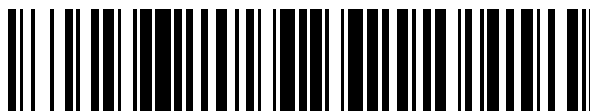


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 694 516**

51 Int. Cl.:

**H04L 29/06** (2006.01)

**H04L 29/08** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **10.06.2011 PCT/US2011/039910**

87 Fecha y número de publicación internacional: **22.12.2011 WO11159558**

96 Fecha de presentación y número de la solicitud europea: **10.06.2011 E 11727599 (0)**

97 Fecha y número de publicación de la concesión europea: **08.08.2018 EP 2580902**

54 Título: **Fichero delta de descripción de presentación de medios para envío por flujo continuo de HTTP**

30 Prioridad:

**14.06.2010 US 815269**

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

**21.12.2018**

73 Titular/es:

**BLACKBERRY LIMITED (100.0%)  
295 Phillip Street  
Waterloo, ON N2L 3W8, CA**

72 Inventor/es:

**FURBECK, DAVID, STUART y  
CHITTURI, SURESH**

74 Agente/Representante:

**CARVAJAL Y URQUIJO, Isabel**

ES 2 694 516 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

Fichero delta de descripción de presentación de medios para envío por flujo continuo de HTTP

### Antecedentes

5 El Proyecto Común de Tecnologías Inalámbricas de la 3ª Generación (3GPP) ha desarrollado una característica conocida como envío por flujo continuo de HTTP, mediante la cual los teléfonos móviles, asistentes digitales personales, ordenadores de bolsillo o portátiles, ordenadores de sobremesa, decodificadores de salón, aparatos de red y dispositivos similares pueden recibir contenido de medios de envío por flujo continuo mediante el protocolo de transferencia de hipertexto (HTTP). Cualquier dispositivo que pueda recibir datos de envío por flujo continuo se denominará en el presente documento como un cliente. El contenido que puede proporcionarse a tales dispositivos cliente mediante HTTP puede incluir vídeo de envío por flujo continuo, audio de envío por flujo continuo y otro contenido multimedia. En algunos casos, el contenido se prepara y a continuación se almacena en un servidor web convencional para envío por flujo continuo posterior mediante HTTP. En otros casos, puede usarse envío por flujo continuo en directo o casi en directo, mediante el cual se coloca contenido en un servidor web en o cerca del momento en el que se crea el contenido. En cualquier caso, los clientes pueden usar tecnología de exploración web convencional y recibir el contenido enviado por flujo continuo en cualquier momento deseado.

15 El documento QUALCOMM INCORPORATED ET AL: "Essential Corrections to 3GPP Adaptive HTTP Streaming", 3GPP DRAFT; vol. SA WG4, n.º Vancouver, Canadá; 20100426 analiza envío por flujo continuo adaptativo de HTTP. De acuerdo con un aspecto de la invención, se proporciona un método como se define en la reivindicación 1 en lo sucesivo. De acuerdo con otro aspecto de la invención, se proporciona un dispositivo cliente como se define en la reivindicación 8 en lo sucesivo. De acuerdo con otro aspecto de la invención, se proporciona un método como se define en la reivindicación 9 en lo sucesivo. De acuerdo con otro aspecto de la invención, se proporciona un servidor como se define en la reivindicación 16 en lo sucesivo. Se exponen realizaciones preferidas en las reivindicaciones dependientes.

### Breve descripción de los dibujos

25 Para un entendimiento más completo de esta divulgación, se hace ahora referencia a la siguiente descripción breve, tomada en relación con los dibujos adjuntos y la descripción detallada, en la que números de referencia similares representan partes similares.

La Figura 1 es una arquitectura de sistema para envío por flujo continuo de HTTP adaptativo.

30 La Figura 2 es un ejemplo de un documento de descripción de presentación de medios de acuerdo con una realización de la divulgación.

La Figura 3 es una arquitectura de sistema para envío por flujo continuo de HTTP adaptativo de acuerdo con una realización de la divulgación,

Las Figuras 4a y 4b ilustran un esquema delta de acuerdo con una realización de la divulgación.

35 La Figura 5 es un diagrama de flujo que ilustra un método para obtener información de descripción de presentación de medios de acuerdo con una realización de la divulgación.

La Figura 6 ilustra un procesador y componentes relacionados adecuados para implementar las varias realizaciones de la presente divulgación.

### Descripción detallada

40 Debería entenderse desde el principio que aunque se proporcionan a continuación implementaciones ilustrativas de una o más realizaciones de la presente divulgación, los sistemas y/o métodos desvelados pueden implementarse usando cualquier número de técnicas, ya sean actualmente conocidas o en existencia. La divulgación no debería limitarse de ninguna manera a las implementaciones ilustrativas, dibujos y técnicas ilustradas a continuación, incluyendo los diseños ejemplares e implementaciones ilustrados y descritos en el presente documento, sino que pueden modificarse dentro del alcance de las reivindicaciones adjuntas.

