fragmento de medios en relación con el tiempo de inicio inicial de los medios entre todos los segmentos de medios,  $T(i)$  representa un tiempo de finalización para el  $i$ -ésimo fragmento (y por lo tanto el tiempo de inicio para el siguiente fragmento), y la desviación de bytes  $B(i-1)$  es el índice de bytes correspondiente al comienzo de los datos dentro de este segmento de origen donde el  $i$ -ésimo fragmento de medios comienza en relación con el comienzo del segmento de origen, y  $B(i)$  es el índice de bytes final correspondiente del  $i$ -ésimo fragmento (y por lo tanto el índice del primer byte del siguiente fragmento). Si el segmento contiene múltiples componentes de medios, entonces se pueden proporcionar  $T(i)$  y  $B(i)$  para cada componente en el segmento de manera absoluta o se pueden expresar en relación con otro componente de medios que sirve un componente de medios de referencia.

**[0108]** En este modo de realización, el número de fragmentos en el segmento de origen es  $n$ , donde  $n$  puede variar de un segmento a otro.

**[0109]** En otro modo de realización, la desviación de tiempo en el índice de segmento para cada fragmento puede determinarse con el tiempo de inicio absoluto del primer fragmento y las duraciones de cada fragmento. En este caso, el índice de segmento puede documentar el tiempo de inicio del primer fragmento y la duración de todos los fragmentos que se incluyen en el segmento. El índice de segmento también puede documentar solo un subconjunto de los fragmentos. En ese caso, el índice del segmento documenta la duración de un subsegmento que se define como uno o más fragmentos consecutivos, que termina al final del segmento que lo contiene o al comienzo del subsegmento siguiente.

**[0110]** Para cada fragmento, también puede haber un valor que indique si el fragmento comienza en un punto de búsqueda, es decir, en un punto en el que ningún medio después de ese punto depende de ningún medio anterior a ese punto y, por lo tanto, los medios desde ese fragmento hacia adelante se pueden reproducir independientemente de los fragmentos anteriores. Los puntos de búsqueda son, en general, puntos en los medios donde la reproducción puede comenzar independientemente de todos los medios anteriores. La Fig. 6 también muestra un ejemplo simple de posible indexación de segmentos para un segmento de origen. En ese ejemplo, el valor de desviación de tiempo está en unidades de milisegundos y, por lo tanto, el primer fragmento de este segmento de origen comienza 20 segundos desde el comienzo del medio, y el primer fragmento tiene un tiempo de reproducción de 485 milisegundos. La desviación de bytes del inicio del primer fragmento es 0, y la desviación de bytes del final del primer fragmento/inicio del segundo fragmento es de 50 245, y por lo tanto el primer fragmento tiene un tamaño de 50 245 bytes. Si el fragmento o el subsegmento no comienzan con un punto de acceso aleatorio, pero el punto de acceso aleatorio está contenido en el fragmento o subsegmento, entonces puede darse la diferencia entre el tiempo de descodificación o el tiempo de presentación entre el tiempo de inicio y el tiempo real de RAP. Esto permite que, en caso de cambiar a este segmento de medios, el cliente pueda saber con precisión el tiempo hasta que deba presentarse el cambio de representación.

**[0111]** Además de, o en lugar de, la indexación simple o jerárquica, se puede usar la indexación en secuencia margarita y/o una indexación híbrida.

**[0112]** Debido a que la duración de la muestra para diferentes pistas puede no ser la misma (por ejemplo, las muestras de vídeo pueden mostrarse durante 33 ms, mientras que una muestra de audio puede durar 80 ms), es posible que las diferentes pistas en un fragmento de película no comiencen ni terminen precisamente a la misma hora, es decir, el audio puede comenzar un poco antes o un poco después del vídeo, dándose lo contrario para el fragmento anterior, para compensar. Para evitar ambigüedades, las marcas de tiempo especificadas en los datos de desviación de tiempo y bytes pueden especificarse en relación con una pista particular y esta puede ser la misma pista para cada representación. Habitualmente esta será la pista de vídeo. Esto permite al cliente identificar exactamente la siguiente trama de vídeo cuando está cambiando las representaciones.

**[0113]** Se debe tener cuidado durante la presentación para mantener una relación estricta entre las escalas de tiempo de la pista y el tiempo de presentación, para garantizar una reproducción y un mantenimiento sin problemas de la sincronización de audio/vídeo a pesar del problema anterior.

**[0114]** La Fig. 7 ilustra algunos ejemplos, como un índice simple 700 y un índice jerárquico 702.

[0115] A continuación se proporcionan dos ejemplos específicos de una ventana que contiene un mapa de segmentos, una denominada ventana de índice de tiempo ("TIDX") y otra llamada ("SIDX"). La definición sigue la estructura de la ventana de acuerdo con el formato de archivo de medios de base ISO. Otros diseños para que dichas ventanas definan una sintaxis similar y con la misma semántica y funcionalidad deben ser evidentes para el lector.

5

### Ventana de índice de tiempo

#### Definición

10 [0116]

Tipo de ventana: "tidx"  
 Contenedor: Archivo  
 Obligatorio: No  
 Cantidad: Cualquier número cero o uno

15

[0117] La ventana de índice de tiempo puede proporcionar un conjunto de índices de desviación de tiempo y bytes que asocian ciertas regiones del archivo con ciertos intervalos de tiempo de la presentación. La ventana de índice de tiempo puede incluir un campo targettype, que indica el tipo de datos referenciados. Por ejemplo, una ventana de índice de tiempo con targettype "moof" proporciona un índice a los fragmentos de medios contenidos en el archivo en términos de desviaciones de tiempo y bytes. Una ventana de índice de tiempo con targettype de ventana de índice de tiempo se puede usar para construir un índice de tiempo jerárquico, lo cual permite a los usuarios del archivo navegar rápidamente a la parte requerida del índice.

20

[0118] El índice de segmento puede contener, por ejemplo, la siguiente sintaxis:

```
aligned(8) class TimeIndexBox
extends FullBox('frai') {
unsigned int(32) targettype;
unsigned int(32) time_reference_track_ID;
unsigned int(32) number_of_elements;
unsigned int(64) first_element_offset;
unsigned int(64) first_element_time;
for(i=1; i <= number_of_elements; i++)
{
bit (1) random_access_flag;
unsigned int(31) length;
unsigned int(32) deltaT;
}
}
```

25 **Semántica**

[0119]

30

targettype: es el tipo de datos de ventana a los que hace referencia esta ventana de índice de tiempo. Puede ser la cabecera de fragmento de película ("moof") o la ventana de índice de tiempo ("tidx").

time-reference\_track\_id: indica la pista con respecto a la cual se especifican las desviaciones de tiempo en este índice.

35

number\_of elements: el número de elementos indexados por esta ventana de índice de tiempo.

first\_element\_offset: La desviación de bytes desde el inicio del archivo del primer elemento indexado.

40

first\_element\_time: el tiempo de inicio del primer elemento indexado, utilizando la escala de tiempo especificada en la ventana de cabecera de medios de la pista identificada por time\_reference\_track\_id.

random\_access\_flag: Uno si el tiempo de inicio del elemento es un punto de acceso aleatorio. Cero de lo contrario.

45

longitud: La longitud del elemento indexado en bytes

deltaT: La diferencia en términos de la escala de tiempo especificada en el la ventana de cabecera de medios de la pista identificada por el identificador de pista de referencia de tiempo entre el tiempo de inicio de este elemento y el tiempo de inicio del siguiente elemento.

## 5 Ventana de índice de segmento

[0120] La ventana de índice de segmento ("sidx") proporciona un índice compacto de los fragmentos de película y otras ventanas de índice de segmento en un segmento. Hay dos estructuras de bucle en la ventana de índice de segmento. El primer bucle documenta la primera muestra del subsegmento, es decir, la muestra en el primer fragmento de película al que hace referencia el segundo bucle. El segundo bucle proporciona un índice del subsegmento. El contenedor para la ventana "sidx" es el archivo o segmento directamente.

### Sintaxis

#### 15 [0121]

```
aligned(8) class SegmentIndexBox extends FullBox('sidx', version, 0) {
  unsigned int(32) reference_track_ID;
  unsigned int(16) track_count;
  unsigned int(16) reference_count;
  for (i=1; i<= track_count; i++)
  {
    unsigned int(32) track_ID;
    if (version==0)
    {
      unsigned int(32) decoding_time;
    } else
    {
      unsigned int(64) decoding_time;
    }
  }
  for(i=1; i <= reference_count; i++)
  {
    bit(1) reference_type;
    unsigned int(31) reference_offset;
    unsigned int(32) subsegment_duration;
    bit(1) contains_RAP;
    unsigned int(31) RAP_delta_time;
  }
}
```

### Semántica:

#### 45 [0122]

reference\_track\_ID proporciona el track\_ID para la pista de referencia.

número de pistas: el número de pistas indexadas en el siguiente bucle (1 o mayor);

50 reference\_count: el número de elementos indexados por segundo bucle (1 o mayor);

ID de pista: el ID de una pista para la cual se incluye un fragmento de pista en el primer fragmento de película identificado por este índice; exactamente un track\_ID en este bucle es igual al reference\_track\_ID;

55 tiempo de decodificación: el tiempo de decodificación para la primera muestra en la pista identificada por track\_ID en el fragmento de película al que hace referencia el primer elemento en el segundo bucle, expresado en la escala de tiempo de la pista (como se documenta en el campo de escala de tiempo de la ventana de cabecera de medios de la pista);

60 reference\_type: cuando se establece en 0, indica que la referencia es a una ventana de fragmento de película ("moof"); cuando se establece en 1, indica que la referencia es a una ventana de índice de segmento ("sidx");

reference\_offset: la distancia en bytes desde el primer byte que sigue a la ventana de índice de segmento que contiene, hasta el primer byte de la ventana referenciada;

65

subsegment\_duration: cuando la referencia es a la ventana de índice de segmento, este campo lleva la suma de los campos subsegment\_duration en el segundo bucle de esa ventana; cuando la referencia es a un fragmento de película, este campo lleva la suma de las duraciones de muestra de las muestras en la pista de referencia, en el fragmento de película indicado y los fragmentos de película subsiguientes hasta el primer fragmento de película documentado por la siguiente entrada en el bucle, o el final del subsegmento, lo que sea anterior; la duración se expresa en la escala de tiempo de la pista (como se documenta en el campo de escala de tiempo de la ventana de cabecera de medios de la pista);

contains\_RAP: cuando la referencia es a un fragmento de película, entonces este bit puede ser 1 si el fragmento de pista dentro de ese fragmento de película para la pista con track\_ID igual a reference\_track\_ID contiene al menos un punto de acceso aleatorio, de lo contrario, este bit se establece en 0; cuando la referencia es a un índice de segmento, entonces este bit se establece en 1 solo si alguna de las referencias en ese índice de segmento tiene este bit establecido en 1, y 0 en caso contrario;

Tiempo delta de RAP: si contains\_RAP es 1, proporciona el tiempo de presentación (composición) de un punto de acceso aleatorio (RAP); reservado con el valor 0 si contains\_RAP es 0. El tiempo se expresa como la diferencia entre el tiempo de decodificación de la primera muestra del subsegmento documentado por esta entrada y el tiempo de presentación (composición) del punto de acceso aleatorio, en la pista con track\_ID igual a reference\_track\_ID.

## **Diferencias entre TIDX y SIDX**

[0123] El SIDX y el SIDX proporcionan la misma funcionalidad con respecto a la indexación. El primer bucle del SIDX proporciona además una temporización global para el primer fragmento de película, pero la temporización global también puede estar contenida en el propio fragmento de la película, ya sea absoluta o relativa a la pista de referencia.

[0124] El segundo bucle del SIDX implementa la funcionalidad del TIDX. Específicamente, el SIDX permite tener una mezcla de objetivos para la referencia para cada índice al que hace referencia reference\_type, mientras que el TIDX solo hace referencia al TIDX o solo al MOOF. El número de elementos en TIDX corresponde al reference\_count en SIDX, el time-reference\_track\_id en TIDX corresponde a reference\_track\_ID en SIDX, el first\_element\_offset en TIDX corresponde al reference\_offset en la primera entrada del segundo bucle, el first\_element\_time en TIDX corresponde al decoding\_time de la pista de referencia en el primer bucle, el indicador de acceso aleatorio en TIDX corresponde al contains\_RAP en el SIDX con la libertad adicional de que en el SIDX el RAP no necesariamente se coloque al inicio del fragmento, y por lo tanto requiriendo RAP\_delta\_time, la longitud en TIDX corresponde al conjunto de referencia en SIDX y, finalmente, el deltaT en TIDX corresponde a la subsegment\_duration en SIDX. Por lo tanto las funcionalidades de las dos ventanas son equivalentes.

## **Tamaño de bloque variable y bloques Sub-GoP**

[0125] Para los medios de vídeo, la relación entre la estructura de codificación de vídeo y la estructura de bloque para las peticiones puede ser importante. Por ejemplo, si cada bloque comienza con un punto de búsqueda, como un punto de acceso aleatorio ("RAP"), y cada bloque representa un período igual de tiempo de vídeo, entonces la posición de al menos algunos puntos de búsqueda en los medios de vídeo es fija y los puntos de búsqueda ocurrirán a intervalos regulares dentro de la codificación del vídeo. Como es bien sabido por los expertos en la técnica de la codificación de vídeo, la eficiencia de compresión puede mejorarse si los puntos de búsqueda se colocan de acuerdo con las relaciones entre las tramas de vídeo, y en particular, si se colocan en tramas que tienen poco en común con las tramas anteriores. Este requisito de que los bloques representan cantidades iguales de tiempo impone una restricción en la codificación del vídeo, por lo que la compresión puede ser subóptima.

[0126] Es deseable permitir que la posición de los puntos de búsqueda dentro de una presentación de vídeo sea elegida por el sistema de codificación de vídeo, en lugar de requerir puntos de búsqueda en posiciones fijas. Al permitir que el sistema de codificación de vídeo elija los puntos de búsqueda, se obtiene una compresión de vídeo mejorada y, por lo tanto, se puede proporcionar una mejor calidad de medios de vídeo utilizando un ancho de banda disponible dado, lo cual da como resultado una experiencia de usuario mejorada. Los sistemas de transmisión de petición de bloques actuales pueden requerir que todos los bloques tengan la misma duración (en tiempo de vídeo), y que cada bloque debe comenzar con un punto de búsqueda y esto es, por lo tanto, una desventaja de los sistemas existentes.

[0127] Ahora se describe un nuevo sistema de transmisión de petición de bloques que ofrece ventajas sobre lo anterior. En un modo de realización, el proceso de codificación de vídeo de una primera versión del componente de vídeo puede configurarse para elegir las posiciones de los puntos de búsqueda con el fin de optimizar la eficiencia de compresión, pero con el requisito de que haya un máximo en la duración entre los puntos de búsqueda. Este último requisito restringe la elección de puntos de búsqueda mediante el proceso de codificación y, por lo tanto, reduce la eficiencia de compresión. Sin embargo, la reducción en la eficiencia de compresión es pequeña en comparación con la incurrida si se requieren posiciones fijas regulares para los puntos de búsqueda, siempre que el máximo en la duración entre los puntos de búsqueda no sea demasiado pequeño (por ejemplo, mayor que aproximadamente un segundo). Además, si el máximo en la duración entre los puntos de búsqueda es de unos pocos segundos, la reducción

de la eficiencia de compresión en comparación con la posición completamente libre de los puntos de búsqueda es en general muy pequeña.

- 5 **[0128]** En muchos modos de realización, incluido este modo de realización, puede ser que algunos RAP no sean puntos de búsqueda, es decir, puede haber una trama que sea un RAP que esté entre dos puntos de búsqueda consecutivos que no se elijan como un punto de búsqueda, por ejemplo porque el RAP está demasiado cerca en tiempo de los puntos de búsqueda circundantes, o porque la cantidad de datos de medios entre el punto de búsqueda que precede o sigue al RAP y el RAP es demasiado pequeña.
- 10 **[0129]** La posición de los puntos de búsqueda dentro de todas las demás versiones de la presentación de medios se puede restringir para que sea la misma que los puntos de búsqueda en una primera versión (por ejemplo, la mayor velocidad de datos de medios). Esto reduce la eficiencia de compresión de estas otras versiones en comparación con permitir que el codificador elija libremente los puntos de búsqueda.
- 15 **[0130]** El uso de los puntos de búsqueda típicamente requeriría que una trama fuera descifrable independientemente, lo cual en general da como resultado una baja eficiencia de compresión para esa trama. Las tramas que no requieren ser descodificables de forma independiente pueden codificarse con referencia a los datos en otras tramas, lo cual en general aumenta la eficiencia de compresión para esa trama en una cantidad que depende de la cantidad de elementos comunes entre la trama a codificar y las tramas de referencia. La elección eficiente del posicionamiento del punto de búsqueda elige preferentemente como trama de punto de búsqueda una trama que tiene un bajo nivel de elementos comunes con las tramas anteriores y, por lo tanto, minimiza la penalización de la eficiencia de compresión en la que se incurre al codificar la trama de forma que sea descodificable de forma independiente.
- 20 **[0131]** Sin embargo, el nivel de elementos comunes entre una trama y las tramas de referencia potenciales está altamente correlacionado entre las diferentes representaciones del contenido, ya que el contenido original es el mismo. Como resultado, la restricción de los puntos de búsqueda en otras variantes para que sean las mismas posiciones que los puntos de búsqueda en la primera variante no crea una gran diferencia en la eficiencia de compresión.
- 25 **[0132]** La estructura del punto de búsqueda se usa preferentemente para determinar la estructura del bloque. Preferentemente, cada punto de búsqueda determina el inicio de un bloque, y puede haber uno o más bloques que abarquen los datos entre dos puntos de búsqueda consecutivos. Dado que la duración entre los puntos de búsqueda no es fija para la codificación con buena compresión, no se requiere que todos los bloques tengan la misma duración de reproducción. En algunos modos de realización, los bloques se alinean entre las versiones del contenido; es decir, si hay un bloque que abarca un grupo específico de tramas en una versión del contenido, entonces hay un bloque que abarca el mismo grupo de tramas en otra versión del contenido. Los bloques para una versión dada del contenido no se superponen y cada trama del contenido está contenida exactamente en un bloque de cada versión.
- 30 **[0133]** Una característica habilitadora que permite el uso eficiente de las duraciones variables entre los puntos de búsqueda y, por lo tanto, los GoP de duración variable, es la indexación de segmentos o el mapa de segmentos que se puede incluir en un segmento o proporcionar a un cliente por otros medios, es decir, se trata de metadatos asociados con este segmento en esta representación que pueden proporcionarse que comprenden el tiempo de inicio y la duración de cada bloque de la presentación. El cliente puede usar estos datos de indexación de segmentos al determinar el bloque en el que comenzar la presentación cuando el usuario ha solicitado que la presentación comience en un punto particular que esté dentro de un segmento. Si no se proporcionan dichos metadatos, entonces la presentación puede comenzar solo al principio del contenido, o en un punto aleatorio o aproximado cercano al punto deseado (por ejemplo, al elegir el bloque de inicio dividiendo el punto de inicio solicitado (en el tiempo) por la duración media del bloque para dar el índice del bloque inicial).
- 35 **[0134]** En un modo de realización, cada bloque puede proporcionarse como un archivo separado. En otro modo de realización, se pueden agregar múltiples bloques consecutivos en un solo archivo para formar un segmento. En este segundo modo de realización, se pueden proporcionar metadatos para cada versión que comprenden el tiempo de inicio y la duración de cada bloque y la desviación de bytes dentro del archivo en el que comienza el bloque. Estos metadatos pueden proporcionarse en respuesta a una petición de protocolo inicial, es decir, disponibles por separado del segmento o archivo, o pueden estar contenidos dentro del mismo archivo o segmento que los bloques mismos, por ejemplo, al principio del archivo. Como quedará claro para los expertos en la técnica, estos metadatos pueden codificarse en una forma comprimida, como la codificación gzip o delta o en forma binaria, para reducir los recursos de red necesarios para transportar los metadatos al cliente.
- 40 **[0135]** La Fig. 6 muestra un ejemplo de indexación de segmentos donde los bloques son de tamaño variable y donde el alcance de los bloques es un GoP parcial, es decir, una cantidad parcial de los datos de medios entre un RAP y el siguiente RAP. En este ejemplo, los puntos de búsqueda están indicados por el indicador RAP, en el que un valor de indicador de RAP de 1 indica que el bloque comienza con o contiene un RAP o punto de búsqueda, y en el que un indicador de RAP de 0 indica que el bloque no contiene un RAP o punto de búsqueda. En este ejemplo, los primeros tres bloques, es decir, los bytes 0 a 157.033, comprenden el primer GoP, que tiene una duración de presentación de 1,623 segundos, con un tiempo de presentación que va desde 20 segundos hasta el contenido hasta 21,623 segundos. En este ejemplo, el primero de los tres primeros bloques comprende ,485 segundos de tiempo de presentación y
- 45   
50   
55   
60   
65

comprende los primeros 50.245 bytes de los datos de medios en el segmento. En este ejemplo, los bloques 4, 5 y 6 comprenden el segundo GoP, los bloques 7 y 8 comprenden el tercer GoP, y los bloques 9, 10 y 11 comprenden el cuarto GoP. Tenga en cuenta que puede haber otros RAP en los datos de medios que no están designados como puntos de búsqueda y, por lo tanto, no se señalan como RAP en el mapa de segmentos.

**[0136]** Refiriéndose de nuevo a la Fig. 6, si el cliente o el receptor desea acceder al contenido comenzando en una desviación de tiempo de aproximadamente 22 segundos en la presentación de medios, entonces el cliente podría primero usar otra información, como el MPD que se describe con más detalle más adelante, para determinar en primer lugar que los datos de medios relevantes están dentro de este segmento. El cliente puede descargar la primera parte del segmento para obtener la indexación de segmentos, que en este caso es solo de unos pocos bytes, por ejemplo, utilizando una petición de rango de bytes HTTP. Usando la indexación de segmentos, el cliente puede determinar que el primer bloque que debe descargar es el primer bloque con una desviación de tiempo que es como máximo de 22 segundos y que comienza con un RAP, es decir, es un punto de búsqueda. En este ejemplo, aunque el bloque 5 tiene una desviación de tiempo inferior a 22 segundos, es decir, su desviación de tiempo es de 21,965 segundos, la indexación de segmentos indica que el bloque 5 no comienza con un RAP y, por lo tanto, basándose en la indexación de segmentos, el cliente selecciona descargar el bloque 4, ya que su tiempo de inicio es como máximo de 22 segundos, es decir, su desviación de tiempo es 21,623 segundos, y comienza con un RAP. Por lo tanto, basándose en la indexación de segmentos, el cliente realizará una petición de rango HTTP a partir de la desviación de bytes 157.034.

**[0137]** Si la indexación de segmentos no estuviera disponible, es posible que el cliente tenga que descargar todos los 157.034 bytes de datos anteriores antes de descargar estos datos, lo cual conlleva un tiempo de inicio, o un tiempo de zapping, mucho mayor, y una descarga inútil de datos que no es útil. De forma alternativa, si la indexación de segmentos no estuviera disponible, el cliente podría aproximarse al inicio de los datos deseados dentro del segmento, pero la aproximación podría ser deficiente y podría perder el tiempo apropiado y a continuación podría requerirse retroceder, lo cual nuevamente aumentaría el retardo de inicio.

**[0138]** En general, cada bloque abarca una parte de los datos de medios que, junto con los bloques anteriores, pueden reproducirse en un reproductor de medios. Por lo tanto, la estructura de bloqueo y la señalización de la estructura de bloqueo de indexación de segmentos para el cliente, ya sea contenida dentro del segmento o proporcionada al cliente por otros medios, puede mejorar significativamente la capacidad del cliente para proporcionar un cambio rápido de canal y una reproducción perfecta ante las interrupciones y variaciones de red. El soporte de los bloques de duración variable y los bloques que abarcan solo partes de un GoP, según lo habilitado por la indexación de segmentos, puede mejorar significativamente la experiencia de transmisión. Por ejemplo, refiriéndose nuevamente a la Fig. 6 y al ejemplo descrito anteriormente, donde el cliente desea comenzar la reproducción aproximadamente a los 22 segundos de la presentación, el cliente puede solicitar, a través de una o más peticiones, los datos dentro del bloque 4, y a continuación enviar esta información al reproductor de medios tan pronto como esté disponible para iniciar la reproducción. Por lo tanto, en este ejemplo, la reproducción comienza tan pronto como se reciban los 42.011 bytes del bloque 4 en el cliente, lo cual permite un rápido tiempo de zapping. Si, por el contrario, el cliente necesitaba solicitar el GoP completo antes de que comenzara la reproducción, el tiempo de zapping sería mayor, ya que se trata de 144.211 bytes de datos.

**[0139]** En otros modos de realización, los RAP o los puntos de búsqueda también pueden ocurrir en la mitad de un bloque, y puede haber datos en la indexación de segmentos que indican dónde está ese RAP o punto de búsqueda dentro del bloque o fragmento. En otros modos de realización, la desviación de tiempo puede ser el tiempo de decodificación de la primera trama dentro del bloque, en lugar del tiempo de presentación de la primera trama dentro del bloque.

**[0140]** Las Figs. 8 (a) y (b) ilustran un ejemplo de tamaño de bloque variable de una estructura de puntos de búsqueda alineada a través de una pluralidad de versiones o representaciones; la Fig. 8(a) ilustra el tamaño del bloque variable con puntos de búsqueda alineados sobre una pluralidad de versiones de un flujo de medios, mientras que la Fig. 8(b) ilustra el tamaño del bloque variable con puntos de búsqueda no alineados sobre una pluralidad de versiones de un flujo de medios.

**[0141]** El tiempo se muestra en la parte superior en segundos, y los bloques y los puntos de búsqueda de los dos segmentos para las dos representaciones se muestran de izquierda a derecha en términos de su temporización con respecto a esta línea de tiempo, y por lo tanto la longitud de cada bloque mostrado es proporcional a su tiempo de reproducción y no proporcional al número de bytes en el bloque. En este ejemplo, la indexación de segmentos para ambos segmentos de las dos representaciones tendría las mismas desviaciones de tiempo para los puntos de búsqueda, pero potencialmente diferentes números de bloques o fragmentos entre los puntos de búsqueda, y diferentes desviaciones de bytes a bloques debido a las diferentes cantidades de datos de medios en cada bloque. En este ejemplo, si el cliente desea pasar de la representación 1 a la representación 2 en el momento de la presentación aproximadamente 23 segundos, entonces el cliente podría solicitar hasta el bloque 1.2 en el segmento para la representación 1, y comenzar a solicitar el segmento para la representación 2 comenzando en el bloque 2.2, y por lo tanto, el cambio ocurriría en la presentación coincidiendo con el punto de búsqueda 1.2 en la representación 1, que es al mismo tiempo que el punto de búsqueda 2.2 en la representación 2.

**[0142]** Como debe quedar claro a partir de los anterior, el sistema de transmisión de petición de bloques descrito no restringe la codificación de vídeo para colocar puntos de búsqueda en posiciones específicas dentro del contenido y esto reduce uno de los problemas de los sistemas existentes.

5 **[0143]** En los modos de realización descritos anteriormente, se organiza de modo que los puntos de búsqueda para las diversas representaciones de la misma presentación de contenido estén alineados. Sin embargo, en muchos casos, es preferible relajar este requisito de alineación. Por ejemplo, a veces ocurre que las herramientas de codificación se han utilizado para generar las representaciones que no tienen la capacidad de generar representaciones alineadas de puntos de búsqueda. Como otro ejemplo, la presentación del contenido puede  
10 codificarse en diferentes representaciones de forma independiente, sin alineación de puntos de búsqueda entre diferentes representaciones. Como otro ejemplo, una representación puede contener más puntos de búsqueda, ya que tiene velocidades más bajas y, más comúnmente, debe cambiarse o contiene puntos de búsqueda para soportar modos de truco como avance rápido, retroceso rápido o búsqueda rápida. Por lo tanto, es deseable proporcionar procedimientos que hagan que un sistema de transmisión de petición de bloques sea capaz de tratar de manera  
15 eficiente y sin problemas los puntos de búsqueda no alineados en las diversas representaciones para una presentación de contenido.

**[0144]** En este modo de realización, las posiciones de los puntos de búsqueda en las representaciones pueden no alinearse. Los bloques se construyen de tal manera que un nuevo bloque comienza en cada punto de búsqueda y, por lo tanto, puede que no haya alineación entre los bloques de diferentes versiones de la presentación. En la Fig. 8 (b) se muestra un ejemplo de dicha estructura de punto de búsqueda no alineada entre diferentes representaciones. El tiempo se muestra en la parte superior en segundos, y los bloques y los puntos de búsqueda de los dos segmentos para las dos representaciones se muestran de izquierda a derecha en términos de su temporización con respecto a esta línea de tiempo, y por lo tanto la longitud de cada bloque mostrado es proporcional a su tiempo de reproducción y no proporcional al número de bytes en el bloque. En este ejemplo, la indexación de segmentos para ambos segmentos de las dos representaciones tendría desviaciones de tiempo potencialmente diferentes para los puntos de búsqueda, y también números de bloques o fragmentos potencialmente diferentes entre los puntos de búsqueda y diferentes desviaciones de bytes a bloques debido a las diferentes cantidades de datos de medios en cada bloque. En este ejemplo, si el cliente desea cambiar de la representación 1 a la representación 2 en el tiempo de presentación aproximadamente 25 segundos, entonces el cliente podría solicitar hasta el bloque 1.3 en el segmento para la representación 1, y comenzar a solicitar el segmento para la representación 2 comenzando en el bloque 2.3, y por lo tanto, el cambio ocurriría en la presentación coincidiendo con el punto de búsqueda 2.3 en la representación 2, que se encuentra en medio de la reproducción del bloque 1.3 en la representación 1, y por lo tanto algunos de los medios para el bloque 1.2 no se reproducirían (aunque los datos de medios para las tramas del bloque 1.3 que no se reproducen pueden tener que cargarse en la memoria intermedia del receptor para descodificar otras tramas del bloque 1.3 que se reproducen).

**[0145]** En este modo de realización, el funcionamiento del selector de bloque 123 se puede modificar de tal manera que siempre que se requiera seleccionar un bloque de una representación que sea diferente de la versión previamente seleccionada, se escoge el último bloque cuya primera trama no sea posterior a la trama posterior a la última trama del último bloque seleccionado.

**[0146]** Este último modo de realización descrito puede eliminar el requisito de restringir las posiciones de los puntos de búsqueda dentro de versiones distintas a la primera versión y, por lo tanto, aumenta la eficiencia de compresión de estas versiones, lo cual da como resultado una presentación de mayor calidad para un ancho de banda disponible dado y esto brinda una mejor experiencia de usuario. Una consideración adicional es que las herramientas de codificación de vídeo que realizan la función de alineación del punto de búsqueda en múltiples codificaciones (versiones) del contenido pueden no estar ampliamente disponibles y, por lo tanto, una ventaja de este último modo de realización descrito es que se pueden usarse las herramientas de codificación de vídeo actualmente disponibles. Otra ventaja es que la codificación de diferentes versiones del contenido puede realizarse en paralelo sin necesidad de coordinación entre los procesos de codificación para las diferentes versiones. Otra ventaja es que las versiones adicionales del contenido se pueden codificar y agregar a la presentación en un momento posterior, sin tener que proporcionar las herramientas de codificación con las listas de posiciones de puntos de búsqueda específicos.

**[0147]** En general, cuando las imágenes se codifican como grupos de imágenes (GoPs), la primera imagen en la secuencia puede ser un punto de búsqueda, pero no siempre es así.

### **División óptima de bloques**

60 **[0148]** Un tema de preocupación en un sistema de transmisión de petición de bloques es la interacción entre la estructura de los medios codificados, por ejemplo, los medios de vídeo, y la estructura de bloque utilizada para las peticiones de bloque. Como sabrán los expertos en la técnica de la codificación de vídeo, a menudo ocurre que el número de bits necesarios para la representación codificada de cada trama de vídeo varía, a veces sustancialmente, de trama a trama. Como resultado, la relación entre la cantidad de datos recibidos y la duración de los medios codificados por esos datos puede no ser sencilla. Además, la división de datos de medios en bloque dentro de un sistema de transmisión de petición de bloques agrega una dimensión adicional de complejidad. En particular, en  
65

algunos sistemas, los datos de medios de un bloque pueden no reproducirse hasta que se haya recibido todo el bloque, por ejemplo, la disposición de los datos de medios dentro de un bloque o las dependencias entre muestras de medios dentro de un bloque del uso de códigos de borrado pueden dar como resultado esta propiedad. Como resultado de estas interacciones complejas entre el tamaño del bloque y la duración del bloque y la posible necesidad de recibir un bloque completo antes de comenzar la reproducción, es común que los sistemas cliente adopten un enfoque conservador en el que los datos de medios se almacenan en memoria intermedia antes de que comience la reproducción. Dicho almacenamiento en memoria intermedia da como resultado un tiempo de zapping largo en el canal y, por lo tanto, una experiencia de usuario deficiente.

[0149] La solicitud de patente de Estados Unidos n.º 12/705,202 presentada el 12 de febrero de 2010 describe "procedimientos de partición de bloques", que son procedimientos nuevos y eficientes para determinar cómo dividir un flujo de datos en bloques contiguos basándose en la estructura subyacente del flujo de datos y describe varias ventajas de estos procedimientos en el contexto de un sistema de transmisión. Ahora se describe un modo de realización adicional de la invención para aplicar los procedimientos de partición de bloques de la solicitud de patente de Estados Unidos n.º 12/705,202 a un sistema de transmisión de bloques. Este procedimiento puede comprender organizar los datos de medios que se presentarán en un orden de tiempo de presentación aproximado, de modo que el tiempo de reproducción de cualquier elemento dado de los datos de medios (por ejemplo, una trama de vídeo o muestra de audio) difiera del de cualquier elemento de datos de medios adyacentes en menos de un umbral proporcionado. Los datos de medios así ordenados pueden considerarse un flujo de datos en el lenguaje de la solicitud de patente de Estados Unidos n.º 12/705,202, y cualquiera de los procedimientos de la Solicitud de Patente de Estados Unidos n.º 12/705,202 aplicados a este flujo de datos identifica límites de bloques con el flujo de datos. Los datos entre cualquier par de límites de bloques adyacentes se consideran un "Bloque" en el lenguaje de esta divulgación y los procedimientos de esta divulgación se aplican para proporcionar la presentación de los datos de medios dentro de un sistema de transmisión de peticiones de bloques. Como quedará claro para los expertos en la técnica al leer esta divulgación, las diversas ventajas de los procedimientos descritos en la solicitud de patente de Estados Unidos n.º 12/705,202 pueden realizarse a continuación en el contexto de un sistema de transmisión de petición de bloques.

[0150] Como se describe en la solicitud de patente de Estados Unidos n.º 12/705,202, la determinación de la estructura de bloque de un segmento, incluidos los bloques que abarcan GoPs parciales o porciones de más que en GoP, puede afectar a la capacidad del cliente para habilitar los rápidos tiempos de zapping. En la solicitud de patente de Estados Unidos n.º 12/705,202, se proporcionaron procedimientos que, dado un tiempo de inicio objetivo, proporcionarían una estructura de bloque y una velocidad de descarga de destino que garantizaría que si el cliente comenzaba a descargar la representación en cualquier punto de búsqueda y comenzaba la reproducción después de que haya transcurrido el tiempo de inicio objetivo, entonces la reproducción continuaría sin interrupciones siempre y cuando en cada momento la cantidad de datos que el cliente haya descargado sea al menos la velocidad de descarga objetivo multiplicada por el tiempo transcurrido desde el comienzo de la descarga. Es ventajoso para el cliente tener acceso al tiempo de inicio objetivo y a la velocidad de descarga de destino, ya que esto proporciona al cliente un medio para determinar cuándo comenzar a reproducir la representación en el momento más temprano y permite al cliente continuar reproduciendo la representación siempre que la descarga cumpla con las condiciones descritas anteriormente. Por lo tanto, el procedimiento descrito más adelante proporciona un medio para incluir el tiempo de inicio objetivo y la velocidad de descarga objetivo dentro de la descripción de presentación de medios, de modo que se pueda utilizar para los fines descritos anteriormente.

#### **Modelo de datos de presentación de medios**

[0151] La Fig. 5 ilustra las posibles estructuras del almacén de contenido que se muestra en la Fig. 1, incluidos los segmentos y los archivos de descripción de presentación de medios ("MPD"), y un desglose de segmentos, temporización y otra estructura dentro de un archivo MPD. Ahora se describirán los detalles de las posibles implementaciones de estructuras o archivos MPD. En muchos ejemplos, el MPD se describe como un archivo, pero también se pueden usar estructuras que no son archivos.

[0152] Como se ilustra allí, el almacén de contenido 110 contiene una pluralidad de segmentos de origen 510, MPDs 500 y segmentos de reparación 512. Un MPD podría comprender registros de período 501, que a su vez podrían comprender registros de representación 502, que contienen información de segmento 503, tales como referencias a segmentos de inicialización 504 y segmentos de medios 505.

[0153] La Fig. 9 (a) ilustra un ejemplo de tabla de metadatos 900, mientras que la Fig. 9 (b) ilustra un ejemplo de cómo un cliente de transmisión HTTP 902 obtiene la tabla de metadatos 900 y los bloques de medios 904 a través de una conexión a un servidor de transmisión HTTP 906.

[0154] En los procedimientos descritos en el presente documento, se proporciona una "Descripción de presentación de medios" que comprende información sobre las representaciones de la presentación de medios que están disponibles para el cliente. Las representaciones pueden ser alternativas en el sentido de que el cliente selecciona una de las diferentes alternativas, o pueden ser complementarias en el sentido de que el cliente selecciona varias de las representaciones, cada una posiblemente también de un conjunto de alternativas, y las presenta de manera conjunta. Las representaciones pueden asignarse ventajosamente a grupos, con el cliente programado o configurado

para comprender que, para las representaciones en un grupo, son alternativas entre sí, mientras que las representaciones de diferentes grupos son tales que más de una representación debe presentarse conjuntamente. En otras palabras, si hay más de una representación en un grupo, el cliente elige una representación de ese grupo, una representación del siguiente grupo, etc., para formar una presentación.

**[0155]** La información que describe las representaciones puede incluir ventajosamente detalles de los códecs de medios aplicados, incluidos los perfiles y niveles de esos códecs que se requieren para descodificar la representación, las frecuencias de tramas de vídeo, la resolución de vídeo y las velocidades de datos. El cliente que recibe la Descripción de Presentación de Medios puede usar esta información para determinar de antemano si una representación es adecuada para descodificación o presentación. Esto representa una ventaja porque si la información diferencial solo estuviera contenida en los datos binarios de la representación, sería necesario solicitar los datos binarios de todas las representaciones y analizar y extraer la información relevante para descubrir información sobre su idoneidad. Estas peticiones múltiples y la extracción de los datos del anexo de análisis pueden tomar algún tiempo, lo cual daría como resultado un tiempo de inicio y, por lo tanto, una experiencia de usuario deficiente.

**[0156]** Además, la descripción de presentación de medios puede incluir información que restringe las peticiones de los clientes basándose en la hora del día. Por ejemplo, para un servicio en vivo, el cliente puede estar restringido a solicitar partes de la presentación que estén cerca del "tiempo de retransmisión actual". Esto representa una ventaja, ya que para la retransmisión en vivo puede ser conveniente eliminar los datos de la infraestructura de servicio para el contenido que se retransmitió más de un umbral proporcionado antes del tiempo de retransmisión actual. Esto puede ser conveniente para la reutilización de los recursos de almacenamiento dentro de la infraestructura de servicio. Esto también puede ser deseable dependiendo del tipo de servicio ofrecido, por ejemplo, en algunos casos, una presentación puede estar disponible solo en vivo debido a un cierto modelo de suscripción de dispositivos cliente de recepción, mientras que otras presentaciones de medios pueden estar disponibles en vivo y bajo demanda. y otras presentaciones pueden estar disponibles solo en vivo para una primera clase de dispositivos cliente, solo bajo demanda para una segunda clase de dispositivos cliente, y una combinación de en vivo o bajo demanda para una tercera clase de dispositivos cliente. Los procedimientos descritos en el Modelo de datos de presentación de medios (a continuación) permiten que el cliente esté informado de tales políticas para que el cliente pueda evitar realizar peticiones y ajustar las ofertas al usuario, para datos que pueden no estar disponibles en la infraestructura de servicio. De forma alternativa, por ejemplo, el cliente puede presentar una notificación al usuario de que estos datos no están disponibles.

**[0157]** En un modo de realización adicional de la invención, los segmentos de medios pueden cumplir con el formato de archivo de medios base ISO descrito en ISO/IEC 14496-12 o especificaciones derivadas (como el formato de archivo 3GP descrito en la especificación técnica 26 244 de 3GPP). La sección Uso del formato de archivo 3GPP (arriba) describe mejoras novedosas del formato de archivo de medios base ISO, lo cual permite un uso eficiente de las estructuras de datos de este formato de archivo dentro de un sistema de transmisión de petición de bloques. Como se describe en esta referencia, se puede proporcionar información dentro del archivo que permita una asignación rápida y eficiente entre los segmentos de tiempo de la presentación de medios y los rangos de bytes dentro del archivo. Los datos de medios en sí pueden estructurarse de acuerdo con la construcción de Fragmento de Película definida en ISO/IEC14496-12. Esta información que proporciona las desviaciones de tiempo y bytes puede estar estructurada jerárquicamente o como un único bloque de información. Esta información puede proporcionarse al inicio del archivo. La provisión de esta información mediante una codificación eficiente como se describe en la sección Uso del formato de archivo 3GPP hace que el cliente pueda recuperar esta información rápidamente, por ejemplo, utilizando peticiones GET parciales de HTTP, en el caso de que el protocolo de descarga de archivos utilizado por el sistema de transmisión de peticiones de bloque seas HTTP, lo cual da como resultado un breve tiempo de cambio de inicio, búsqueda o transmisión y, por lo tanto, una mejor experiencia de usuario.

**[0158]** Las representaciones en una presentación de medios se sincronizan en una línea de tiempo global para asegurar un cambio perfecto entre las representaciones, que típicamente son alternativas, y para asegurar la presentación síncrona de dos o más representaciones. Por lo tanto, el tiempo de muestra de los medios contenidos en las representaciones dentro de una presentación de medios de transmisión HTTP adaptable puede relacionarse con una línea de tiempo global continua en múltiples segmentos.

**[0159]** Un bloque de medios codificados que contienen medios de múltiples tipos, por ejemplo, audio y vídeo, puede tener diferentes tiempos de finalización de presentación para los diferentes tipos de medios. En un sistema de transmisión de petición de bloques, dichos bloques de medios pueden reproducirse consecutivamente de tal manera que cada tipo de medios se reproduzca continuamente y, por lo tanto, se puedan reproducir muestras de medios de un tipo de un bloque antes de muestras de medios de otro tipo del bloque anterior, que aquí se denomina "empalme de bloques continuo". De forma alternativa, tales bloques de medios pueden reproducirse de tal manera que la muestra más temprana de cualquier tipo de un bloque se reproduzca después de la muestra más reciente de cualquier tipo del bloque anterior, que se denomina en el presente documento "empalme de bloques discontinuo". El empalme de bloques continuo puede ser apropiado cuando ambos bloques contienen medios del mismo elemento de contenido y la misma representación, codificados en secuencia, o en otros casos. Típicamente, dentro de una representación, el empalme de bloques continuo se puede aplicar al empalmar dos bloques. Esto es ventajoso, ya que se puede aplicar la codificación existente y se puede realizar la segmentación sin necesidad de alinear las pistas de los medios en los

límites de los bloques. Esto se ilustra en la Fig. 10, donde el flujo de vídeo 1000 comprende el bloque 1202 y otros bloques, con RAP como RAP 1204.

**Descripción de presentación de medios**

5 [0160] Una presentación de medios puede verse como una colección estructurada de archivos en un servidor de transmisión HTTP. El cliente de transmisión HTTP puede descargar información suficiente para presentar el servicio de transmisión al usuario. Las representaciones alternativas pueden comprender uno o más archivos 3GP o partes de  
10 archivos 3GP que se ajustan al formato de archivo 3GPP o al menos a un conjunto bien definido de estructuras de datos que pueden convertirse fácilmente desde o hacia un archivo 3GP.

15 [0161] Una presentación de medios puede describirse mediante una descripción de presentación de medios. La descripción de presentación de medios (MPD) puede contener metadatos que el cliente puede usar para construir peticiones de archivos apropiadas, por ejemplo, peticiones GET de HTTP, para acceder a los datos en el momento adecuado y para proporcionar el servicio de transmisión al usuario. La descripción de presentación de medios puede proporcionar información suficiente para que el cliente de transmisión HTTP seleccione los archivos 3GPP y los fragmentos de archivos apropiados. Las unidades que se señalan al cliente para que sean accesibles se denominan segmentos.

20 [0162] Entre otros, una descripción de presentación de medios puede contener elementos y atributos de la siguiente manera.

***Elemento MediaPresentationDescription***

25 [0163] Un elemento que encapsula los metadatos utilizados por el cliente de transmisión HTTP para proporcionar el servicio de transmisión al usuario final. El elemento MediaPresentationDescription puede contener uno o más de los siguientes atributos y elementos.

30 [0164] Versión: Número de versión del protocolo para asegurar la extensibilidad.

[0165] PresentationIdentifier: Información tal que la presentación pueda identificarse de manera única entre otras presentaciones. También puede contener campos o nombres privados.

35 [0166] UpdateFrequency: Frecuencia de actualización de la descripción de presentación de medios, es decir, con qué frecuencia el cliente puede volver a cargar la descripción de presentación de medios real. Si no está presente, la presentación de medios puede ser estática. Actualizar la presentación de medios puede significar que la presentación de medios no se puede almacenar en memoria caché.

40 [0167] MediaPresentationDescriptionURI: URI para fechar la descripción de presentación de los medios.

[0168] Flujo: Describe el tipo de flujo o presentación de medios: vídeo, audio o texto. Un tipo de transmisión de vídeo puede contener audio y puede contener texto.

45 [0169] Servicio: Describe el tipo de servicio con atributos adicionales. Los tipos de servicio pueden ser en vivo y bajo demanda. Esto se puede usar para informar al cliente que no se permite la búsqueda y el acceso más allá de la hora actual.

50 [0170] MaximumClientPreBufferTime: Una cantidad máxima de tiempo que el cliente puede pre-almacenar en memoria intermedia el flujo de medios. Esta temporización puede diferenciar la transmisión de la descarga progresiva si el cliente está restringido a la descarga más allá de este tiempo máximo de pre-almacenamiento en memoria intermedia. Es posible que el valor no esté presente, lo cual indica que no se pueden aplicar restricciones en términos de pre-almacenamiento en memoria intermedia.

55 [0171] SafetyGuardIntervalLiveService: Información sobre el tiempo máximo de entrega de un servicio en vivo en el servidor. Proporciona una indicación al cliente sobre qué información ya está disponible en el momento actual. Esta información puede ser necesaria si se espera que el cliente y el servidor funcionen en la hora UTC y no se proporciona ninguna sincronización de hora estricta.

60 [0172] TimeShiftBufferDepth: Información sobre cuánto puede ir atrás el cliente en un servicio en vivo en relación con la hora actual. Mediante la extensión de esta profundidad, se pueden permitir los servicios de visualización y actualización de turnos sin cambios específicos en el aprovisionamiento del servicio.

[0173] LocalCachingPermitted: Este indicador indica si el cliente HTTP puede almacenar en memoria caché los datos descargados localmente después de que se haya reproducido.

65

**[0174]** LivePresentationInterval: Contiene intervalos de tiempo durante los cuales la presentación puede estar disponible especificando StartTimes y EndTimes. StartTime indica el tiempo de inicio de los servicios y EndTime indica el tiempo de finalización del servicio. Si no se especifica el tiempo de finalización, entonces el tiempo de finalización es desconocida en el momento actual y UpdateFrequency puede garantizar que los clientes tengan acceso al tiempo de finalización antes del tiempo de finalización real del servicio.

**[0175]** IntervaloDeDisponibilidadDisponibilidad: El intervalo de presentación indica la disponibilidad del servicio en la red. Se pueden proporcionar múltiples intervalos de presentación. Es posible que el cliente HTTP no pueda acceder al servicio fuera de una ventana de tiempo específica. Mediante el aprovisionamiento del intervalo OnDemand, se puede especificar una visualización adicional de turnos. Este atributo también puede estar presente para un servicio en vivo. En caso de que esté presente para un servicio en vivo, el servidor puede garantizar que el cliente pueda acceder al servicio como servicio OnDemand durante todos los intervalos de disponibilidad proporcionados. Por lo tanto, el LivePresentationInterval no se puede superponer con ningún OnDemandA AvailabilityInterval.

**[0176]** MPDFileInfoDynamic: Describe la construcción dinámica predeterminada de los archivos en la presentación de medios. A continuación se proporcionan más detalles. La especificación predeterminada en el nivel MPD puede ahorrar repeticiones innecesarias si se usan las mismas reglas para varias o todas las representaciones alternativas.

**[0177]** Descripción de MPDCodec: Describe los principales códecs predeterminados en la presentación de medios. A continuación se proporcionan más detalles. La especificación predeterminada en el nivel MPD puede ahorrar repeticiones innecesarias si se utilizan los mismos códecs para varias o todas las representaciones.

**[0178]** MPDMoveBoxHeaderSizeDoesNotChange: Un indicador para indicar si la cabecera de MoveBox cambia de tamaño entre los archivos individuales dentro de la presentación de medios completa. Este indicador se puede utilizar para optimizar la descarga y solo puede estar presente en el caso de formatos de segmentos específicos, especialmente aquellos para los cuales los segmentos contienen la cabecera moov.

**[0179]** FileURIPattern: Un patrón utilizado por el cliente para generar mensajes de petición para archivos dentro de la presentación de medios. Los diferentes atributos permiten la generación de URI únicos para cada uno de los archivos dentro de la presentación de medios. El URI base puede ser un URI HTTP.

**[0180]** Representación alternativa: Describe una lista de representaciones.

#### ***Elemento AlternativeRepresentation:***

**[0181]** Un elemento XML que encapsula todos los metadatos para una representación. El elemento AlternativeRepresentation puede contener los siguientes atributos y elementos.

**[0182]** RepresentationID: Una ID única para esta representación alternativa específica dentro de la presentación de los medios.

**[0183]** FilesInfoStatic: Proporciona una lista explícita de los tiempos de inicio y el URI de todos los archivos de una presentación alternativa. El aprovisionamiento estático de la lista de archivos puede proporcionar la ventaja de una descripción exacta del momento de la presentación del medio, pero puede no ser tan compacta, especialmente si la representación alternativa contiene muchos archivos. Además, los nombres de los archivos pueden tener nombres arbitrarios.

**[0184]** FilesInfoDynamic: Proporciona una forma implícita de construir la lista de los tiempos de inicio y el URI de una presentación alternativa. El aprovisionamiento dinámico de la lista de archivos puede proporcionar la ventaja de una representación más compacta. Si solo se proporciona la secuencia de los tiempos de inicio, entonces las ventajas de tiempo también se mantienen aquí, pero los nombres de los archivos se construirán de forma dinámica basándose en FilePatternURI. Si solo se proporciona la duración de cada segmento, entonces la representación es compacta y puede ser adecuada para su uso dentro de los servicios en vivo, pero la generación de los archivos puede estar gobernada por el tiempo global.

**[0185]** APMoveBoxHeaderSizeDoesNotChange: Un indicador que indica si la cabecera de MoveBox cambia de tamaño entre los archivos individuales dentro de la descripción alternativa. Este indicador se puede utilizar para optimizar la descarga y solo puede estar presente en el caso de formatos de segmentos específicos, especialmente aquellos para los cuales los segmentos contienen la cabecera moov.

**[0186]** APCodecDescription: Describe los principales códecs de archivos en la presentación alternativa.

#### ***Elemento de descripción de medios***

**[0187]** MediaDescription: Un elemento que puede encapsular todos los metadatos para los medios contenidos en esta representación. Específicamente, puede contener información sobre las pistas en esta presentación alternativa,

así como la agrupación recomendada de pistas, si corresponde. El atributo MediaDescription contiene los siguientes atributos:

TrackDescription: Un atributo XML que encapsula todos los metadatos para los medios contenidos en esta representación. El atributo TrackDescription contiene los siguientes atributos:

5 Identificador de pista: Una ID única para la pista dentro de la representación alternativa. Esto se puede usar en caso de que la pista sea parte de una descripción de agrupación.

[0188] Velocidad de transmisión de bits: La velocidad de transmisión de bits de la pista.

10 [0189] TrackCodecDescripción: Un atributo XML que contiene todos los atributos en el códec utilizado en esta pista. El atributo TrackCodecDescription contiene los siguientes atributos:

MediaName: Un atributo que define el tipo de medio. Los tipos de medios incluyen "audio", "vídeo", "texto", "aplicación" y "mensaje".

15 [0190] Códec: CodecType incluyendo perfil y nivel.

[0191] LanguageTag: LanguageTag si es aplicable.

20 [0192] MaxWidth, MaxHeight: Para vídeo, altura y ancho de vídeo contenido en píxel.

[0193] SamplingRate: Para audio, frecuencia de muestreo

GroupDescription: Un atributo que proporciona recomendaciones al cliente para la agrupación adecuada basándose en diferentes parámetros.

25

[0194] GroupType: Un tipo basado en el cual el cliente puede decidir cómo agrupar pistas.

[0195] La información en una descripción de presentación de medios es utilizada ventajosamente por un cliente de transmisión HTTP para realizar peticiones de archivos/segmentos o partes de los mismos en momentos apropiados, seleccionando los segmentos de representaciones adecuadas que se adapten con sus capacidades, por ejemplo en términos de ancho de banda de acceso, capacidades de visualización, capacidades de códec, etc., así como las preferencias del usuario, como el idioma, etc. Además, dado que la descripción de presentación de medios describe representaciones que están alineadas en el tiempo y asignadas a una línea de tiempo global, el cliente también puede usar la información en el MPD durante una presentación de medios en curso para iniciar acciones apropiadas para cambiar las representaciones, presentar representaciones conjuntamente o buscar dentro de la presentación de los medios.

35

### **Tiempos de inicio del segmento de señalización**

40 [0196] Una representación se puede dividir, en el tiempo, en múltiples segmentos. Existe un problema de temporización entre pistas entre el último fragmento de un segmento y el siguiente fragmento del siguiente segmento. Además, existe otro problema de temporización en el caso de que se utilicen segmentos de duración constante.

45 [0197] Usar la misma duración para cada segmento puede tener la ventaja de que el MPD es compacto y estático. Sin embargo, cada segmento aún puede comenzar en un punto de acceso aleatorio. Por lo tanto, la codificación de vídeo puede estar restringida para proporcionar puntos de acceso aleatorio en estos puntos específicos, o las duraciones reales de los segmentos puede no ser exactamente como se especifica en el MPD. Puede ser deseable que el sistema de transmisión no imponga restricciones innecesarias en el proceso de codificación de vídeo, por lo que la segunda opción puede ser la preferida.

50

[0198] Específicamente, si la duración del archivo se especifica en el MPD como  $d$  segundos, entonces el noveno archivo puede comenzar con el punto de acceso aleatorio en el momento  $(n-1)d$  o inmediatamente después.

55 [0199] En este enfoque, cada archivo puede incluir información sobre el tiempo de inicio exacto del segmento en términos de tiempo de presentación global. Tres formas posibles de señalar esto incluyen:

(1) En primer lugar, restringir el tiempo de inicio de cada segmento a la temporización exacta especificado en el MPD. Pero entonces el codificador de medios puede no tener ninguna flexibilidad en la ubicación de las tramas IDR y puede requerir una codificación especial para la transmisión de archivos.

60

(2) En segundo lugar, agregar el tiempo exacto de inicio al MPD para cada segmento. Para el caso bajo demanda, la compacidad de MPD puede reducirse. Para el caso en vivo, esto puede requerir una actualización regular del MPD, lo cual puede reducir la escalabilidad.

65 (3) En tercer lugar, agregar el tiempo global o el tiempo de inicio exacto en relación con el tiempo de inicio anunciado de la representación o el tiempo de inicio anunciado del segmento en el MPD al segmento en el sentido

de que el segmento contiene esta información. Esto podría agregarse a una nueva ventana dedicada a la transmisión adaptable. Esta ventana también puede incluir la información provista por la ventana "TIDX" o "SIDX". Una consecuencia de este tercer enfoque es que al buscar una posición particular cerca del comienzo de uno de los segmentos, el cliente puede, con base en el MPD, elegir el segmento subsiguiente al que contiene el punto de búsqueda requerido. Una respuesta simple en este caso puede ser mover el punto de búsqueda hacia el inicio del segmento recuperado (es decir, al siguiente punto de acceso aleatorio después del punto de búsqueda). En general, los puntos de acceso aleatorio se proporcionan al menos cada pocos segundos (y con frecuencia hay poca ganancia de codificación al hacerlos menos frecuentes) y, en el peor de los casos, el punto de búsqueda puede moverse unos segundos más tarde de lo especificado. De forma alternativa, el cliente podría determinar, al recuperar la información de la cabecera para el segmento, que el punto de búsqueda solicitado está, de hecho, en el segmento anterior y solicitar ese segmento. Esto puede dar como resultado un aumento ocasional en el tiempo requerido para ejecutar la operación de búsqueda.

### Lista de segmentos accesibles

**[0200]** La presentación de medios comprende un conjunto de representaciones, cada una de las cuales proporciona una versión diferente de la codificación para el contenido de medios original. Las representaciones en sí contienen ventajosamente información sobre los parámetros de diferenciación de la representación en comparación con otros parámetros. También contienen, ya sea explícita o implícitamente, una lista de segmentos accesibles.

**[0201]** Los segmentos pueden diferenciarse en segmentos atemporales que contienen solo metadatos y segmentos de medios que contienen principalmente datos de medios. La Descripción de Presentación de los Medios ("MPD") identifica y asigna ventajosamente diferentes atributos a cada uno de los segmentos, ya sea de forma implícita o explícita. Los atributos asignados ventajosamente a cada segmento comprenden el período durante el cual un segmento es accesible, los recursos y protocolos a través de los cuales los segmentos son accesibles. Además, los segmentos de medios se asignan ventajosamente a atributos tales como el tiempo de inicio del segmento en la presentación de medios y la duración del segmento en la presentación de medios.

**[0202]** Cuando la presentación de medios es del tipo "bajo demanda", como lo indica ventajosamente un atributo en la descripción de presentación de medios, como el `OnDemandAvailabilityInterval`, la descripción de presentación de medios típicamente describe los segmentos completos y también proporciona una indicación de cuándo los segmentos son accesibles y cuándo los segmentos no son accesibles. Los tiempos de inicio de los segmentos se expresan ventajosamente en relación con el inicio de la presentación de medios, de modo que dos clientes que inician la reproducción de las mismas presentaciones de medios, pero en diferentes momentos, pueden usar la misma descripción de presentación de medios, así como los mismos segmentos de medios. Esto mejora ventajosamente la capacidad de almacenar en memoria caché los segmentos.

**[0203]** Cuando la presentación de los medios es del tipo "en vivo", como lo indica ventajosamente un atributo en la descripción de presentación de los medios, como el `Servicio` de atributos, los segmentos que comprenden la presentación de los medios más allá de la hora real del día en general no se generan o al menos no son accesibles a pesar de los segmentos están completamente descritos en el MPD. Sin embargo, con la indicación de que el servicio de presentación de medios es del tipo "en vivo", el cliente puede producir una lista de segmentos accesibles junto con los atributos de temporización para un tiempo interno del cliente AHORA en tiempo de reloj de pared basándose en la información contenida en el MPD y el tiempo de descarga del MPD. El servidor opera ventajosamente en un sentido que hace que los recursos sean accesibles de manera tal que un cliente de referencia que opera con la instancia del MPD en el tiempo de reloj de pared AHORA puede acceder a los recursos.

**[0204]** Específicamente, el cliente de referencia produce una lista de segmentos accesibles junto con los atributos de tiempo para un tiempo interno del cliente AHORA en tiempo de reloj de pared basándose en la información contenida en el MPD y el tiempo de descarga del MPD. A medida que avanza el tiempo, el cliente usará el mismo MPD y creará una nueva lista de segmentos accesibles que se puede usar para reproducir continuamente la presentación de medios. Por lo tanto, el servidor puede anunciar segmentos en un MPD antes de que estos segmentos sean realmente accesibles. Esto es ventajoso, ya que reduce la frecuente actualización y descarga del MPD.

**[0205]** Supongamos que una lista de segmentos, cada uno con el tiempo de inicio,  $tS$ , se describe explícitamente mediante una lista de reproducción en elementos como `FileInfoStatic` o implícitamente mediante el uso de un elemento como `FileInfoDynamic`. A continuación se describe un procedimiento ventajoso para generar una lista de segmentos usando `FileInfoDynamic`. Basándose en esta regla de construcción, el cliente tiene acceso a una lista de URI para cada representación,  $r$ , referida aquí como `FileURI(r,i)`, y un tiempo de inicio  $tS(r,i)$  para cada segmento con índice  $i$ .

**[0206]** El uso de la información en el MPD para crear la ventana de tiempo accesible de los segmentos se puede realizar usando las siguientes reglas.

**[0207]** Para un servicio de tipo "bajo demanda", como lo indica ventajosamente un atributo como `Servicio`, si la hora actual del reloj de pared en el cliente AHORA está dentro de cualquier rango de disponibilidad, expresado ventajosamente por un elemento MPD como `OnDemandAvailabilityInterval`, entonces todos los segmentos descritos

de esta presentación bajo petición son accesibles. Si la hora actual del reloj de pared en el cliente AHORA está fuera de cualquier rango de disponibilidad, entonces no se puede acceder a ninguno de los segmentos descritos de esta presentación bajo petición.

5 **[0208]** Para un servicio de tipo "en vivo", como lo indica ventajosamente un atributo como *Servicio*, el tiempo de inicio  $tS(r,i)$  expresa ventajosamente el tiempo de disponibilidad en tiempo de reloj de pared. El tiempo de inicio de la disponibilidad puede obtenerse como una combinación del tiempo de servicio en vivo del evento y algún tiempo de entrega en el servidor para la captura, codificación y publicación. El tiempo para este proceso puede, por ejemplo, especificarse en el MPD, por ejemplo, utilizando un intervalo de seguridad tG especificado por ejemplo, especificado como `SafetyGuardIntervalLiveService` en el MPD. Esto proporcionaría la diferencia mínima entre la hora UTC y la disponibilidad de los datos en el servidor de transmisión HTTP. En otro modo de realización, el MPD especifica explícitamente el tiempo de disponibilidad del segmento en el MPD sin proporcionar el tiempo de respuesta como una diferencia entre el tiempo en vivo del evento y el tiempo de respuesta. En las siguientes descripciones, se supone que los tiempos globales se especifican como tiempos de disponibilidad. Una u otra habilidad ordinaria en la técnica de la retransmisión de medios en vivo puede obtener esta información a partir de la información adecuada en la descripción de presentación de los medios después de leer esta descripción.

10 **[0209]** Si la hora actual del reloj de pared en el cliente AHORA está fuera de cualquier rango del intervalo de presentación en vivo, expresado ventajosamente por un elemento MPD como `LivePresentationInterval`, entonces no se puede acceder a ninguno de los segmentos descritos de esta presentación en vivo. Si la hora actual del reloj de pared en el cliente AHORA está dentro del intervalo de presentación en vivo, entonces al menos ciertos segmentos de los segmentos descritos de esta presentación en vivo pueden ser accesibles.

15 **[0210]** La restricción de los segmentos accesibles se rige por los siguientes valores:  
La hora del reloj de pared AHORA (según disponibilidad para el cliente).

20 **[0211]** La profundidad de memoria intermedia de turnos permitida tTSB, por ejemplo, especificada como `TimeShiftBufferDepth` en la descripción de presentación de medios.

25 **[0212]** A un cliente en el tiempo de evento relativo  $t_i$  solamente se le puede permitir solicitar segmentos con tiempos de inicio  $tS(r,i)$  en el intervalo de (AHORA - tTSB) y AHORA o en un intervalo tal que el tiempo de finalización del segmento con la duración  $d$  también se incluye, lo cual da como resultado un intervalo de (NOW - tTSB-d) y AHORA.

### **Actualización del MPD**

30 **[0213]** En algunos modos de realización, el servidor no conoce de antemano el archivo o el localizador de segmentos y los tiempos de inicio de los segmentos, como por ejemplo la ubicación del servidor, cambiarán, o la presentación de medios incluye algún anuncio de un servidor diferente, o la duración de la presentación de medios es desconocida, o el servidor quiere ofuscar el localizador para los siguientes segmentos.

35 **[0214]** En tales modos de realización, el servidor solo puede describir segmentos que ya son accesibles o se pueden obtener poco después de que se haya publicado esta instancia del MPD. Además, en algunos modos de realización, el cliente consume ventajosamente los medios cerca de los medios descritos en el MPD, de manera que el usuario experimenta el programa de medios contenido lo más cerca posible de la generación del contenido de medios. Tan pronto como el cliente prevé que llega al final de los segmentos de medios descritos en el MPD, solicita ventajosamente una nueva instancia del MPD para continuar la reproducción continua con la expectativa de que el servidor haya publicado un nuevo MPD que describa nuevos segmentos de medios. El servidor genera ventajosamente nuevas instancias del MPD y actualiza el MPD de tal manera que los clientes pueden confiar en los procedimientos para las actualizaciones continuas. El servidor puede adaptar sus procedimientos de actualización de MPD junto con la generación de segmentos y la publicación a los procedimientos de un cliente de referencia que actúa como un cliente común que puede actuar.

40 **[0215]** Si una nueva instancia del MPD solo describe un avance de tiempo breve, entonces los clientes deben solicitar con frecuencia nuevas instancias de MPD. Esto puede dar como resultado problemas de escalabilidad y tráfico innecesario de enlace ascendente y descendente debido a peticiones frecuentes innecesarias.

45 **[0216]** Por lo tanto, es importante, por un lado, describir los segmentos en la medida de lo posible en el futuro sin necesariamente hacerlos accesibles todavía, y por otro lado, permitir actualizaciones imprevistas en el MPD para expresar nuevas ubicaciones de servidores, permitir la inserción de nuevo contenido como Como anuncios o proporcionar cambios en los parámetros de códec.

50 **[0217]** Además, en algunos modos de realización, la duración de los segmentos de medios puede ser pequeña, como en el intervalo de varios segundos. La duración de los segmentos de medios es ventajosamente flexible para ajustarse a tamaños de segmentos adecuados que pueden optimizarse para las propiedades de entrega o almacenamiento en memoria caché, para compensar el retardo de extremo a extremo en servicios en vivo u otros aspectos relacionados con el almacenamiento o la entrega de segmentos, o por otras razones. Especialmente en los

casos en que los segmentos son pequeños en comparación con la duración de la presentación de medios, entonces una cantidad significativa de recursos de segmentos de medios y tiempos de inicio deben describirse en la descripción de presentación de medios. Como resultado, el tamaño de la descripción de presentación de medios puede ser grande, lo cual puede afectar negativamente al tiempo de descarga de la descripción de presentación de medios y, por lo tanto, afectar el retardo de inicio de la presentación de medios y también al uso de ancho de banda en el enlace de acceso. Por lo tanto, es ventajoso no solo permitir la descripción de una lista de segmentos de medios usando listas de reproducción, sino también permitir la descripción usando plantillas o reglas de construcción de URL. Las plantillas y las reglas de construcción de URL se utilizan como sinónimos en esta descripción.

**[0218]** Además, las plantillas pueden usarse ventajosamente para describir los localizadores de segmentos en casos reales más allá de la hora actual. En tales casos, las actualizaciones del MPD por sí mismas no son necesarias, ya que las plantillas describen los localizadores y la lista de segmentos. Sin embargo, aún pueden ocurrir eventos imprevistos que requieran cambios en la descripción de las representaciones o los segmentos. Es posible que se necesiten cambios en una descripción de presentación de medios de transmisión HTTP adaptable cuando el contenido de múltiples fuentes diferentes se empalma, por ejemplo, cuando se ha insertado publicidad. El contenido de diferentes fuentes puede diferir de varias formas. Otra razón, durante las presentaciones en vivo, es que puede ser necesario cambiar las direcciones URL utilizadas para los archivos de contenido para evitar el cambio por error de un servidor de origen en vivo a otro.

**[0219]** En algunos modos de realización, es ventajoso que si se actualiza el MPD, entonces las actualizaciones del MPD se lleven a cabo de manera tal que el MPD actualizado sea compatible con el MPD anterior en el siguiente sentido que el cliente de referencia y, por lo tanto, cualquier cliente implementado genere una lista funcional de segmentos accesibles desde el MPD actualizado para cualquier momento hasta el tiempo de validez del MPD anterior como lo habría hecho desde la instancia anterior del MPD. Este requisito garantiza que (a) los clientes pueden comenzar a usar el nuevo MPD inmediatamente sin sincronización con el antiguo MPD, ya que es compatible con el antiguo MPD antes de la hora de actualización; y (b) el tiempo de actualización no necesita estar sincronizado con el momento en que se produce el cambio real en el MPD. En otras palabras, las actualizaciones del MPD se pueden anunciar con anticipación y el servidor puede reemplazar la instancia anterior del MPD una vez que la nueva información esté disponible sin tener que mantener diferentes versiones del MPD.

**[0220]** Pueden existir dos posibilidades para la temporización de medios a través de una actualización de MPD para un conjunto de representaciones o todas las representaciones. Ya sea (a) la línea de tiempo global existente continúa a través de la actualización de MPD (denominada en el presente documento como una "actualización continua de MPD"), o (b) la línea de tiempo actual finaliza y una nueva línea de tiempo comienza con el segmento que sigue al cambio (denominado en el presente documento una "actualización de MPD discontinua").

**[0221]** La diferencia entre estas opciones puede ser evidente cuando se considera que las pistas de un fragmento de medios, y por lo tanto de un segmento, en general no comienzan y terminan al mismo tiempo debido a la granularidad de la muestra diferente en las pistas. Durante la presentación normal, las muestras de una pista de un fragmento pueden renderizarse *antes que* algunas muestras de otra pista del fragmento *anterior*, es decir, hay algún tipo de superposición entre los fragmentos, aunque puede que no haya superposición *dentro de una sola pista*.

**[0222]** La diferencia entre (a) y (b) es si dicha superposición puede habilitarse a través de una actualización de MPD. Cuando la actualización de MPD se debe a un empalme de contenido completamente separado, tal superposición es en general difícil de lograr ya que el nuevo contenido necesita una nueva codificación para empalmarse con el contenido anterior. Por lo tanto, es ventajoso proporcionar la capacidad de actualizar de forma discontinua la presentación de medios reiniciando la línea de tiempo para ciertos segmentos y posiblemente también definir un nuevo conjunto de representaciones después de la actualización. Además, si el contenido se ha codificado y segmentado de manera independiente, también se evita ajustar las marcas de tiempo para que se ajusten a la línea de tiempo global del contenido anterior.

**[0223]** Cuando la actualización se realiza por razones menores, como solo agregar nuevos segmentos de medios a la lista de segmentos de medios descritos, o si se cambia la ubicación de las URL, se superponen y se pueden permitir actualizaciones continuas.

**[0224]** En el caso de una actualización de MPD discontinua, la línea de tiempo del último segmento de la representación anterior termina en el último tiempo de finalización de la presentación de cualquier muestra en el segmento. La línea de tiempo de la siguiente representación (o, más precisamente, el primer tiempo de presentación del primer segmento de medios de la nueva parte de la presentación de medios, también conocido como nuevo período) comienza típicamente y de forma ventajosa en este mismo instante que el final de la presentación del último período de modo que se garantice la reproducción continua y perfecta.

**[0225]** Los dos casos se ilustran en la Fig. 11.

**[0226]** Es preferible y ventajoso restringir las actualizaciones de MPD a los límites del segmento. La razón para restringir tales cambios o actualizaciones a los límites del segmento es la siguiente. En primer lugar, los cambios en

los metadatos binarios para cada representación, típicamente la cabecera de película, pueden tener lugar al menos en los límites del segmento. En segundo lugar, la descripción de presentación de medios puede contener los punteros (URL) a los segmentos. En cierto sentido, el MPD es la estructura de datos "paraguas" que agrupa todos los archivos de segmentos asociados con la presentación de medios. Para mantener esta relación de contención, cada segmento puede ser referenciado por un solo MPD y cuando el MPD se actualiza, ventajosamente solo se actualiza en un límite del segmento.

**[0227]** En general no se requiere que los límites de los segmentos estén alineados, sin embargo, para el caso de contenido empalmado de diferentes fuentes, y para las actualizaciones de MPD discontinuas en general, tiene sentido alinear los límites de los segmentos (específicamente, que el último segmento de cada representación puede terminar en la misma trama de vídeo y no pueda contener muestras de audio con un tiempo de inicio de presentación posterior al tiempo de presentación de esa trama). Una actualización discontinua puede iniciar a continuación un nuevo conjunto de representaciones en un instante de tiempo común, denominado período. El tiempo de inicio de la validez de este nuevo conjunto de representaciones se proporciona, por ejemplo, mediante un tiempo de inicio de período. El tiempo de inicio relativo de cada representación se restablece a cero y el tiempo de inicio del período coloca el conjunto de representaciones en este nuevo período en la línea de tiempo de la presentación de medios global.

**[0228]** Para actualizaciones continuas de MPD, no se requiere que los límites del segmento estén alineados. Cada segmento de cada representación alternativa puede estar gobernado por una sola Descripción de Presentación de Medios y, por lo tanto, las peticiones de actualización para nuevas instancias de la Descripción de Presentación de Medios, en general activadas por la previsión de que no se describen segmentos de medios adicionales en el MPD operativo, pueden tener lugar en diferentes momentos, dependiendo del conjunto de representaciones consumido, incluido el conjunto de representaciones que se prevé consumir.

**[0229]** Para soportar actualizaciones en elementos y atributos MPD en un caso más general, cualquier elemento, no solo las representaciones o el conjunto de representaciones, puede asociarse con un tiempo de validez. Por lo tanto, si ciertos elementos del MDP deben actualizarse, por ejemplo, cuando se cambia el número de representaciones o se cambian las reglas de construcción de la URL, cada uno de estos elementos puede actualizarse individualmente en momentos específicos, proporcionando múltiples copias del elemento con tiempos de validez diferentes.

**[0230]** La validez se asocia ventajosamente con el tiempo global de los medios, de modo que el elemento descrito asociado con un tiempo de validez es válido en un período de la línea de tiempo global de la presentación de los medios.

**[0231]** Como se analizó anteriormente, en un modo de realización, los tiempos de validez solo se agregan a un conjunto completo de representaciones. Cada conjunto completo forma un período. A continuación, el tiempo de validez forma el tiempo de inicio del período. En otras palabras, en un caso específico del uso del elemento de validez, un conjunto completo de representaciones puede ser válido por un período de tiempo, indicado por un tiempo de validez global para un conjunto de representaciones. El tiempo de validez de un conjunto de representaciones se denomina período. Al comienzo de un nuevo período, la validez de la representación del conjunto anterior caduca y el nuevo conjunto de representaciones es válido. Tenga en cuenta una vez más que los tiempos de validez de los períodos son preferentemente diferentes.

**[0232]** Como se indicó anteriormente, los cambios en la descripción de presentación de los medios tienen lugar en los límites del segmento y, por lo tanto, para cada representación, el cambio de un elemento tiene lugar realmente en el siguiente límite del segmento. A continuación, el cliente puede formar un MPD válido que incluya una lista de segmentos para cada instante de tiempo dentro del tiempo de presentación de los medios.

**[0233]** El empalme de bloques discontinuos puede ser apropiado en los casos en que los bloques contengan datos de medios de diferentes representaciones, o de contenido diferente, por ejemplo, de un segmento de contenido y un anuncio, o en otros casos. Puede ser necesario en un sistema de transmisión de petición de bloques que los cambios en los metadatos de presentación solo se realicen en los límites del bloque. Esto puede ser ventajoso por razones de implementación porque actualizar los parámetros del descodificador de medios dentro de un bloque puede ser más complejo que actualizarlos solo entre bloques. En este caso, puede especificarse ventajosamente que los intervalos de validez descritos anteriormente pueden interpretarse como aproximados, de tal manera que un elemento se considera válido desde el primer límite del bloque no antes del inicio del intervalo de validez especificado hasta el primer límite del bloque no antes que el final del intervalo de validez especificado.

**[0234]** Un modo de realización de ejemplo de lo anterior describe mejoras novedosas en un sistema de transmisión de petición de bloques que se describe en la sección presentada más adelante titulada Cambios en las presentaciones de medios.

### **Segmento de señalización de la duración**

**[0235]** Las actualizaciones discontinuas dividen efectivamente la presentación en una serie de intervalos diferentes, a los que se hace referencia como punto. Cada período tiene su propia línea de tiempo para el tiempo de muestra de

medios. La temporización de representación de los medios dentro de un período puede indicarse ventajosamente especificando una lista compacta separada de duraciones de segmentos para cada período o para cada representación en un período.

5 **[0236]** Un atributo, por ejemplo denominado tiempo de inicio del período, asociado a elementos dentro del MPD puede especificar el tiempo de validez de ciertos elementos dentro del tiempo de presentación de los medios. Este atributo se puede agregar a cualquier elemento (los atributos a los que se les puede asignar una validez se pueden cambiar a elementos) del MPD.

10 **[0237]** Para actualizaciones de MPD discontinuas, los segmentos de todas las representaciones pueden terminar en la discontinuidad. Esto en general implica al menos que el último segmento antes de la discontinuidad tiene una duración diferente que los anteriores. La duración del segmento de señalización puede implicar indicar que todos los segmentos tienen la misma duración o indicar una duración separada para cada segmento. Puede ser deseable tener una representación compacta para una lista de duraciones de segmentos que sea eficiente en el caso de que muchos de ellos tengan la misma duración.

15 **[0238]** Las duraciones de cada segmento en una representación o un conjunto de representaciones se pueden llevar a cabo ventajosamente con una sola secuencia que especifica todas las duraciones de los segmentos para un único intervalo desde el inicio de la actualización discontinua, es decir, el inicio del período hasta el último segmento de medios descrito en el MPD. En un modo de realización, el formato de este elemento es una secuencia de texto que se ajusta a una producción que contiene una lista de entradas de duración de segmento donde cada entrada contiene un atributo de duración *dur* y un multiplicador opcional *mult* del atributo que indica que esta representación contiene <mult> de los primeros segmentos de entrada de duración <dur> de la primera entrada, a continuación <mult> de los segundos segmentos de entrada de duración <dur> de la segunda entrada y así sucesivamente.

20 **[0239]** Cada entrada de duración especifica la duración de uno o más segmentos. Si al valor <dur> le sigue el carácter "\*" y un número, este número especifica el número de segmentos consecutivos con esta duración, en segundos. Si el signo multiplicador "\*" está ausente, el número de segmentos es uno. Si el "\*" está presente sin el número siguiente, entonces todos los segmentos subsiguientes tienen la duración especificada y es posible que no haya más entradas en la lista. Por ejemplo, la secuencia "30 \*" significa que todos los segmentos tienen una duración de 30 segundos. La secuencia "30\*12 10,5" indica 12 segmentos de 30 segundos de duración, seguidos de uno de 10,5 segundos de duración.

25 **[0240]** Si las duraciones de los segmentos se especifican por separado para cada representación alternativa, entonces la suma de las duraciones de los segmentos dentro de cada intervalo puede ser la misma para cada representación. En el caso de las pistas de vídeo, el intervalo puede terminar en la misma trama en cada representación alternativa.

30 **[0241]** Los expertos en la técnica, después de leer esta divulgación, pueden encontrar formas similares y equivalentes para expresar las duraciones de los segmentos de manera compacta.

35 **[0242]** En otro modo de realización, se señala que la duración de un segmento es constante para todos los segmentos en la representación, excepto para el último por un atributo de duración de la señal <duration>. La duración del último segmento antes de una actualización discontinua puede ser más corta siempre que se proporcione el punto de inicio de la próxima actualización discontinua o el inicio de un nuevo período, lo cual implica la duración del último segmento que llega hasta el inicio del siguiente período.

### **Cambios y actualizaciones de metadatos de representación**

40 **[0243]** Los cambios indicativos de metadatos de representación codificados en binario, como los cambios de "moov" de cabecera de película, se pueden lograr de diferentes maneras: (a) puede haber una ventana moov para todas las representaciones en un archivo separado referenciado en el MPD, (b) puede haber la ventana moov para cada representación alternativa en un archivo separado referenciado en cada representación alternativa, (c) cada segmento puede contener una ventana moov y, por lo tanto, es independiente, (d) puede haber una ventana moov para toda la representación en un archivo 3GP junto con MPD.

45 **[0244]** Tenga en cuenta que en el caso de (a) y (b), el único "moov" puede combinarse ventajosamente con el concepto de validez desde arriba en el sentido de que se puede hacer referencia a más ventanas "moov" en un MPD siempre que su validez sea diferente. Por ejemplo, con la definición de un límite de período, la validez de "moov" en el período anterior puede caducar con el inicio del nuevo período.

50 **[0245]** En el caso de la opción (a), se puede asignar un elemento de validez a la referencia a la ventana moov única. Se pueden permitir múltiples cabeceras de presentación, pero solo una puede ser válida a la vez. En otro modo de realización, el tiempo de validez de todo el conjunto de representaciones en un período o todo el período tal como se definió anteriormente se puede usar como tiempo de validez para estos metadatos de representación, típicamente proporcionados como la cabecera moov.

[0246] En el caso de la opción (b), se puede asignar un elemento de validez a la referencia a la ventana moov de cada representación. Se pueden permitir cabeceras de representación múltiple, pero solo una puede ser válida a la vez. En otro modo de realización, el tiempo de validez de la representación completa o el período completo tal como se definió anteriormente se puede usar como tiempo de validez para estos metadatos de representación, típicamente proporcionados como la cabecera moov.

[0247] En el caso de las opciones (c), no se puede agregar ninguna señalización en el MPD, pero se puede agregar una señalización adicional en el flujo de medios para indicar si la ventana moov cambiará para cualquiera de los próximos segmentos. Esto se explica con más detalle a continuación en el contexto de "Actualizaciones de señalización dentro de los metadatos del segmento".

### **Actualizaciones de señalización dentro de los metadatos del segmento**

[0248] Para evitar actualizaciones frecuentes de la descripción de presentación de medios para obtener conocimiento sobre posibles actualizaciones, es conveniente señalar dichas actualizaciones junto con los segmentos de medios. Se puede proporcionar un elemento o elementos adicionales dentro de los propios segmentos de medios que pueden indicar que los metadatos actualizados, como la descripción de presentación de medios, están disponibles y se debe acceder a ellos dentro de un cierto período de tiempo para continuar con éxito la creación de listas de segmentos accesibles. Además, dichos elementos pueden proporcionar un identificador de archivo, como una URL, o información que se puede usar para construir un identificador de archivo, para el archivo de metadatos actualizado. El archivo de metadatos actualizado puede incluir metadatos iguales a los proporcionados en el archivo de metadatos original para la presentación modificada para indicar intervalos de validez junto con metadatos adicionales también acompañados por intervalos de validez. Dicha indicación se puede proporcionar en segmentos de medios de todas las representaciones disponibles para una presentación de medios. Un cliente que acceda a un sistema de transmisión de petición de bloques, al detectar dicha indicación dentro de un bloque de medios, puede utilizar el protocolo de descarga de archivos u otros medios para recuperar el archivo de metadatos actualizado. De este modo, se proporciona al cliente información sobre los cambios en la descripción de presentación de los medios y el momento en que ocurrirán o han ocurrido. Ventajosamente, cada cliente solicita la descripción actualizada de la presentación de medios solo una vez cuando ocurren tales cambios en lugar de "sondear" y recibir el archivo muchas veces para posibles actualizaciones o cambios.