45 La Figura 1 ilustra una arquitectura de sistema típica para envío por flujo continuo de HTTP adaptativo. En una fase de preparación de contenido 110, se prepara una presentación de medios para envío por flujo continuo de HTTP. El contenido se almacena a continuación en un servidor de envío por flujo continuo de HTTP 120 y/o posiblemente en una caché de HTTP 130. Un cliente de envío por flujo continuo de HTTP 140 puede usar una solicitud HTTP GET o

un mensaje similar para descargar la presentación de medios desde el servidor 120 o la caché 130.

La presentación de medios puede describirse en un documento de lenguaje de marcas extensible (XML), que en las especificaciones del 3GPP se denomina una descripción de presentación de medios (MPD). La MPD contiene metadatos que informan al cliente de los formatos en los que se codifica el contenido de medios, tal como las tasas de bits, códecs, resoluciones de pantalla e idiomas. Cada combinación diferente de tales parámetros puede denominarse como una representación del contenido, y múltiples diferentes representaciones pueden representar el mismo contenido. Esto permite que el cliente elija una representación particular basándose posiblemente en su resolución de pantalla, el ancho de banda de canal actual, las condiciones de recepción de canal actuales, la preferencia de idioma del usuario, y otros parámetros.

En casos donde no se usa el envío por flujo continuo en directo, puede proporcionarse una MPD que describe una presentación completa al cliente, y el cliente puede usar los metadatos en la MPD a través de toda la presentación. En casos de envío por flujo continuo, sin embargo, puede no ser posible proporcionar los metadatos acerca del flujo de medios completo antes del comienzo del envío por flujo continuo, puesto que los metadatos pueden no haberse conocido aún. También, los parámetros relacionados con una sesión de envío por flujo continuo pueden cambiar durante el curso de la sesión. Por ejemplo, el cliente puede moverse en un área con recepción pobre, y la tasa de datos puede ralentizarse. En un caso de este tipo, el cliente puede necesitar conmutar a una representación con una tasa de bits inferior. En otro ejemplo, el dispositivo de cliente puede elegir conmutar la visualización del contenido enviado por flujo continuo desde modo vertical a horizontal, caso en el que puede requerirse una representación diferente.

Por estas razones, puede no ser posible proporcionar al cliente con una MPD que describe una presentación completa. En tales casos, una presentación de medios puede descomponerse en segmentos, y una porción de la MPD puede describir cada segmento. Es decir, con envío por flujo continuo de HTTP, los medios pueden descargarse un segmento cada vez de modo que la reproducción de contenido en directo quede muy lejos de la codificación en directo y de modo que el cliente pueda conmutar a un contenido diferente que se codifica de manera adaptativa de acuerdo con condiciones de canal u otros factores, como se ha descrito anteriormente.

Los segmentos son direccionables con la información en el elemento de XML "Url" en la MPD. El elemento "Url" contiene el URL del segmento y puede restringirse por un atributo "rango" que podría usarse con el encabezamiento Rangos de una solicitud HTTP GET. Un nuevo segmento de contenido puede añadirse a intervalos regulares, tal como cada diez segundos, y puede añadirse nueva información correspondiente a la MPD a medida que se añade cada nuevo segmento de contenido. Por lo tanto, el tamaño de la MPD puede crecer a medida que progresa el envío por flujo continuo del contenido.

Un ejemplo MPD se ilustra en la Figura 2 y puede denominarse como "exampleMPD.xml". Múltiples porciones 210 de la MPD de ejemplo cada una describen un segmento de la presentación de medios. En interés de ahorrar espacio en la figura, únicamente se muestran dos de tales porciones 210, pero en una MPD real pueden estar presentes un número mayor de tales porciones. En este ejemplo, puede suponerse que hay tres periodos de diez minutos de duración, para un total de 30 minutos de contenido. Cada periodo contiene tres diferentes codificaciones o representaciones, y cada representación contiene segmentos de diez segundos de duración. Por lo tanto, hay 60 segmentos para cada representación en un periodo y un total de 540 diferentes segmentos. El tamaño de fichero de exampleMPD.xml es de 52 kilobytes. Para una presentación de medios en directo, la MPD en el servidor se actualiza cada vez que se añade un nuevo segmento, que para el ejemplo es cada diez segundos.

En este ejemplo, se supone que únicamente se proporcionan tres diferentes codificaciones o representaciones de contenido en un periodo. En otros casos, un servidor podría contener un número mayor de diferentes resoluciones, múltiples diferentes tasas de bits, múltiples idiomas, etc. Por lo que en lugar de las tres codificaciones diferentes del ejemplo, podría estar disponible un número mucho mayor de representaciones. También, en la MPD de ejemplo, únicamente se describen 30 minutos de contenido. Otra MPD puede describir, por ejemplo, un evento deportivo que es de tres horas o mayor de duración.

En un segundo ejemplo, puede suponerse que se describen 15 representaciones y 2,5 horas de contenido en la MPD. En comparación con el primer ejemplo de tres representaciones y 30 minutos de contenido, el tamaño de la MPD almacenada en el servidor se ha multiplicado por 25 (es decir, cinco veces como tantas representaciones multiplicadas por cinco veces la duración del contenido). Por lo tanto, en el segundo ejemplo, se necesitaría una MPD de aproximadamente 1,3 megabytes de datos (25 veces 52 kilobytes) para almacenarse en el servidor y proporcionarse y/u obtenerse cada diez segundos. Proporcionar y/u obtener un fichero grande de este tipo de manera tan frecuente a tal vez cientos de clientes podría consumir grandes cantidades de ancho de banda.

Una posible solución a este gran consumo de ancho de banda podría ser comprimir la MPD con una aplicación de compresión. Sin embargo, en el caso de envío por flujo continuo en directo, el cliente típicamente descarga la MPD cada diez segundos o así, para hallar la localización del último segmento codificado, y la MPD crece con cada nuevo

segmento que se añade al contenido enviado por flujo continuo. La compresión de la MPD no resuelve de manera adecuada el problema de una MPD creciente. Además, el coste de recurso de compresión y descompresión de la MPD para cada segmento puede ser significativo.

5 La mayoría de la información en la MPD permanece invariable cuando se actualiza la MPD. Por ejemplo, cada vez que se añade un nuevo segmento en un periodo, únicamente se añade el elemento de "Url" a cada representación. Por ejemplo, en la MPD de la Figura 2, hay hasta 540 elementos "Url" diferentes, y cada diez segundos únicamente se añaden tres. La otra información en la MPD típicamente permanece igual.

10 En una realización, el cliente descarga únicamente la información que ha cambiado realmente desde la última descarga de información de MPD. Esto se ilustra en la Figura 3, donde el servidor 120 incluye una MPD 310 que describe una presentación de medios disponible desde el servidor 120. El cliente 140 también incluye una MPD 320, y la MPD del cliente 320 debería estar sincronizada con la MPD del servidor 310 de modo que el cliente 140 tiene información actualizada acerca de la presentación de medios. Se crea un fichero delta 330 (tal vez por el proveedor de contenido) que incluye la información que ha cambiado desde la última actualización de la MPD. El cliente 140 puede descargar el fichero delta 330 cada vez que se añade un nuevo segmento a la presentación en lugar de  
15 descargar toda la MPD cada vez que se añade un nuevo segmento. Es decir, una vez que el cliente 140 ha descargado la MPD 310 completa, el cliente 140 puede simplemente descargar el fichero delta 330 cada vez que se actualiza el fichero delta 330.

20 En una realización, el servidor 120 es un servidor web convencional que cumple con HTTP 1.1 o superior. Es decir, el servidor 120 no tiene capacidades tales como la funcionalidad del Protocolo de Acceso de Configuración de XML (XCAP) o la funcionalidad del Protocolo de Iniciación de Sesión (SIP).

El Lenguaje de Ruta de XML (XPath) es un lenguaje definido por el Consorcio World Wide Web (W3C) para definir partes de un documento de XML. El RFC 5261 describe tipos de esquemas de elemento para "parches" o cambios a un documento de XML que usa selectores de XPath. Los "parches" definidos en el RFC 5261 son <añadir> (<add>), <25 sustituir> (<replace>), y <eliminar> (<remove>). Estos pueden usarse para describir cambios a elementos o atributos. Estos tipos de esquemas de elemento pueden usarse en un esquema de XML que describe "ficheros delta" de XML (ficheros de XML que describen cambios a un documento de XML). En una realización, es por lo tanto posible describir al cliente únicamente lo que ha cambiado en la última actualización a la MPD. Los clientes que están simplemente sintonizando podrían aún descargar la MPD completa. Sin embargo, otros clientes podrían descargar únicamente un fichero delta para hallar lo que se ha cambiado después de la última actualización. La misma MPD podría contener  
30 una duración de "actualización mínima" que indica la frecuencia con la que la MPD puede posiblemente actualizarse. Por lo que si la MPD se actualiza cada 10 segundos, el cliente descargaría la MPD completa cuando se sintoniza en primer lugar, y posteriormente podría simplemente descargar el fichero delta. Es posible crear varios tipos de ficheros delta. En otras realizaciones, los lenguajes distintos de XPath podrían usarse para identificar un punto en la MPD.