[0249] Entre los ejemplos de cambios se incluyen la adición o eliminación de representaciones, cambios en una o más representaciones, tales como cambios en la velocidad de transmisión de bits, resolución, relación de aspecto, pistas o parámetros de códec incluidos, y cambios en las reglas de construcción de URL, por ejemplo, un servidor de origen diferente para un anuncio. Algunos cambios pueden afectar solo al segmento de inicialización, como el átomo de cabecera de película ("moov") asociado con una representación, mientras que otros cambios pueden afectar la descripción de presentación de medios (MPD).

[0250] En el caso de contenido bajo demanda, estos cambios y su temporización pueden ser conocidos de antemano y se pueden señalar en la descripción de presentación de medios.

[0251] Para el contenido en vivo, los cambios pueden no ser conocidos hasta el punto en que ocurren. Una solución es permitir que la descripción de presentación de medios disponible en una URL específica se actualice dinámicamente y requerir a los clientes que soliciten regularmente este MPD para detectar cambios. Esta solución tiene desventajas en términos de escalabilidad (carga del servidor de origen y eficiencia de memoria caché). En un escenario con un gran número de espectadores, las memorias caché pueden recibir muchas peticiones para el MPD después de que la versión anterior haya caducado de la memoria caché y antes de que se haya recibido la nueva versión y todas estas puedan reenviarse al servidor de origen. Es posible que el servidor de origen necesite procesar constantemente las peticiones de las memorias caché para cada versión actualizada del MPD. Además, las actualizaciones pueden no ser fácilmente alineadas en el tiempo con los cambios en la presentación de los medios.

[0252] Dado que una de las ventajas de la transmisión HTTP es la capacidad de utilizar la infraestructura de Internet estándar y los servicios para la escalabilidad, una solución preferida puede incluir solo archivos "estáticos" (es decir, en memoria caché) y no depender de los archivos de "sondeo" de los clientes para ver si han cambiado.

[0253] Las soluciones se analizan y se proponen para resolver la actualización de los metadatos, incluida la descripción de presentación de los medios y los metadatos de representación binaria, como los átomos "moov", en una presentación de medios de transmisión HTTP adaptable.

[0254] Para el caso de contenido en vivo, los puntos en los que el MPD o "moov" pueden cambiar pueden desconocerse cuando se construye el MPD. Como en general se debe evitar el "sondeo" frecuente del MPD para buscar actualizaciones, por razones de ancho de banda y escalabilidad, las actualizaciones del MPD se pueden indicar "en banda" en los propios archivos del segmento, es decir, cada segmento de medios puede tener la opción de indicar actualizaciones. Dependiendo de los formatos de segmento (a) a (c) de arriba, se pueden señalar diferentes actualizaciones.

**[0255]** En general, la siguiente indicación puede proporcionarse ventajosamente en una señal dentro del segmento: un indicador de que el MPD puede actualizarse antes de solicitar el siguiente segmento dentro de esta representación o cualquier segmento siguiente que tenga un tiempo de inicio mayor que el tiempo de inicio del segmento actual. La actualización se puede anunciar con anticipación, lo cual indica que la actualización solo debe realizarse en cualquier segmento posterior al siguiente. Esta actualización de MPD también se puede usar para actualizar metadatos de representación binarios, como las cabeceras de películas, en caso de que se cambie el localizador del segmento de medios. Otra señal puede indicar que con la finalización de este segmento, no se deben solicitar más segmentos que adelanten el tiempo.

**[0256]** En el caso de que los segmentos estén formateados de acuerdo con el formato del segmento (c), es decir, cada segmento de medios puede contener metadatos autoinicializados, como la cabecera de película, a continuación se puede agregar otra señal que indique que el segmento subsiguiente contiene una cabecera de película actualizada (moov). Esto permite ventajosamente que la cabecera de película se incluya en el segmento, pero la cabecera de película solo debe ser solicitada por el cliente si el segmento anterior indica una actualización de la cabecera de película o en el caso de búsqueda o acceso aleatorio al cambiar de representación. En otros casos, el cliente puede emitir una petición de rango de bytes al segmento que excluye la cabecera de película de la descarga, por lo tanto, ventajosamente ahorrando ancho de banda.

**[0257]** En otro modo de realización adicional, si se señala la indicación de actualización de MPD, entonces la señal también puede contener un localizador como el URL para la Descripción de Presentación de Medios actualizada. El MPD actualizado puede describir la presentación tanto *antes como después* de la actualización, utilizando los atributos de validez, como un período nuevo y antiguo en caso de actualizaciones discontinuas. Esto puede usarse ventajosamente para permitir la visualización de turnos como se describe más adelante, pero también permite ventajosamente que se señalice la actualización de MPD en cualquier momento antes de que los cambios que contiene surtan efecto. El cliente puede descargar inmediatamente el nuevo MPD y aplicarlo a la presentación en curso.

**[0258]** En un modo de realización específico, la señalización de cualquier cambio en la descripción de presentación de medios, las cabeceras Moov o el final de la presentación puede estar contenida en una ventana de información de transmisión que se formatea siguiendo las reglas del formato de segmento utilizando la estructura de ventana del formato de archivo de medios de base ISO. Esta ventana puede proporcionar una señal específica para cualquiera de las diferentes actualizaciones.

### ***Ventana de información de transmisión***

#### **Definición**

##### **[0259]**

Tipo de ventana: "sinf"  
 Contenedor: Ninguno  
 Obligatorio: No  
 Cantidad: Cero o uno.

**[0260]** La ventana de información de transmisión contiene información sobre la presentación de transmisión de la que forma parte el archivo.

#### **Sintaxis**

##### **[0261]**

```
aligned(8) class StreamingInformationBox
    extends FullBox('sinf') {
    unsigned int(32) streaming_information_flags;
    /// The following are optional fields
    string mpd_location
}
```

#### **Semántica**

**[0262]** streaming\_information\_flags contiene el OR lógico de cero o más de los siguientes:

0x00000001	La actualización de la cabecera de película sigue
0x00000002	Actualización de descripción de presentación
0x00000004	Fin de la presentación

[0263] La ubicación de mpd está presente si y solo si los indicadores de *Actualización de Descripción de Presentación* están establecidos y proporciona un Localizador de Recursos Uniforme para la nueva Descripción de Presentación de Medios.

5

### **Ejemplo de caso de uso para actualizaciones de MPD para servicios en vivo**

[0264] Supongamos que un proveedor de servicios desea proporcionar un evento de fútbol en vivo utilizando la transmisión de petición de bloques mejorada que se describe en el presente documento. Quizá millones de usuarios quieran acceder a la presentación del evento. El evento en vivo es interrumpido esporádicamente por las pausas cuando se pide un tiempo muerto o durante otra pausa en la acción, durante la cual se pueden agregar anuncios. En general, no se notifica con antelación o se notifica con muy poca el momento exacto de las pausas.

10

[0265] Es posible que el proveedor de servicios necesite proporcionar infraestructura redundante (por ejemplo, codificadores y servidores) para habilitar un cambio sin interrupciones en caso de que alguno de los componentes falle durante el evento en vivo.

15

[0266] Supongamos que un usuario, Anna, accede al servicio en un bus con su dispositivo móvil, y el servicio está disponible de inmediato. Junto a ella se sienta otro usuario, Paul, que ve el evento en su ordenador portátil. Se marca un gol y ambos celebran este evento al mismo tiempo. Paul le dice a Anna que el primer gol del partido fue aún más emocionante y Anna usa el servicio para poder ver el evento 30 minutos atrás en el tiempo. Después de haber visto el gol, vuelve al evento en vivo.

20

[0267] Para abordar ese caso de uso, el proveedor del servicio debe poder actualizar el MPD, señalar a los clientes que hay un MPD actualizado y permitir que los clientes accedan al servicio de transmisión de manera que pueda presentar los datos casi en tiempo real.

25

[0268] La actualización del MPD es factible de manera asíncrona a la entrega de segmentos, como se explica en el presente documento en otra parte. El servidor puede proporcionar garantías al receptor de que un MPD no se actualiza durante algún tiempo. El servidor puede confiar en el MPD actual. Sin embargo, no se necesita una señalización explícita cuando el MPD se actualiza antes del período mínimo de actualización.

30

[0269] Difícilmente se logra una reproducción completamente síncrona, ya que el cliente puede operar en diferentes instancias de actualización de MPD y, por lo tanto, los clientes pueden tener desviaciones. Usando las actualizaciones de MPD, el servidor puede transmitir cambios y los clientes pueden recibir alertas de cambios, incluso durante una presentación. La señalización en banda segmento a segmento se puede usar para indicar la actualización del MPD, por lo que las actualizaciones podrían limitarse a los límites del segmento, pero eso debería ser aceptable en la mayoría de las aplicaciones.

35

[0270] Se puede agregar un elemento MPD que proporciona el tiempo de publicación en el tiempo de reloj de pared del MPD, así como una ventana de actualización de MPD opcional que se agrega al comienzo de los segmentos para indicar que se requiere una actualización de MPD. Las actualizaciones se pueden realizar jerárquicamente, como con los MPD.

40

[0271] El "Tiempo de publicación" de MPD proporciona un identificador único para el MPD y cuándo se emitió el MPD. También proporciona un anclaje para los procedimientos de actualización.

45

[0272] La ventana de actualización de MPD puede encontrarse en el MPD después de la ventana "styp", y definirse mediante un Tipo de ventana = "mupe", sin necesidad de contenedor, sin ser obligatorio y teniendo una cantidad de cero o uno. La ventana de actualización de MPD contiene información sobre la presentación de medios de los que forma parte el segmento.

50

[0273] La sintaxis de ejemplo es la siguiente:

```
aligned(8) class MPDUpdateBox
  extends FullBox('mupe') {
    unsigned int(3) mpd information flags;
    unsigned int(1) new-location flag;
    unsigned int(28) latest mpd update time;
    /// The following are optional fields
    string mpd location
  }
```

55

60

[0274] La semántica de los diversos objetos de la clase MPDUpdateBox podría ser la siguiente:  
Indicadores de información de mpd: el OR lógico de cero o más de los siguientes:

65

0x00	Actualización de descripción de presentación de medios ahora
0x01	Actualización de descripción de presentación de medios por delante
0x02	Fin de la presentación
0x03-0x07	Reservado

[0275] new\_location flag: si se establece en 1, la nueva descripción de presentación de medios está disponible en una nueva ubicación especificada en mpd\_location.

5 [0276] latest\_mpd\_update time: especifica el tiempo (en ms) cuando la actualización de MPD es necesaria en relación con el tiempo de emisión de MPD del último MPD. El cliente puede optar por actualizar el MPD en cualquier momento entre ahora.

10 [0277] mpd\_location: está presente si y solo si se establece new\_location\_flag y, si es así, mpd\_location proporciona un localizador de recursos uniforme para la nueva descripción de presentación de medios.

[0278] Si el ancho de banda utilizado por las actualizaciones es un problema, el servidor puede ofrecer MPD para ciertas capacidades del dispositivo, de modo que solo se actualicen estas partes.

15 **Visualización de turnos y PVR de red**

[0279] Cuando se soporta la visualización de turnos, puede suceder que durante la vida útil de la sesión sean válidos dos o más MPD o cabeceras de película. En este caso, actualizando el MPD cuando sea necesario, pero agregando el mecanismo de validez o el concepto de período, puede existir un MPD válido para toda la ventana de tiempo. Esto significa que el servidor puede garantizar que cualquier MPD y cabecera de película se anuncien para cualquier período de tiempo que se encuentre dentro de la ventana de tiempo válida para la visualización de turnos. Es responsabilidad del cliente asegurarse de que su MPD y sus metadatos disponibles para su tiempo de presentación actual sean válidos. También se puede soportar la migración de una sesión en vivo a una sesión de PVR en red utilizando solo actualizaciones menores de MPD.

25 **Segmentos de medios especiales**

[0280] Un problema cuando el formato de archivo de ISO/IEC 14496-12 se usa dentro de un sistema de transmisión de petición de bloques es que, como se ha descrito anteriormente, puede ser ventajoso almacenar los datos de medios para una versión única de la presentación en varios archivos, dispuestos en segmentos de tiempo consecutivos. Además, puede ser ventajoso organizar que cada archivo comience con un punto de acceso aleatorio. Además, puede ser ventajoso elegir las posiciones de los puntos de búsqueda durante el proceso de codificación del vídeo y segmentar la presentación en múltiples archivos, cada uno de los cuales comienza con un punto de búsqueda basándose en la selección de puntos de búsqueda que se realizó durante el proceso de codificación, en el que cada punto de acceso aleatorio puede colocarse al principio de un archivo, pero en el que cada archivo comienza con un punto de acceso aleatorio. En un modo de realización con las propiedades descritas anteriormente, los metadatos de presentación, o la Descripción de Presentación de Medios, pueden contener la duración exacta de cada archivo, donde la duración se toma, por ejemplo, como la diferencia entre el tiempo de inicio de los medios de vídeo de un archivo y el tiempo de inicio de los medios de vídeo del siguiente archivo. Basándose en esta información en los metadatos de presentación, el cliente puede construir una asignación entre la línea de tiempo global para la presentación de medios y la línea de tiempo local para los medios dentro de cada archivo.

[0281] En otro modo de realización, el tamaño de los metadatos de presentación puede reducirse ventajosamente especificando en cambio que cada archivo o segmento tiene la misma duración. Sin embargo, en este caso y donde los archivos de medios se construyen de acuerdo con el procedimiento anterior, la duración de cada archivo puede no ser exactamente igual a la duración especificada en la descripción de presentación de medios, ya que es posible que no exista un punto de acceso aleatorio en el punto que es exactamente la duración especificada desde el inicio del archivo.

[0282] Ahora se describe un modo de realización adicional de la invención para proporcionar el funcionamiento correcto del sistema de transmisión de petición de bloques a pesar de la discrepancia mencionada anteriormente. En este procedimiento, se puede proporcionar un elemento dentro de cada archivo que especifique la asignación de la línea de tiempo local de los medios dentro del archivo (lo cual significa que la línea de tiempo comienza desde la marca de tiempo cero respecto a la cual se especifican las marcas de tiempo de descodificación y composición de las muestras de medios en el archivos de acuerdo con ISO/IEC 14496-12) para la línea de tiempo de presentación global. Esta información de asignación puede comprender una sola marca de tiempo en el tiempo de presentación global que corresponde a la marca de tiempo cero en la línea de tiempo del archivo local. La información de asignación también puede comprender de forma alternativa un valor de desviación que especifica la diferencia entre el tiempo de presentación global correspondiente a la marca de tiempo cero en la línea de tiempo del archivo local y el tiempo de presentación global correspondiente al inicio del archivo de acuerdo con la información proporcionada en los metadatos de la presentación.

[0283] El ejemplo para tales ventanas puede ser, por ejemplo, la ventana de tiempo de descodificación del fragmento de pista ("tfdt") o la ventana de ajuste del fragmento de pista ("tfad") junto con la ventana de ajuste de medios del fragmento de pista ("tfma").

5

#### **Cliente de ejemplo que incluye la generación de listas de segmentos**

[0284] Se describirá ahora un ejemplo de cliente. Puede usarse como un cliente de referencia para el servidor para asegurar la generación y actualizaciones correctas del MPD.

10

[0285] Un cliente de transmisión HTTP se guía por la información proporcionada en el MPD. Se supone que el cliente tiene acceso al MPD que recibió en el tiempo  $T$ , es decir, el momento en que pudo recibir un MPD con éxito. La determinación de la recepción exitosa puede incluir que el cliente obtenga un MPD actualizado o que el cliente verifique que el MPD no se ha actualizado desde la recepción exitosa anterior.

15

[0286] Se presenta un ejemplo de comportamiento del cliente. Para proporcionar un servicio de transmisión continua al usuario, el cliente primero analiza el MPD y crea una lista de segmentos accesibles para cada representación para la hora local del cliente en un tiempo actual del sistema, teniendo en cuenta los procedimientos de generación de listas de segmentos que se detallan a continuación, posiblemente utilizando listas de reproducción o usando reglas de construcción de URL. A continuación, el cliente selecciona una o varias representaciones basándose en la información en los atributos de representación y otra información, por ejemplo, el ancho de banda disponible y las capacidades del cliente. Dependiendo de la agrupación, las representaciones pueden presentarse de manera independiente o en conjunto con otras representaciones.

20

25

[0287] Para cada representación, el cliente adquiere los metadatos binarios, como la cabecera "moov" para la representación, si está presente, y los segmentos de medios de las representaciones seleccionadas. El cliente accede al contenido de medios solicitando segmentos o rangos de bytes de segmentos, posiblemente utilizando la lista de segmentos. Inicialmente, el cliente puede almacenar en la memoria intermedia los medios antes de comenzar la presentación y, una vez que la presentación ha comenzado, el cliente continúa consumiendo el contenido de medios solicitando continuamente segmentos o partes de segmentos teniendo en cuenta los procedimientos de actualización de MPD.

30

[0288] El cliente puede cambiar las representaciones teniendo en cuenta la información actualizada de MPD y/o la información actualizada de su entorno, por ejemplo, el cambio del ancho de banda disponible. Con cualquier petición de un segmento de medios que contenga un punto de acceso aleatorio, el cliente puede cambiar a una representación diferente. Cuando se avanza, es decir, el tiempo actual del sistema (conocido como "tiempo AHORA" para representar el tiempo relativo a la presentación) avanza, el cliente consume los segmentos accesibles. Con cada avance en el tiempo AHORA, el cliente posiblemente amplía la lista de segmentos accesibles para cada representación de acuerdo con los procedimientos especificados en el presente documento.

35

40

[0289] Si aún no se ha llegado al final de la presentación de medios y si el tiempo de reproducción actual se encuentra dentro de un umbral para el cual el cliente prevé que se agotarán los medios de los medios descritos en el MPD para cualquier representación que consuma o a ser consumida, el cliente puede solicitar una actualización del MPD, con un nuevo tiempo de recepción de tiempo de recuperación  $T$ . Una vez recibido, el cliente tiene en cuenta el MPD posiblemente actualizado y el nuevo tiempo  $T$  en la generación de las listas de segmentos accesibles. La Figura 29 ilustra un procedimiento para servicios en vivo en diferentes momentos en el cliente.

45

#### **Generación de lista de segmentos accesibles**

[0290] Supongamos que el cliente de transmisión HTTP tiene acceso a un MPD y puede querer generar una lista de segmentos accesibles por un tiempo de reloj de pared AHORA. El cliente está sincronizado con una referencia de tiempo global con cierta precisión, pero ventajosamente no se requiere una sincronización directa con el servidor de transmisión HTTP.

50

[0291] La lista de segmentos accesibles para cada representación se define preferentemente como un par de listas de un tiempo de inicio de segmento y un localizador de segmentos donde el tiempo de inicio de segmento puede definirse como relativo al inicio de la representación sin pérdida de generalidad. El inicio de la representación puede alinearse con el inicio de un período o si se aplica este concepto. De lo contrario, el inicio de la representación puede ser al comienzo de la presentación de medios.

55

60

[0292] El cliente utiliza las reglas de construcción de URL y la temporización como, por ejemplo, se define más adelante en el presente documento. Una vez que se obtiene una lista de segmentos descritos, esta lista se restringe aún más a los accesibles, que pueden ser un subconjunto de los segmentos de la presentación completa de los medios. La construcción se rige por el valor actual del reloj en el momento del cliente AHORA. En general, los segmentos solo están disponibles para cualquier momento AHORA dentro de un conjunto de tiempos de disponibilidad. Para tiempos AHORA fuera de esta ventana, no hay segmentos disponibles. Además, para los servicios

65

en vivo, suponga que el tiempo de comprobación proporciona información sobre hasta qué punto se describen los medios en el futuro. El tiempo de comprobación se define en el eje de tiempo de medios documentado por MPD; cuando el tiempo de reproducción del cliente alcanza el tiempo de comprobación, solicita ventajosamente un nuevo MPD.

5 ; cuando el tiempo de reproducción del cliente alcanza el tiempo de comprobación, solicita ventajosamente un nuevo MPD.

10 **[0293]** A continuación, la lista de segmentos se restringe aún más por el tiempo de comprobación junto con el atributo MPD TimeShiftBufferDepth, de modo que solo los segmentos de medios disponibles son aquellos para los cuales la suma del tiempo de inicio del segmento de medios y el tiempo de inicio de la representación cae en el intervalo entre AHORA menos timeShiftBufferDepth menos la duración del último segmento descrito y el valor más pequeño de tiempo de comprobación o AHORA.

### 15 **Bloques escalables**

20 **[0294]** A veces, el ancho de banda disponible es tan bajo que es improbable que el bloque o los bloques que se reciben actualmente en un receptor se reciban completamente a tiempo para reproducirse sin pausar la presentación. El receptor puede detectar tales situaciones por adelantado. Por ejemplo, el receptor podría determinar que está recibiendo bloques que codifican 5 unidades de medios cada 6 unidades de tiempo, y tiene una memoria intermedia de 4 unidades de medios, de modo que el receptor podría esperar que la presentación se detenga o se pause, unas 24 unidades de tiempo después. Con suficiente antelación, el receptor puede reaccionar ante tal situación, por ejemplo, abandonando el flujo actual de bloques y comenzando a solicitar un bloque o bloques de una representación diferente del contenido, como uno que usa menos ancho de banda por unidad de tiempo de reproducción. Por ejemplo, si el receptor cambia a una representación en la que los bloques se codifican durante al menos un 20 % más de tiempo de vídeo para bloques del mismo tamaño, el receptor podría eliminar la necesidad de detenerse hasta que la situación del ancho de banda mejore.

30 **[0295]** Sin embargo, podría ser inútil que el receptor descarte por completo los datos ya recibidos de la representación abandonada. En un modo de realización de un sistema de transmisión de bloques descrito en el presente documento, los datos dentro de cada bloque pueden codificarse y disponerse de tal manera que ciertos prefijos de los datos dentro del bloque puedan usarse para continuar la presentación sin que el resto del bloque se haya recibido. Por ejemplo, se pueden utilizar técnicas bien conocidas de codificación de vídeo escalable. Los ejemplos de dichos procedimientos de codificación de vídeo incluyen la codificación de vídeo escalable H.264 (SVC) o la escalabilidad temporal de la codificación de vídeo avanzada H.264 (AVC). Ventajosamente, este procedimiento permite que la presentación continúe basándose en la parte de un bloque que se haya recibido, incluso cuando la recepción de un bloque o bloques pueda abandonarse, por ejemplo debido a cambios en el ancho de banda disponible. Otra ventaja es que se puede usar un solo archivo de datos como fuente para múltiples representaciones diferentes del contenido. Esto es posible, por ejemplo, haciendo uso de las peticiones de GET parciales de HTTP que seleccionan el subconjunto de un bloque correspondiente a la representación requerida.

40 **[0296]** Una mejora detallada en el presente documento es un segmento mejorado, un mapa de segmentos escalable. El mapa de segmentos escalable contiene las ubicaciones de las diferentes capas en el segmento de tal manera que el cliente puede acceder a las partes de los segmentos en consecuencia y extraer las capas. En otro modo de realización, los datos de medios en el segmento se ordenan de tal manera que la calidad del segmento aumenta mientras se descargan los datos gradualmente desde el principio del segmento. En otro modo de realización, el aumento gradual de la calidad se aplica a cada bloque o fragmento contenido en el segmento, de manera que las peticiones de fragmentos se pueden hacer para abordar el enfoque escalable.

50 **[0297]** La Fig. 12 es una figura que muestra un aspecto de bloques escalables. En esa figura, un transmisor 1200 genera los metadatos 1202, la capa escalable 1 (1204), la capa escalable 2 (1206) y la capa escalable 3 (1208), con este último retardado. Un receptor 1210 puede usar los metadatos 1202, la capa escalable 1 (1204) y la capa escalable 2 (1206) para presentar la presentación de medios 1212.

### 55 **Capas de escalabilidad independientes**

60 **[0298]** Como se explicó anteriormente, no es deseable que un sistema de transmisión de petición de bloques tenga que detenerse cuando el receptor no puede recibir los bloques solicitados de una representación específica de los datos de medios a tiempo para su reproducción, ya que eso a menudo crea una experiencia de usuario deficiente. Las paradas se pueden evitar, reducirse o disminuirse su efecto restringiendo la velocidad de datos de las representaciones elegidas para que sean mucho menores que el ancho de banda disponible, por lo que es muy improbable que una parte determinada de la presentación no se reciba a tiempo, pero esta estrategia la desventaja de que la calidad de los medios es necesariamente mucho más baja de lo que en principio podría ser soportado por el ancho de banda disponible. Una presentación de menor calidad de la que es posible también puede interpretarse como una mala experiencia de usuario. Por lo tanto, el diseñador de un sistema de transmisión de petición de bloques se enfrenta con una elección en el diseño de los procedimientos del cliente, la programación del cliente o la configuración del hardware, ya sea para solicitar una versión de contenido que tenga una velocidad de datos mucho

menor que el ancho de banda disponible, en cuyo caso el usuario puede sufrir una mala calidad de medios, o para solicitar una versión del contenido que tenga una velocidad de datos cercana al ancho de banda disponible, en cuyo caso el usuario puede sufrir una alta probabilidad de pausas durante la presentación a medida que cambie el ancho de banda disponible.

5 **[0299]** Para manejar tales situaciones, los sistemas de transmisión de bloques descritos en el presente documento pueden configurarse para manejar múltiples capas de escalabilidad de manera independiente, de manera que un receptor puede realizar peticiones en capas y un transmisor puede responder a peticiones en capas.

10 **[0300]** En tales modos de realización, los datos de medios codificados para cada bloque se pueden dividir en múltiples partes diferentes, denominadas aquí "capas", de tal manera que una combinación de capas comprende la totalidad de los datos de medios para un bloque y de tal manera que un cliente que tiene algunos subconjuntos recibidos de las capas puede realizar la descodificación y presentación de una representación del contenido. En este enfoque, el orden de los datos en el flujo es tal que los rangos contiguos aumentan en la calidad y los metadatos reflejan esto.

15 **[0301]** Un ejemplo de una técnica que puede utilizarse para generar capas con la propiedad anterior es la técnica de codificación de vídeo escalable, por ejemplo, como se describe en las normas H.264/SVC de ITU-T. Otro ejemplo de una técnica que se puede usar para generar capas con la propiedad anterior es la técnica de capas de escalabilidad temporal según lo dispuesto en la norma ITU-T H.264/AVC.

20 **[0302]** En estos modos de realización, los metadatos pueden proporcionarse en el MPD o en el propio segmento que permite la construcción de peticiones de capas individuales de cualquier bloque y/o combinación de capas y/o una capa de múltiples bloques y/o una combinación de capas de bloques múltiples. Por ejemplo, las capas que comprenden un bloque pueden almacenarse dentro de un solo archivo y se pueden proporcionar metadatos especificando los rangos de bytes dentro del archivo correspondiente a las capas individuales.

25 **[0303]** Se puede usar un protocolo de descarga de archivos capaz de especificar rangos de bytes, por ejemplo HTTP 1.1, para solicitar capas individuales o capas múltiples. Además, como quedará claro para un experto en la técnica al revisar esta divulgación, las técnicas descritas anteriormente relacionadas con la construcción, petición y descarga de bloques de tamaño variable y combinaciones variables de bloques también se pueden aplicar en este contexto.

### **Combinaciones**

30 **[0304]** Ahora se describen varios modos de realización que pueden ser empleados ventajosamente por un cliente de transmisión de peticiones de bloque para lograr una mejora en la experiencia del usuario y/o una reducción en los requisitos de capacidad de la infraestructura de servicio en comparación con las técnicas existentes mediante el uso de datos de medios divididos en capas como se ha descrito anteriormente.

35 **[0305]** En un primer modo de realización, las técnicas conocidas de un sistema de transmisión de petición de bloques pueden aplicarse con la modificación de que diferentes versiones del contenido en algunos casos son reemplazadas por diferentes combinaciones de las capas. Es decir, cuando un sistema existente puede proporcionar dos representaciones distintas del contenido, el sistema mejorado descrito aquí puede proporcionar dos capas, donde una representación del contenido en el sistema existente es similar en cuanto a velocidad de transmisión de bits, calidad y posiblemente otras métricas a la primera capa en el sistema mejorado y la segunda representación del contenido en el sistema existente es similar en cuanto a velocidad de transmisión de bits, calidad y posiblemente otras métricas a la combinación de las dos capas en el sistema mejorado. Como resultado, la capacidad de almacenamiento requerida dentro del sistema mejorado se reduce en comparación con la requerida en el sistema existente. Además, mientras que los clientes del sistema existente pueden emitir peticiones de bloques de una representación u otra representación, los clientes del sistema mejorado pueden emitir peticiones para la primera capa o para ambas capas de un bloque. Como resultado, la experiencia del usuario en los dos sistemas es similar. Además, se proporciona un almacenamiento en memoria caché mejorado ya que, incluso para diferentes calidades, se utilizan segmentos comunes que a continuación se almacenan en memoria caché con mayor probabilidad.

40 **[0306]** En un segundo modo de realización, un cliente en un sistema de transmisión de petición de bloques mejorado que emplea el procedimiento de capas ahora descrito puede mantener una memoria intermedia de datos separada para cada una de varias capas de la codificación de medios. Como quedará claro para los expertos en la técnica de la gestión de datos dentro de los dispositivos del cliente, estas memorias intermedias "separadas" pueden implementarse mediante la asignación de regiones de memoria separadas física o lógicamente para las memorias intermedias separadas o mediante otras técnicas en las que se almacenan los datos de la memoria intermedia en una o varias regiones de memoria y la separación de datos de diferentes capas se logra de manera lógica mediante el uso de estructuras de datos que contienen referencias a las ubicaciones de almacenamiento de datos de las capas separadas y, por lo tanto, a partir de aquí, debe entenderse que el término "memorias intermedias separadas" incluye cualquier procedimiento en el que los datos de las distintas capas se pueden identificar por separado. El cliente emite peticiones de capas individuales de cada bloque basándose en la ocupación de cada memoria intermedia; por ejemplo, las capas pueden ordenarse en un orden de prioridad tal que no se pueda emitir una petición de datos de una capa si

la ocupación de cualquier memoria intermedia para una capa inferior en el orden de prioridad está por debajo de un umbral para esa capa inferior. En este procedimiento, se da prioridad a recibir datos de las capas inferiores en el orden de prioridad, de manera que si el ancho de banda disponible cae por debajo del requerido para recibir también las capas más altas en el orden de prioridad, solo se solicitan las capas inferiores. Además, los umbrales asociados con las diferentes capas pueden ser diferentes, de modo que, por ejemplo, las capas más bajas tienen umbrales más altos. En el caso de que el ancho de banda disponible cambie de modo que los datos para una capa superior no puedan recibirse antes del tiempo de reproducción del bloque, entonces los datos para las capas inferiores ya se habrán recibido necesariamente y la presentación puede continuar solo con las capas inferiores. Los umbrales para la ocupación de la memoria intermedia se pueden definir en términos de bytes de datos, la duración de la reproducción de los datos contenidos en la memoria intermedia, el número de bloques o cualquier otra medida adecuada.

[0307] En un tercer modo de realización, los procedimientos del primer y segundo modos de realización pueden combinarse de manera que se proporcionen múltiples representaciones de medios, cada una de las cuales comprendiendo un subconjunto de las capas (como en el primer modo de realización) y de tal manera que el segundo modo de realización se aplique a un subconjunto de las capas dentro de una representación.

[0308] En un cuarto modo de realización, los procedimientos del primer, segundo y/o tercer modo de realización pueden combinarse con el modo de realización en el que se proporcionan múltiples representaciones independientes del contenido de manera que, por ejemplo, al menos una de las representaciones independientes comprenda múltiples capas en las que se aplican las técnicas del primer, segundo y/o tercer modo de realización.

### **Gestor de memoria intermedia avanzado**

[0309] En combinación con el monitor de memoria intermedia 126 (ver Fig. 2), se puede usar un administrador de memoria intermedia avanzado para optimizar una memoria intermedia del lado del cliente. Los sistemas de transmisión de peticiones de bloque desean garantizar que la reproducción de medios pueda comenzar rápidamente y continuar sin problemas, proporcionando al mismo tiempo la máxima calidad de medios al usuario o dispositivo de destino. Esto puede requerir que el cliente solicite bloques que tengan la mejor calidad de medios, pero que también puedan iniciarse rápidamente y recibirse a tiempo para reproducirse sin forzar una pausa en la presentación.

[0310] En los modos de realización que utilizan el administrador de memoria intermedia avanzado, el administrador determina qué bloques de datos de medios solicitar y cuándo realizar esas peticiones. Un administrador de memoria intermedia avanzado podría, por ejemplo, contar con un conjunto de metadatos para el contenido que se presentará, incluyendo estos metadatos una lista de representaciones disponibles para el contenido y los metadatos de cada representación. Los metadatos para una representación pueden comprender información sobre la velocidad de datos de la representación y otros parámetros, como el vídeo, audio u otros códecs y parámetros de códec, resolución de vídeo, complejidad de descodificación, lenguaje de audio y cualquier otro parámetro que pueda afectar la elección de representación en el cliente.

[0311] Los metadatos para una representación también pueden comprender identificadores para los bloques en los que se ha segmentado la representación, proporcionando estos identificadores la información necesaria para que el cliente solicite un bloque. Por ejemplo, cuando el protocolo de petición es HTTP, el identificador podría ser una URL HTTP posiblemente junto con información adicional que identifica un rango de bytes o un intervalo de tiempo dentro del archivo identificado por la URL, este rango de bytes o un intervalo de tiempo que identifica el bloque específico dentro del archivo identificado por la URL.

[0312] En una implementación específica, el administrador de memoria intermedia avanzado determina cuándo un receptor realiza una petición de nuevos bloques y podría manejar el envío de las peticiones. En un aspecto novedoso, el administrador de memoria intermedia avanzado realiza peticiones de nuevos bloques de acuerdo con el valor de una proporción de equilibrio que se equilibra entre usar demasiado ancho de banda y quedarse sin medios durante una reproducción de transmisión.

[0313] La información recibida por el monitor de memoria intermedia 126 de la memoria intermedia de bloques 125 puede incluir indicaciones de cada evento cuando se reciben datos de medios, cuánto se ha recibido, cuándo se inició o detuvo la reproducción de los datos de medios y la velocidad de reproducción de los medios. Basándose en esta información, el monitor de memoria intermedia 126 puede calcular una variable que representa un tamaño de memoria intermedia actual,  $B_{actual}$ . En estos ejemplos,  $B_{actual}$  representa la cantidad de medios contenidos en una memoria intermedia o memorias intermedias de un dispositivo u otro dispositivo y puede medirse en unidades de tiempo, de modo que  $B_{actual}$  representa la cantidad de tiempo que tardaría la reproducción de todos los medios representados por los bloques o bloques parciales almacenados en la memoria intermedia o memorias intermedias si no se recibieron bloques adicionales o bloques parciales. Por lo tanto,  $B_{actual}$  representa la "duración de reproducción", a la velocidad de reproducción normal, de los datos de medios disponibles en el cliente, pero aún no reproducidos.

[0314] A medida que pasa el tiempo, el valor de  $B_{actual}$  disminuirá a medida que se reproduzcan los medios y puede aumentar cada vez que se reciban nuevos datos para un bloque. Tenga en cuenta que, para los fines de esta explicación, se supone que se recibe un bloque cuando todos los datos de ese bloque están disponibles en el

solicitante de bloque 124, pero se pueden usar otras medidas en su lugar, por ejemplo, para tener en cuenta la recepción de bloques parciales. En la práctica, la recepción de un bloque puede tener lugar durante un período de tiempo.

5 **[0315]** La Fig. 13 ilustra una variación del valor de  $B_{actual}$  a lo largo del tiempo, a medida que se reproducen los medios y se reciben los bloques. Como se muestra en la Fig. 13, el valor de  $B_{actual}$  es cero para tiempos menores que  $t_0$ , lo cual indica que no se han recibido datos. En  $t_0$ , se recibe el primer bloque y el valor de  $B_{actual}$  aumenta para igualar la duración de la reproducción del bloque recibido. En este momento, la reproducción aún no ha comenzado y, por lo tanto, el valor de  $B_{actual}$  permanece constante, hasta el tiempo  $t_1$ , en el cual llega un segundo bloque y  $B_{actual}$  aumenta según el tamaño de este segundo bloque. En este momento, comienza la reproducción y el valor de  $B_{actual}$  comienza a disminuir linealmente, hasta el tiempo  $t_2$ , momento en el cual llega un tercer bloque.

15 **[0316]** La progresión de  $B_{actual}$  continúa de esta manera "diente de sierra", aumentando paso a paso cada vez que se recibe un bloque (en los tiempos  $t_2, t_3, t_4, t_5$  y  $t_6$ ) y disminuye suavemente a medida que los datos se reproducen en el medio. Tenga en cuenta que en este ejemplo, la reproducción avanza a la velocidad de reproducción normal para el contenido, por lo que la pendiente de la curva entre la recepción del bloque es exactamente -1, lo cual significa que se reproduce un segundo de datos de medios por cada segundo de tiempo real que pasa. Con los medios basados en tramas reproducidos en un número determinado de tramas por segundo, por ejemplo, 24 tramas por segundo, la pendiente de -1 se aproximará mediante funciones de pasos pequeños que indican la reproducción de cada trama de datos individual, por ejemplo, pasos de  $-1/24$  de segundo cuando se reproduce cada trama.

25 **[0317]** La Fig. 14 muestra otro ejemplo de la evolución de  $B_{actual}$  a lo largo del tiempo. En ese ejemplo, el primer bloque llega a  $t_0$  y la reproducción comienza de inmediato. La llegada y la reproducción del bloque continúan hasta el tiempo  $t_3$ , en el que el valor de  $B_{actual}$  llega a cero. Cuando eso sucede, no hay más datos de medios disponibles para su reproducción, lo cual obliga a una pausa en la presentación de medios. En el momento  $t_4$ , se recibe un cuarto bloque y se puede reanudar la reproducción. Por lo tanto, este ejemplo muestra un caso en el que la recepción del cuarto bloque fue posterior a la deseada, lo cual dio como resultado una pausa en la reproducción y, por lo tanto, una experiencia de usuario deficiente. Por lo tanto, un objetivo del administrador de memoria intermedia avanzado y otras características es reducir la probabilidad de este evento, al mismo tiempo que se mantiene una alta calidad de medios.

30 **[0318]** El monitor de memoria intermedia 126 también puede calcular otra métrica,  $B_{relación}(t)$ , que es la relación de los medios recibidos en un período de tiempo dado hasta la duración del período de tiempo. Más específicamente,  $B_{relación}(t)$  es igual a  $T_{recibida}(T_{ahora}-t)$ , donde  $T_{recibida}$  es la cantidad de medios (medida por su tiempo de reproducción) recibida en el período de tiempo desde  $t$ , un tiempo antes que el actual tiempo hasta el tiempo actual,  $T_{ahora}$ .

35 **[0319]**  $B_{relación}(t)$  se puede usar para medir la velocidad de cambio de  $B_{relación}$ .  $B_{relación}(t)=0$  es el caso donde no se han recibido datos desde el tiempo  $t$ ;  $B_{relación}$  se habrá reducido en  $(T_{ahora}-t)$  desde ese momento, suponiendo que los medios se están reproduciendo.  $B_{relación}(t) = 1$  es el caso en que los medios se reciben en la misma cantidad que se está reproduciendo, para tiempo  $(T_{ahora}-t)$ ;  $B_{relación}$  tendrá el mismo valor en el tiempo  $T_{ahora}$  que en el tiempo  $t$ .  $B_{relación}(t) > 1$  es el caso en el que se han recibido más datos de los necesarios para reproducir durante el tiempo  $(T_{ahora}-t)$ ;  $B_{relación}$  habrá aumentado desde el tiempo  $t$  hasta el tiempo  $T_{ahora}$ .

45 **[0320]** El monitor de memoria intermedia 126 calcula además un estado de valor, que puede tomar un número discreto de valores. El monitor de memoria intermedia 126 está equipado además con una función, **NewState** ( $B_{actual}, B_{relación}$ ), que, dado el valor actual de  $B_{actual}$  y los valores de  $B_{relación}$  para  $t < T_{ahora}$ , proporciona un nuevo valor de estado como salida. Cada vez que  $B_{actual}$  y  $B_{relación}$  hacen que esta función devuelva un valor diferente del valor actual de Estado, el nuevo valor se asigna a Estado y este nuevo valor de Estado se indica al selector de bloque 123.

50 **[0321]** La función **NewState** puede evaluarse con referencia al espacio de todos los valores posibles del par ( $B_{actual}, B_{relación}(T_{ahora} - T_x)$ ) donde  $T_x$  puede ser un valor fijo (configurado), o puede obtenerse a partir de  $B_{actual}$ , por ejemplo, mediante una tabla de configuración que asigna valores de  $B_{actual}$  a valores de  $T_x$ , o puede depender del valor anterior de Estado. El monitor de memoria intermedia 126 se suministra con una o más particiones de este espacio, donde cada partición comprende conjuntos de regiones diferentes, estando cada región anotada con un valor de Estado. A continuación, la evaluación de la función NewState comprende la operación de identificar una partición y determinar la región en la que cae el par ( $B_{actual}, B_{relación}(T_{ahora} - T_x)$ ). A continuación, el valor de retorno es la anotación asociada con esa región. En un caso simple, solo se proporciona una partición. En un caso más complejo, la partición puede depender del par ( $B_{actual}, B_{relación}(T_{ahora} - T_x)$ ) en el momento previo de la evaluación de la función **NewState** o de otros factores.