35 Lo siguiente es un ejemplo de una expresión XPath que indica el elemento al que se anexa el elemento de URL en la representación del periodo 3 con ancho de banda de 239K:  
//Period[3]/\*[@bandwidth='239000']/SegmentInfo]

Por lo que después de que se añade el elemento de "Url" para el último segmento para esta MPD, esto podría describirse, usando las operaciones de parche de RFC 5261 por ejemplo, con una sintaxis tal como:  
40 <add sel="//Period[3]/Representation[@bandwidth='239000']/SegmentInfo"><Url sourceURL="p3rep1.3 gp" range="17642842-17943394"/></add>

En este ejemplo, el "delta" consiste en tres partes. El hecho de que el cambio a la MPD consiste en algo añadido que se indica por el elemento <add>. El atributo "sel" indica dónde se ha de añadir el elemento. En otras palabras, el valor del atributo "sel" es una expresión XPath que indica dónde se ha de añadir en la MPD el elemento. En este caso, se añade como el último nodo hijo del SegmentInfo. El elemento que se añadió es el texto dentro del elemento <add>.  
45 Por lo tanto, como un ejemplo, los cambios realizados a la MPD podrían describirse con las siguientes tres expresiones delta:

```
AppendElement
//Period[3]/Representation[@bandwidth='239000']/SegmentInfo/Url[last()]
<Url sourceURL="p3rep1.3 gp" range="17642842-17943394"/>
AppendElement
//Period[3]/Representation[@bandwidth='478000']/SegmentInfo/Url[last()]
<Url sourceURL="p3rep2. 3 gp" range="35284728-35885833"/>
AppendElement
//Period[3]/Representation[@bandwidth='892000']/SegmentInfo/Url[last()]
55 <Url sourceURL="p3rep3. 3 gp" range="65844317-66966044"/>
```

Como alternativa, como otro ejemplo, los cambios realizados a la MPD podrían describirse con el siguiente lenguaje de XML:

```

5   <add sel="//Period[3]/*[@bandwidth='239000']SegmentInfo">
    <Url sourceURL="p3rep1. 3 gp" range="17642842-17943394"/></add>
    <add sel="//Period[3]/*[@bandwidth='478000']SegmentInfo">
    <Url sourceURL="p3rep2. 3 gp" range="35284728-35885833"/></add>
    <add sel="//Period[3]/*[@bandwidth='892000']SegmentInfo">
    <Url sourceURL="p3rep3, 3 gp" range="65844317-66966044"/></add>

```

10 En este caso, el atributo "sel" contiene la expresión XPath y el texto dentro del elemento <add> contiene el elemento añadido a la MPD.

15 El segundo ejemplo anterior tiene un tamaño total de 362 bytes. Por lo que en lugar de que cada cliente tenga que descargar o recibir algo en el orden de 52 kilobytes cada diez segundos como en el ejemplo de la Figura 2, ahora cada cliente únicamente tiene que descargar o recibir 362 bytes. Esto es menos del 1/100 partes del tamaño y por lo tanto es mucho más eficaz en ancho de banda. Esto podría ahorrar recursos computacionales también para los clientes que no tienen que comprimir y descomprimir cada diez segundos.

La localización de los ficheros delta podría obtenerse por el cliente de la misma manera que el cliente determina la localización de la MPD. Como alternativa, la localización del fichero delta podría referenciarse en la MPD. Podría haber un fichero delta o una pluralidad de ficheros delta numerados de manera secuencial.

20 En una realización, la MPD puede incluir un indicador que indica que la MPD soporta el uso de ficheros delta. Es decir, para hacer uso de los ficheros delta, el cliente puede necesitar conocer que los ficheros delta están disponibles. El cliente puede haber tenido conocimiento de este hecho por un indicador en la MPD. Por ejemplo, un elemento en la MPD puede indicar que se están usando ficheros delta para describir actualizaciones a la MPD. Los atributos del elemento pueden indicar cuántos ficheros delta se almacenan, los nombres de los ficheros delta, la frecuencia con la que se actualizan los ficheros delta y/u otra información. En otras realizaciones, el indicador puede indicar que las actualizaciones están disponibles en alguna otra forma, como se describe a continuación.

25 El cliente tiene diversos medios de conocer que no se ha perdido una actualización delta. La MPD puede contener una duración de "actualización mínima" que indica la frecuencia con la que la MPD puede posiblemente actualizarse. Si hubiera una duración de actualización mínima descrita en la MPD, a continuación el cliente conoce que la MPD no se actualizará de manera más frecuente que la duración, y el cliente no necesita buscar los ficheros delta a una frecuencia mayor que esa duración. Por lo que si la MPD se actualiza cada diez segundos, el cliente descargaría la MPD completa cuando sintoniza en primer lugar, y por lo tanto simplemente descargaría el fichero delta cada diez segundos. Si el cliente continúa descargando ficheros delta cada diez segundos, puede asegurarse que el cliente que no ha perdido un fichero delta.

30 Como alternativa, al mismo tiempo que se actualiza la MPD, puede ponerse a disposición un fichero delta en el servidor al cliente. Si el cliente actualiza a la frecuencia del periodo de actualización mínimo, entonces sabe que no perderá una actualización de fichero delta. También, pueden almacenarse versiones anteriores del fichero delta de modo que el cliente no necesita solicitar el fichero delta a la frecuencia de actualización mínima. Por ejemplo, el fichero delta puede tener nombre de fichero delta.xml. La versión anterior de delta.xml puede almacenarse en delta-1.xml y la versión anterior de delta1.xml puede almacenarse en un fichero nombrado delta-2.xml. delta.xml (y por lo tanto delta-1.xml y delta-2.xml) contiene números de secuencia de modo que el cliente puede conocer si ha perdido una versión. Por lo que, si el cliente descargara ficheros delta a dos veces el periodo de actualización mínimo, puede estar seguro de que no ha habido más de dos actualizaciones diferentes a la MPD, pero no puede conocer seguro si ha habido 0, 1 o 2 actualizaciones. La secuencia en el fichero delta proporciona esta indicación.

35 Como alternativa, en otra realización, podría proporcionarse a los ficheros delta nombres de fichero que se numeran de manera secuencial, y el cliente podría confirmar que no ha perdido un número en la secuencia. Por ejemplo, para que un cliente aproveche el concepto de un fichero delta sin tener que sintonizar cada diez segundos (o cualquiera que pasara a ser la duración de actualización mínima), podrían crearse múltiples ficheros delta. Por ejemplo, si la actualización mínima pasara a ser diez segundos, entonces "delta\_10" podría ser un fichero delta que especifica qué ha cambiado en la versión más reciente de la MPD con relación a la versión inmediatamente precedente de la MPD, "delta\_20" podría ser un fichero delta que especifica qué ha cambiado en las dos versiones más recientes de la MPD, y así sucesivamente. Si el cliente desea un fichero delta que se ha actualizado cada diez segundos, el cliente podría descargar el fichero delta\_10, si el cliente desea un fichero delta que se ha actualizado cada 20 segundos, el cliente podría descargar el fichero delta\_20, y así sucesivamente. También, si el cliente tiene conocimiento de que ha perdido un fichero delta\_10, por ejemplo, el cliente podría descargar un fichero delta\_20 y permanecer en sincronización con la MPD en el servidor. Podría usarse una convención de nomenclatura convencional para los ficheros delta, o los nombres de los ficheros podrían colocarse en la MPD.

En otra realización alternativa, a los ficheros delta pueden proporcionarse diferentes números de versión, nombres de versión o números de secuencia que indican qué actualización en una pluralidad de actualizaciones representa los ficheros delta. En este caso, los ficheros delta no sería necesario que se crearan de una manera periódica. Por ejemplo, a los ficheros delta pueden proporcionarse nombres tales como "revisión 1" para una primera revisión a la MPD original, "revisión 2" para una revisión que incluye todas las actualizaciones desde la "revisión 1", "revisión 3" para una revisión que incluye todas las actualizaciones desde la "revisión 2", y así sucesivamente. Las revisiones no se generarían necesariamente a intervalos que corresponden a los intervalos a los que se actualiza el contenido descrito por la MPD. Es decir, si se añade un nuevo segmento de contenido cada diez segundos, los ficheros delta pueden crearse a algún otro intervalo periódico o pueden crearse sin ninguna periodicidad regular. Si el cliente deseara actualizar su MPD, el cliente podría simplemente descargar todos los ficheros delta con números de revisión mayores que el número de revisión del fichero delta más reciente que descargó. El cliente no necesitaría descargar los ficheros delta sin ninguna frecuencia regular.

En una realización, si el cliente fallara al obtener un fichero delta por cualquier razón, el cliente podría descargar toda la MPD de nuevo. Posteriormente, el cliente podría reanudar la descarga únicamente de los ficheros delta.

Aunque los comandos de Añadir Elemento y Borrar Elemento pueden describir los deltas más comunes, tal como descripciones de segmento añadidas o borradas de la MPD, son posibles muchos otros comandos. Por ejemplo Añadir Hermano, Añadir Hijo, Añadir Padre, Borrar Hermano, Borrar Hijo, Borrar Padre, Añadir Atributo, Sustituir Atributo, Eliminar Atributo, etc.

Como una alternativa a tener los ficheros delta, una etiqueta <delta> podría colocarse cerca de la parte superior o inferior de la MPD, de modo que un cliente podría usar un GET parcial, o un mensaje similar, para descargar únicamente una porción de la MPD, es decir, información que ha cambiado en la MPD desde la última actualización de la MPD que pudiera colocarse entre las etiquetas de apertura y cierre de <delta>. El cliente podría a continuación recuperar únicamente la información dentro de las etiquetas delta y podría ignorar el resto de la MPD.