60 **[0322]** En un modo de realización específico, la partición descrita anteriormente se puede basar en una tabla de configuración que contiene un número de valores de umbral para  $B_{actual}$  y un número de valores de umbral para  $B_{relación}$ . Específicamente, deje que los valores de umbral para  $B_{actual}$  sean  $B_{trilla}(0) = 0, B_{trilla}(1), \dots, B_{trilla}(n_1), B_{trilla}(n_1+1) = \infty$ , donde  $n_1$  es el número de umbral distintos de cero para  $B_{actual}$ . Deje que los valores de umbral para  $B_{relación}$  sean  $B_{r-trilla}(0) = 0, B_{r-trilla}(1), \dots, B_{r-trilla}(n_2), B_{r-trilla}(n_2+1) = \infty$ , donde  $n_2$  es el número de valores de umbral para  $B_{relación}$ . Estos valores de umbral definen una partición que comprende una cuadrícula de células  $(n_1+1)$  por  $(n_2+1)$ , donde la  $i$ -ésima célula de la fila  $j$ -ésima corresponde a la región en la que  $B_{trilla}(i-1) \leq B_{actual} < B_{trilla}(i)$  y  $B_{r-trilla}(j-1) \leq B_{relación} < B_{r-trilla}(j)$ .

$trilla(j)$ ). Cada célula de la cuadrícula descrita anteriormente está anotada con un valor de estado, por ejemplo, al estar asociada con valores particulares almacenados en la memoria, y a continuación la función **NewState** devuelve el valor de estado asociado con la célula indicada por los valores  $B_{actual}$  y  $B_{relación}(T_{ahora} - T_x)$ .

5 [0323] En un modo de realización adicional, un valor de histéresis puede estar asociado a cada valor de umbral. En este procedimiento mejorado, la evaluación de la función **NewState** puede basarse en una partición temporal construida utilizando un conjunto de valores de umbral modificados temporalmente, de la siguiente manera. Para cada valor de umbral  $B_{actual}$  que sea menor que el rango de  $B_{actual}$  correspondiente a la célula elegida en la última evaluación de **NewState**, el valor de umbral se reduce al restar el valor de histéresis asociado con ese umbral. Para cada valor de umbral  $B_{actual}$  que sea mayor que el rango de  $B_{actual}$  correspondiente a la célula elegida en la última evaluación de **NewState**, el valor de umbral aumenta al agregar el valor de histéresis asociado con ese umbral. Para cada valor de umbral de  $B_{relación}$  que es menor que el rango de  $B_{relación}$  correspondiente a la célula elegida en la última evaluación de **NewState**, el valor de umbral se reduce al restar el valor de histéresis asociado con ese umbral. Para cada valor de umbral de  $B_{relación}$  que sea mayor que el rango de  $B_{relación}$  correspondiente a la célula elegida en la última evaluación de **NewState**, el valor de umbral aumenta al agregar el valor de histéresis asociado con ese umbral. Los valores de umbral modificados se utilizan para evaluar el valor de **NewState** y a continuación los valores de umbral se devuelven a sus valores originales.

20 [0324] Otras formas de definir las particiones del espacio serán obvias para los expertos en la técnica al leer esta divulgación. Por ejemplo, una partición puede definirse mediante el uso de desigualdades basadas en combinaciones lineales de  $B_{relación}$  y  $B_{actual}$ , por ejemplo, umbrales de desigualdad lineal de la forma  $\alpha_1 \bullet B_{relación} + \alpha_2 \bullet B_{actual} \leq \alpha_0$  para  $\alpha_0$  de valor real,  $\alpha_1$  y  $\alpha_2$ , para definir espacios intermedios dentro del espacio general y definir cada conjunto diferente como la intersección de un número de dichos espacios intermedios.

25 [0325] La descripción anterior es ilustrativa del proceso básico. Como quedará claro para los expertos en la técnica de la programación en tiempo real después de leer esta divulgación, son posibles implementaciones eficientes. Por ejemplo, en cada momento en que se proporciona nueva información al monitor de memoria intermedia 126, es posible calcular el tiempo futuro en el que **NewState** realizará la transición a un nuevo valor si, por ejemplo, no se reciben más datos para los bloques. A continuación se establece un temporizador para este tiempo y, en ausencia de más entradas, la expiración de este temporizador hará que el nuevo valor de *Estado* se envíe al selector de bloque 123. Como resultado, los cálculos solo deben realizarse cuando se proporciona nueva información al monitor de memoria intermedia 126 o cuando el temporizador expira, en lugar de hacerlo de forma continua.

35 [0326] Los valores adecuados de *Estado* pueden ser "Bajo", "Estable" y "Completo". En la Fig. 15 se muestra un ejemplo de un conjunto adecuado de valores de umbral y la cuadrícula de células resultante.

40 [0327] En la Fig. 15, los umbrales de  $B_{actual}$  se muestran en el eje horizontal en milisegundos, con los valores de histéresis mostrados a continuación como "+/-valor". Los umbrales de  $B_{relación}$  se muestran en el eje vertical en por mil (es decir, multiplicados por 1000) con los valores de histéresis mostrados a continuación como "valor+/-". Los valores de estado se anotan en las células de la cuadrícula como "L", "S" y "F" para "Bajo", "Estable" y "Completo" respectivamente.

45 [0328] El selector de bloque 123 recibe notificaciones del solicitante de bloque 124 cuando existe la oportunidad de solicitar un nuevo bloque. Como se describió anteriormente, el selector de bloque 123 está provisto de información sobre la pluralidad de bloques disponibles y los metadatos para esos bloques, que incluyen, por ejemplo, información sobre la velocidad de datos de medios de cada bloque.

50 [0329] La información sobre la velocidad de datos de medios de un bloque puede comprender la velocidad de datos de medios real del bloque específico (es decir, el tamaño del bloque en bytes dividido por el tiempo de reproducción en segundos), la velocidad media de datos de medios de la representación a la que pertenece el bloque o una medida del ancho de banda disponible requerido, de forma sostenida, para reproducir la representación a la que pertenece el bloque sin pausas, o una combinación de lo anterior.

55 [0330] El selector de bloque 123 selecciona bloques basándose en el último valor de *estado* indicado por el monitor 126 de la memoria intermedia. Cuando este valor de estado es "Estable", el selector de bloque 123 selecciona un bloque de la misma representación que el bloque seleccionado anteriormente. El bloque seleccionado es el primer bloque (en orden de reproducción) que contiene datos de medios durante un período de tiempo en la presentación para el cual no se han solicitado datos de medios previamente.

60 [0331] Cuando el valor del estado es "Bajo", el selector de bloque 123 selecciona un bloque de una representación con una velocidad de datos de medios más baja que la del bloque seleccionado previamente. Varios factores pueden influir en la elección exacta de la representación en este caso. Por ejemplo, el selector de bloque 123 podría estar provisto de una indicación de la velocidad agregada de datos entrantes y puede elegir una representación con una velocidad de datos de medios que sea menor que ese valor.

65

[0332] Cuando el valor del estado es "Completo", el selector de bloque 123 selecciona un bloque de una representación con una velocidad de datos de medios más alta que la del bloque seleccionado previamente. Varios factores pueden influir en la elección exacta de la representación en este caso. Por ejemplo, el selector de bloque 123 puede proporcionarse con una indicación de la velocidad agregada de datos entrantes y puede elegir una representación con una velocidad de datos de medios que no sea mayor que ese valor.

[0333] Varios factores adicionales pueden influir aún más en el funcionamiento del selector de bloque 123. En particular, la frecuencia con la que se aumenta la velocidad de datos de medios del bloque seleccionado puede limitarse, incluso si el monitor 126 de la memoria intermedia continúa indicando el estado "Completo". Además, es posible que el selector de bloque 123 reciba una indicación de estado "Completo" pero no haya bloques de mayor velocidad de datos de medios disponible (por ejemplo, porque el último bloque seleccionado ya tenía la mayor velocidad de datos de medios disponible). En este caso, el selector de bloque 123 puede retardar la selección del siguiente bloque en un tiempo elegido de tal manera que la cantidad total de datos de medios almacenados en la memoria intermedia de bloques 125 esté delimitada anteriormente.

[0334] Factores adicionales pueden influir en el conjunto de bloques que se consideran durante el proceso de selección. Por ejemplo, los bloques disponibles pueden limitarse a aquellos de representaciones cuya resolución de codificación se encuentra dentro de un rango específico provisto para el selector de bloque 123.

[0335] El selector de bloque 123 también puede recibir entradas de otros componentes que supervisan otros aspectos del sistema, como la disponibilidad de recursos computacionales para la descodificación de medios. Si tales recursos se vuelven escasos, el selector de bloque 123 puede elegir bloques cuya descodificación se indica que es de menor complejidad computacional dentro de los metadatos (por ejemplo, las representaciones con resolución o frecuencia de tramas más baja en general son de menor complejidad de descodificación).

[0336] El modo de realización descrito anteriormente aporta una ventaja sustancial en que el uso del valor  $B_{relación}$  en la evaluación de la función **NewState** dentro del monitor de memoria intermedia 126 permite un aumento más rápido en la calidad en el comienzo de la presentación en comparación con un procedimiento que solo considera  $B_{actual}$ . Sin considerar  $B_{relación}$ , se puede acumular una gran cantidad de datos almacenados en memoria intermedia antes de que el sistema pueda seleccionar bloques con una mayor velocidad de datos de medios y, por lo tanto, una mayor calidad. Sin embargo, cuando el valor de  $B_{relación}$  es grande, esto indica que el ancho de banda disponible es mucho más alto que la velocidad de datos de medios de los bloques recibidos anteriormente y que incluso con relativamente pocos datos almacenados (es decir, un valor bajo para  $B_{actual}$ ), sigue siendo seguro solicitar bloques de mayor velocidad de datos de medios y por lo tanto de mayor calidad. Igualmente, si el valor de  $B_{relación}$  es bajo ( $<1$ , por ejemplo), esto indica que el ancho de banda disponible ha descendido por debajo de la velocidad de datos de los bloques solicitados anteriormente y, por lo tanto, incluso si  $B_{actual}$  es alta, el sistema cambiará a velocidad de datos de medios más baja y, por lo tanto, una calidad más baja, por ejemplo, para evitar alcanzar el punto donde  $B_{actual} = 0$  y la reproducción de los medios se detiene. Este comportamiento mejorado puede ser especialmente importante en entornos donde las condiciones de la red y, por lo tanto, las velocidades de entrega pueden variar de forma rápida y dinámica, por ejemplo, con los usuarios transmitiendo a dispositivos móviles.

[0337] Otra ventaja es conferida por el uso de datos de configuración para especificar la partición del espacio de valores de ( $B_{actual}$ ,  $B_{relación}$ ). Dichos datos de configuración se pueden proporcionar al monitor de memoria intermedia 126 como parte de los metadatos de presentación o por otros medios dinámicos. Dado que, en implementaciones prácticas, el comportamiento de las conexiones de red del usuario puede ser muy variable entre los usuarios y con el tiempo para un solo usuario, puede ser difícil predecir particiones que funcionarán bien para todos los usuarios. La posibilidad de proporcionar dicha información de configuración a los usuarios dinámicamente permite que se desarrollen buenos ajustes de configuración a lo largo del tiempo de acuerdo con la experiencia acumulada.

#### **Petición de tamaño variable**

[0338] Puede requerirse una alta frecuencia de peticiones si cada petición es para un solo bloque y si cada bloque se codifica para un segmento de medios corto. Si los bloques de medios son cortos, la reproducción del vídeo se está moviendo de bloque a bloque rápidamente, lo cual brinda oportunidades más frecuentes para que el receptor ajuste o cambie su velocidad de datos seleccionada al cambiar la representación, lo cual mejora la probabilidad de que la reproducción pueda continuar sin detenerse. Sin embargo, un inconveniente de una alta frecuencia de peticiones es que podrían no ser sostenibles en ciertas redes en las que el ancho de banda disponible está restringido en la red del cliente al servidor, por ejemplo, en redes WAN inalámbricas como las WAN inalámbricas 3G y 4G, donde la capacidad del enlace de datos del cliente a la red es limitada o puede ser limitada durante períodos cortos o largos de tiempo debido a cambios en las condiciones de la radio.

[0339] Una alta frecuencia de peticiones también implica una gran carga en la infraestructura de servicio, lo cual genera costes asociados en términos de requisitos de capacidad. Por lo tanto, sería deseable tener algunos de los beneficios de una alta frecuencia de peticiones sin todas las desventajas.

5 [0340] En algunos modos de realización de un sistema de transmisión en bloque, la flexibilidad de la alta frecuencia de petición se combina con peticiones menos frecuentes. En este modo de realización, los bloques pueden construirse como se describió anteriormente y agregarse en segmentos que contienen múltiples bloques, también como se describió anteriormente. Al comienzo de la presentación, los procesos descritos anteriormente en los que cada petición hace referencia a un solo bloque o se realizan múltiples peticiones simultáneas para solicitar que se apliquen partes de un bloque para garantizar un rápido tiempo de zapping y, por lo tanto, una buena experiencia de usuario al comienzo de la presentación. Posteriormente, cuando se cumple una determinada condición, que se describirá a continuación, el cliente puede emitir peticiones que abarcan varios bloques en una sola petición. Esto es posible porque los bloques se han agregado en archivos o segmentos más grandes y se pueden solicitar mediante bytes o intervalos de tiempo. 10 Los bytes o intervalos de tiempo consecutivos se pueden agregar en un solo byte o intervalo de tiempo más grande, lo cual da como resultado una petición única para múltiples bloques, e incluso los bloques discontinuos se pueden solicitar en una petición.

15 [0341] Una configuración básica que puede manejarse al decidir si solicitar un bloque único (o un bloque parcial) o solicitar varios bloques consecutivos es hacer que la configuración base la decisión en si es probable que los bloques solicitados se ejecuten o no. Por ejemplo, si es probable que haya una necesidad de cambiar a otra representación pronto, entonces es mejor para el cliente realizar peticiones de bloques individuales, es decir, pequeñas cantidades de datos de medios. Una razón para esto es que si se realiza una petición de bloques múltiples cuando un cambio a otra representación puede ser inminente es que el cambio puede realizarse antes de que se ejecuten los últimos bloques de la petición. Por lo tanto, la descarga de estos últimos bloques podría retardar la entrega de los datos de medios de la representación hacia la cual se realiza el cambio, lo cual podría causar que la reproducción de los medios se detenga.

25 [0342] Sin embargo, las peticiones de bloques individuales dan como resultado una mayor frecuencia de peticiones. Por otro lado, si es poco probable que haya una necesidad de cambiar a otra representación pronto, entonces se puede preferir realizar peticiones de bloques múltiples, ya que es probable que todos estos bloques se reproduzcan, y esto da como resultado una menor frecuencia de peticiones, lo cual puede reducir sustancialmente la sobrecarga de peticiones, especialmente si es típico que no haya un cambio inminente en la representación.

30 [0343] En los sistemas de agregación de bloques convencionales, la cantidad solicitada en cada petición no se ajusta dinámicamente, es decir, típicamente cada petición es para un archivo completo, o cada petición es para aproximadamente la misma cantidad del archivo de una representación (algunas veces se mide en el tiempo, algunas veces en bytes). Por lo tanto, si todas las peticiones son más pequeñas, entonces la sobrecarga de peticiones es alta, mientras que si todas las peticiones son más grandes, esto aumenta las posibilidades de que los medios se detengan y/o se proporcione una reproducción de medios de menor calidad si se eligen representaciones de menor calidad para evitar tener que cambiar rápidamente las representaciones cuando las condiciones de la red varían. 35

40 [0344] Un ejemplo de una condición que, cuando se cumple, puede hacer que las peticiones posteriores hagan referencia a varios bloques, es un umbral en el tamaño de la memoria intermedia,  $B_{actual}$ . Si  $B_{actual}$  está por debajo del umbral, entonces cada petición emitida hace referencia a un solo bloque. Si  $B_{actual}$  es mayor o igual que el umbral, entonces cada petición emitida hace referencia a varios bloques. Si se emite una petición que hace referencia a varios bloques, entonces el número de bloques solicitados en cada petición individual se puede determinar de una de las varias formas posibles. Por ejemplo, el número puede ser constante, por ejemplo, dos. De forma alternativa, el número de bloques solicitados en una sola petición puede depender del estado de la memoria intermedia y, en particular, de  $B_{actual}$ . Por ejemplo, se puede establecer una cantidad de umbrales, obteniéndose la cantidad de bloques solicitados en una sola petición a partir del más alto de los umbrales múltiples que es menor que  $B_{actual}$ . 45

50 [0345] Otro ejemplo de una condición que, cuando se cumple, puede hacer que las peticiones hagan referencia a múltiples bloques, es la variable de *estado* de valor descrita anteriormente. Por ejemplo, cuando el *estado* es "Estable" o "Completo", se pueden emitir peticiones para varios bloques, pero cuando el *estado* es "Bajo", todas las peticiones pueden ser para un bloque.

55 [0346] Otro modo de realización se muestra en la Fig. 16. En este modo de realización, cuando se debe emitir la siguiente petición (determinada en el paso 1300), el valor actual del estado y  $B_{actual}$  se utilizan para determinar el tamaño de la siguiente petición. Si el valor actual del estado es "Bajo" o el valor actual del estado es "Completo" y la representación actual no es la más alta disponible (determinada en el paso 1310, la respuesta es "Sí"), entonces se elige la siguiente petición como breve, por ejemplo, solo para el siguiente bloque (bloque determinado y petición realizada en el paso 1320). La razón detrás de esto es que estas son condiciones donde es probable que muy pronto haya un cambio de representación. Si el valor actual del estado es "Estable" o el valor del estado actual es "Completo" y la representación actual es la más alta disponible (determinada en el paso 1310, la respuesta es "No"), entonces la duración de los bloques consecutivos solicitados en la siguiente la petición se elige para que sea proporcional a una fracción  $\alpha$  de  $B_{actual}$  para algunos  $\alpha < 1$  fijos (bloques determinados en el paso 1330, petición realizada en el paso 1340); por ejemplo, para  $\alpha = 0,4$ , si  $B_{actual} = 5$  segundos, entonces la siguiente petición puede ser de aproximadamente 2 segundos de bloques, mientras que si  $B_{actual}$  es de 10 segundos, la siguiente petición puede ser de aproximadamente 65 4 segundos de bloques. Una razón para esto es que en estas condiciones puede ser poco probable que se realice un cambio a una nueva representación durante un período de tiempo proporcional a  $B_{actual}$ .

**Yuxtaposición flexible**

5 [0347] Los sistemas de transmisión de bloques pueden usar un protocolo de petición de archivos que tiene un protocolo de transporte subyacente particular, por ejemplo, TCP/IP. Al comienzo de una conexión de protocolo TCP/IP u otro protocolo de transporte, puede tardarse un tiempo considerable en lograr la utilización del ancho de banda completo disponible. Esto puede dar como resultado una "penalización de inicio de conexión" cada vez que se inicie una nueva conexión. Por ejemplo, en el caso de TCP/IP, la penalización de inicio de la conexión se debe tanto al tiempo que tarda el protocolo de enlace TCP inicial en establecer la conexión como al tiempo necesario para que el protocolo de control de congestión logre la utilización completa del ancho de banda disponible.

15 [0348] En este caso, puede ser conveniente emitir múltiples peticiones utilizando una sola conexión, para reducir la frecuencia con la que se incurre en la penalización de inicio de la conexión. Sin embargo, algunos protocolos de transporte de archivos, por ejemplo HTTP, no proporcionan un mecanismo para cancelar una petición, aparte de cerrar por completo la conexión de la capa de transporte y, por lo tanto, incurrir en una penalización de inicio de la conexión cuando se establece una nueva conexión en lugar de la anterior. Es posible que deba cancelarse una petición emitida si se determina que el ancho de banda disponible ha cambiado y se requiere una velocidad de datos de medios diferente, es decir, hay una decisión de cambiar a una representación diferente. Otra razón para cancelar una petición emitida puede ser si el usuario ha solicitado que se finalice la presentación de medios y se inicie una nueva presentación (tal vez del mismo elemento de contenido en un punto diferente de la presentación o tal vez de un nuevo elemento de contenido).

25 [0349] Como se sabe, la penalización de inicio de la conexión se puede evitar manteniendo la conexión abierta y reutilizando la misma conexión para las peticiones posteriores, y como también se sabe, la conexión se puede mantener plenamente utilizada si se emiten múltiples peticiones al mismo tiempo en la misma conexión (una técnica conocida como "yuxtaposición" en el contexto de HTTP). Sin embargo, una desventaja de emitir múltiples peticiones al mismo tiempo, o más en general de tal manera que se emiten múltiples peticiones antes de que se completen las peticiones anteriores a través de una conexión, puede ser que la conexión se comprometa a llevar la respuesta a esas peticiones y por lo tanto, si los cambios para los que se deben emitir peticiones se convierten en deseables, entonces la conexión puede cerrarse si es necesario cancelar las peticiones ya emitidas que ya no se desean.

35 [0350] La probabilidad de que una petición emitida deba cancelarse puede depender en parte de la duración del intervalo de tiempo entre la emisión de la petición y el tiempo de reproducción del bloque solicitado, en el sentido de que cuando este intervalo de tiempo es alto, la probabilidad de que una petición emitida deba cancelarse también es alta (porque es probable que el ancho de banda disponible cambie durante el intervalo).

40 [0351] Como se sabe, algunos protocolos de descarga de archivos tienen la propiedad de que una única conexión de capa de transporte subyacente pueda usarse ventajosamente para múltiples peticiones de descarga. Por ejemplo, HTTP tiene esta propiedad, ya que la reutilización de una sola conexión para múltiples peticiones evita la "penalización de inicio de conexión" descrita anteriormente para peticiones distintas a la primera. Sin embargo, una desventaja de este enfoque es que la conexión se compromete a transportar los datos solicitados en cada petición emitida y, por lo tanto, si una petición o peticiones deben cancelarse, entonces la conexión puede cerrarse, incurriendo en la penalización de inicio de la conexión cuando se establece una conexión de reemplazo, o el cliente puede esperar a recibir los datos que ya no son necesarios, lo cual puede provocar un retardo en la recepción de los datos posteriores.

45 [0352] Ahora describimos un modo de realización que conserva las ventajas de la reutilización de la conexión sin incurrir en esta desventaja y que también mejora adicionalmente la frecuencia con la que se pueden reutilizar las conexiones.

50 [0353] Los modos de realización de los sistemas de transmisión de bloques descritos en el presente documento están configurados para reutilizar una conexión para múltiples peticiones sin tener que confirmar la conexión al inicio con un conjunto particular de peticiones. Esencialmente, una nueva petición se emite en una conexión existente cuando las peticiones ya emitidas en la conexión aún no se han completado, pero están cerca de completarse. Una razón para no esperar hasta que se completen las peticiones existentes es que si las peticiones anteriores se completan, entonces la velocidad de conexión podría disminuir, es decir, la sesión TCP subyacente podría pasar a un estado inactivo, o la variable *cwnd* de TCP podría reducirse sustancialmente, por lo tanto sustancialmente reduciendo la velocidad de descarga inicial de la nueva petición emitida en esa conexión. Una razón para esperar hasta que casi se complete antes de emitir una petición adicional es porque si se emite una nueva petición mucho antes de que se completen las peticiones anteriores, es posible que la nueva petición emitida ni siquiera comience durante un período de tiempo considerable, y podría darse el caso que durante este período de tiempo antes de que la nueva petición emitida comience, la decisión de realizar la nueva petición ya no sea válida, por ejemplo, debido a una decisión de cambiar de representación. Por lo tanto, el modo de realización de los clientes que implementan esta técnica emitirá una nueva petición en una conexión lo más tarde posible sin ralentizar las capacidades de descarga de la conexión.

**[0354]** El procedimiento comprende supervisar el número de bytes recibidos en una conexión en respuesta a la última petición emitida en esta conexión y aplicar una prueba a este número. Esto se puede hacer configurando el receptor (o el transmisor, si corresponde) para supervisar y probar.

5 **[0355]** Si la prueba pasa, entonces se puede emitir una petición adicional en la conexión. Un ejemplo de una prueba adecuada es si el número de bytes recibidos es mayor que una fracción fija del tamaño de los datos solicitados. Por ejemplo, esta fracción podría ser del 80 %. Otro ejemplo de una prueba adecuada se basa en el siguiente cálculo, como se ilustra en la Fig. 17. En el cálculo,  $R$  debe ser una estimación de la velocidad de datos de la conexión,  $T$  una estimación del tiempo de ida y vuelta ("RTT") y  $X$  un factor numérico que, por ejemplo, podría ser una constante establecida en un valor entre 0,5 y 2, donde las estimaciones de  $R$  y  $T$  se actualizan regularmente (actualizado en el paso 1410).  $S$  debe ser el tamaño de los datos solicitados en la última petición,  $B$  el número de bytes de los datos solicitados recibidos (calculado en el paso 1420).

15 **[0356]** Una prueba adecuada sería hacer que el receptor (o el transmisor, si corresponde) ejecute una rutina para evaluar la desigualdad  $(S-B) < X \bullet R \bullet T$  (probada en el paso 1430), y si la respuesta es "Sí", emprender una acción. Por ejemplo, se podría hacer una prueba para ver si hay otra petición lista para ser emitida en la conexión (probada en el paso 1440), y si el resultado es "Sí", entonces emitir esa petición a la conexión (paso 1450) y si es "No", entonces el proceso regresa al paso 1410 para continuar actualizando y probando. Si el resultado de la prueba en el paso 1430 es "No", el proceso regresa al paso 1410 para continuar la actualización y las pruebas.

20 **[0357]** La prueba de desigualdad en el paso 1430 (realizada por elementos debidamente programados, por ejemplo) hace que cada petición subsiguiente se emita cuando la cantidad de datos restantes a recibir sea igual a  $X$  veces la cantidad de datos que se pueden recibir a la velocidad de recepción estimada actual dentro de un RTT. En la técnica se conocen varios procedimientos para estimar la velocidad de datos,  $R$ , en el paso 1410. Por ejemplo, la velocidad de datos puede estimarse como  $Dt/t$ , donde  $Dt$  es el número de bits recibidos en los  $t$  segundos precedentes y donde  $t$  puede ser, por ejemplo, 1s o 0,5 s o algún otro intervalo. Otro procedimiento es una media ponderada exponencial, o el filtro de Respuesta de Impulso Infinito (IIR) de primer orden de la velocidad de datos entrantes. En la técnica se conocen varios procedimientos para estimar el RTT,  $T$ , en el paso 1410.

30 **[0358]** La prueba en el paso 1430 se puede aplicar al agregado de todas las conexiones activas en una interfaz, como se explica con más detalle a continuación.

35 **[0359]** El procedimiento comprende además la construcción de una lista de peticiones candidatas, asociando cada petición candidata a un conjunto de servidores adecuados a los cuales se puede realizar la petición y ordenando la lista de peticiones candidatas en orden de prioridad. Algunas entradas en la lista de peticiones candidatas pueden tener la misma prioridad. Los servidores en la lista de servidores adecuados asociados con cada petición candidata se identifican mediante nombres de dispositivo principal. Cada nombre de dispositivo principal corresponde a un conjunto de direcciones de protocolo de Internet que se pueden obtener del sistema de nombres de dominio como es bien conocido. Por lo tanto, cada petición posible en la lista de peticiones candidatas está asociada con un conjunto de direcciones de protocolo de Internet, específicamente la unión de los conjuntos de direcciones de protocolo de Internet asociadas con los nombres de dispositivo principal asociados con los servidores asociados con la petición candidata. Siempre que se cumpla la prueba descrita en el paso 1430 para una conexión, y aún no se haya emitido una nueva petición en esa conexión, se escoge la petición de prioridad más alta en las listas de peticiones candidatas a las que está asociada la dirección de protocolo de Internet del destino de la conexión, y esta petición se emite en la conexión. La petición también se elimina de la lista de peticiones candidatas.

50 **[0360]** Las peticiones candidatas se pueden eliminar (cancelar) de la lista de peticiones candidatas, se pueden agregar nuevas peticiones a la lista de candidatas con una prioridad que sea mayor que las peticiones existentes en la lista de candidatas, y las peticiones existentes en la lista de candidatas pueden tener su prioridad cambiada. La naturaleza dinámica de qué peticiones se encuentran en la lista de peticiones candidatas y la naturaleza dinámica de su prioridad en la lista de candidatas pueden alterar qué peticiones se pueden emitir a continuación dependiendo de cuándo se satisfaga una prueba del tipo descrito en el paso 1430.

55 **[0361]** Por ejemplo, podría ser posible que si la respuesta a la prueba descrita en el paso 1430 es "Sí" en algún tiempo  $t$ , la siguiente petición emitida sea una petición A, mientras que si la respuesta a la prueba descrita en el paso 1430 no es "Sí" hasta que algún tiempo  $t' > t$ , la siguiente petición emitida sea en cambio una petición B, ya sea porque la petición A se eliminó de la lista de peticiones candidatas entre el tiempo  $t$  y  $t'$ , o porque la petición B se agregó a la lista de peticiones candidatas con mayor prioridad que la petición A entre el tiempo  $t$  y  $t'$ , o porque la petición B estaba en la lista de candidatas en el tiempo  $t$  pero con menor prioridad que la petición A, y entre el tiempo  $t$  y  $t'$  se realizó la prioridad de la petición B superior a la de la petición A.

65 **[0362]** La Fig. 18 ilustra un ejemplo de una lista de peticiones en la lista de peticiones candidatas. En este ejemplo, hay tres conexiones, y hay seis peticiones en la lista de candidatas, etiquetadas A, B, C, D, E y F. Cada una de las peticiones en la lista de candidatas puede emitirse en un subconjunto de conexiones como se indica; por ejemplo, la petición A se puede emitir en la conexión 1, mientras que la petición F se puede emitir en la conexión 2 o la conexión 3. La prioridad de cada petición también se etiqueta en la Fig. 18, y un valor de prioridad más bajo indica que una

petición es de mayor prioridad. Por lo tanto, las peticiones A y B con prioridad 0 son las peticiones de prioridad más alta, mientras que la petición F con un valor de prioridad 3 es la prioridad más baja entre las peticiones en la lista de candidatas.

5 **[0363]** Si, en este tiempo  $t$ , la conexión 1 pasa la prueba descrita en el paso 1430, entonces la petición A o la petición B se emiten en la conexión 1. Si, en cambio, la conexión 3 pasa la prueba descrita en el paso 1430 en este tiempo  $t$ , entonces la petición D se emite en la conexión 3, ya que la petición D es la petición con la prioridad más alta que se puede emitir en la conexión 3.

10 **[0364]** Supongamos que para todas las conexiones, la respuesta a la prueba descrita en el paso 1430 desde el tiempo  $t$  hasta un tiempo posterior  $t'$  es "No", y entre el tiempo  $t$  y  $t'$  la petición A cambia su prioridad de 0 a 5, la petición B se elimina de la lista de candidatas y una nueva petición G con prioridad 0 se agrega a la lista de candidatas. A continuación, en el tiempo  $t'$ , la nueva lista de candidatas podría ser como se muestra en la Fig. 19.

15 **[0365]** Si en el tiempo  $t'$  la conexión 1 pasa la prueba descrita en el paso 1430, la petición C con prioridad 4 se emite en la conexión 1, ya que es la petición de prioridad más alta en la lista de candidatas que se puede emitir en la conexión 1 en este momento.

20 **[0366]** Supongamos que, en esta misma situación, la petición A hubiera sido emitida en la conexión 1 en el tiempo  $t$  (que fue una de las dos opciones de mayor prioridad para la conexión 1 en el tiempo  $t$  como se muestra en la Fig. 18). Dado que la respuesta a la prueba descrita en el paso 1430 desde el tiempo  $t$  hasta el tiempo  $t'$  posterior es "No" para todas las conexiones, la conexión 1 sigue entregando datos hasta al menos el tiempo  $t'$  para peticiones emitidas antes del tiempo  $t$ , y por lo tanto la petición A no habría comenzado hasta al menos el tiempo  $t'$ . La emisión de la petición C en el tiempo  $t'$  es una mejor decisión de lo que habría sido la emisión de la petición A en el tiempo  $t$ , ya que la petición C comienza al mismo tiempo después de que  $t'$  que la petición A hubiera comenzado, y dado que en ese tiempo la petición C es de mayor prioridad que la petición A.

25 **[0367]** Como otra alternativa, si la prueba del tipo descrito en el paso 1430 se aplica al conjunto de las conexiones activas, se puede elegir una conexión que tenga un destino cuya Dirección de Protocolo de Internet esté asociada con la primera petición en la lista de peticiones candidatas u otra petición con la misma prioridad que dicha primera petición.

30 **[0368]** Una serie de procedimientos son posibles para la construcción de la lista de peticiones candidatas. Por ejemplo, la lista de candidatas podría contener  $n$  peticiones que representen peticiones para las siguientes  $n$  partes de datos de la representación actual de la presentación en orden de secuencia de tiempo, donde la petición de la parte más antigua de datos tiene la mayor prioridad y la petición de la parte más reciente de datos tiene la prioridad más baja. En algunos casos  $n$  puede ser uno. El valor de  $n$  puede depender del tamaño de la memoria intermedia  $B_{actual}$ , o la variable de *estado* u otra medida del estado de la ocupación de la memoria intermedia del cliente. Por ejemplo, se pueden establecer una serie de valores de umbral para  $B_{actual}$  y un valor asociado con cada umbral y a continuación se toma el valor de  $n$  como el valor asociado con el umbral más alto que es menor que  $B_{actual}$ .

35 **[0369]** El modo de realización descrito anteriormente garantiza una asignación flexible de las peticiones a las conexiones, asegurando que se da preferencia a la reutilización de una conexión existente incluso si la petición de prioridad más alta no es adecuada para esa conexión (porque la dirección IP de destino de la conexión no es una asignada a cualquiera de los nombres de dispositivo principal asociados con la petición). La dependencia de  $n$  en  $B_{actual}$  o en el *estado* u otra medida de la ocupación de la memoria intermedia del cliente garantiza que dichas peticiones "fuera de orden prioritario" no se emitan cuando el cliente necesita urgentemente la emisión y la finalización de la petición asociada con la siguiente parte de datos a reproducir en la secuencia de tiempo.

40 **[0370]** Estos procedimientos pueden combinarse ventajosamente con HTTP y FEC cooperativos.

### **Selección coherente del servidor**

45 **[0371]** Como es bien sabido, los archivos que se descargan utilizando un protocolo de descarga de archivos se identifican comúnmente por un identificador que comprende un nombre de dispositivo principal y un nombre de archivo. Por ejemplo, este es el caso del protocolo HTTP, en cuyo caso el identificador es un identificador de recursos uniforme (URI). Un nombre de dispositivo principal puede corresponder a múltiples dispositivos principales, identificados por direcciones de protocolo de Internet. Por ejemplo, este es un procedimiento común para distribuir la carga de peticiones de varios clientes en varias máquinas físicas. En particular, este enfoque es comúnmente adoptado por las redes de distribución de contenido (CDN). En este caso, se espera que una petición emitida en una conexión a cualquiera de los dispositivos principales físicos tenga éxito. Se conocen varios procedimientos por los cuales un cliente puede seleccionar entre las Direcciones de Protocolo de Internet asociadas con un nombre de dispositivo principal. Por ejemplo, estas direcciones típicamente se proporcionan al cliente a través del sistema de nombres de dominio y se proporcionan en orden de prioridad. Un cliente puede a continuación elegir la dirección de protocolo de Internet de prioridad más alta (primera). Sin embargo, en general no existe una coordinación entre los clientes sobre cómo se hace esta elección, con el resultado de que diferentes clientes pueden solicitar el mismo archivo de diferentes

servidores. Esto puede hacer que el mismo archivo se almacene en la memoria caché de varios servidores cercanos, lo cual reduce la eficiencia de la infraestructura de la memoria caché.

5 **[0372]** Esto puede ser manejado por un sistema que ventajosamente aumenta la probabilidad de que dos clientes que soliciten el mismo bloque soliciten este bloque al mismo servidor. El nuevo procedimiento descrito aquí incluye la selección de entre las Direcciones de Protocolo de Internet disponibles de una manera determinada por el identificador del archivo a solicitar y de tal manera que diferentes clientes a los que se les presenten las mismas o similares opciones de direcciones de Protocolo de Internet e identificadores de archivo realicen la misma elección.

10 **[0373]** Una primer modo de realización del procedimiento se describe con referencia a la Fig. 20. El cliente primero obtiene un conjunto de direcciones de protocolo de Internet  $IP_1, IP_2, \dots, IP_n$ , como se muestra en el paso 1710. Si hay un archivo para el que se emitirán las peticiones, como se decidió en el paso 1720, entonces el cliente determina qué dirección de Protocolo de Internet emitirá las peticiones para el archivo, según lo determinado en los pasos 1730-1770. Dado un conjunto de direcciones de Protocolo de Internet y un identificador para un archivo a solicitar, el  
15 procedimiento comprende ordenar las direcciones de Protocolo de Internet de una manera determinada por el identificador de archivo. Por ejemplo, para cada dirección de Protocolo de Internet se construye una secuencia de bytes que comprende la concatenación de la dirección de Protocolo de Internet y el identificador de archivo, como se muestra en el paso 1730. Se aplica una función hash a esta secuencia de bytes, como se muestra en el paso 1740, y los valores hash resultantes se organizan de acuerdo con un orden fijo, como se muestra en el paso 1750, por ejemplo, aumentando el orden numérico, induciendo un orden en las direcciones de Protocolo de Internet. Todos los clientes pueden utilizar la misma función hash, lo cual garantiza que la función hash produzca el mismo resultado en una entrada determinada de todos los clientes. La función hash puede configurarse estáticamente en todos los clientes en un conjunto de clientes, o todos los clientes en un conjunto de clientes pueden obtener una descripción parcial o completa de la función hash cuando los clientes obtienen la lista de direcciones de Protocolo de Internet, o todos los  
20 clientes en un conjunto de clientes puede obtener una descripción parcial o completa de la función hash cuando los clientes obtienen el identificador de archivo, o la función hash puede determinarse por otros medios. Se elige la dirección del Protocolo de Internet que es la primera en este orden y a continuación se usa para establecer una conexión y emitir peticiones para todo o partes del archivo, como se muestra en los pasos 1760 y 1770.

30 **[0374]** El procedimiento anterior puede aplicarse cuando se establece una nueva conexión para solicitar un archivo. También se puede aplicar cuando se dispone de una serie de conexiones establecidas y se puede elegir una de ellas para emitir una nueva petición.

35 **[0375]** Además, cuando una conexión establecida está disponible y se puede elegir una petición de entre un conjunto de peticiones candidatas con la misma prioridad, se induce un orden en las peticiones candidatas, por ejemplo, mediante el mismo procedimiento de valores hash descritos anteriormente y se elige la petición candidata que aparece primero en este orden. Los procedimientos pueden combinarse para seleccionar una conexión y una petición candidata de entre un conjunto de conexiones y peticiones de igual prioridad, nuevamente, calculando un hash para cada combinación de conexión y petición, ordenando estos valores de hash de acuerdo con un orden fijo y eligiendo la combinación que ocurre primero en el orden inducido en el conjunto de combinaciones de peticiones y conexiones.  
40

**[0376]** Este procedimiento tiene la ventaja por la siguiente razón: un enfoque típico adoptado por una infraestructura de servicio de bloques como el que se muestra en la Fig. 1 (BSI 101) o en la Fig. 2 (BSIs 101), y en particular un enfoque comúnmente adoptado por las CDN, es proporcionar múltiples servidores proxy de almacenamiento en memoria caché que reciben peticiones de los clientes. Es posible que no se proporcione a un servidor proxy de almacenamiento en memoria caché el archivo solicitado en una petición determinada y, en este caso, dichos servidores típicamente reenvían la petición a otro servidor, reciben la respuesta de ese servidor, que típicamente incluye el archivo solicitado, y reenvían la respuesta al cliente. El servidor proxy de almacenamiento en memoria caché también puede almacenar (en memoria caché) el archivo solicitado para que pueda responder inmediatamente a las peticiones posteriores del archivo. El enfoque común descrito anteriormente tiene la propiedad de que el conjunto de archivos almacenados en un servidor proxy de almacenamiento en memoria caché determinado está determinado en gran medida por el conjunto de peticiones que el servidor proxy de almacenamiento en memoria caché ha recibido.  
45  
50

**[0377]** El procedimiento descrito anteriormente tiene la siguiente ventaja. Si a todos los clientes en un conjunto de clientes se les proporciona la misma lista de direcciones de Protocolo de Internet, estos clientes utilizarán la misma dirección de Protocolo de Internet para todas las peticiones emitidas para el mismo archivo. Si hay dos listas diferentes de direcciones de Protocolo de Internet y a cada cliente se le proporciona una de estas dos listas, los clientes utilizarán a lo sumo dos direcciones de Protocolo de Internet diferentes para todas las peticiones emitidas para el mismo archivo. En general, si las listas de direcciones de Protocolo de Internet proporcionadas a los clientes son similares, los clientes utilizarán un pequeño conjunto de direcciones de Protocolo de Internet proporcionadas para todas las peticiones emitidas para el mismo archivo. Dado que los clientes próximos tienden a recibir listas similares de direcciones de Protocolo de Internet, es probable que los clientes próximos emitan peticiones de un archivo desde solo una pequeña parte de los servidores proxy de almacenamiento en memoria caché disponibles para esos clientes. Por lo tanto, solo habrá una pequeña fracción de los servidores proxy de almacenamiento en memoria caché que almacenan en memoria caché el archivo, lo cual minimiza ventajosamente la cantidad de recursos de almacenamiento en memoria caché utilizados para almacenar en memoria caché el archivo.  
55  
60  
65

5 **[0378]** Preferentemente, la función hash tiene la propiedad de que una fracción muy pequeña de las diferentes entradas se asignan a la misma salida, y que las diferentes entradas se asignan a salidas esencialmente aleatorias, para garantizar que para un conjunto dado de direcciones de Protocolo de Internet, la proporción de archivos para los cuales una de las direcciones de Protocolo de Internet figura en primer lugar en la lista ordenada producida por el paso 1750 es aproximadamente la misma para todas las direcciones de Protocolo de Internet en la lista. Por otro lado, es importante que la función hash sea determinista, en el sentido de que para una entrada dada, la salida de la función hash es la misma para todos los clientes.

10 **[0379]** Otra ventaja del procedimiento descrito anteriormente es la siguiente. Supongamos que todos los clientes en un conjunto de clientes reciben la misma lista de direcciones de Protocolo de Internet. Debido a las propiedades de la función hash que se acaba de describir, es probable que las peticiones de diferentes archivos de estos clientes se distribuyan de manera uniforme en el conjunto de direcciones de Protocolo de Internet, lo cual a su vez significa que las peticiones se distribuirán de manera uniforme en los servidores proxy de almacenamiento en memoria caché. Por lo tanto, los recursos de almacenamiento en memoria caché utilizados para almacenar estos archivos se distribuyen de manera uniforme entre los servidores proxy de almacenamiento en memoria caché, y las peticiones de archivos se distribuyen de manera uniforme entre los servidores proxy de almacenamiento en memoria caché. Por lo tanto, el procedimiento proporciona equilibrio de almacenamiento y equilibrio de carga en toda la infraestructura de almacenamiento en memoria caché.

20 **[0380]** Los expertos en la técnica conocen varias variaciones del enfoque descrito anteriormente y, en muchos casos, estas variaciones conservan la propiedad de que el conjunto de archivos almacenados en un proxy determinado está determinado, al menos en parte, por el conjunto de peticiones que el servidor proxy de almacenamiento en memoria caché ha recibido. En el caso común en el que un nombre de dispositivo principal determinado se resuelve en varios servidores proxy de almacenamiento en memoria caché físico, será común que todos estos servidores finalmente almacenen una copia de cualquier archivo dado que se solicite con frecuencia. Dicha duplicación puede ser indeseable, ya que los recursos de almacenamiento en los servidores proxy de almacenamiento en memoria caché son limitados y, como resultado, los archivos pueden, en ocasiones, eliminarse (depurarse) de la memoria caché. El nuevo procedimiento descrito aquí garantiza que las peticiones de un archivo determinado se dirijan a los servidores proxy de almacenamiento en memoria caché de tal manera que se reduzca esta duplicación, lo cual reduce la necesidad de eliminar archivos de la memoria caché y, por lo tanto, aumenta la probabilidad de que un archivo determinado esté presente en (es decir, no se haya eliminado) en el memoria caché de proxy.

35 **[0381]** Cuando un archivo está presente en la memoria caché de proxy, la respuesta enviada al cliente es más rápida, lo cual tiene la ventaja de reducir la probabilidad de que el archivo solicitado llegue tarde, lo cual puede provocar una pausa en la reproducción de los medios y, por lo tanto, una mala experiencia del usuario. Además, cuando un archivo no está presente en la memoria caché de proxy, la petición puede enviarse a otro servidor, lo cual provoca una carga adicional tanto en la infraestructura de servicio como en las conexiones de red entre servidores. En muchos casos, el servidor al que se envía la petición puede estar en una ubicación distante y la transmisión del archivo desde este servidor de vuelta al servidor proxy de almacenamiento en memoria caché puede incurrir en costes de transmisión. Por lo tanto, el nuevo procedimiento descrito aquí da como resultado una reducción de estos costes de transmisión.

#### **Peticiones probabilísticas de archivo completo**

45 **[0382]** Un problema particular en el caso de que el protocolo HTTP se use con las peticiones de rango es el comportamiento de los servidores de memoria caché que se usan comúnmente para proporcionar escalabilidad en la infraestructura de servicio. Si bien puede ser común que los servidores de memoria caché HTTP soporten la cabecera de rango HTTP, el comportamiento exacto de los diferentes servidores de memoria caché HTTP varía según la implementación. La mayoría de las implementaciones del servidor de memoria caché atienden las peticiones de rango desde la memoria caché en caso de que el archivo esté disponible en la memoria caché. Una implementación común de los servidores de memoria caché HTTP siempre reenvía las peticiones HTTP en sentido descendente que contienen la cabecera de rango a un nodo en sentido ascendente a menos que el servidor de memoria caché tenga una copia del archivo (servidor de memoria caché o servidor de origen). En algunas implementaciones, la respuesta ascendente a la petición de rango es el archivo completo, y todo este archivo se almacena en memoria caché y la respuesta a la petición de rango descendente se extrae de este archivo y se envía. Sin embargo, en al menos una implementación, la respuesta ascendente a la petición de rango es solo los bytes de datos en la propia petición de rango, y estos bytes de datos no se almacenan en memoria caché, sino que se envían como respuesta a la petición de rango descendente. Como resultado, el uso de cabeceras de rango por parte de los clientes puede tener la consecuencia de que el archivo en sí nunca se pone en memoria caché y se perderán las propiedades de escalabilidad deseables de la red.

60 **[0383]** Anteriormente, se describió el funcionamiento de los servidores proxy de almacenamiento en memoria caché y también se describió el procedimiento para solicitar bloques de un archivo que es una agregación de múltiples bloques. Por ejemplo, esto se puede lograr mediante el uso de la cabecera de petición de rango HTTP. Dichas peticiones se denominan "peticiones parciales" a continuación. Ahora se describe un modo de realización adicional que tiene ventaja en el caso de que la infraestructura de servicio de bloque 101 no proporcione soporte completo para

la cabecera de rango HTTP. Comúnmente, los servidores dentro de una infraestructura de servicio de bloques, por ejemplo una red de entrega de contenido, soportan peticiones parciales, pero pueden no almacenar la respuesta a peticiones parciales en el almacenamiento local (memoria caché). Dicho servidor puede cumplir una petición parcial enviando la petición a otro servidor, a menos que el archivo completo se almacene en un almacenamiento local, en cuyo caso la respuesta puede enviarse sin reenviar la petición a otro servidor.

**[0384]** Un sistema de transmisión de peticiones de bloque que hace uso de la mejora novedosa de la agregación de bloques descrita anteriormente puede funcionar mal si el bloque que sirve a la infraestructura muestra este comportamiento, ya que todas las peticiones, que son peticiones parciales, se reenviarán a otro servidor y no se atenderán peticiones mediante los servidores proxy de almacenamiento en memoria caché, venciendo al objeto de proporcionar los servidores proxy de almacenamiento en memoria caché en primer lugar. Durante el proceso de transmisión de petición de bloques como se ha descrito anteriormente, un cliente puede en algún momento solicitar un bloque que se encuentra al principio de un archivo.

**[0385]** De acuerdo con el nuevo procedimiento descrito aquí, cada vez que se cumple una determinada condición, dichas peticiones pueden convertirse de peticiones para el primer bloque en un archivo a peticiones para todo el archivo. Cuando un servidor proxy de almacenamiento en memoria caché recibe una petición de todo el archivo, el servidor proxy típicamente almacena la respuesta. Por lo tanto, el uso de estas peticiones hace que el archivo se lleve a la memoria caché de los servidores proxy locales de almacenamiento en memoria caché, de modo que las peticiones posteriores, ya sea para el archivo completo o las peticiones parciales, pueden ser atendidas directamente por el servidor proxy de almacenamiento en memoria caché. La condición puede ser tal que entre un conjunto de peticiones asociadas con un archivo dado, por ejemplo, el conjunto de peticiones generado por un conjunto de clientes que ven el elemento de contenido en cuestión, la condición se cumplirá para al menos una fracción proporcionada de estas peticiones.

**[0386]** Un ejemplo de una condición adecuada es que un número elegido al azar está por encima de un umbral proporcionado. Este umbral se puede establecer de tal manera que la conversión de una petición de bloques único en una petición de archivo completo ocurra de media para una fracción proporcionada de las peticiones, por ejemplo, una vez de cada diez (en cuyo caso el número aleatorio se puede elegir del intervalo [0,1] y el umbral puede ser 0,9). Otro ejemplo de una condición adecuada es que una función hash calculada sobre cierta información asociada con el bloque y cierta información asociada con el cliente toma uno de los valores proporcionados. Este procedimiento tiene la ventaja de que para un archivo que se solicita con frecuencia, el archivo se llevará a la memoria caché de un servidor proxy local; sin embargo, el funcionamiento del sistema de transmisión de petición de bloques no se altera significativamente respecto al funcionamiento estándar en la que se realiza cada petición por un solo bloque. En muchos casos, donde ocurre la conversión de la petición de un solo bloque a una petición de archivo completo, los procedimientos del cliente de lo contrario continuarían para solicitar los otros bloques dentro del archivo. Si este es el caso, entonces tales peticiones pueden suprimirse porque los bloques en cuestión serán recibidos en cualquier caso como resultado de la petición del archivo completo.

#### **Construcción de URL y generación de listas de segmentos y búsqueda**

**[0387]** La generación de la lista de segmentos se ocupa del problema de cómo un cliente puede generar una lista de segmentos desde el MPD en una hora local específica del cliente AHORA para una representación específica que comienza en algún tiempo de inicio startTime en relación con el inicio de la presentación de medios para casos de demanda o expresados en tiempo de reloj de pared. Una lista de segmentos puede comprender un localizador, por ejemplo, una URL para metadatos de representación inicial opcional, así como una lista de segmentos de medios. A cada segmento de medios se le puede asignar un startTime, una duración y un localizador. El startTime típicamente expresa una aproximación de la hora de los medios de comunicación contenidos en un segmento, pero no necesariamente una muestra precisa de la hora. El startTime es utilizado por el cliente de transmisión HTTP para emitir la petición de descarga en el momento adecuado. La generación de la lista de segmentos, incluido el tiempo de inicio de cada uno, puede realizarse de diferentes maneras. Las URL pueden proporcionarse como una lista de reproducción o una regla de construcción de URL y pueden usarse ventajosamente para una representación compacta de la lista de segmentos.

**[0388]** Una lista de segmentos basada en la construcción de URL puede, por ejemplo, llevarse a cabo si el MPD señala eso mediante un atributo o elemento específico como FileDynamicInfo o una señal equivalente. A continuación, se proporciona una forma genérica de crear una lista de segmentos a partir de una construcción de URL en la sección "Descripción general de la construcción de URL". Una construcción basada en listas de reproducción puede, por ejemplo, señalarse mediante una señal diferente. La búsqueda en la lista de segmentos y la obtención de un tiempo de medios preciso también se implementa ventajosamente en este contexto.

#### **Descripción general de constructor de URL**

**[0389]** Como se describió anteriormente, en un modo de realización de la presente invención se puede proporcionar un archivo de metadatos que contiene reglas de construcción de URL que permiten a los dispositivos clientes construir los identificadores de archivo para los bloques de la presentación. Ahora describimos otra mejora novedosa en el

sistema de transmisión de petición de bloques que proporciona cambios en el archivo de metadatos, incluidos los cambios en las reglas de construcción de la URL, cambios en el número de codificaciones disponibles, cambios en los metadatos asociados con las codificaciones disponibles, como la velocidad de transmisión de bits, relación de aspecto, resolución, códecs de audio o vídeo, parámetros de códecs u otros parámetros.

**[0390]** En esta mejora novedosa, se pueden proporcionar datos adicionales asociados con cada elemento del archivo de metadatos que indiquen un intervalo de tiempo dentro de la presentación general. Dentro de este intervalo de tiempo, el elemento puede considerarse válido y, dentro de otro intervalo de tiempo, puede ignorarse el elemento. Además, la sintaxis de los metadatos se puede mejorar de tal manera que los elementos que antes se permitían que aparecieran solo una vez o como máximo una vez pueden aparecer varias veces. Se puede aplicar una restricción adicional en este caso que establece que para tales elementos los intervalos de tiempo especificados deben ser diferentes. En cualquier momento dado, considerar solo los elementos cuyo intervalo de tiempo contiene el instante de tiempo dado, da como resultado un archivo de metadatos que es coherente con la sintaxis de metadatos original. A estos intervalos de tiempo los llamamos intervalos de validez. Por lo tanto, este procedimiento proporciona para la señalización dentro de un solo archivo de metadatos cambios del tipo descrito anteriormente. Ventajosamente, este procedimiento se puede utilizar para proporcionar una presentación de medios que soporte cambios del tipo descrito en puntos específicos dentro de la presentación.

### **Constructor de URL**

**[0391]** Como se describe en el presente documento, una característica común de los sistemas de transmisión de petición de bloques es la necesidad de proporcionar al cliente "metadatos" que identifiquen las codificaciones de medios disponibles y proporcionen la información que necesita el cliente para solicitar los bloques de esas codificaciones. Por ejemplo, en el caso de HTTP, esta información puede comprender direcciones URL para los archivos que contienen los bloques de medios. Se puede proporcionar un archivo de lista de reproducción que enumere las URL de los bloques para una codificación determinada. Se proporcionan múltiples archivos de listas de reproducción, uno para cada codificación, junto con una lista de reproducción de listas de reproducción maestra que enumera las listas de reproducción correspondientes a las diferentes codificaciones. Una desventaja de este sistema es que los metadatos pueden llegar a ser bastante grandes y, por lo tanto, tardan en solicitarse cuando el cliente comienza la transmisión. Otra desventaja de este sistema es evidente en el caso de contenido en vivo, cuando los archivos correspondientes a los bloques de datos de medios se generan "al vuelo" desde un flujo de medios que se captura en tiempo real (en vivo), por ejemplo, un programa de noticias o un evento deportivo en vivo. En este caso, los archivos de la lista de reproducción pueden actualizarse cada vez que haya un nuevo bloque disponible (por ejemplo, cada pocos segundos). Los dispositivos cliente pueden recuperar repetidamente el archivo de la lista de reproducción para determinar si hay nuevos bloques disponibles y obtener sus URL. Esto puede suponer una carga significativa para la infraestructura de servicio y, en particular, significa que los archivos de metadatos no pueden almacenarse en memoria caché durante más tiempo que el intervalo de actualización, que es igual al tamaño de bloque que suele ser del orden de unos pocos segundos.

**[0392]** Un aspecto importante de un sistema de transmisión de petición de bloques es el procedimiento utilizado para informar a los clientes sobre los identificadores de archivos, por ejemplo, las direcciones URL, que deben usarse, junto con el protocolo de descarga de archivos, para solicitar bloques. Por ejemplo, un procedimiento en el que para cada representación de una presentación se proporciona un archivo de lista de reproducción que enumera las URL de los archivos que contienen los bloques de datos de medios. Una desventaja de este procedimiento es que al menos parte del propio archivo de la lista de reproducción debe descargarse antes de que pueda comenzar la reproducción, lo cual aumenta el tiempo de zapping y, por lo tanto, provoca una experiencia de usuario deficiente. Para una presentación de medios larga con varias o muchas representaciones, la lista de URL de archivos puede ser grande y, por lo tanto, el archivo de lista de reproducción puede ser grande, lo cual aumenta aún más el tiempo de zapping.

**[0393]** Otra desventaja de este procedimiento ocurre en el caso de contenido en vivo. En este caso, la lista completa de URL no está disponible de antemano y el archivo de la lista de reproducción se actualiza periódicamente a medida que los nuevos bloques están disponibles y los clientes solicitan periódicamente el archivo de la lista de reproducción para recibir la versión actualizada. Debido a que este archivo se actualiza con frecuencia, no se puede almacenar por mucho tiempo dentro de los servidores proxy de almacenamiento en memoria caché. Esto significa que muchas de las peticiones de este archivo se reenviarán a otros servidores y, finalmente, al servidor que genera el archivo. En el caso de una presentación de medios popular, esto puede dar como resultado una gran carga en este servidor y en la red, lo cual a su vez puede dar como resultado un tiempo de respuesta lento y, por lo tanto, un tiempo alto de zapping en el canal y una experiencia de usuario deficiente. En el peor de los casos, el servidor se sobrecarga y esto hace que algunos usuarios no puedan ver la presentación.

**[0394]** Es deseable en el diseño de un sistema de transmisión de petición de bloques evitar poner restricciones en la forma de los identificadores de archivo que se pueden usar. Esto se debe a que una serie de consideraciones pueden motivar el uso de identificadores de una forma particular. Por ejemplo, en el caso de que la infraestructura de servicio en bloque sea una red de entrega de contenido, puede haber convenciones de almacenamiento o denominación de archivos relacionadas con el deseo de distribuir el almacenamiento o la carga de servicio a través

de la red u otros requisitos que conlleven formas particulares de identificador de archivos que no pueden predecirse en el momento del diseño del sistema.

5 **[0395]** Ahora se describe un modo de realización adicional que reduce las desventajas mencionadas anteriormente, a la vez que conserva la flexibilidad para elegir las convenciones de identificación de archivos apropiadas. En este procedimiento, se pueden proporcionar metadatos para cada representación de la presentación de medios que comprende una regla de construcción de identificador de archivo. La regla de construcción del identificador de archivo puede, por ejemplo, comprender una secuencia de texto. Para determinar el identificador de archivo para un bloque dado de la presentación, se puede proporcionar un procedimiento de interpretación de la regla de construcción del identificador de archivo, comprendiendo este procedimiento la determinación de los parámetros de entrada y la evaluación de la regla de construcción de identificación de archivo junto con los parámetros de entrada. Los parámetros de entrada pueden incluir, por ejemplo, un índice del archivo a identificar, donde el primer archivo tiene índice cero, el segundo tiene índice uno, el tercero tiene índice dos y así sucesivamente. Por ejemplo, en el caso de que cada archivo abarque la misma duración (o aproximadamente la misma duración), se puede determinar fácilmente el índice del archivo asociado con un tiempo determinado dentro de la presentación. De forma alternativa, el tiempo dentro de la presentación abarcado por cada archivo puede proporcionarse dentro de la presentación o los metadatos de la versión.

20 **[0396]** En un modo de realización, la regla de construcción del identificador de archivo puede comprender una secuencia de texto que puede contener ciertos identificadores especiales correspondientes a los parámetros de entrada. El procedimiento de evaluación de la regla de construcción del identificador de archivo comprende determinar las posiciones de los identificadores especiales dentro de la secuencia de texto y reemplazar cada uno de dichos identificadores especiales con una representación de secuencia del valor del parámetro de entrada correspondiente.

25 **[0397]** En otro modo de realización, la regla de construcción del identificador de archivo puede comprender una secuencia de texto que se ajuste a un lenguaje de expresión. Un lenguaje de expresión comprende una definición de una sintaxis a la que pueden ajustarse las expresiones en el lenguaje y un conjunto de reglas para evaluar una secuencia conforme a la sintaxis.

30 **[0398]** Ahora se describirá un ejemplo específico, con referencia a las Fig. 21 y ss. Un ejemplo de una definición de sintaxis para un lenguaje de expresión adecuado, definido en la forma de Backus-Naur aumentada, es como se muestra en la Fig. 21. Un ejemplo de reglas para evaluar una secuencia conforme a la producción <expression> en la Fig. 21 comprende la transformación recursiva de la secuencia conforme a la producción <expression> (una <expression>) en una secuencia conforme a la producción <literal> de la siguiente manera:

35 **[0399]** Una <expression> conforme a la producción <literal> no ha cambiado.

**[0400]** Una <expression> conforme a la producción <variable> se sustituye por el valor de la variable identificada por la secuencia <token> de la producción <variable>.

40 **[0401]** Una <expression> conforme a la producción <function> se evalúa evaluando cada uno de sus argumentos de acuerdo con estas reglas y aplicando una transformación a estos argumentos dependiente del elemento <token> de la producción <function> como se describe a continuación.

45 **[0402]** Una <expression> conforme a la última alternativa de la producción <expression> se evalúa evaluando los dos elementos <expression> y aplicando una operación a estos argumentos dependiendo del elemento <operator> de la última alternativa de la producción <expression> como se describe a continuación.

50 **[0403]** En el procedimiento descrito anteriormente, se supone que la evaluación tiene lugar en un contexto en el que se puede definir una pluralidad de variables. Una variable es un par (nombre, valor) donde "nombre" es una secuencia conforme a la producción <token> y "valor" es una secuencia conforme a la producción <literal>. Algunas variables pueden definirse fuera del proceso de evaluación antes de que comience la evaluación. Otras variables pueden definirse dentro del propio proceso de evaluación. Todas las variables son "globales" en el sentido de que solo existe una variable con cada posible "nombre".

55 **[0404]** Un ejemplo de una función es la función "printf ". Esta función acepta uno o más argumentos. El primer argumento puede ser conforme a la producción <string> (en adelante una "secuencia"). La función printf se evalúa como una versión transformada de su primer argumento. La transformación aplicada es la misma que la función "printf "de la biblioteca estándar de C, con los argumentos adicionales incluidos en el producción <function> que proporciona los argumentos adicionales esperados por la función printf de biblioteca estándar de C.

60 **[0405]** Otro ejemplo de una función es la función "hash". Esta función acepta dos argumentos, el primero de los cuales puede ser una secuencia y el segundo de los cuales puede ser compatible con la producción <number> (en adelante un "número"). La función "hash" aplica un algoritmo de hash al primer argumento y devuelve un resultado que es un número entero no negativo menor que el segundo argumento. Un ejemplo de una función hash adecuada se da en la función C mostrada en la Fig. 22, cuyos argumentos son la secuencia de entrada (excluyendo las comillas) y el valor de entrada numérico. Otros ejemplos de funciones hash son bien conocidos por los expertos en la técnica.

65

[0406] Otro ejemplo de una función es la función "Subst" que toma uno, dos o tres argumentos de secuencia. En el caso de que se proporcione un argumento, el resultado de la función "Subst" es el primer argumento. En el caso de que se proporcionen dos argumentos, el resultado de la función "Subst" se calcula borrando cualquier ocurrencia del segundo argumento (excluyendo las comillas) dentro del primer argumento y devolviendo el primer argumento así modificado. En el caso de que se proporcionen tres argumentos, el resultado de la función "Subst" se calcula reemplazando cualquier ocurrencia del segundo argumento (excluyendo las comillas) dentro del primer argumento con el tercer argumento (excluyendo las comillas adjuntas) y devolviendo el primer argumento así modificado.

[0407] Algunos ejemplos de operadores son los operadores de suma, resta, división, multiplicación y módulo, identificados por las producciones <operator> "+", "-", "/", "\*", "%" respectivamente. Estos operadores requieren que las producciones <expression> a ambos lados de la producción <operator> se evalúen a números. La evaluación del operador comprende aplicar la operación aritmética apropiada (suma, resta, división, multiplicación y módulo, respectivamente) a estos dos números de la manera habitual y devolver el resultado en una forma compatible con la producción <number>.

[0408] Otro ejemplo de un operador es el operador de asignación, identificado por la producción <operator> "=". Este operador requiere que el argumento de la izquierda evalúe una secuencia cuyo contenido sea compatible con la producción <token>. El contenido de una secuencia se define como la secuencia de caracteres dentro de las comillas que la contienen. El operador de igualdad ocasiona la variable cuyo nombre es el <token> igual al contenido del argumento izquierdo al que se le asigna un valor igual al resultado de evaluar el argumento correcto. Este valor también es el resultado de evaluar la expresión del operador.

[0409] Otro ejemplo de un operador es el operador de secuencia, identificado por la producción <operator> ";". El resultado de evaluar este operador es el argumento correcto. Tenga en cuenta que, como con todos los operadores, ambos argumentos se evalúan y el argumento de la izquierda se evalúa primero.

[0410] En un modo de realización de esta invención, el identificador de un archivo puede obtenerse evaluando una regla de construcción de identificador de archivo de acuerdo con la regla anterior con un conjunto específico de variables de entrada que identifican el archivo requerido. Un ejemplo de una variable de entrada es la variable con el nombre "índice" y un valor igual al índice numérico del archivo dentro de la presentación. Otro ejemplo de una variable de entrada es la variable con el nombre de "velocidad de transmisión de bits" y un valor igual a la velocidad de transmisión de bits media de la versión requerida de la presentación.

[0411] La Fig. 23 ilustra algunos ejemplos de reglas de construcción de identificador de archivo, donde las variables de entrada son "id", dando un identificador para la representación de la presentación deseada y "seq", dando un número de secuencia para el archivo

[0412] Como quedará claro para los expertos en la técnica al leer esta divulgación, son posibles numerosas variaciones del procedimiento anterior. Por ejemplo, no se pueden proporcionar todas las funciones y operadores descritos anteriormente o se pueden proporcionar funciones u operadores adicionales.

### **Reglas y temporización de construcción de URL**

[0413] Esta sección proporciona reglas básicas de construcción de URI para asignar un URI de archivo o segmento, así como un tiempo de inicio para cada segmento dentro de una representación y la presentación de medios.

[0414] Para esta cláusula, se supone la disponibilidad de una descripción de presentación de medios en el cliente.

[0415] Supongamos que el cliente de transmisión HTTP está reproduciendo los medios que se descargan dentro de una presentación de medios. El tiempo de presentación real del cliente HTTP se puede definir en cuanto a dónde el tiempo de presentación es relativo al inicio de la presentación. En la inicialización, el tiempo de presentación  $t=0$  puede suponerse.

[0416] En cualquier punto  $t$ , el cliente HTTP puede descargar cualquier dato con el tiempo de reproducción  $tP$  (también en relación con el inicio de la presentación) a lo sumo `MaximumClientPreBufferTime` antes del tiempo de presentación real  $t$  y cualquier dato que sea necesario debido a la interacción del usuario, por ejemplo buscar, adelantar, etc. En algunos modos de realización, el `MaximumClientPreBufferTime` ni siquiera puede especificarse en el sentido de que un cliente puede descargar datos antes del tiempo de reproducción actual  $tP$  sin restricciones.

[0417] El cliente HTTP puede evitar la descarga de datos innecesarios, por ejemplo, los segmentos de las representaciones que no se espera que se reproduzcan típicamente no se pueden descargar.

[0418] El proceso básico para proporcionar los servicios de transmisión puede ser la descarga de datos mediante la generación de peticiones apropiadas para descargar archivos/segmentos completos o un subconjunto de archivos/segmentos, por ejemplo, mediante peticiones GET de HTTP o peticiones GET parciales de HTTP. Esta

descripción explica cómo acceder a los datos para un tiempo de reproducción específico  $t_P$ , pero en general el cliente puede descargar los datos durante un intervalo de tiempo de reproducción mayor para evitar peticiones ineficientes. El cliente HTTP puede minimizar el número/frecuencia de las peticiones HTTP al proporcionar el servicio de transmisión.

5 **[0419]** Para acceder a los datos de medios en el tiempo de reproducción  $t_P$  o al menos cerca del tiempo de reproducción  $t_P$  en una representación específica, el cliente determina la URL del archivo que contiene este tiempo de reproducción y, además, determina el rango de bytes en el archivo para acceder a este tiempo de reproducción.

10 **[0420]** La descripción de presentación de medios puede asignar un ID de representación,  $r$ , a cada representación, por ejemplo, mediante el uso del atributo RepresentationID. En otras palabras, el contenido del MPD, cuando está escrito por el sistema de ingestión o cuando es leído por el cliente, se interpretará de manera tal que haya una asignación. Para descargar datos durante un tiempo de reproducción específico  $t_P$  para una representación específica con id  $r$ , el cliente puede construir un URI apropiado para un archivo.

15 **[0421]** La descripción de presentación de los medios puede asignar a cada archivo o segmento de cada representación  $r$  los siguientes atributos:

20 **[0422]** (a) un número de secuencia  $i$  del archivo dentro de la representación  $r$ , con  $i=1, 2, \dots, N_r$ , (b) el tiempo de inicio relativo del archivo con ID de representación  $r$  y el índice de archivo  $i$  en relación con el tiempo de presentación, definido como  $ts(r,i)$ , (c) el URI de archivo para el archivo/segmento con ID de representación  $r$  y el índice de archivo  $i$ , indicado como FileURI( $r, i$ ).

25 **[0423]** En un modo de realización, el tiempo de inicio del archivo y los URI del archivo pueden proporcionarse explícitamente para una representación. En otro modo de realización, puede proporcionarse explícitamente una lista de URI de archivo donde a cada URI de archivo se le asigna inherentemente el índice  $i$  de acuerdo con la posición en la lista y el tiempo de inicio del segmento se obtiene como la suma de todas las duraciones de segmento para los segmentos de 1 a  $i-1$ . La duración de cada segmento puede proporcionarse de acuerdo con cualquiera de las reglas analizadas anteriormente. Por ejemplo, cualquier experto en matemáticas básicas puede usar otros procedimientos para obtener una metodología para obtener fácilmente el tiempo de inicio de un elemento o atributo único y la posición/índice del URI del archivo en la representación.

30 **[0424]** Si se proporciona una regla de construcción de URI dinámica en el MPD, entonces el tiempo de inicio de cada archivo y cada URI de archivo se pueden construir dinámicamente usando una regla de construcción, el índice del archivo solicitado y posiblemente algunos parámetros adicionales proporcionados en la descripción de presentación de medios. La información se puede proporcionar, por ejemplo, en atributos y elementos MPD como FileURIPattern y FileInfoDynamic. FileURIPattern proporciona información sobre cómo construir los URI basándose en el número de secuencia de índice de archivo  $i$  y el ID de representación  $r$ . El FileURIFormat se construye como:

40 `FileURIFormat=sprintf("%s%s%s%s%s.%s", BaseURI, BaseFileName,`  
`RepresentationIDFormat,`  
`SeparatorFormat,`  
`FileSequenceIDFormat,`  
`FileExtension);`  
 45 `and the FileURI(r,i) is constructed as`  
`FileURI(r,i)=sprintf(FileURIFormat, r, i);`

50 **[0425]** El tiempo de inicio relativo  $ts(r, i)$  para cada archivo/segmento puede obtenerse de algún atributo contenido en el MPD que describe la duración de los segmentos en esta representación, por ejemplo, el atributo FileInfoDynamic. El MPD también puede contener una secuencia de atributos de FileInfoDynamic que es global para todas las representaciones en la presentación de medios o al menos para todas las representaciones en un período de la misma manera que se especificó anteriormente. Si se solicitan datos de medios para un tiempo de reproducción específico  $t_P$  en la representación  $r$ , el índice correspondiente  $i(r, t_P)$  se puede obtener como  $i(r, t_p)$  de modo que el tiempo de reproducción de este índice se encuentre en el intervalo del tiempo de inicio de  $ts(r, i(r, t_P))$  y  $ts(r, i(r, t_P)+1)$ . El acceso al segmento se puede restringir aún más por los casos anteriores, por ejemplo, el segmento no es accesible.

60 **[0426]** Acceder al tiempo de reproducción exacto  $t_P$  una vez que se obtiene el índice y el URI del segmento correspondiente depende del formato del segmento real. En este ejemplo, suponga que los segmentos de medios tienen una línea de tiempo local que comienza en 0 sin pérdida de generalidad. Para acceder y presentar los datos en tiempo de reproducción  $t_P$ , el cliente puede descargar los datos correspondientes a la hora local desde el archivo/segmento al que se puede acceder a través del URI FileURI( $r,i$ ) con  $i=i(r, t_p)$ .

65 **[0427]** En general, los clientes pueden descargar el archivo completo y luego pueden acceder al tiempo de reproducción  $t_P$ . Sin embargo, no necesariamente es necesario descargar toda la información del archivo 3GP, ya que el archivo 3GP proporciona estructuras para asignar la temporización local a los rangos de bytes. Por lo tanto, solo los rangos de bytes específicos para acceder al tiempo de reproducción  $t_P$  pueden ser suficientes para reproducir los

medios siempre que haya suficiente información de acceso aleatorio disponible. También se puede proporcionar información suficiente sobre la estructura y la asignación del rango de bytes y la temporización local del segmento de medios en la parte inicial del segmento, por ejemplo, utilizando un índice de segmento. Al tener acceso a los, por ejemplo, 1200 bytes, iniciales del segmento, el cliente puede tener información suficiente para acceder directamente al rango de bytes necesario para el tiempo de reproducción tP.

[0428] En un ejemplo adicional, suponga que el índice de segmento, posiblemente especificado como la ventana "tidx" como se muestra a continuación, puede usarse para identificar las desviaciones de bytes del Fragmento o Fragmentos requeridos. Se pueden formar peticiones GET parciales para el Fragmento o Fragmentos requeridos. Hay otras alternativas, por ejemplo, el cliente puede emitir una petición estándar para el archivo y cancelarla cuando se haya recibido la primera ventana "tidx".

### **Buscando**

[0429] Un cliente puede intentar buscar un tiempo de presentación específico tp en una representación. Basándose en el MPD, el cliente tiene acceso al tiempo de inicio del segmento de medios y la URL del segmento de medios de cada segmento en la representación. El cliente puede obtener el índice de segmento del segmento que probablemente contenga muestras de medios durante el tiempo de presentación tp como el índice de segmento máximo i, para el cual el tiempo de inicio tS(r,i) es menor o igual al tiempo de presentación tp, es decir,  $\text{segment\_index} = \max \{ i \mid tS(r,i) \leq tp \}$ . La URL del segmento se obtiene como FileURI(r,i).

[0430] Tenga en cuenta que la información de tiempo en el MPD puede ser aproximada, debido a problemas relacionados con la ubicación de los puntos de acceso aleatorio, la alineación de las pistas de los medios y la desviación de los tiempos de los medios. Como resultado, el segmento identificado por el procedimiento anterior puede comenzar en un momento ligeramente posterior a tp y los datos de medios para el tiempo de presentación tp pueden estar en el segmento de medios anterior. En el caso de la búsqueda, el tiempo de búsqueda puede actualizarse para igualar el tiempo de la primera muestra del archivo recuperado, o el archivo anterior puede recuperarse en su lugar. Sin embargo, tenga en cuenta que durante la reproducción continua, incluidos los casos en los que hay un cambio entre representaciones/versiones alternativas, los datos de medios del tiempo entre tp y el inicio del segmento recuperado están sin embargo disponibles.

[0431] Para una búsqueda precisa de un tiempo de presentación tp, el cliente de transmisión HTTP necesita acceder a un punto de acceso aleatorio (RAP). Para determinar el punto de acceso aleatorio en un segmento de medios en el caso de transmisión HTTP adaptable 3GPP, el cliente puede, por ejemplo, usar la información en la ventana "tidx" o "sidx", si está presente, para ubicar los puntos de acceso aleatorio y el tiempo de presentación correspondiente en la presentación de medios. En los casos en que un segmento es un fragmento de película 3GPP, también es posible que el cliente use información dentro de las ventanas "moof" y "mdat", por ejemplo, para localizar RAP y obtener el tiempo de presentación necesario a partir de la información en el fragmento de película. y el tiempo de inicio del segmento obtenido del MPD. Si no está disponible un RAP con tiempo de presentación antes del tiempo de presentación solicitado tp, el cliente puede acceder al segmento anterior o simplemente puede usar el primer punto de acceso aleatorio como resultado de la búsqueda. Cuando los segmentos de medios comienzan con un RAP, estos procedimientos son simples.

[0432] También tenga en cuenta que no necesariamente toda la información del segmento de medios debe descargarse para acceder al tiempo de presentación tp. El cliente puede, por ejemplo, solicitar inicialmente la ventana "tidx" o "sidx" desde el principio del segmento de medios utilizando peticiones de rango de bytes. Mediante el uso de las ventanas "tidx" o "sidx", la temporización del segmento se puede asignar a los rangos de bytes del segmento. Al utilizar continuamente las peticiones HTTP parciales, solo es necesario acceder a las partes relevantes del segmento de medios, para mejorar la experiencia del usuario y reducir los retardos en el inicio.

### **Generación de lista de segmentos**

[0433] Como se describe en el presente documento, debería ser evidente cómo implementar un cliente de transmisión HTTP sencillo que utilice la información proporcionada por el MPD para crear una lista de segmentos para una representación que tiene una duración de segmento aproximada señalada de *dur*. En algunos modos de realización, el cliente puede asignar los segmentos de medios dentro de índices consecutivos de representación  $i = 1, 2, 3, \dots$ , es decir, al primer segmento de medios se le asigna un índice  $i=1$ , al segundo segmento de medios se le asigna el índice  $i=2$ , y así sucesivamente. A continuación, a la lista de segmentos de medios con índices de segmento  $i$  se le asigna  $\text{startTime}[i]$  y la URL[i] se genera, por ejemplo, de la siguiente manera. En primer lugar, el índice  $i$  se establece en 1. El tiempo de inicio del primer segmento de medios se obtiene como 0,  $\text{startTime}[1] = 0$ . La URL del segmento de medios  $i$ , URL[i], se obtiene como FileURI(r, i). El proceso continúa para todos los segmentos de medios descritos con índice  $i$  y el  $\text{startTime}[i]$  del segmento de medios  $i$  se obtiene como  $(i-1) \cdot \text{dur}$  y la URL[i] se obtiene como FileURI(r,i).

### **Peticiones HTTP/TCP concurrentes**

**[0434]** Un problema en un sistema de transmisión de petición de bloques es un deseo de solicitar siempre los bloques de la más alta calidad que se pueden recibir completamente a tiempo para la reproducción. Sin embargo, la velocidad de llegada de datos puede no conocerse de antemano y, por lo tanto, puede suceder que un bloque solicitado no llegue a tiempo para ser ejecutado. Esto da como resultado una necesidad de pausar la reproducción de los medios, lo cual da como resultado una experiencia de usuario deficiente. Este problema puede ser reducido por algoritmos de clientes que adoptan un enfoque conservador para la selección de bloques a solicitar mediante la petición de bloques de menor calidad (y por lo tanto de menor tamaño) que es más probable que se reciban a tiempo, incluso si la velocidad de llegada de datos disminuye durante la recepción del bloque. Sin embargo, este enfoque conservador tiene la desventaja de posiblemente ofrecer una reproducción de menor calidad al usuario o dispositivo de destino, lo cual también es una mala experiencia para el usuario. El problema puede ampliarse cuando se usan múltiples conexiones HTTP al mismo tiempo para descargar diferentes bloques, como se describe a continuación, ya que los recursos de red disponibles se comparten entre conexiones y, por lo tanto, se usan simultáneamente para bloques con diferentes tiempos de reproducción.

**[0435]** Puede ser ventajoso para el cliente emitir peticiones para múltiples bloques simultáneamente, donde en este contexto "concurrentemente" significa que las respuestas a las peticiones se producen en intervalos de tiempo superpuestos, y no es necesariamente el caso que las peticiones se realicen con precisión o incluso aproximadamente al mismo tiempo. En el caso del protocolo HTTP, este enfoque puede mejorar la utilización del ancho de banda disponible debido al comportamiento del protocolo TCP (como es bien sabido). Esto puede ser especialmente importante para mejorar el tiempo de zapping de contenido, ya que cuando se solicita un nuevo contenido por primera vez, las conexiones HTTP/TCP correspondientes sobre las cuales se solicitan los datos para los bloques pueden comenzar lentamente, y por lo tanto usar varias conexiones HTTP/TCP en este punto puede acelerar de manera espectacular el tiempo de entrega de datos de los primeros bloques. Sin embargo, la petición de diferentes bloques o fragmentos a través de diferentes conexiones HTTP/TCP también puede ocasionar un rendimiento degradado, ya que las peticiones de los bloques que se van a reproducir primero compiten con las peticiones de los bloques posteriores, las descargas HTTP/TCP que compiten varían enormemente en su velocidad de entrega y, por lo tanto, el tiempo de finalización de la petición puede ser muy variable, y en general no es posible controlar qué descargas de HTTP/TCP se realizarán con mayor rapidez y cuáles serán más lentas, y por lo tanto es probable que al menos parte del tiempo, las descargas de HTTP/TCP de los primeros bloques serán las últimas en completarse, lo cual conllevará tiempos de zapping de grandes y variables.

**[0436]** Supongamos que cada bloque o fragmento de un segmento se descarga a través de una conexión HTTP/TCP independiente, y que el número de conexiones paralelas es  $n$  y la duración de reproducción de cada bloque es  $t$  segundos, y que la velocidad de transmisión del contenido asociado con el segmento es  $S$ . Cuando el cliente comienza a transmitir el contenido por primera vez, se pueden emitir peticiones para los primeros  $n$  bloques, que representan  $n \cdot t$  segundos de datos de medios.

**[0437]** Como saben los expertos en la técnica, existe una gran variación en la velocidad de datos de las conexiones TCP. Sin embargo, para simplificar este análisis, suponga que lo ideal es que todas las conexiones se realicen en paralelo, de modo que el primer bloque se reciba completamente casi al mismo tiempo que los otros  $n-1$  bloques solicitados. Para simplificar aún más el análisis, suponga que el ancho de banda agregado utilizado por las  $n$  conexiones de descarga se fija en un valor  $B$  durante toda la descarga, y que la velocidad de transmisión  $S$  es constante en toda la representación. Supongamos además que la estructura de datos de medios es tal que la reproducción de un bloque se puede hacer cuando todo el bloque está disponible en el cliente, es decir, la reproducción de un bloque solo puede comenzar después de que se recibe el bloque completo, por ejemplo, debido a la estructura de la codificación de vídeo subyacente, o porque el cifrado se está empleando para cifrar cada fragmento o bloque por separado, y por lo tanto, todo el fragmento o bloque debe recibirse antes de que se pueda descifrar. Por lo tanto, para simplificar el análisis siguiente, suponemos que se debe recibir un bloque completo antes de que se pueda reproducir cualquiera de los bloques. A continuación, el tiempo requerido antes de que llegue el primer bloque y se pueda reproducir es aproximadamente  $n \cdot t \cdot S / B$ .