El envío por flujo continuo adaptativo de HTTP está actualmente controlado por cliente y, como se ha descrito anteriormente, el servidor es un servidor web convencional. Sin embargo, en el futuro, el servidor puede incluir funcionalidad adicional. Por lo tanto, en una realización, el cliente podría descargar inicialmente la MPD, y a continuación actualizar en forma de un fichero delta o podría insertarse de alguna otra forma al cliente por el servidor en lugar de que el cliente compruebe el servidor para actualizaciones.

En otra realización donde el servidor tiene funcionalidad adicional, en lugar de que el cliente use la combinación del enfoque "delta" y XPath para acceder a la MPD, podría usarse XCAP para proporcionar la MPD al cliente. En este caso, no necesitarían prepararse ficheros delta, en su lugar, podría proporcionarse un identificador o número de secuencia predecible o conocido a cada segmento de metadatos en la MPD. El cliente podría acceder directamente al fichero MPD en un servidor activado para XCAP y usar XCAP GET para apuntar al elemento en la MPD que sigue el elemento más reciente que descargó el cliente. El cliente podría a continuación recuperar un número especificado de elementos empezando en ese punto. Los mecanismos XCAP para acceder a documentos de XML se describen en el RFC 4825. Como alternativa, el servidor podría usar un mensaje basado en XCAP para insertar actualizaciones de MPD al cliente.

En otra alternativa donde el servidor tiene funcionalidad adicional, podría usarse un modelo de suscripción/notificación para notificar al cliente de cambios a la MPD. Es decir, el cliente podría suscribirse para ser notificado de cambios a la MPD, y el servidor podría notificar al cliente de cualesquiera cambios cuando los cambios tuvieran lugar. El modelo de suscripción/notificación puede implementarse usando SIP:SUSCRIBE/SIP:NOTIFY o usando XDCCP (Protocolo de Comando de Documento de XML), que es un mecanismo alternativo a SIP que está basado en HTTP. Ambos de estos mecanismos se describen en las especificaciones del activador 2.1 de OMA XDM (Gestión de Documento de XML). Para soportar estos mecanismos, se requiere un uso de aplicación para MPD para alojar el fichero de MPD en el servidor de XDM. Y el cliente de XDM necesita implementarse en el UE (Equipo de Usuario) para poder acceder y suscribirse al fichero de MPD como se especifica en el uso de aplicación.

En cualquiera de estas alternativas donde el servidor tiene funcionalidad adicional, podría incluirse un indicador en la MPD para indicar que el servidor soporta el uso de actualizaciones a la MPD. Es decir, similar al indicador anteriormente descrito que indica el soporte de ficheros delta, un indicador podría indicar que el servidor tiene la capacidad de insertar actualizaciones de MPD al cliente, la capacidad de usar XCAP para proporcionar actualizaciones de MPD al cliente, la capacidad de usar un modelo de suscripción/notificación para informar al cliente de actualizaciones de MPD, u otras capacidades para proporcionar actualizaciones de MPD al cliente. El cliente podría a continuación tener funcionalidad apropiada para interpretar el indicador y tomar acciones apropiadas para recibir actualizaciones de la MPD sin recibir la MPD completa.

Las Figuras 4a y 4b ilustran un ejemplo de un esquema delta que puede usarse para los ficheros delta anteriormente analizados.

La Figura 5 ilustra una realización de un método 500 para obtener información de descripción de presentación de medios. En el bloque 510, un cliente obtiene información relacionada con un cambio en una primera MPD en un servidor. En el bloque 520, el cliente actualiza una segunda MPD en el cliente con la información relacionada con el cambio.

5 En resumen, puede añadirse un elemento a la MPD para indicar soporte de ficheros delta. El nombre del elemento añadido a la MPD puede ser "delta". El elemento "delta" puede contener un atributo nombrado "numberOfStoredDeltas", que toma un valor entero positivo. El nombre del fichero delta puede ser "delta.xml" y su versión anterior puede ser "delta-1.xml". La versión anterior de "delta-1.xml" puede ser "delta-2.xml", etc. Por lo que si el valor del atributo "numberOfStoredDeltas" es "3", entonces el cliente conoce que los ficheros delta.xml, delta-1.xml, y delta-2.xml existen en el servidor.

También, el esquema en la Figura 4 podría usarse para los ficheros delta. El valor del atributo "sel" en cualquier elemento <add>, <replace>, o <remove> puede ser una expresión XPath válida. El esquema contiene un atributo "xpathVersion" que indica la versión de XPath a usarse. El valor por defecto es "1.0". Si se usa otra versión de XPath, podría indicarse mediante el valor del atributo "xpathVersion". Si el cliente, tras descargar un fichero delta, halla un error en la expresión XPath en un atributo "sel", puede descargar la MPD completa.