**[0438]** Dado que es deseable minimizar el tiempo de zapping de contenido, por lo tanto, es deseable minimizar  $n \cdot t \cdot S / B$ . El valor de  $t$  puede determinarse mediante factores tales como la estructura de codificación de vídeo subyacente y cómo se utilizan los procedimientos de ingestión, y por lo tanto,  $t$  puede ser razonablemente pequeño, pero los valores muy pequeños de  $t$  ocasionan un mapa de segmentos demasiado complicado y posiblemente pueden ser incompatibles con codificación y descifrado de vídeo eficiente, si se usan. El valor de  $n$  también puede afectar al valor de  $B$ , es decir,  $B$  puede ser mayor para un número mayor de conexiones  $n$ , y por lo tanto, reducir el número de conexiones,  $n$ , tiene el efecto secundario negativo de reducir potencialmente la cantidad de ancho de banda disponible que se utiliza,  $B$ , y por lo tanto puede no ser efectivo para lograr el objetivo de reducir el tiempo de zapping del contenido. El valor de  $S$  depende de la representación elegida para descargar y reproducir, e idealmente  $S$  debería estar lo más cerca posible de  $B$  para maximizar la calidad de reproducción de los medios para las condiciones de red dadas. Por lo tanto, para simplificar este análisis, suponga que  $S$  es aproximadamente igual a  $B$ . A continuación, el tiempo de zapping es proporcional a  $n \cdot t$ . Por lo tanto, utilizar más conexiones para descargar diferentes fragmentos puede degradar el tiempo de zapping si el ancho de banda agregado utilizado por las conexiones es sublinealmente proporcional al número de conexiones, que suele ser el caso típicamente.

[0439] Como ejemplo, suponga que  $t = 1$  segundo, y con  $n = 1$  el valor de  $B = 500$  Kbps, y con  $n = 2$  el valor de  $B = 700$  Kbps, y con  $n = 3$  el valor de  $B = 800$  Kbps. Supongamos que se elige la representación con  $S = 700$  Kbps. A continuación, con  $n = 1$  el tiempo de descarga para el primer bloque es  $1 \cdot 700 / 500 = 1,4$  segundos, con  $n = 2$  el tiempo de descarga para el primer bloque es  $2 \cdot 700 / 700 = 2$  segundos y con  $n = 3$  el tiempo de descarga para el primer bloque es  $3 \cdot 700 / 800 = 2,625$  segundos. Además, a medida que aumenta el número de conexiones, es probable que aumente la variabilidad en las velocidades de descarga individuales de las conexiones (aunque incluso con una conexión es probable que haya una variabilidad significativa). Así, en este ejemplo, el tiempo de zapping y la variabilidad en el tiempo de zapping aumentan a medida que aumenta el número de conexiones. Intuitivamente, los bloques que se están entregando tienen diferentes prioridades, es decir, el primer bloque tiene la fecha límite de entrega más temprana, el segundo bloque tiene la segunda fecha límite más temprana, etc., mientras que las conexiones de descarga sobre las cuales se entregan los bloques compiten por la red recursos durante la entrega, y por lo tanto los bloques con los primeros plazos se retardan a medida que se solicitan más bloques de la competencia. Por otro lado, incluso en este caso, el uso de más de una conexión de descarga en última instancia permite el soporte de una velocidad de transmisión más alta de manera sostenible; por ejemplo, con tres conexiones, puede soportarse una velocidad de transmisión de hasta 800 Kbps en este ejemplo, mientras que solo puede soportarse una transmisión de 500 Kbps con una conexión.

[0440] En la práctica, como se indicó anteriormente, la velocidad de datos de una conexión puede ser altamente variable tanto dentro de la misma conexión a lo largo del tiempo y entre las conexiones y, como resultado, los  $n$  bloques solicitados en general no se completan al mismo tiempo y, de hecho, pueden comúnmente ser el caso que un bloque pueda completarse en la mitad del tiempo que otro bloque. Este efecto da como resultado un comportamiento impredecible, ya que en algunos casos el primer bloque puede completarse mucho antes que otros bloques y en otros casos el primer bloque puede completarse mucho más tarde que otros bloques y, como resultado, el inicio de la reproducción en algunos casos puede ocurrir con relativa rapidez y en otros casos puede tardar en ocurrir. Este comportamiento impredecible puede ser frustrante para el usuario y, por lo tanto, puede considerarse una experiencia de usuario deficiente.

[0441] Por lo tanto, lo que se necesita son procedimientos en los que se puedan utilizar múltiples conexiones TCP para mejorar el tiempo de zapping y la variabilidad en el tiempo de zapping, mientras que al mismo tiempo se soporte una velocidad de transmisión de la mejor calidad posible. Lo que también se necesita son procedimientos para permitir que la proporción del ancho de banda disponible asignado a cada bloque se ajuste a medida que se aproxima el tiempo de reproducción de un bloque, de modo que, si es necesario, se pueda asignar una mayor parte del ancho de banda disponible hacia el bloque con el tiempo de reproducción más cercano.

### **Petición cooperativa de HTTP/TCP**

[0442] Ahora describimos procedimientos para usar peticiones HTTP/TCP concurrentes de manera cooperativa. Un receptor puede emplear múltiples peticiones HTTP/TCP cooperativas concurrentes, por ejemplo, utilizando una pluralidad de peticiones de rango de bytes HTTP, en el que cada petición de este tipo es para una parte de un fragmento en un segmento de origen, o todo un fragmento de un segmento de origen, o una parte o un fragmento de reparación de un segmento de reparación, o para todo un fragmento de reparación de un segmento de reparación.

[0443] Las ventajas de las peticiones HTTP/TCP cooperativas junto con el uso de los datos de reparación FEC pueden ser especialmente importantes para proporcionar tiempos de zapping rápidos de forma constante. Por ejemplo, es probable que las conexiones TCP se hayan iniciado recientemente o hayan estado inactivas durante un período de tiempo en el tiempo de zapping, en cuyo caso la ventana de congestión,  $cwnd$ , se encuentra en su valor mínimo para las conexiones, y por lo tanto la velocidad de entrega de estas conexiones TCP tardará varios tiempos de ida y vuelta (RTT) para aumentar, y habrá una gran variabilidad en las velocidades de entrega en las diferentes conexiones TCP durante este tiempo de aumento.

[0444] Ahora se describe una descripción general del procedimiento No-FEC, que es un procedimiento cooperativo de petición HTTP/TCP en el que solo se solicitan datos de medios de los bloques de origen que utilizan múltiples conexiones HTTP/TCP concurrentes, es decir, no se solicitan datos de reparación FEC. Con el procedimiento No-FEC, se solicitan partes del mismo fragmento a través de diferentes conexiones, por ejemplo, utilizando peticiones de rango de bytes HTTP para partes del fragmento, y por lo tanto, por ejemplo, cada petición de rango de bytes HTTP es para una parte del rango de bytes indicada en el mapa de segmentos para el fragmento. Puede darse el caso de que una petición HTTP/TCP individual aumente su velocidad de entrega para utilizar completamente el ancho de banda disponible en varios RTT (tiempos de ida y vuelta), y por lo tanto hay un período de tiempo relativamente largo en el que la velocidad de entrega es menor que el ancho de banda disponible y, por lo tanto, si se utiliza una única conexión HTTP/TCP para descargar, por ejemplo, el primer fragmento de un contenido que se va a reproducir, el tiempo de zapping podría ser grande. Usando el procedimiento No-FEC, la descarga de diferentes porciones del mismo fragmento a través de diferentes conexiones HTTP/TCP puede reducir significativamente el tiempo de zapping.

[0445] Ahora se describe una descripción general del procedimiento FEC, que es un procedimiento cooperativo de petición HTTP/TCP en el que los datos de medios de un segmento de origen y los datos de reparación FEC generados a partir de los datos de medios se solicitan mediante múltiples conexiones HTTP/TCP concurrentes. Con el

procedimiento FEC, se solicitan partes del mismo fragmento y datos de reparación FEC generados a partir de ese fragmento a través de diferentes conexiones, utilizando peticiones de rango de bytes HTTP para partes del fragmento, y por lo tanto, por ejemplo, cada petición de rango de bytes HTTP es para una parte del rango de bytes indicada en el mapa de segmentos para el fragmento. Puede darse el caso de que una petición HTTP/TCP individual aumente su velocidad de entrega para utilizar completamente el ancho de banda disponible en varios RTT (tiempos de ida y vuelta), y por lo tanto hay un período de tiempo relativamente largo en el que la velocidad de entrega es menor que el ancho de banda disponible y, por lo tanto, si se utiliza una única conexión HTTP/TCP para descargar, por ejemplo, el primer fragmento de un contenido que se va a reproducir, el tiempo de zapping podría ser grande. El uso del procedimiento FEC tiene las mismas ventajas que el procedimiento No-FEC, y tiene la ventaja adicional de que no es necesario que todos los datos solicitados lleguen antes de poder recuperar el fragmento, lo cual reduce aún más el tiempo de zapping y la variabilidad en el tiempo de zapping. Al realizar peticiones a través de diferentes conexiones TCP, y realizar peticiones adicionales también solicitando datos de reparación FEC en al menos una de las conexiones, la cantidad de tiempo que se tarda en entregar una cantidad suficiente de datos para, por ejemplo, recuperar el primer fragmento solicitado que permite a los medios el comienzo de la reproducción, puede reducirse considerablemente y ser mucho más coherente que si no se usaran conexiones TCP cooperativas y datos de reparación FEC.

**[0446]** Las Figs. 24 (a)-(e) muestran un ejemplo de las fluctuaciones en la velocidad de entrega de 5 conexiones TCP que se ejecutan en el mismo enlace al mismo cliente desde el mismo servidor web HTTP de una red optimizada de datos de evolución emulados (EVDO). En las Figs. 24 (a)-(e), el eje X muestra el tiempo en segundos, y el eje Y muestra la velocidad a la que se reciben los bits en el cliente en cada una de las 5 conexiones TCP medidas en intervalos de 1 segundo, para cada conexión. En esta emulación en particular, había 12 conexiones TCP en total en este enlace, y por lo tanto la red se cargó relativamente durante el tiempo que se muestra, lo cual puede ser típico cuando más de un cliente está transmitiendo dentro de la misma célula de una red móvil. Tenga en cuenta que aunque las velocidades de entrega están algo correlacionadas con el tiempo, hay una gran diferencia en las velocidades de entrega de las 5 conexiones en muchos puntos en el tiempo.

**[0447]** La Fig. 25 muestra una posible estructura de petición para un fragmento que tiene un tamaño de 250.000 bits (aproximadamente 31,25 kilobytes), donde hay 4 peticiones de rango de bytes HTTP realizadas en paralelo para diferentes partes del fragmento, es decir, la primera conexión HTTP solicita los primeros 50.000 bits, la segunda conexión HTTP solicita los siguientes 50.000 bits, la tercera conexión HTTP solicita los siguientes 50.000 bits y la cuarta conexión HTTP solicita los siguientes 50.000 bits. Si no se utiliza FEC, es decir, el procedimiento No-FEC, estas son las únicas 4 peticiones para el fragmento en este ejemplo. Si se usa FEC, es decir, el procedimiento FEC, entonces en este ejemplo hay una conexión HTTP adicional que solicita 50.000 bits adicionales de datos de reparación FEC de un segmento de reparación generado a partir del fragmento.

**[0448]** La Fig. 26 es una explosión de los primeros dos segundos de las 5 conexiones TCP mostradas en las Figs. 24(a) - (e), donde en la Fig. 26 el eje X muestra el tiempo a intervalos de 100 milisegundos, y el eje Y muestra la velocidad a la que se reciben los bits en el cliente en cada una de las 5 conexiones TCP medidas en intervalos de 100 milisegundos. Una línea muestra la cantidad agregada de bits que se han recibido en el cliente para el fragmento de las primeras 4 conexiones HTTP (excluyendo la conexión HTTP a través de la cual se solicitan los datos de FEC), es decir, lo que llega usando el procedimiento No-FEC. Otra línea muestra la cantidad agregada de bits que se ha recibido en el cliente para el fragmento desde las 5 conexiones HTTP (incluida la conexión HTTP a través de la cual se solicitan los datos de FEC), es decir, lo que llega usando el procedimiento FEC. Para el procedimiento FEC, se supone que el fragmento puede ser descodificado por FEC a partir de la recepción de 200.000 bits de los 250.000 bits solicitados, lo cual puede realizarse si, por ejemplo, se utiliza un código FEC Reed-Solomon, y puede realizarse esencialmente si por ejemplo, se utiliza el código RaptorQ descrito en la solicitud de patente de Estados Unidos n.º 12/859,161 presentada el 18 de agosto de 2010. Para el procedimiento FEC en este ejemplo, se reciben suficientes datos para recuperar el fragmento utilizando la descodificación FEC después de 1 segundo, lo que permite un tiempo de zapping de 1 segundo (suponiendo que los datos para los fragmentos subsiguientes pueden solicitarse y recibirse antes de que el primer fragmento se reproduzca completamente). Para el procedimiento No-FEC en este ejemplo, todos los datos para las 4 peticiones deben recibirse antes de que se pueda recuperar el fragmento, lo cual ocurre después de 1,7 segundos, lo cual conlleva un tiempo de zapping de 1,7 segundos. Por lo tanto, en el ejemplo que se muestra en la Fig. 26, el procedimiento No-FEC es un 70 % peor en términos de tiempo de zapping que el procedimiento FEC. Una de las razones de la ventaja mostrada por el procedimiento FEC en este ejemplo es que, para el procedimiento FEC, la recepción de cualquier 80 % de los datos solicitados permite la recuperación del fragmento, mientras que para el procedimiento No-FEC, se requiere la recepción del 100 % de los datos solicitados. Por lo tanto, el procedimiento No-FEC tiene que esperar a que la conexión TCP más lenta finalice la entrega, y debido a las variaciones naturales en la velocidad de entrega de TCP, es probable que haya una amplia variación en la velocidad de entrega de la conexión TCP más lenta en comparación con una conexión TCP media. Con el procedimiento FEC en este ejemplo, una conexión TCP lenta no determina cuándo se puede recuperar el fragmento. En cambio, para el procedimiento FEC, la entrega de datos suficientes es mucho más una función de la velocidad de entrega de TCP media que la velocidad de entrega de TCP en el peor de los casos.

**[0449]** Hay muchas variaciones del procedimiento No-FEC y del procedimiento FEC descritos anteriormente. Por ejemplo, las peticiones HTTP/TCP cooperativas se pueden usar solo para los primeros fragmentos después de que se haya producido un zapping, y posteriormente solo se usa una sola petición HTTP/TCP para descargar más

fragmentos, fragmentos múltiples o segmentos completos. Como otro ejemplo, el número de conexiones HTTP/TCP cooperativas utilizadas puede ser una función tanto de la urgencia de los fragmentos solicitados, es decir, cuán inminente es el tiempo de reproducción de estos fragmentos, como de las condiciones actuales de la red.

5 **[0450]** En algunas variaciones, se puede usar una pluralidad de conexiones HTTP para solicitar datos de reparación de segmentos de reparación. En otras variaciones, se pueden solicitar diferentes cantidades de datos en diferentes conexiones HTTP, por ejemplo, dependiendo del tamaño actual de la memoria intermedia de medios y la velocidad de recepción de datos en el cliente. En otra variación, las representaciones de origen no son independientes entre sí, sino que representan una codificación de medios en capas, donde, por ejemplo, una representación de origen mejorada  
10 puede depender de una representación de origen base. En este caso, puede haber una representación de reparación correspondiente a la representación de origen de base, y otra representación de reparación correspondiente a la combinación de las representaciones de origen de base y de mejora.

15 **[0451]** Los elementos generales adicionales se suman a las ventajas que uno puede obtener mediante los procedimientos divulgados anteriormente. Por ejemplo, la cantidad de conexiones HTTP utilizadas puede variar dependiendo de la cantidad actual de medios en la memoria intermedia de medios y/o la velocidad de recepción en la memoria intermedia de medios. Las peticiones HTTP cooperativas que usan FEC, es decir, el procedimiento FEC descrito anteriormente y las variantes de ese procedimiento, se pueden usar agresivamente cuando la memoria intermedia de medios está relativamente vacía; por ejemplo, se realizan peticiones HTTP más cooperativas en paralelo  
20 para diferentes partes del primer fragmento, solicitando todo el fragmento de origen y una fracción relativamente grande de los datos de reparación del fragmento de reparación correspondiente, y a continuación haciendo transición hacia un número reducido de peticiones HTTP concurrentes, solicitando partes más grandes de los datos de medios por petición, y solicitando una fracción más pequeña de datos de reparación, por ejemplo, haciendo transición hacia 1, 2 o 3 peticiones HTTP simultáneas, haciendo transición hacia la realización de peticiones de fragmentos completos  
25 o múltiples fragmentos consecutivos por petición, y haciendo transición hacia la petición de datos sin reparación, a medida que crece la memoria intermedia de medios.

**[0452]** Como otro ejemplo, la cantidad de datos de reparación FEC puede variar en función del tamaño de la memoria intermedia de medios, es decir, cuando la memoria intermedia de medios es pequeña, se pueden solicitar más datos de reparación FEC y, a medida que aumenta la memoria intermedia de medios, entonces la cantidad de datos de reparación FEC solicitada podría disminuir, y en algún momento cuando la memoria intermedia de medios es lo suficientemente grande, entonces no se pueden solicitar datos de reparación FEC, solo datos de segmentos de origen de representaciones de origen. El beneficio de estas técnicas mejoradas es que pueden permitir tiempos de zapping más rápidos y constantes, y mayor resistencia contra posibles detenciones o reproducciones entrecortadas de medios,  
30 minimizando al mismo tiempo la cantidad de ancho de banda adicional utilizada más allá de la cantidad que consumiría solo la entrega de los medios en los segmentos de origen al reducir el tráfico de mensajes de petición y los datos de reparación FEC, permitiendo al mismo tiempo el soporte de las velocidades de medios más altas posibles para las condiciones de red dadas.

#### 40 **Mejoras adicionales al usar conexiones HTTP concurrentes**

**[0453]** Una petición HTTP/TCP puede abandonarse si se cumple una condición adecuada y se puede realizar otra petición HTTP/TCP para descargar datos que pueden reemplazar a los datos solicitados en la petición abandonada, en el que la segunda petición HTTP/TCP puede solicitar exactamente los mismos datos. que en la petición original,  
45 por ejemplo, datos de origen; o datos superpuestos, por ejemplo, algunos de los mismos datos de origen y datos de reparación que no se solicitaron en la primera petición; o datos completamente diferentes, por ejemplo, datos de reparación que no se habían solicitado en la primera petición. Un ejemplo de una condición adecuada es que una petición falla debido a la ausencia de una respuesta de la Infraestructura del Servidor de Bloques (BSI) dentro de un tiempo determinado o un fallo en el establecimiento de una conexión de transporte a la BSI o la recepción de un mensaje de fallo explícito desde el servidor u otra condición de fallo.  
50

**[0454]** Otro ejemplo de una condición adecuada es que la recepción de datos avanza de manera inusualmente lenta, de acuerdo con una comparación de una medida de la velocidad de conexión (velocidad de llegada de datos en respuesta a la petición en cuestión) con la velocidad de conexión esperada o con una estimación de la velocidad de conexión requerida para recibir la respuesta antes del tiempo de reproducción de los datos de medios contenidos en el mismo u otro tiempo que dependa de ese tiempo.  
55

**[0455]** Este enfoque tiene ventajas en el caso de que el BSI a veces muestre fallos o un rendimiento deficiente. En este caso, el enfoque anterior aumenta la probabilidad de que el cliente pueda continuar la reproducción fiable de los datos de medios a pesar de los fallos o el bajo rendimiento dentro del BSI. Tenga en cuenta que, en algunos casos, puede ser ventajoso diseñar el BSI de tal manera que muestre tales fallos o un rendimiento deficiente en ocasiones; por ejemplo, un diseño de este tipo puede tener un coste menor que un diseño alternativo que no exhibe tales fallos o mal rendimiento o que los exhibe con menos frecuencia. En este caso, el procedimiento descrito en el presente documento tiene una ventaja adicional, ya que permite la utilización de un diseño de menor coste para el BSI sin la consiguiente degradación en la experiencia del usuario.  
60  
65

[0456] En otro modo de realización, el número de peticiones emitidas para datos correspondientes a un bloque dado puede depender de si se cumple una condición adecuada con respecto al bloque. Si no se cumple la condición, entonces se puede restringir al cliente para que realice más peticiones del bloque si la finalización exitosa de todas las peticiones de datos actualmente incompletas para el bloque permitiría la recuperación del bloque con alta probabilidad. Si se cumple la condición, se puede emitir un mayor número de peticiones para el bloqueo, es decir, la restricción anterior no se aplica. Un ejemplo de una condición adecuada es que el tiempo hasta el tiempo de reproducción programado del bloque u otro tiempo dependiente de ese tiempo cae por debajo del umbral proporcionado. Este procedimiento tiene la ventaja de que las peticiones adicionales de datos para un bloque se emiten cuando la recepción del bloque se vuelve más urgente, debido a que el tiempo de reproducción de los datos de medios que comprenden el bloque está cerca. En el caso de los protocolos de transporte comunes, como HTTP/TCP, estas peticiones adicionales tienen el efecto de aumentar la proporción del ancho de banda disponible dedicado a los datos que contribuyen a la recepción del bloque en cuestión. Esto reduce el tiempo requerido para la recepción de datos suficientes para recuperar el bloque y, por lo tanto, reduce la probabilidad de que el bloque no pueda recuperarse antes del tiempo de reproducción programado de los datos de medios que comprenden el bloque. Como se describió anteriormente, si el bloque no puede recuperarse antes del tiempo de reproducción programado de los datos de medios que comprenden el bloque, la reproducción puede pausarse, lo cual da como resultado una experiencia de usuario deficiente y, por lo tanto, el procedimiento descrito aquí reduce ventajosamente la probabilidad de esta mala experiencia de usuario.

[0457] Debe entenderse que, a lo largo de esta especificación, las referencias al tiempo de reproducción programado de un bloque se refieren al momento en que los datos de medios codificados que comprenden el bloque pueden estar disponibles primero en el cliente para lograr la reproducción de la presentación sin pausa. Como quedará claro para los expertos en la técnica de los sistemas de presentación de medios, este tiempo es en la práctica un poco anterior al tiempo real de aparición de los medios que comprenden el bloque en los transductores físicos utilizados para la reproducción (pantalla, altavoz, etc.), ya que es posible que se deban aplicar varias funciones de transformación a los datos de medios que comprenden el bloque para efectuar la reproducción real de ese bloque y estas funciones pueden requerir una cierta cantidad de tiempo para completarse. Por ejemplo, los datos de medios en general se transportan en forma comprimida y se puede aplicar una transformación de descompresión.

### Procedimientos para generar estructuras de archivos compatibles con procedimientos cooperativos de HTTP/FEC

[0458] Ahora se describe un modo de realización para generar una estructura de archivos que puede ser utilizada ventajosamente por un cliente que emplea procedimientos cooperativos de HTTP/FEC. En este modo de realización, para cada segmento de origen hay un segmento de reparación correspondiente generado de la siguiente manera. El parámetro R indica de media la cantidad de datos de reparación FEC que se generan para los datos de origen en los segmentos de origen. Por ejemplo,  $R = 0,33$  indica que si un segmento de origen contiene 1.000 kilobytes de datos, entonces el segmento de reparación correspondiente contiene aproximadamente 330 kilobytes de datos de reparación. El parámetro S indica el tamaño del símbolo en bytes utilizados para la codificación y decodificación FEC. Por ejemplo,  $S = 64$  indica que los datos de origen y los datos de reparación comprenden símbolos de tamaño de 64 bytes cada uno para los fines de codificación y decodificación FEC.

[0459] El segmento de reparación se puede generar para un segmento de origen de la siguiente manera. Cada fragmento del segmento de origen se considera como un bloque de origen para fines de codificación de FEC y, por lo tanto, cada fragmento se trata como una secuencia de símbolos de origen de un bloque de origen a partir del cual se generan los símbolos de reparación. El número de símbolos de reparación en total generado para los primeros fragmentos  $i$  se calcula como  $TNRS(i) = \text{ceiling}(R \cdot B(i)/S)$ , en el que  $\text{ceiling}(x)$  es la función que genera el entero más pequeño con un valor que es al menos  $x$ . Por lo tanto, el número de símbolos de reparación generados para el fragmento  $i$  es  $NRS(i) = TNRS(i) - TNRS(i-1)$ .

[0460] El segmento de reparación comprende una concatenación de los símbolos de reparación para los fragmentos, en el que el orden de los símbolos de reparación dentro de un segmento de reparación es el orden de los fragmentos a partir de los cuales se generan, y dentro de un fragmento los símbolos de reparación están en el orden de su identificador de símbolo de codificación (ESI). La estructura del segmento de reparación correspondiente a la estructura del segmento de origen se muestra en la Fig. 27, incluido un generador de segmento de reparación 2700.

[0461] Tenga en cuenta que al definir el número de símbolos de reparación para un fragmento como se ha descrito anteriormente, el número total de símbolos de reparación para todos los fragmentos anteriores, y por lo tanto el índice de bytes en el segmento de reparación, solo depende de R, S,  $B(i-1)$  y  $B(i)$ , y no depende de ninguna de las estructuras anteriores o posteriores de los fragmentos dentro del segmento de origen. Esto es ventajoso porque le permite a un cliente calcular rápidamente la posición del inicio de un bloque de reparación dentro del segmento de reparación, y también calcular rápidamente el número de símbolos de reparación dentro de ese bloque de reparación, utilizando solo información local sobre la estructura del fragmento correspondiente del segmento de origen a partir del cual se genera el bloque de reparación. Por lo tanto, si un cliente decide comenzar a descargar y reproducir un fragmento desde la mitad de un segmento de origen, también puede generar y acceder rápidamente al bloque de reparación correspondiente desde dentro del segmento de reparación correspondiente.

[0462] El número de símbolos de origen en el bloque de origen correspondiente al fragmento  $i$  se calcula como  $NSS(i) = \text{ceiling}((B(i)-B(i-1))/S)$ . El último símbolo de origen se completa con cero bytes para los fines de la codificación y decodificación FEC si  $B(i)-B(i-1)$  no es un múltiplo de  $S$ , es decir, el último símbolo de origen se rellena con cero bytes, por lo que tiene un tamaño de  $S$  bytes para los fines de codificación y decodificación FEC, pero estos bytes de relleno cero no se almacenan como parte del segmento de origen. En este modo de realización, las ESI para el símbolo de origen son  $0, 1, \dots, NSS(i)-1$  y las ESI para los símbolos de reparación son  $NSS(i), \dots, NSS(i)+NRS(i)-1$ .

[0463] La URL para un segmento de reparación en este modo de realización puede generarse a partir de la URL para el segmento de origen correspondiente simplemente agregando, por ejemplo, el sufijo ".repair" a la URL del segmento de origen.

[0464] La información de indexación de reparación y la información de FEC para un segmento de reparación están definidas implícitamente por la información de indexación para el segmento de origen correspondiente, y a partir de los valores de  $R$  y  $S$ , como se describe en el presente documento. Las desviaciones de tiempo y la estructura de fragmentos que comprende el segmento de reparación están determinadas por las desviaciones de tiempo y la estructura del segmento de origen correspondiente. La desviación de bytes al final de los símbolos de reparación en el segmento de reparación correspondiente al fragmento  $i$  se puede calcular como  $RB(i) = S \cdot \text{ceiling}(R \cdot B(i)/S)$ . El número de bytes en el segmento de reparación correspondiente al fragmento  $i$  es entonces  $RB(i)-RB(i-1)$  y, por lo tanto, el número de símbolos de reparación correspondientes al fragmento  $i$  se calcula como  $NRS(i) = (RB(i) - RB(i-1))/S$ . El número de símbolos de origen correspondientes al fragmento  $i$  se puede calcular como  $NSS(i) = \text{ceiling}((B(i)-B(i-1))/S)$ . Por lo tanto, en este modo de realización, la información de indexación de reparación para un bloque de reparación dentro de un segmento de reparación y la información FEC correspondiente pueden obtenerse implícitamente de  $R$ ,  $S$  y la información de indexación para el fragmento correspondiente del segmento de origen correspondiente.

[0465] Como ejemplo, considere el ejemplo que se muestra en la Fig. 28, que muestra un fragmento 2 que comienza en la desviación de bytes  $B(1) = 6.410$  y termina en la desviación de bytes  $B(2) = 6.770$ . En este ejemplo, el tamaño del símbolo es  $S = 64$  bytes, y las líneas verticales de puntos muestran las desviaciones de bytes dentro del segmento de origen que corresponden a los múltiplos de  $S$ . El tamaño total del segmento de reparación global como una fracción del tamaño del segmento de origen se establece en  $R = 0,5$  en este ejemplo. El número de símbolos de origen en el bloque de origen para el fragmento 2 se calcula como  $NSS(2) = \text{ceiling}((6.770-6.410)/64) = \text{ceil}(5,625) = 6$ , y estos 6 símbolos de origen tienen ESI  $0, \dots, 5$ , respectivamente, en el que el primer símbolo de origen son los primeros 64 bytes del fragmento 2 que comienza en el índice de bytes 6.410 dentro del segmento de origen, el segundo símbolo de origen son los siguientes 64 bytes del fragmento 2 que comienza en el índice de bytes 6.474 dentro del segmento de origen, etc. La desviación de bytes final del bloque de reparación correspondiente al fragmento 2 se calcula como  $RB(2) = 64 \cdot \text{ceiling}(0,5 \cdot 6.770/64) = 64 \cdot \text{ceiling}(52,89\dots) = 64 \cdot 53 = 3392$ , y la desviación de bytes de inicio del bloque de reparación correspondiente al fragmento 2 se calcula como  $RB(1) = 64 \cdot \text{ceiling}(0,5 \cdot 6.410/64) = 64 \cdot \text{ceiling}(50,07\dots) = 64 \cdot 51 = 3.264$ , y por lo tanto, en este ejemplo hay dos símbolos de reparación en el bloque de reparación correspondientes al fragmento 2 con ESI  $6$  y  $7$ , respectivamente, comenzando en la desviación de bytes 3.264 dentro del segmento de reparación y terminando en la desviación de bytes 3.392.

[0466] Tenga en cuenta que, en el ejemplo que se muestra en la Fig. 28, aunque  $R = 0,5$  y hay 6 símbolos de origen correspondientes al fragmento 2, el número de símbolos de reparación no es 3, como cabría esperar si uno simplemente usara el número de símbolos de origen para calcular el número de símbolos de reparación, sino que en cambio resultó ser 2 de acuerdo con los procedimientos descritos en el presente documento. A diferencia de simplemente usar el número de símbolos de origen de un fragmento para determinar el número de símbolos de reparación, los modos de realización descritos anteriormente permiten calcular la posición del bloque de reparación dentro del segmento de reparación únicamente a partir de la información de índice asociada con el bloque de origen correspondiente del segmento de origen correspondiente. Además, a medida que crece el número  $K$  de símbolos de origen en un bloque de origen, el número de símbolos de reparación,  $KR$ , del bloque de reparación correspondiente se aproxima mucho a  $K \cdot R$ , ya que en general,  $KR$  es a lo sumo  $\text{ceil}(K \cdot R)$  y  $KR$  es al menos  $\text{floor}((K-1) \cdot R)$ , donde  $\text{floor}(x)$  es el entero más grande que tiene como máximo  $x$ .

[0467] Existen muchas variaciones de los modos de realización anteriores para generar una estructura de archivos que puede ser utilizada ventajosamente por un cliente que emplea procedimientos cooperativos de HTTP/FEC, como reconocerá un experto en la técnica. Como ejemplo de un modo de realización alternativo, un segmento original para una representación puede dividirse en  $N > 1$  segmentos paralelos, en el que para  $i = 1, \dots, N$ , una fracción específica  $F_i$  del segmento original está contenida en el  $i$ -ésimo segmento paralelo, y donde la suma para  $i = 1, \dots, N$  de  $F_i$  es igual a 1. En este modo de realización, puede haber un mapa de segmentos principal que se usa para obtener los mapas de segmentos para todos los segmentos paralelos, de manera similar a cómo se obtiene el mapa de segmentos de reparación a partir del mapa de segmentos de origen en el modo de realización descrito anteriormente. Por ejemplo, el mapa de segmentos principal puede indicar la estructura del fragmento si todos los datos de medios de origen no se dividieron en segmentos paralelos, sino que se incluyeron en el segmento original, y a continuación el mapa de segmentos para el segmento paralelo  $i$ -ésimo puede obtenerse a partir del mapa de segmentos principal calculando que, si la cantidad de datos de medios  $i$  en un primer prefijo de fragmentos del segmento original es  $L$  bytes, entonces

el número total de bytes de este prefijo en conjunto entre el primer segmento paralelo  $i$  es  $\text{ceil}(L \cdot G_i)$ , donde  $G_i$  es la suma sobre  $j = 1, \dots, i$  de  $F_j$ . Como otro ejemplo de un modo de realización alternativo, los segmentos pueden consistir en la combinación de los datos de medios de origen originales para cada fragmento seguidos inmediatamente por los datos de reparación de ese fragmento, dando como resultado un segmento que contiene una mezcla de datos de medios de origen y los datos de reparación generados utilizando un código FEC de los datos de medios de origen.

5 Como otro ejemplo de un modo de realización alternativo, un segmento que contiene una mezcla de datos de medios de origen y datos de reparación puede dividirse en múltiples segmentos paralelos que contienen una mezcla de datos de medios de origen y datos de reparación.

10 **[0468]** Se pueden contemplar modos de realización adicionales para un experto en la técnica después de leer esta divulgación. En otros modos de realización, pueden realizarse ventajosamente combinaciones o sub-combinaciones de la invención divulgada anteriormente. Los ejemplos de disposiciones de componentes se muestran con fines ilustrativos y debe entenderse que las combinaciones, adiciones, reorganizaciones y similares se contemplan en modos de realización alternativos de la presente invención. Por lo tanto, aunque la invención se ha descrito con respecto a modos de realización a modo de ejemplo, un experto en la técnica reconocerá que son posibles numerosas modificaciones.

15 **[0469]** Por ejemplo, los procesos descritos en el presente documento pueden implementarse utilizando componentes de hardware, componentes de software y/o cualquier combinación de los mismos. En algunos casos, los componentes del software se pueden proporcionar en medios tangibles, no transitorios, para su ejecución en el hardware que se proporciona con los medios o está separado de los medios. La especificación y los dibujos, en consecuencia, deben considerarse en un sentido ilustrativo, en lugar de en un sentido restrictivo.

20 **[0470]** A continuación se describen ejemplos adicionales para facilitar el entendimiento de la invención, considerando que la invención sólo se define mediante las reivindicaciones adjuntas.

25 1. En un sistema de comunicación en el que un dispositivo cliente recibe bloques de medios de uno o más sistemas de ingestión de medios, un procedimiento que comprende:

30 identificar, en el dispositivo cliente, el tamaño de una memoria intermedia disponible para almacenar bloques de medios recibidos antes de ser consumidos por una salida del dispositivo cliente;

identificar, en el dispositivo cliente, una tasa de consumo de los medios en la salida del dispositivo cliente;

35 determinar una tasa de cambio del tamaño de la memoria intermedia disponible; y

determinar, para una pluralidad de peticiones de bloques correspondientes a bloques de los medios consumidos por la salida del dispositivo cliente, un bloque a solicitar y una temporización de la petición, en el que el bloque específico solicitado y la temporización de la petición dependen del tamaño de la memoria intermedia disponible y la tasa de cambio del tamaño de la memoria intermedia disponible.

40

2. En un sistema de comunicación en el que un dispositivo cliente recibe bloques de medios de uno o más sistemas de ingestión de medios, un procedimiento que comprende:

45 identificar, en el dispositivo cliente, la cantidad de una memoria intermedia que se utiliza para almacenar bloques de medios recibidos antes de que sean consumidos por una salida del dispositivo cliente;

determinar, en el dispositivo cliente, una tasa de consumo de los bloques recibidos almacenados;

50 determinar, basándose en la tasa de consumo y la cantidad de memoria intermedia que se utiliza,

el tamaño de la petición a efectuar a un sistema de ingestión de medios; y

emisión de la petición al sistema de ingestión de medios.

55

3. En un sistema de comunicación en el que un dispositivo cliente solicita datos de medios de un sistema de ingestión de medios, un procedimiento que comprende:

60 determinar, en el dispositivo cliente, una cantidad de bytes recibidos del sistema de ingestión de medios en una petición previa; y

solicitar datos de medios adicionales del sistema de ingestión de medios basándose en una prueba de la cantidad de bytes recibidos.

65 4. En un sistema de comunicación en el que un dispositivo cliente solicita archivos de medios de uno o más servidores, un procedimiento que comprende:

- determinar, en el dispositivo cliente, un identificador del archivo de medios a solicitar;
- 5 determinar un conjunto de direcciones IP de los servidores;
- ordenar las direcciones IP basándose en un cálculo que involucre el identificador; y
- seleccionar un servidor basándose en el orden.
- 10 5. En un sistema de comunicación en el que un dispositivo cliente recibe bloques de medios de un archivo almacenado en un sistema de ingestión de medios, un procedimiento que comprende:
- identificar, en el dispositivo cliente, una serie de bloques que se solicitarán al sistema de ingestión de medios;
- 15 determinar si el rango de bloques incluye un primer bloque en el archivo;
- calcular un parámetro estadístico; y
- 20 convertir la petición de un rango de bloques en una petición para el archivo, basándose en la determinación y basándose en el parámetro estadístico.
6. El procedimiento según el ejemplo 5, en el que el parámetro estadístico corresponde a un umbral.
7. El procedimiento según el ejemplo 5, en el que el parámetro estadístico corresponde a la salida de una función hash que se basa en el rango de bloques.
- 25 8. El procedimiento según el ejemplo 5, en el que un umbral y un valor se establecen de manera que se convierta un porcentaje deseado de peticiones para el archivo, a partir de múltiples receptores.

**REIVINDICACIONES**

- 5

1. En un sistema de comunicación en el que un dispositivo cliente recibe bloques de medios de uno o más sistemas de ingestión de medios, un procedimiento que comprende:

identificar (126), en el dispositivo cliente, un tamaño de una memoria intermedia disponible (125) para almacenar bloques de medios recibidos antes de ser consumidos por una salida del dispositivo cliente (127, 128);

10

determinar (126) una tasa de cambio del tamaño de la memoria intermedia disponible; y

determinar (123), para una pluralidad de peticiones de bloques correspondientes a bloques de los medios consumidos por la salida del dispositivo cliente, un bloque a solicitar y una temporización de la petición, en el que tanto el bloque a solicitar como la temporización de la petición dependen del tamaño de la memoria intermedia disponible y de la tasa de cambio del tamaño de la memoria intermedia disponible, en el que la tasa de cambio del tamaño de la memoria intermedia disponible se mide mediante la relación entre la cantidad de medios recibidos en un período de tiempo determinado y la duración de dicho período de tiempo.
- 20

2. El procedimiento según la reivindicación 1, en el que el bloque solicitado y la temporización de la petición dependen además de una variable de estado.
- 25

3. El procedimiento según la reivindicación 2, en el que la variable de estado asume un número discreto de valores.
- 30

4. El procedimiento según la reivindicación 2, que comprende además:

determinar el valor de la variable de estado en función del tamaño de la memoria intermedia disponible y la tasa de cambio del tamaño de la memoria intermedia disponible.
- 35

5. El procedimiento según la reivindicación 4, en el que dicha función usa una partición de sus parámetros para proporcionar un nuevo valor para la variable de estado.
- 40

6. El procedimiento según la reivindicación 5, en el que la partición se basa en una tabla de configuración, donde dicha tabla de configuración comprende conjuntos de valores de umbral, en el que dichos valores de umbral definen regiones desunidas, estando cada región anotada con un nuevo valor de la variable de estado.
- 45

7. El procedimiento según la reivindicación 6, en el que un valor de histéresis está asociado a cada valor de umbral.
- 50

8. El procedimiento según la reivindicación 1, en el que el bloque solicitado y la temporización de la petición dependen adicionalmente de la disponibilidad de recursos computacionales en el dispositivo cliente.
- 55

9. El procedimiento según la reivindicación 7, en el que la tabla de configuración se proporciona al dispositivo cliente como parte de los metadatos.
- 60

10. En un sistema de comunicación en el que un dispositivo cliente recibe bloques de medios de uno o más sistemas de ingestión de medios, un aparato que comprende:

medios para identificar (126), en el dispositivo cliente, un tamaño de una memoria intermedia disponible (125) para almacenar bloques de medios recibidos antes de ser consumidos por una salida del dispositivo cliente (127, 128);

medios para determinar (126) una tasa de cambio del tamaño de la memoria intermedia disponible; y

medios para determinar (123), para una pluralidad de peticiones de bloques correspondientes a bloques de los medios consumidos por la salida del dispositivo cliente, un bloque a solicitar y una temporización de la petición, en el que tanto el bloque a solicitar como la temporización de la petición dependen del tamaño de la memoria intermedia disponible y de la tasa de cambio del tamaño de la memoria intermedia disponible, en el que la tasa de cambio del tamaño de la memoria intermedia disponible se mide mediante la relación entre la cantidad de medios recibidos en un período de tiempo determinado y la duración de dicho período de tiempo.
- 65

11. Un programa informático que comprende instrucciones ejecutables para provocar que al menos un ordenador realice un procedimiento de acuerdo con una de las reivindicaciones 1 a 9 cuando se ejecuten.

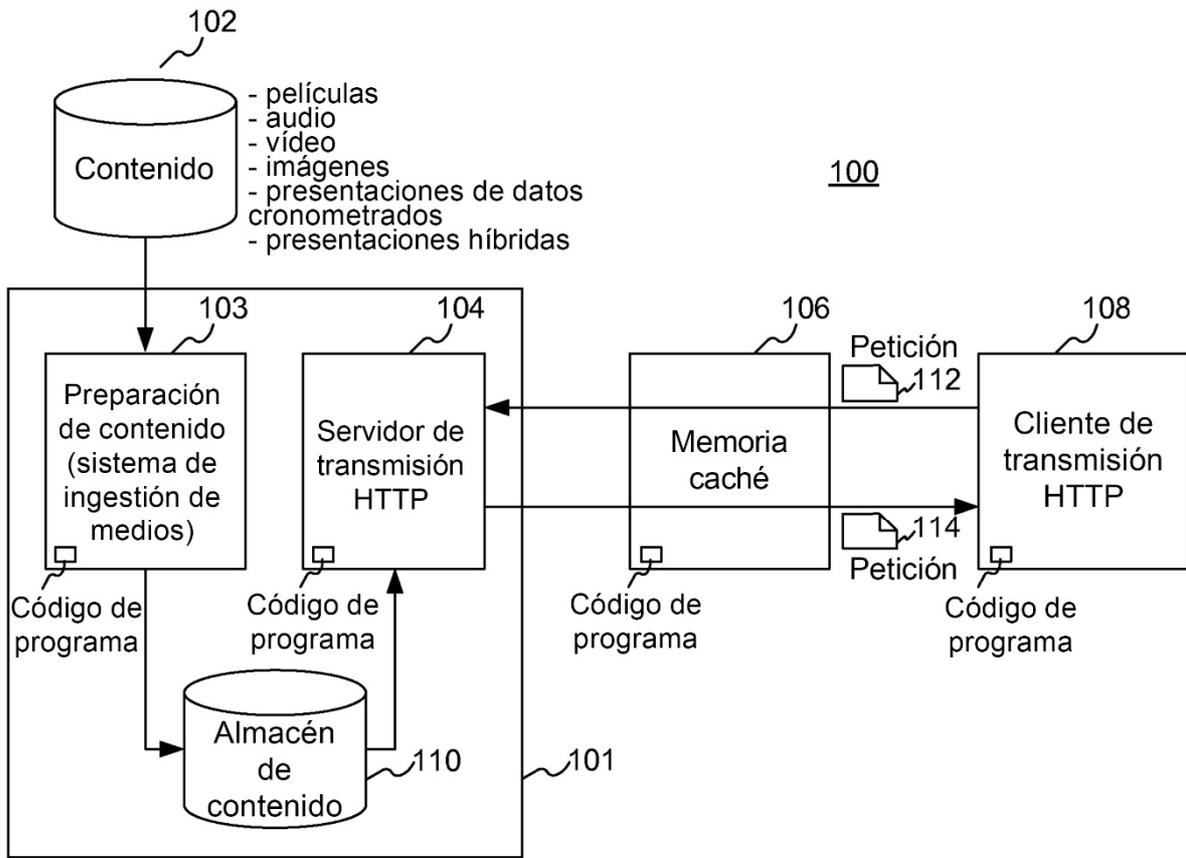


FIG. 1

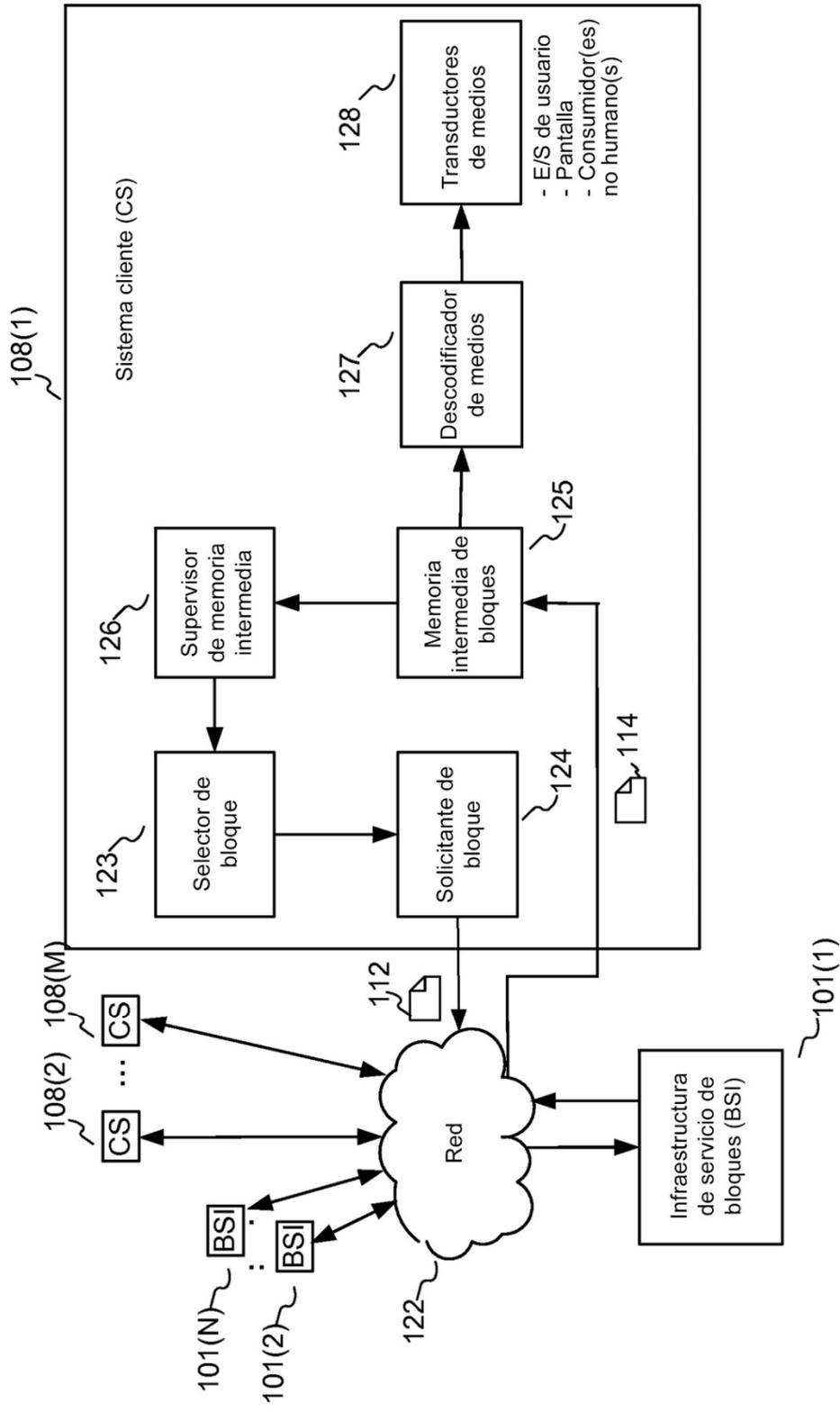


FIG. 2

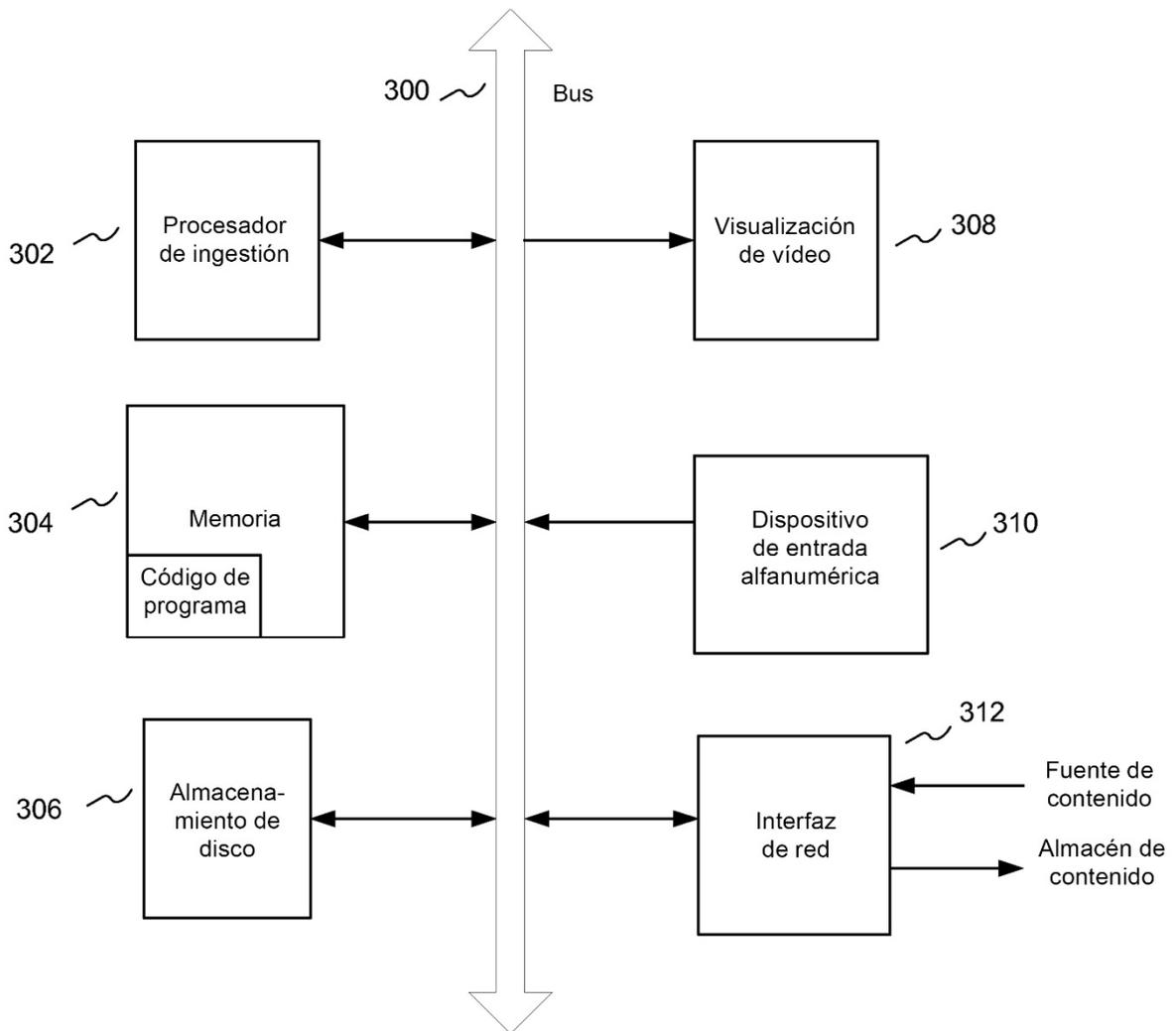


FIG. 3

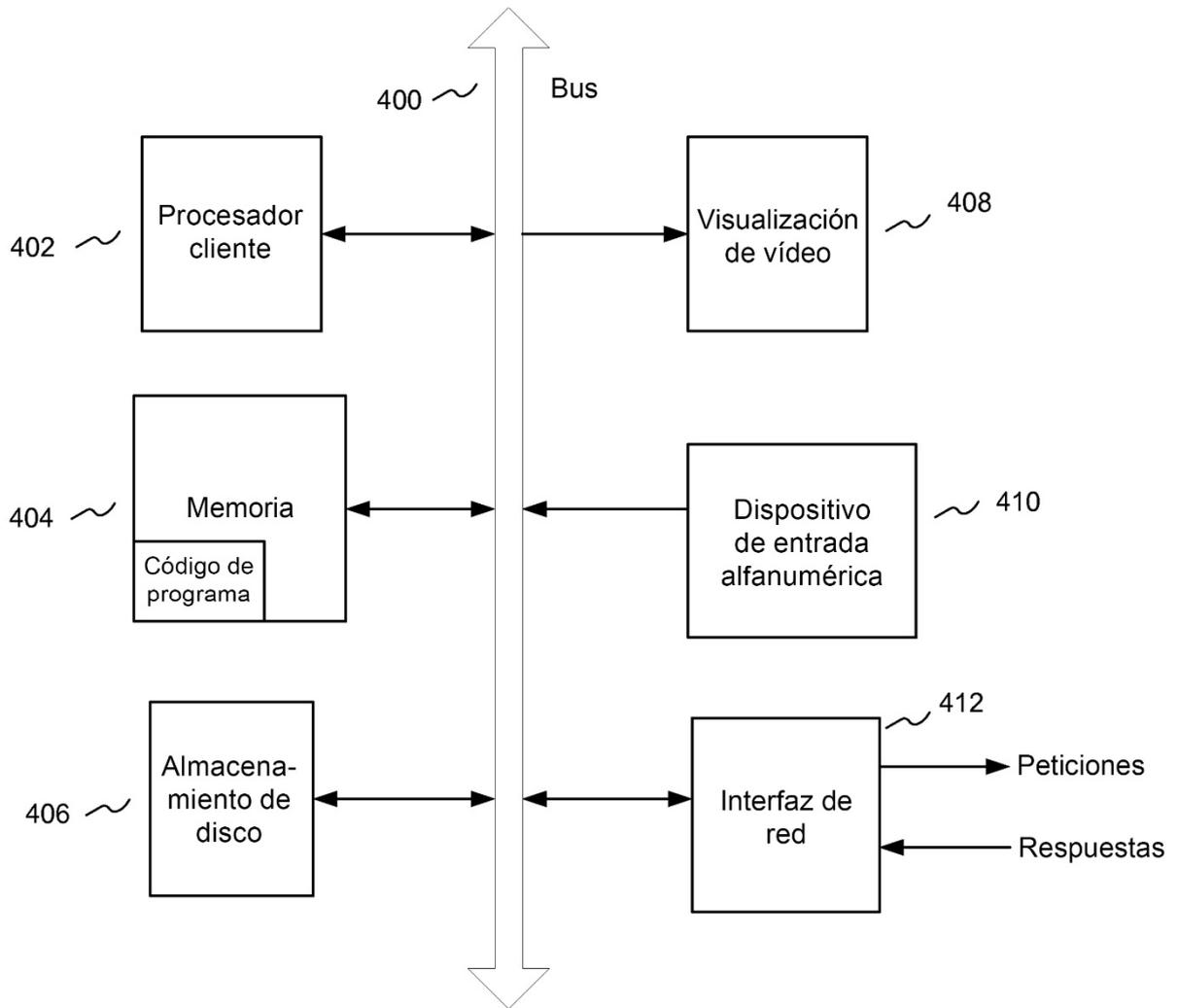


FIG. 4

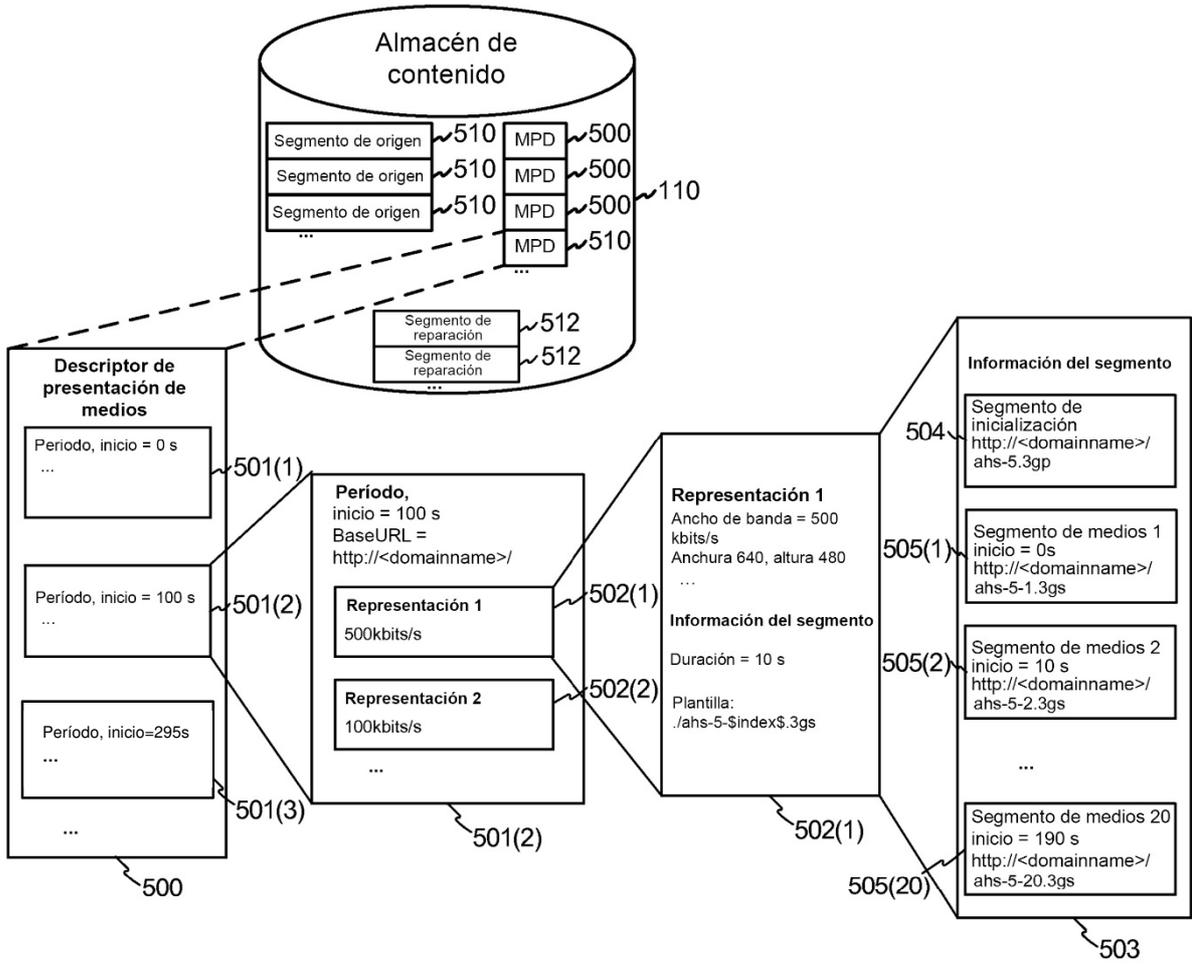


FIG. 5

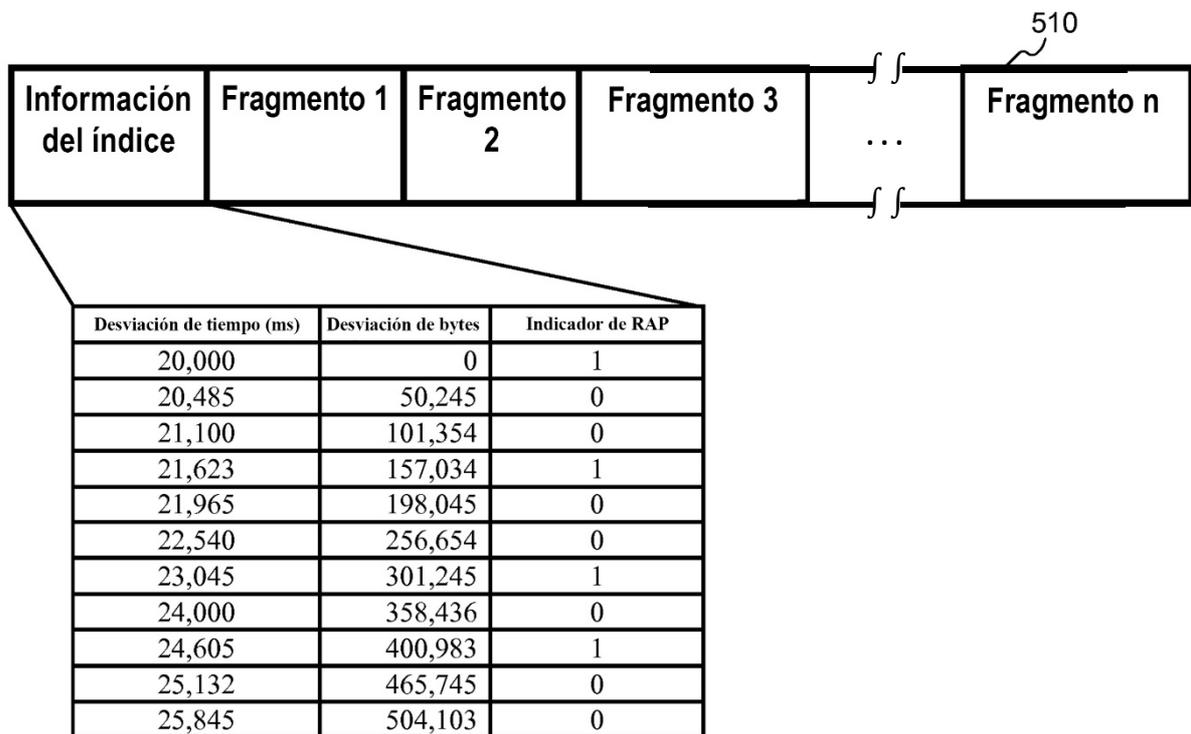


FIG. 6

700  
Índice simple

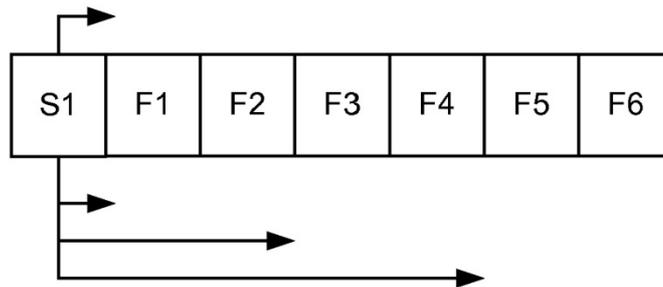


FIG. 7(a)

702  
Índice jerárquico

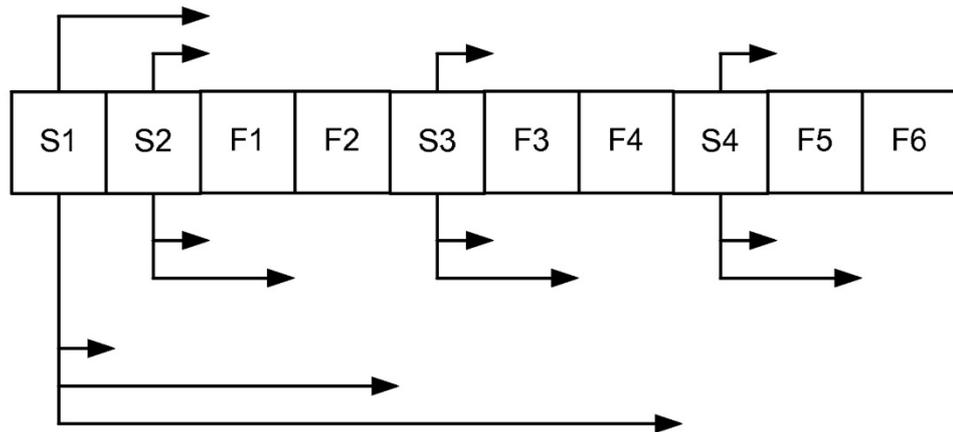
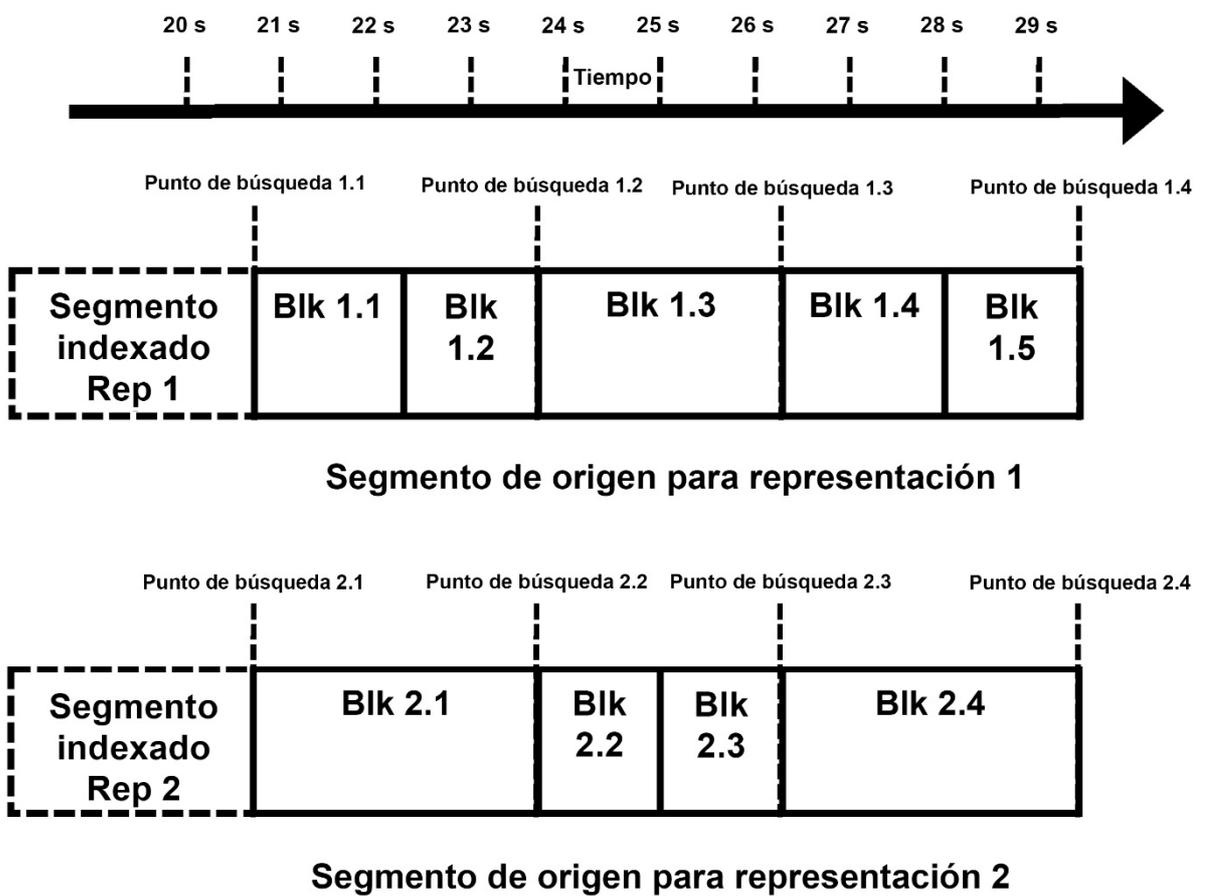


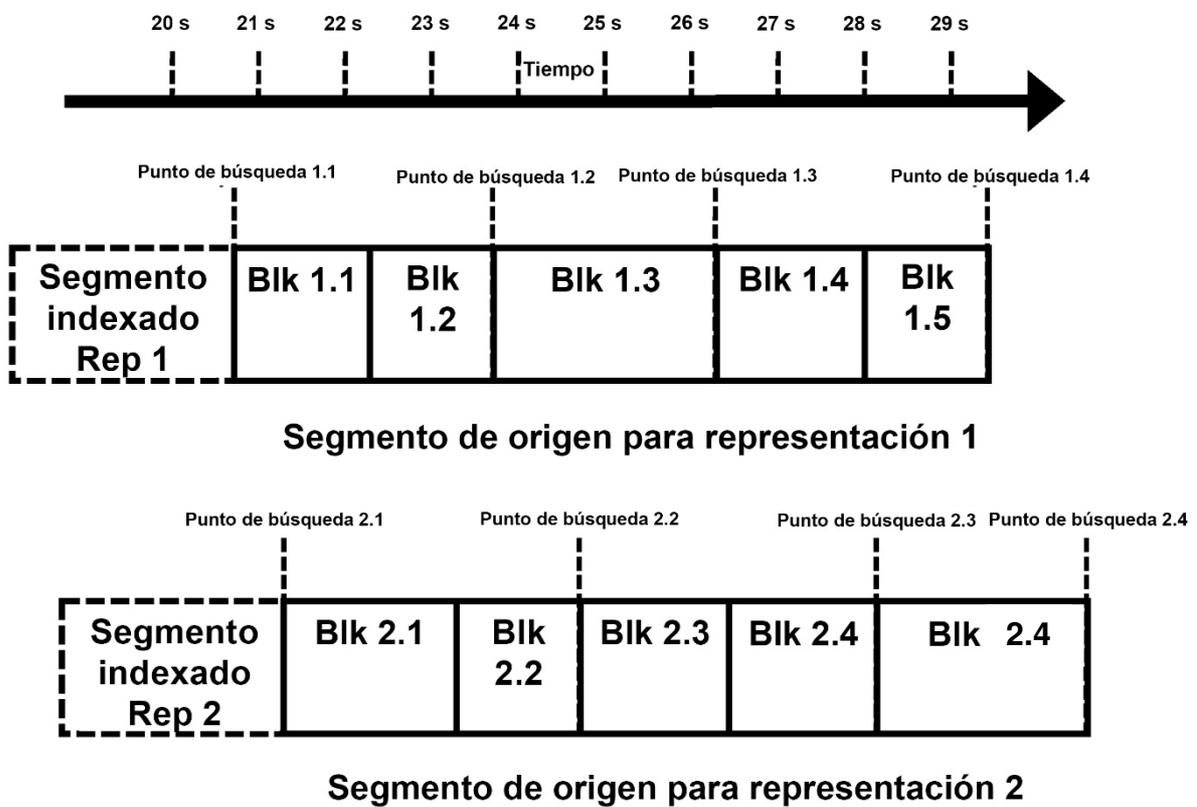
FIG. 7(b)

**Estructura del segmento de origen para representaciones con puntos de búsqueda alineados**



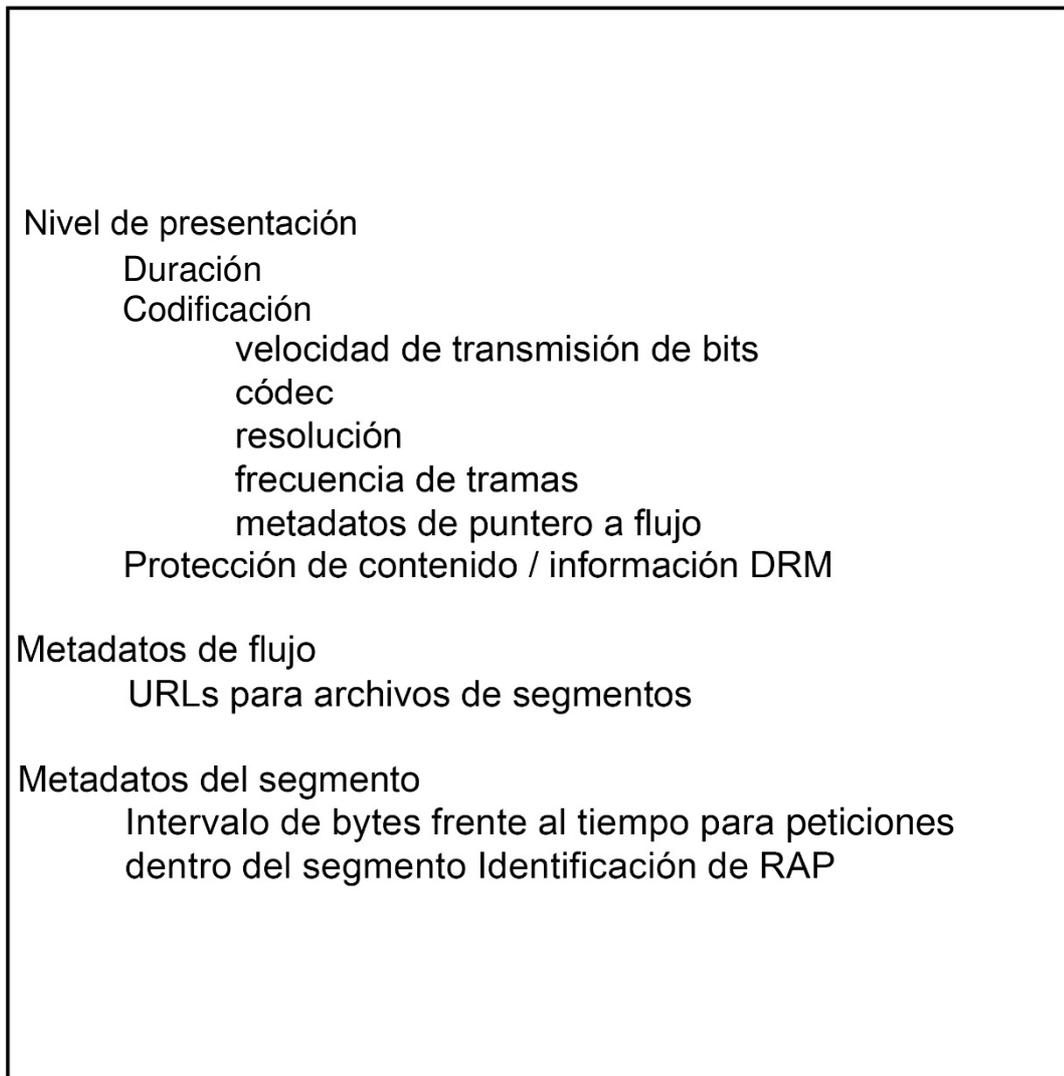
*FIG. 8(a)*

**Estructura del segmento de origen para representaciones con puntos de búsqueda no alineados**

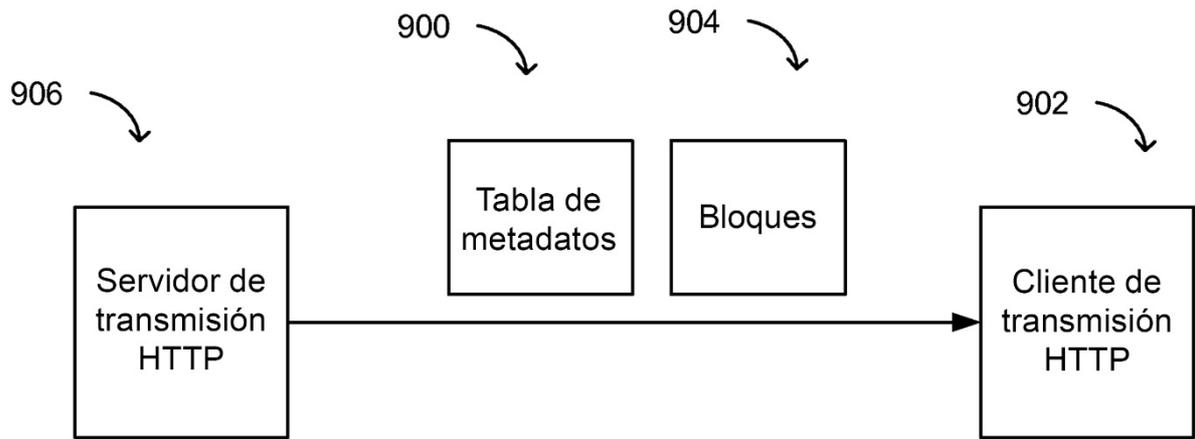


*FIG. 8(b)*

900



*FIG. 9(a)*



*FIG. 9(b)*

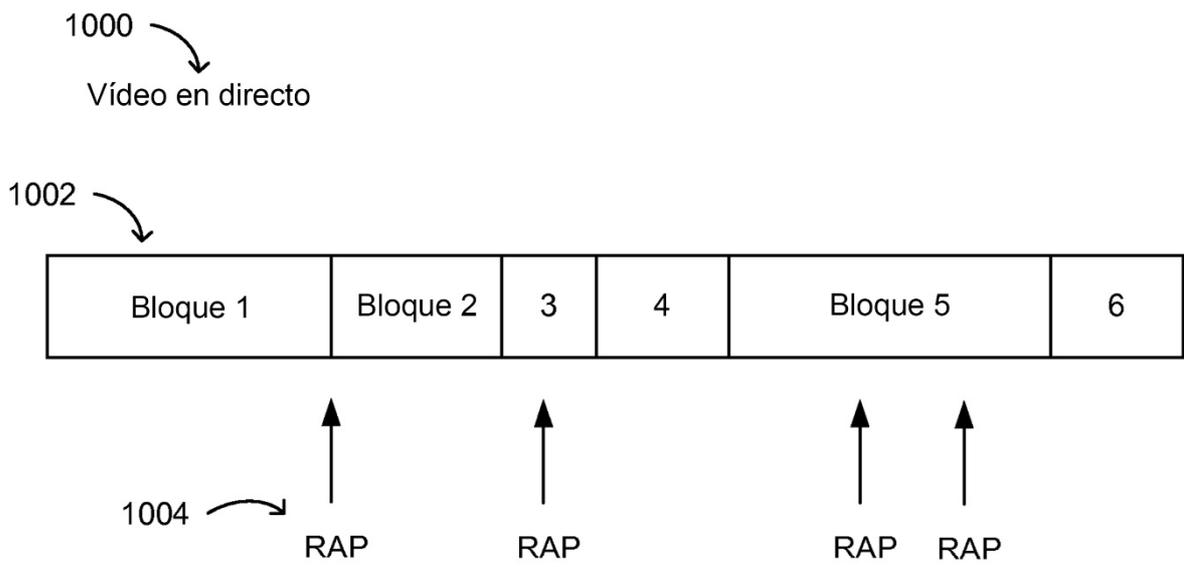
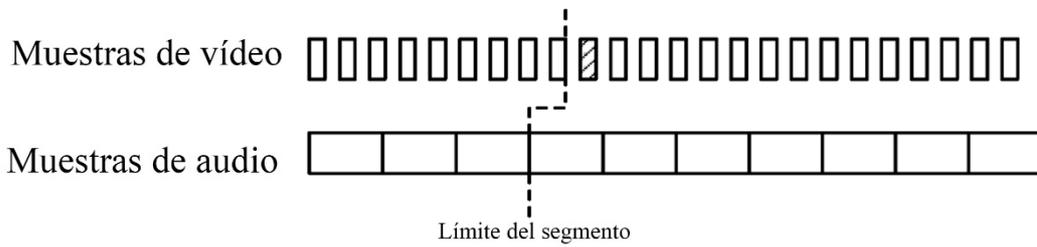
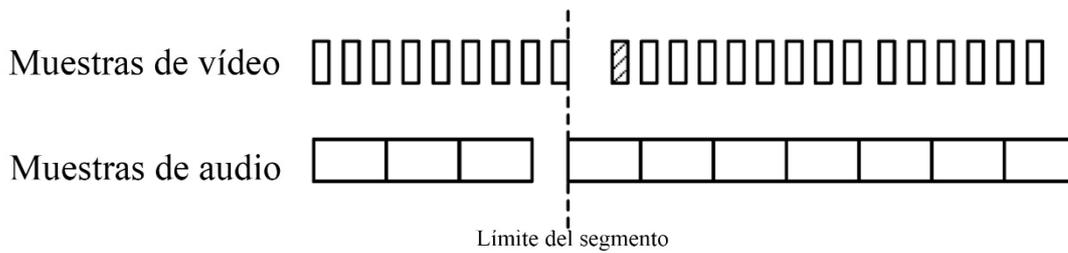


FIG. 10

**Temporización continua entre segmentos**



**Temporización discontinua entre segmentos**



Suposiciones:

- Trama sombreada elegida como punto límite por razones independientes
- El audio nunca debe comenzar más tarde que el vídeo

Notas:

- En el caso continuo, el segundo segmento tiene el *mismo contenido* que el primero
- En el caso discontinuo, el segundo segmento tiene un *contenido diferente* del primero