El cliente, servidor, y otros componentes anteriormente descritos pueden incluir un componente de procesamiento que puede ejecutar instrucciones relacionadas con las acciones anteriormente descritas. La Figura 6 ilustra un ejemplo de un sistema 1300 que incluye un componente de procesamiento 1310 adecuado para implementar una o más realizaciones desveladas en el presente documento. Además del procesador 1310 (que puede denominarse como una unidad de procesador central o CPU), el sistema 1300 puede incluir dispositivos de conectividad de red 1320, memoria de acceso aleatorio (RAM) 1330, memoria de sólo lectura (ROM) 1340, almacenamiento secundario 1350, y dispositivos de entrada/salida (E/S) 1360. Estos componentes pueden comunicarse entre sí mediante un bus 1370. En algunos casos, algunos de estos componentes pueden no estar presentes o pueden combinarse en diversas combinaciones entre sí o con otros componentes no mostrados. Estos componentes pueden localizarse en una única entidad física o en más de una entidad física. Cualesquiera acciones descritas en el presente documento como que se toman por el procesador 1310 pueden tomarse por el procesador 1310 en solitario o por el procesador 1310 en conjunto con uno o más componentes mostrados o no mostrados en el dibujo, tal como un procesador de señales digitales (DSP) 1380. Aunque el DSP 1380 se muestra como un componente separado, el DSP 1380 puede incorporarse en el procesador 1310.

30 El procesador 1310 ejecuta instrucciones, códigos, programas informáticos, o guiones que puede acceder desde los dispositivos de conectividad de red 1320, RAM 1330, ROM 1340, o el almacenamiento secundario 1350 (que puede incluir diversos sistemas basados en disco tal como disco duro, disco flexible o disco óptico). Aunque únicamente se muestra una CPU 1310, pueden estar presentes múltiples procesadores. Por lo tanto, mientras que las instrucciones pueden analizarse según se ejecutan por un procesador, las instrucciones pueden ejecutarse simultáneamente, en serie o de otra manera por uno o múltiples procesadores. El procesador 1310 puede implementarse como uno o más chips de CPU.

Los dispositivos de conectividad de red 1320 pueden tomar la forma de módems, bancos de módem, dispositivos de Ethernet, dispositivos de interfaz de Bus Serie Universal (USB), interfaces serie, dispositivos de anillo con paso de testigo, dispositivos de Interfaz de Datos Distribuidos por Fibra (FDDI), dispositivos de red de área local inalámbrica (WLAN), dispositivos de transceptor de radio tales como dispositivos de Acceso Múltiple por División de Código (CDMA), dispositivos de transceptor de radio de Sistema Global para Comunicación Móvil (GSM), dispositivos de Interoperabilidad Mundial para Acceso por Microondas (WiMAX), y/u otros dispositivos bien conocidos para conectar a redes. Estos dispositivos de conectividad de red 1320 pueden posibilitar que el procesador 1310 comunique con la Internet o una o más redes de telecomunicaciones u otras redes a partir de las cuales el procesador 1310 puede recibir información o a las que el procesador 1310 puede emitir información. Los dispositivos de conectividad de red 1320 pueden incluir también uno o más componentes de transceptor 1325 que pueden transmitir y/o recibir datos inalámbricamente.

La RAM 1330 puede usarse para almacenar datos volátiles y tal vez para almacenar instrucciones que se ejecutan por el procesador 1310. La ROM 1340 es un dispositivo de memoria no volátil que típicamente tiene una capacidad de memoria menor que la capacidad de memoria del almacenamiento secundario 1350. La ROM 1340 puede usarse para almacenar instrucciones y tal vez datos que se leen durante la ejecución de las instrucciones. El acceso tanto a la RAM 1330 como a la ROM 1340 es típicamente más rápido que al almacenamiento secundario 1350. El almacenamiento secundario 1350 está típicamente comprendido de una o más unidades de disco o unidades de cinta y puede usarse para almacenamiento no volátil de datos o como un dispositivo de almacenamiento de datos de desbordamiento si la RAM 1330 no es lo suficientemente grande para mantener todos los datos de funcionamiento. El almacenamiento secundario 1350 puede usarse para almacenar programas que se cargan en RAM 1330 cuando tales programas se seleccionan para ejecución.

5 Los dispositivos de E/S 1380 pueden incluir pantallas de cristal líquido (LCD), pantallas táctiles, teclados, teclados numéricos, conmutadores, selectores, ratón, bolas de mando, reconocedores de voz, lectores de tarjetas, lectores de cinta de papel, impresoras, monitores de vídeo u otros dispositivos de entrada/salida bien conocidos. También, el transceptor 1325 puede considerarse que es un componente de los dispositivos de E/S 1360 en lugar de o además de ser un componente de los dispositivos de conectividad de red 1320.