**FIG. 11**

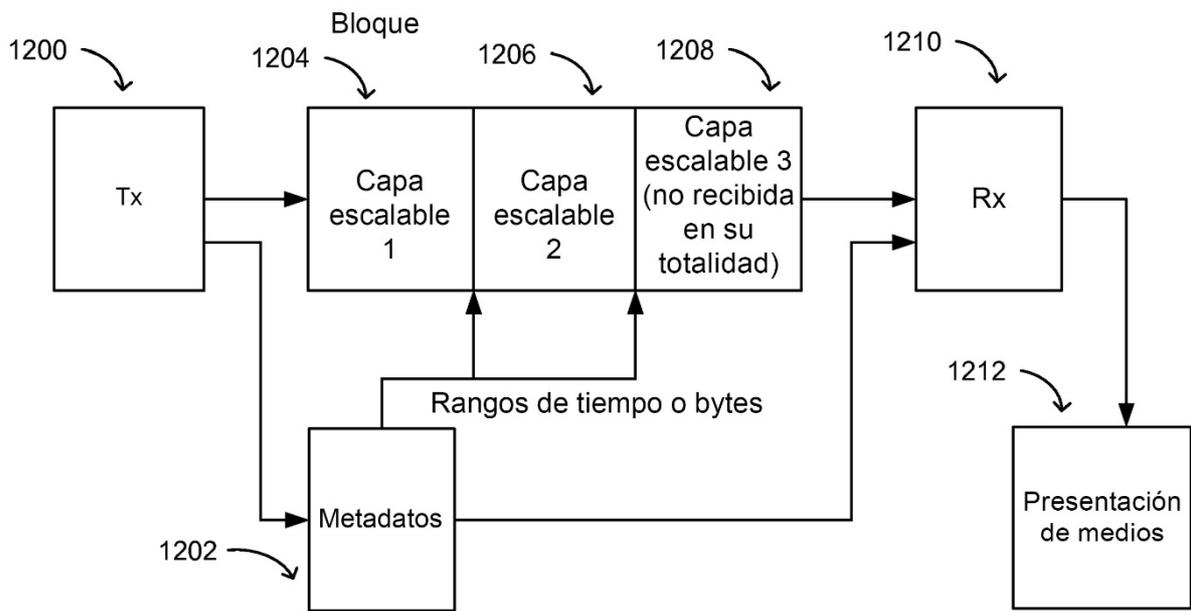


FIG. 12

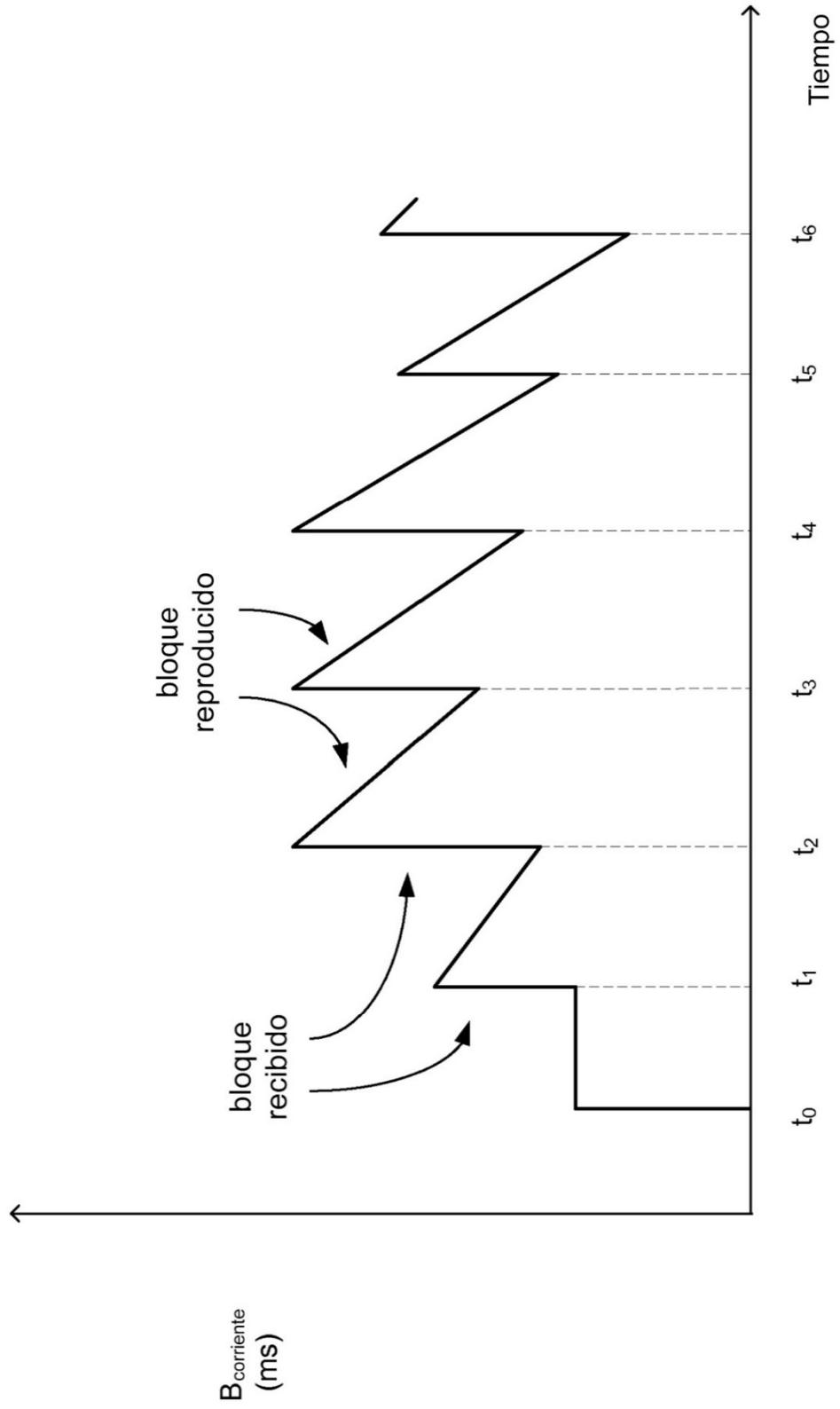


FIG. 13

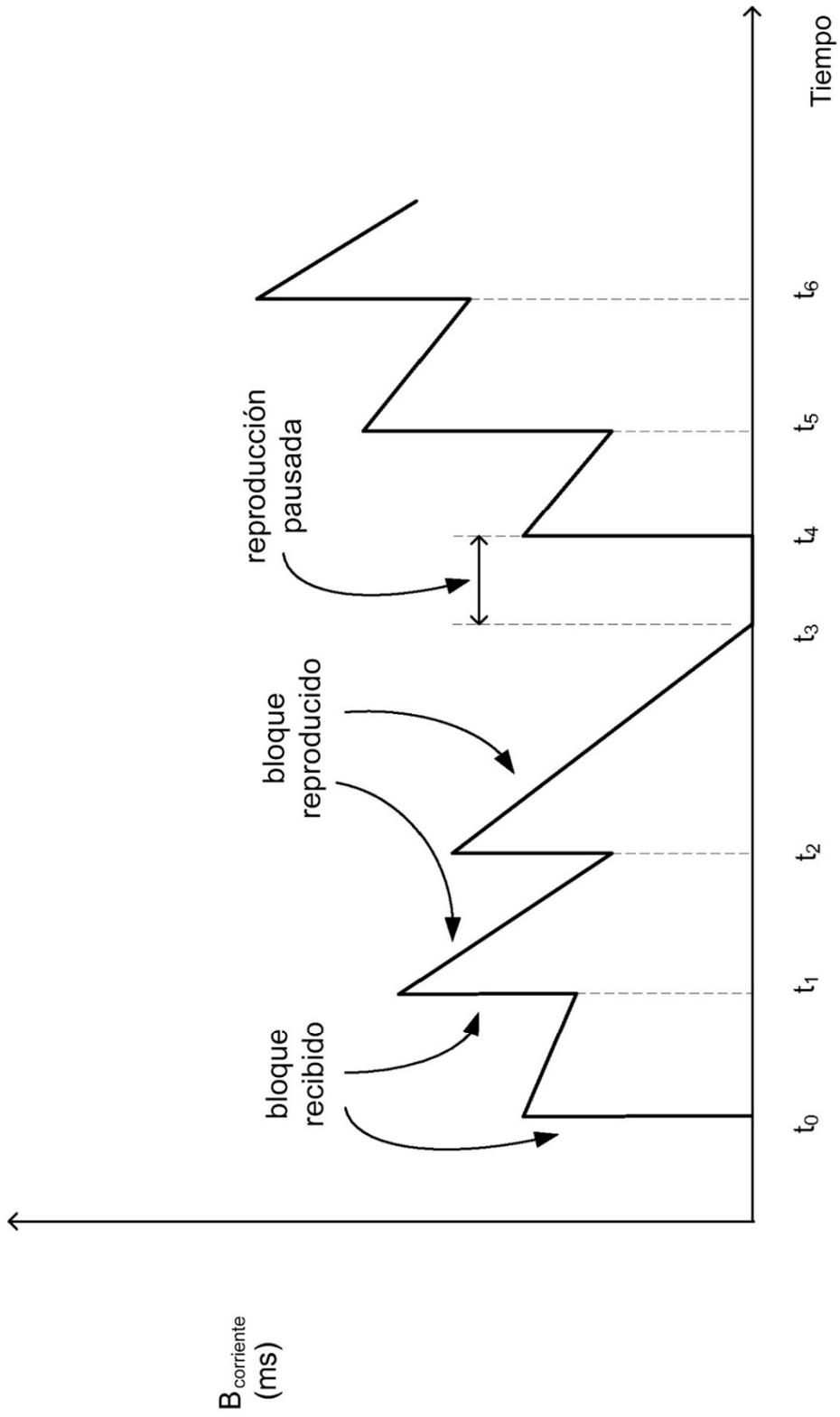


FIG. 14

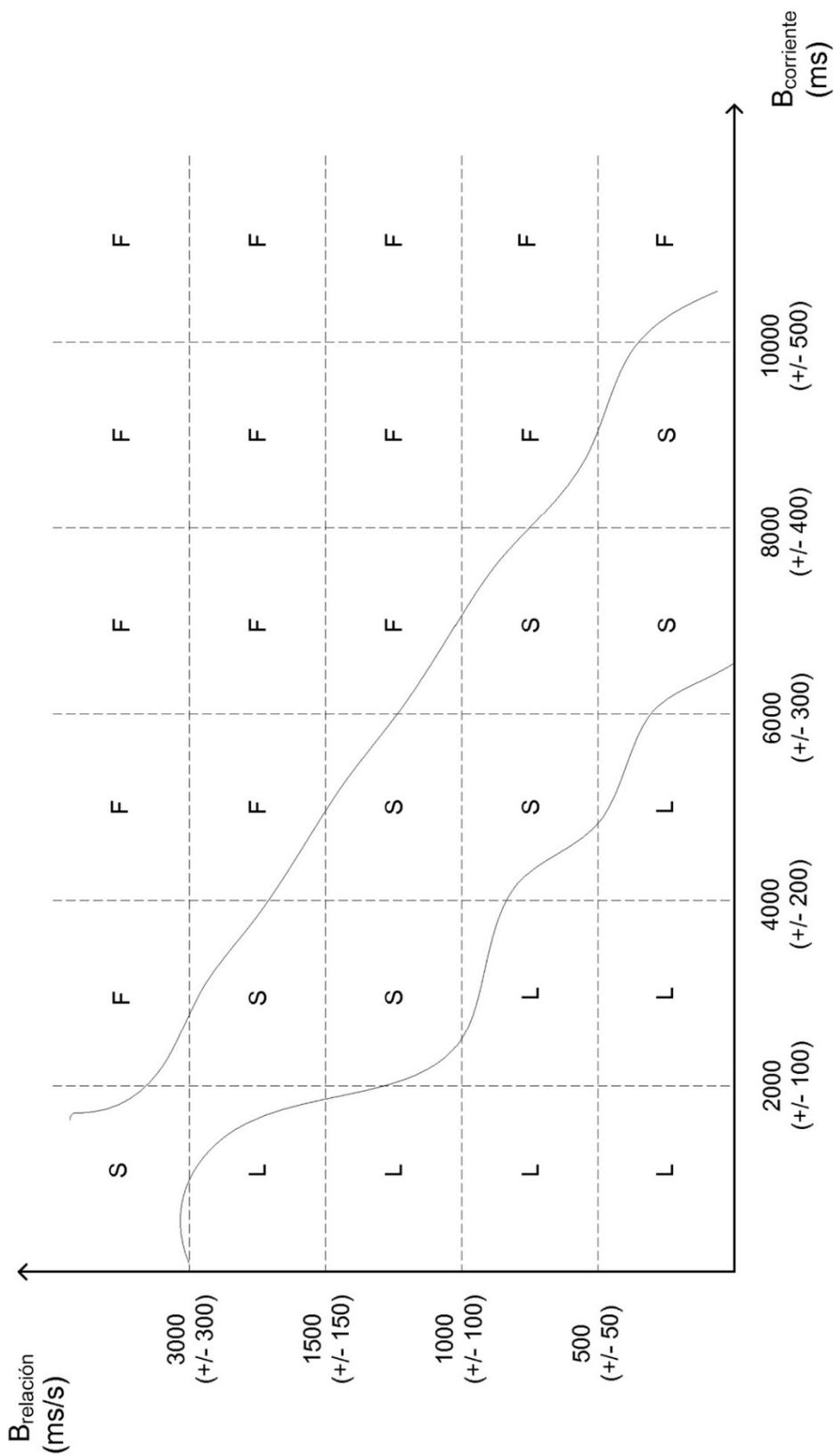


Fig. 15 L = Bajo, S = Estable, F = Completo

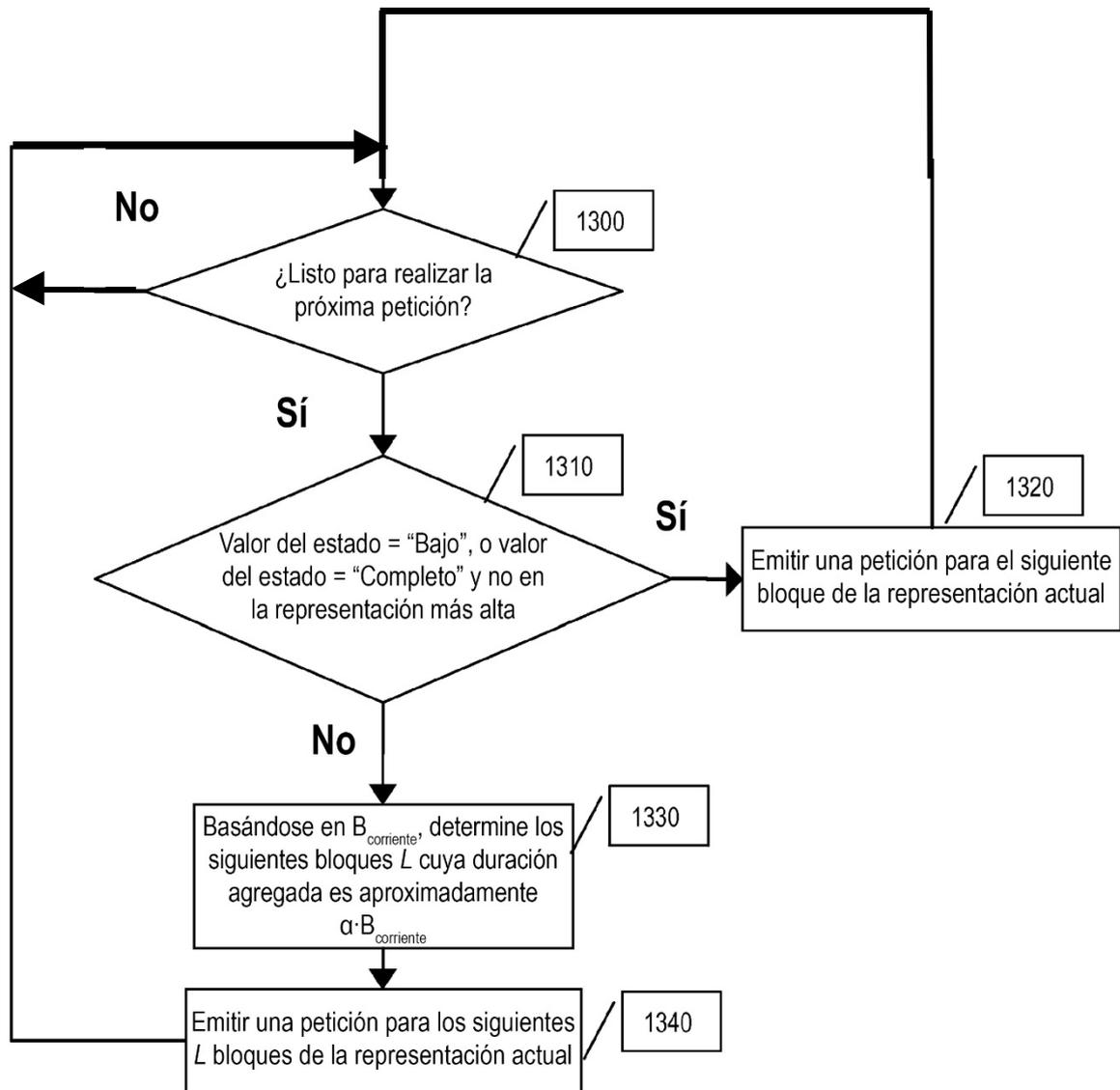


FIG. 16

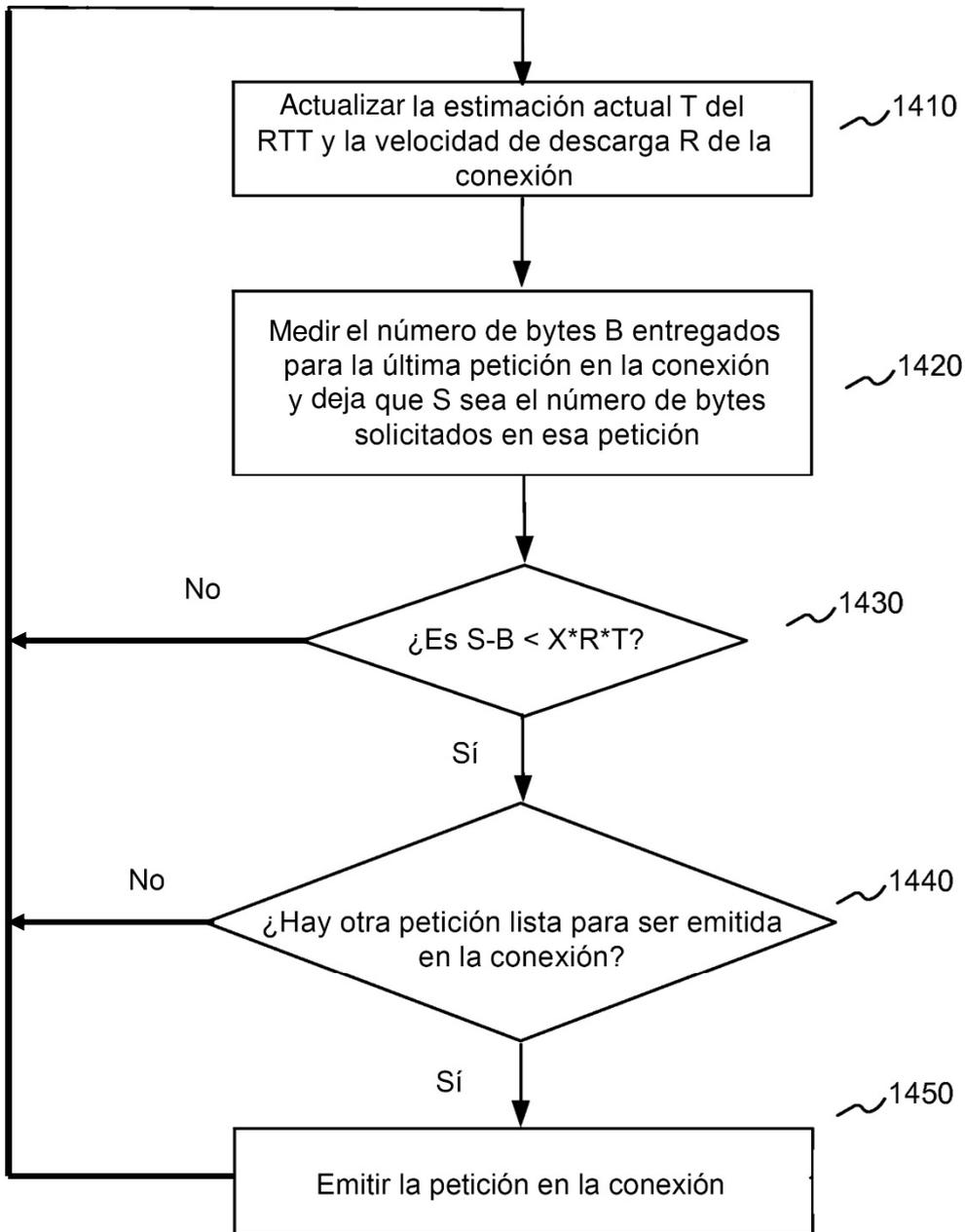


FIG. 17

<b>Lista de peticiones de candidatos</b>	<b>Prioridad</b>	<b>¿Ok para la conexión 1?</b>	<b>¿Ok para la conexión 2?</b>	<b>¿Ok para la conexión 3?</b>
Petición A	0	X		
Petición B	0	X	X	
Petición C	1	X		
Petición D	1			X
Petición E	2			X
Petición F	3		X	X

FIG. 18

<b>Lista de peticiones de candidatos</b>	<b>Prioridad</b>	<b>¿Ok para la conexión 1?</b>	<b>¿Ok para la conexión 2?</b>	<b>¿Ok para la conexión 3?</b>
Petición A	5	X		
Petición G	0			X
Petición C	4	X		
Petición D	1			X
Petición E	2			X
Petición F	3		X	X

FIG. 19

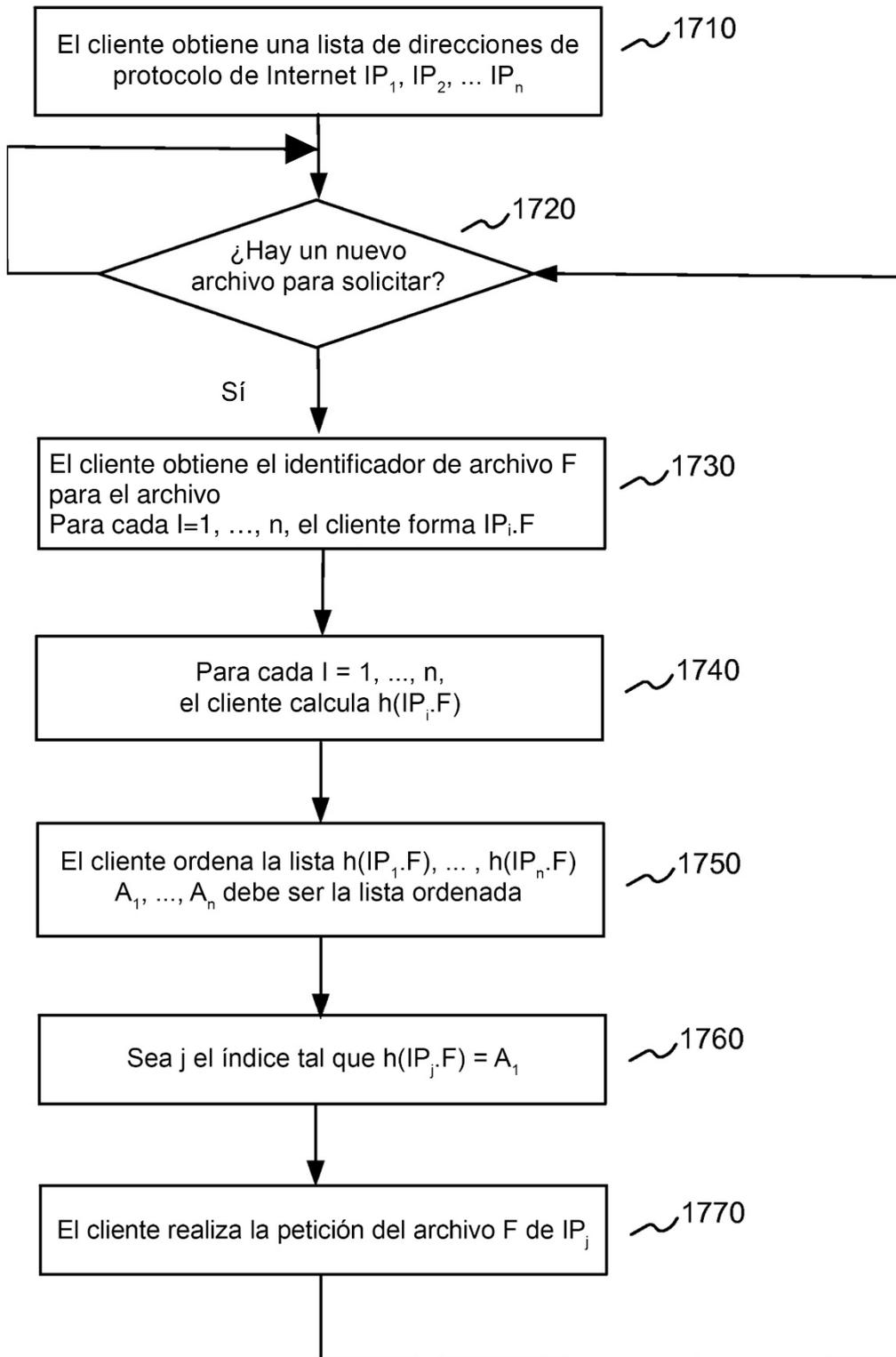


FIG. 20

```

<expression> ::= <literal> |
               <variable> |
               <function> |
               '(' <expression> ')' |
               <expression> <operator> <expression>

<literal> ::= <string> | <number>

<variable> ::= <token>

<function> ::= <token>'(' [ <expression> *(',') <expression> ] ')'

<operator> ::= 1*<opchar>

<token> ::= <tokenchar> *( <tokenchar> | <digit> )

<string> ::= "" * <char> ""

<number> ::= [ '-' ] 1*<digit> [ '.' 1*<digit> ]

<digit> ::= '0-9'

<char> ::= <any ASCII char except ""> | '/' ""

<tokenchar> ::= 'A-Z' | 'a-z' | '_'

<opchar> ::= '-' | '+' | '=' | '/' | ';' | '%' | '*' |

```

FIG. 21

```

unsigned long Hash( const char *p, unsigned long max )
{
    unsigned long hash = 0;
    while( *p != 0 )
        hash = ( hash << 5 ) ^ ( ( hash & 0xf8000000 ) >> 27 ) ^ *p++;
    return hash % max;
}

```

FIG. 22

**Ejemplo 1**

Título:	Asignación de dominio fijo simple
Regla de construcción del identificador de archivo:	<code>printf("http://&lt;domainname&gt;/%s_%3d.dfx", id, seq)</code>
ID de archivo:	SPEED_PAO_S2_E1_5gsyd75
# de secuencia:	22
Resultado:	<code>http://&lt;domainname&gt;/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

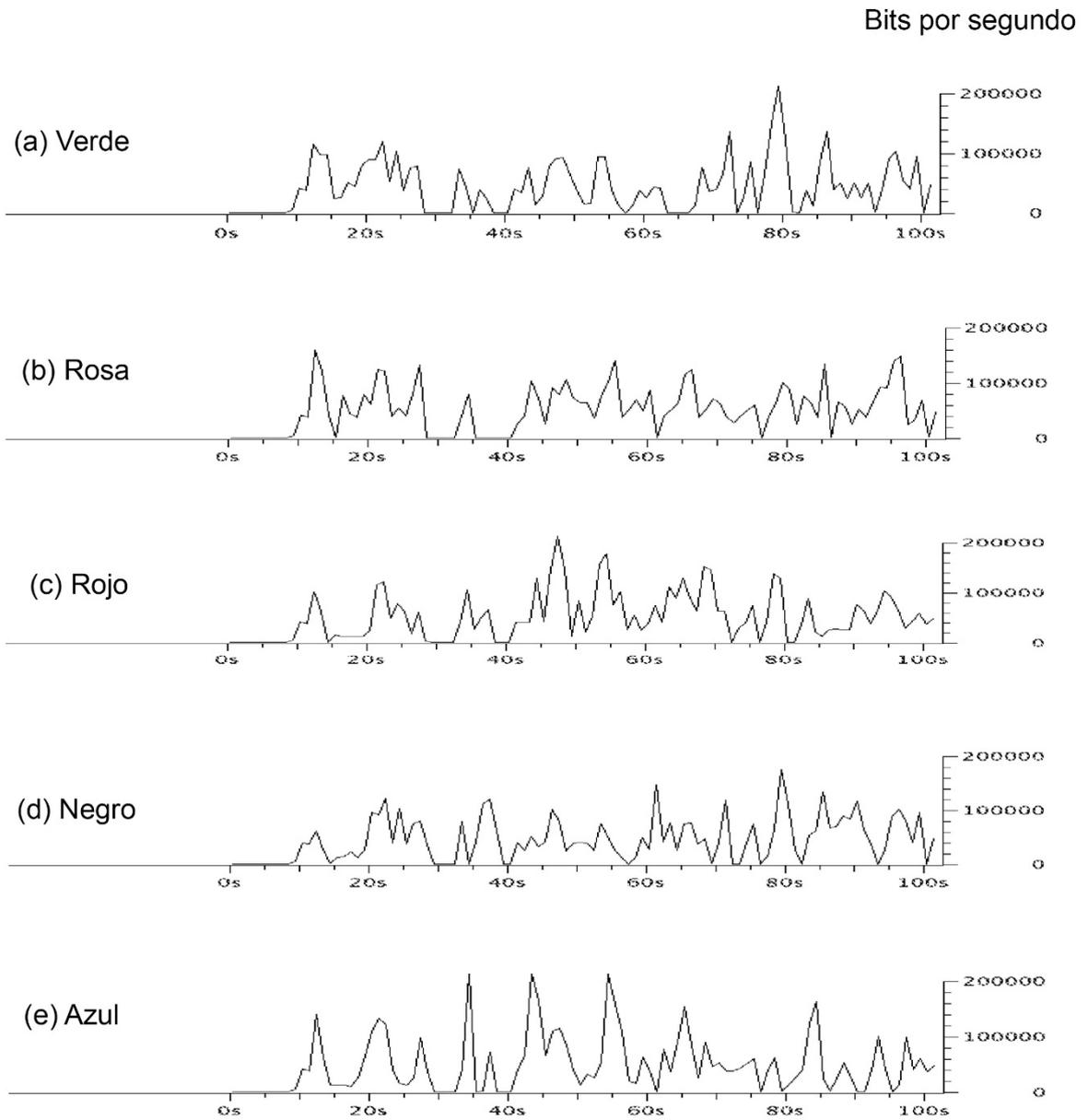
**Ejemplo 2**

Título:	Hash sobre 1000 dominios basados en ID de archivo
Regla de construcción del identificador de archivo:	<code>printf("http://%3d.&lt;domainname&gt;/%s_%3d.dfx", hash(id, 1000), id, seq)</code>
ID de archivo:	SPEED_PAO_S2_E1_5gsyd75
# de secuencia:	22
Resultado:	<code>http://564.&lt;domainname&gt;/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

**Ejemplo 3**

Título:	Hash sobre 10 dominios y 1000 directorios
Regla de construcción del identificador de archivo:	<code>printf("http://%2d.&lt;domainname&gt;/%3d/%s_%3d.dfx", hash(id, 10), hash(id, 1000), id, seq)</code>
ID de archivo:	SPEED_PAO_S2_E1_5gsyd75
# de secuencia:	22
Resultado:	<code>http://4.&lt;domainname&gt;/564/SPEED_PAO_S2_E1_5gsyd75_022.dfx</code>

**FIG. 23**

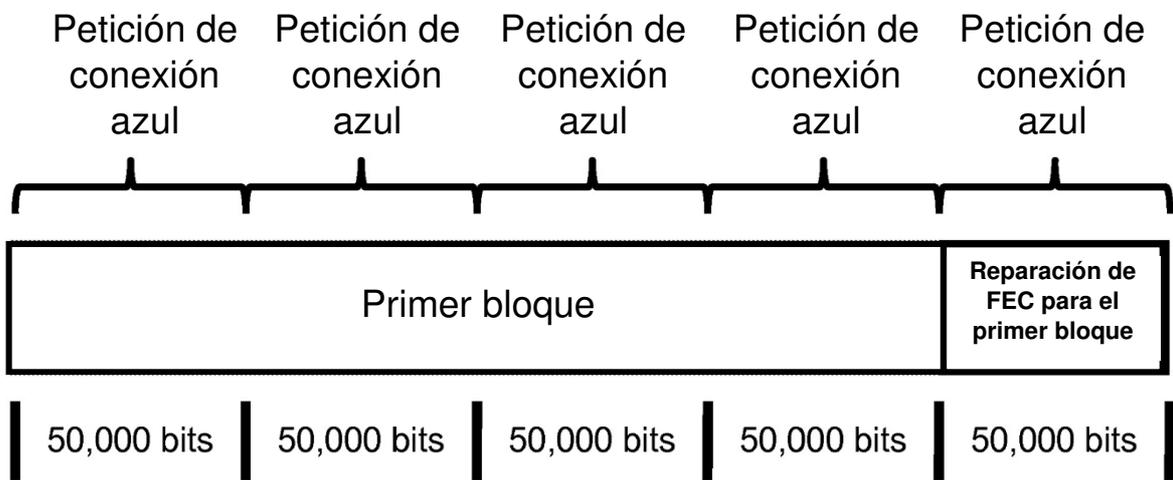


Línea de tiempo de conexiones TCP

**fluctuación típica del ancho de banda de 5 conexiones TCP**

**FIG. 24**

**Múltiples peticiones HTTP para datos de fuente + reparación**



*FIG. 25*

### Ejemplo de tiempo de zapping de canal con y sin FEC

Bits recibidos

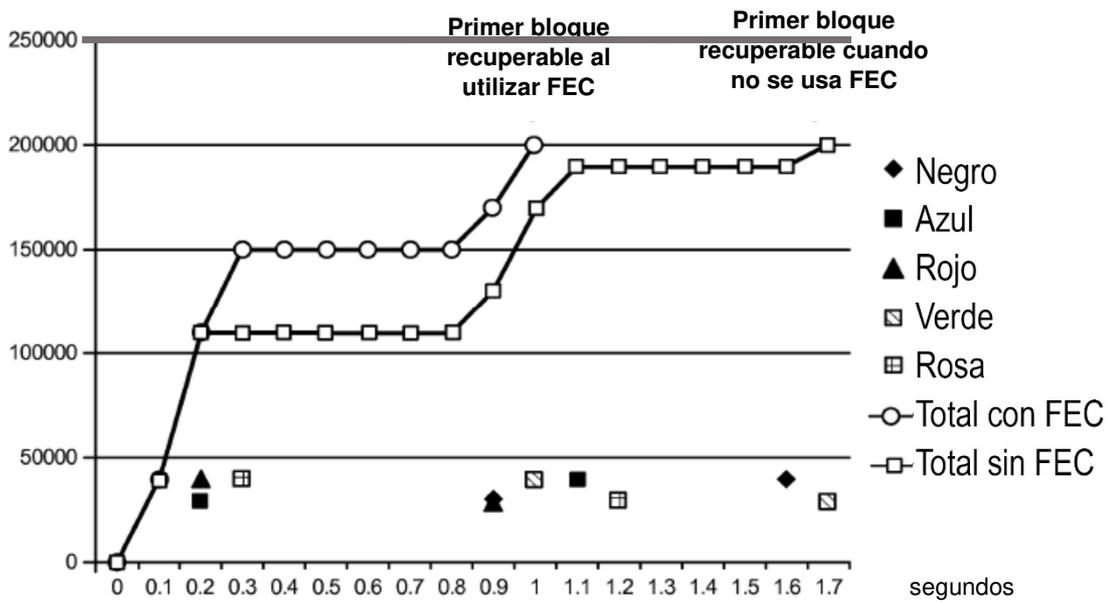


FIG. 26

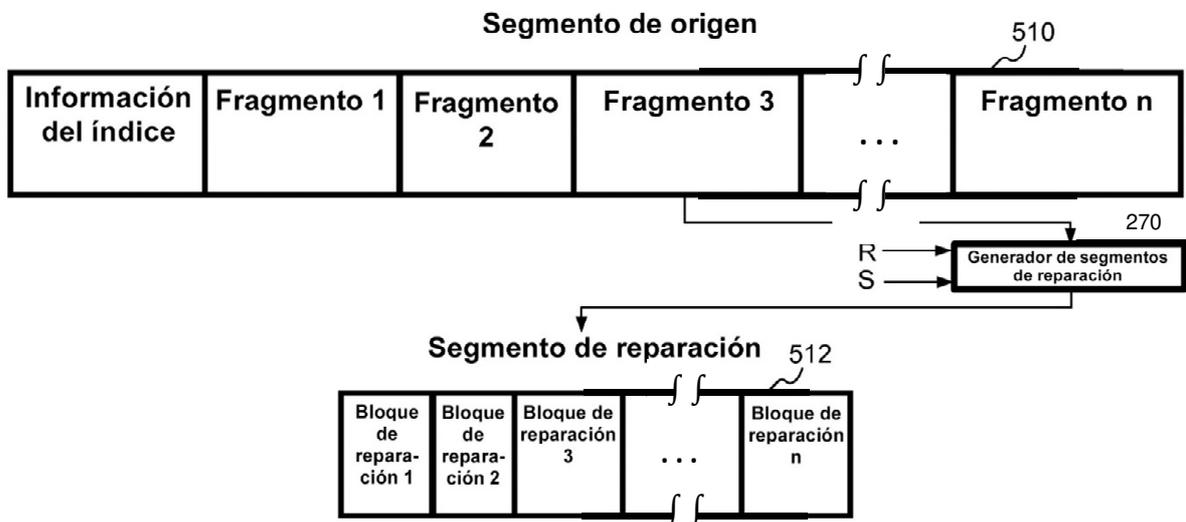


FIG. 27

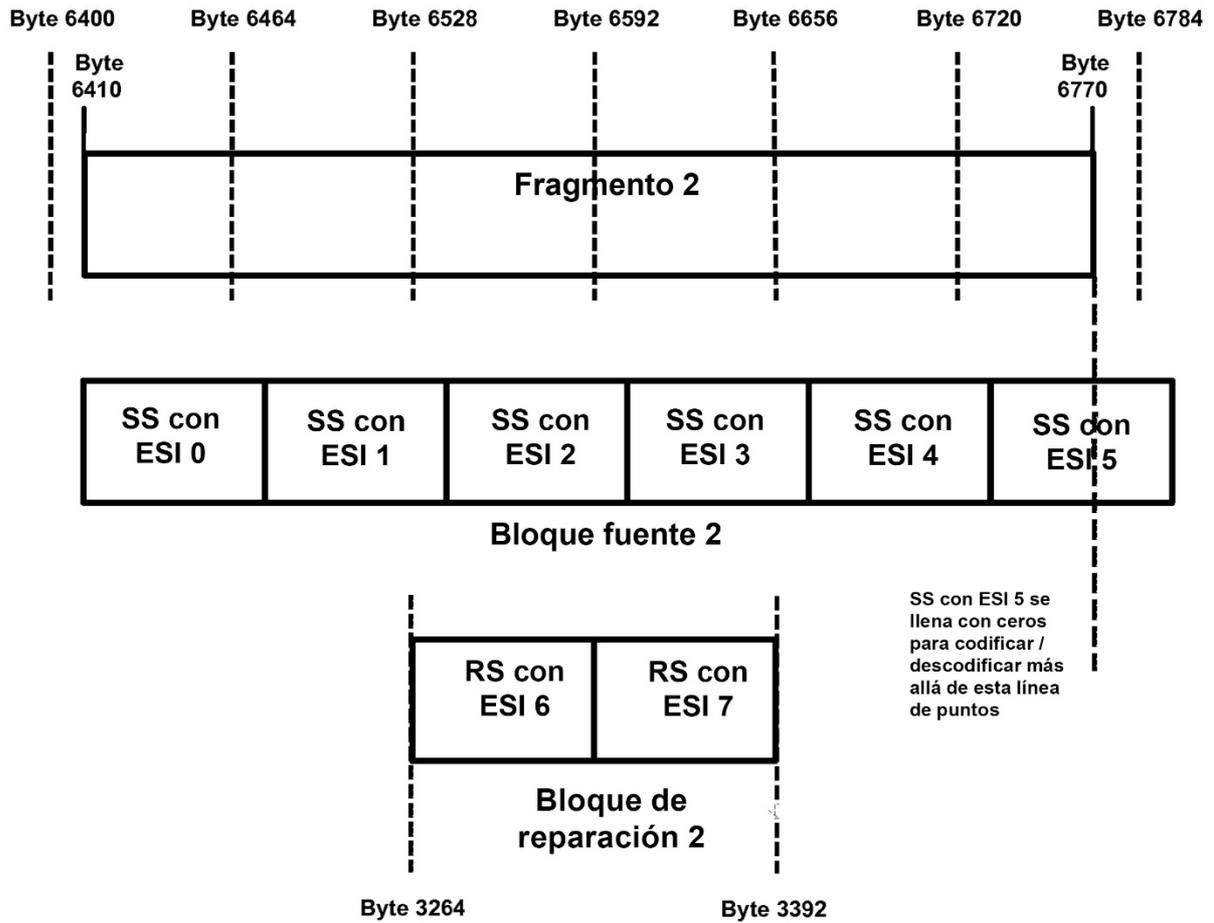


FIG. 28

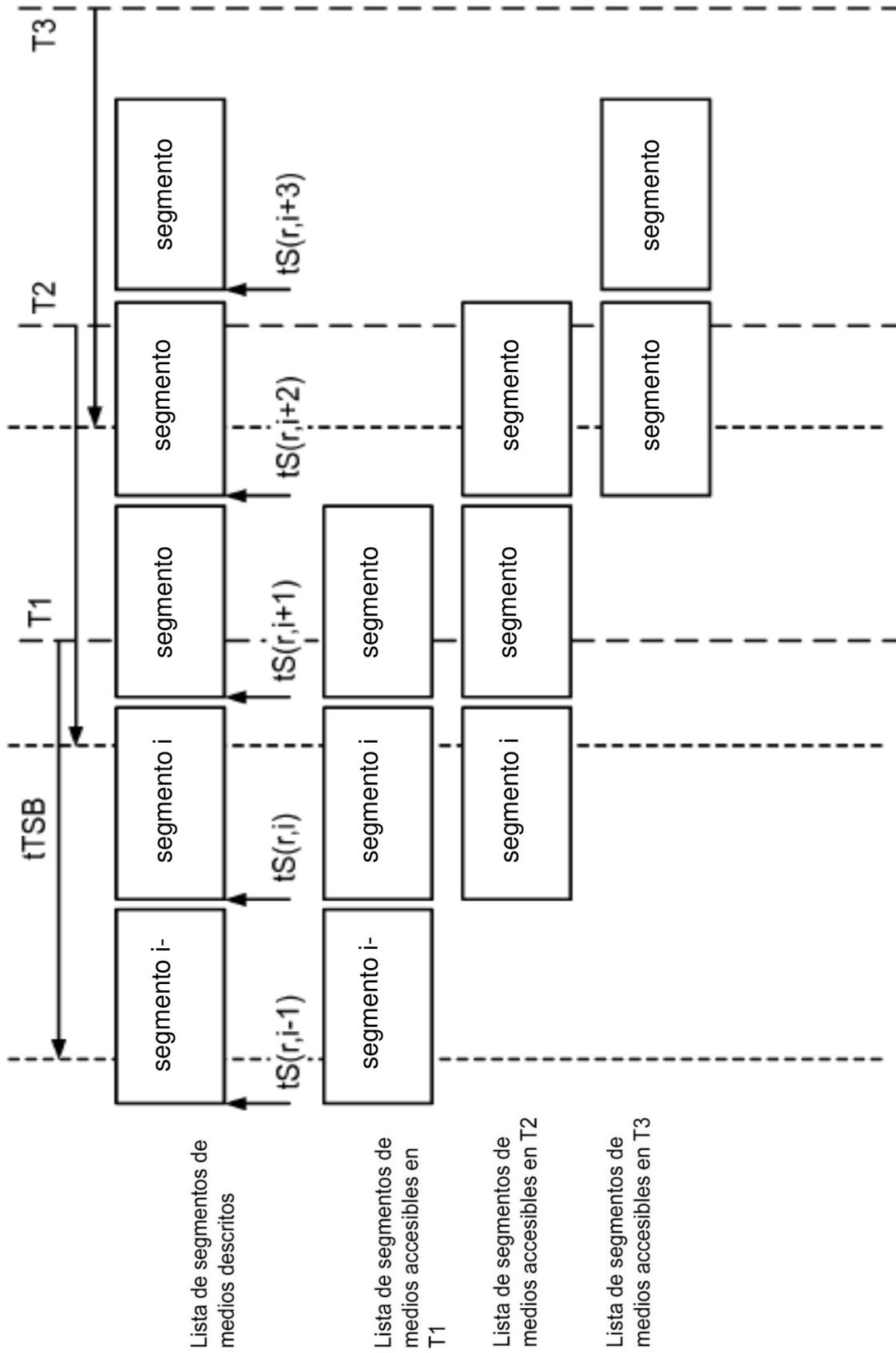


FIG. 29