En una realización, se proporciona un método para obtener información de descripción de presentación de medios. El método incluye un cliente que obtiene información relacionada con un cambio en una primera MPD en un servidor. El método incluye adicionalmente que el cliente actualice una segunda MPD en el cliente con la información relacionada con el cambio.

10 En otra realización, se proporciona un cliente. El cliente incluye un procesador configurado de manera que el cliente obtiene información relacionada con un cambio en una primera MPD en un servidor, y configurado de manera que el cliente actualiza una segunda MPD en el cliente con la información relacionada con el cambio.

15 Las siguientes referencias son relevantes para la presente divulgación: 3GPP Especificación Técnica (TS) 26.234, 3GPP TS 26.244, ISO/IEC 14496-12, Grupo Especial sobre Ingeniería de Internet (IETF) Solicitud de Comentarios (RFC) 5874 e IETF RFC 5261.

20 Aunque se han proporcionado varias realizaciones en la presente divulgación, debería entenderse que los sistemas y métodos desvelados pueden realizarse en muchas otras formas específicas sin alejarse del alcance de la presente divulgación, que se define mediante las reivindicaciones. Los presentes ejemplos se han de considerar como ilustrativos y no restrictivos, y la intención es que no estén limitados a los detalles proporcionados en el presente documento. Por ejemplo, los diversos elementos o componentes pueden combinarse o integrarse en otro sistema o ciertas características pueden omitirse, o no implementarse.

25 También, técnicas, sistemas, subsistemas y métodos descritos e ilustrados en las diversas realizaciones como discretos o separados pueden combinarse o integrarse con otros sistemas, módulos, técnicas, o métodos sin alejarse del alcance de la presente divulgación, que se define mediante las reivindicaciones. Otros elementos mostrados o analizados como acoplados o directamente acoplados o que comunican entre sí pueden acoplarse indirectamente o comunicar a través de alguna interfaz, dispositivo o componente intermedio, ya sea eléctrica, mecánicamente o de otra manera. Otros ejemplos de cambios, sustituciones y modificaciones pueden determinarse por un experto en la materia y podrían realizarse sin alejarse del alcance de las reivindicaciones adjuntas.



**REIVINDICACIONES**

1. Un método que comprende:
  - 5 recibir, en un cliente (140), una descripción de presentación de medios MPD que incluye un elemento de Lenguaje de Marcas Extensible, XML, que indica soporte, por un servidor (120), de ficheros delta de MPD, en el que el elemento de XML comprende un primer atributo de XML que comprende un Localizador de Recurso Uniforme, URL, de un fichero delta de MPD que incluye un cambio para la MPD; y
  - enviar, por el cliente (140), una solicitud de Protocolo de Transferencia de Hipertexto HTTP GET para el fichero delta de MPD, en el que la solicitud HTTP GET comprende el URL.
- 10 2. El método de la reivindicación 1, en el que el fichero delta de MPD comprende adicionalmente un comando de cambio y una localización en la que se aplica el comando de cambio.
3. El método de la reivindicación 2, en el que el comando de cambio comprende uno de:
  - un comando añadir;
  - un comando borrar; o
  - un comando sustituir.
- 15 4. El método de la reivindicación 1, en el que el elemento de XML comprende un segundo atributo de XML indicativo de una duración de actualización mínima.
5. El método de la reivindicación 1 que comprende adicionalmente enviar, por el cliente (140), una segunda solicitud HTTP GET para una segunda MPD basándose en un fallo de la solicitud HTTP GET para el fichero delta de MPD.
- 20 6. El método de la reivindicación 1, en el que el elemento de XML comprende un segundo atributo de XML indicativo de una disponibilidad del fichero delta de MPD.
7. El método de la reivindicación 1, en el que el cliente (140) se suscribe para recibir el fichero delta de MPD y se notifica al cliente (140) cuando el fichero delta de MPD se hace disponible.
8. Un dispositivo cliente configurado para realizar las etapas de cualquiera de las reivindicaciones del método 1-7.
9. Un método, que comprende:
  - 25 transmitir, desde un servidor (120), una descripción de presentación de medios MPD que incluye un elemento de Lenguaje de Marcas Extensible, XML, que indica soporte, por el servidor (120), de ficheros delta de MPD, en el que el elemento de XML comprende un primer atributo de XML que comprende un Localizador de Recurso Uniforme, URL, de un fichero delta de MPD que incluye un cambio para la MPD; y
  - 30 recibir, por el servidor (120), una solicitud de Protocolo de Transferencia de Hipertexto HTTP GET para un fichero delta de MPD, en el que la solicitud HTTP GET comprende el URL.
10. El método de la reivindicación 9, en el que el servidor (120) es compatible con HTTP norma 1.1 y superiores.
11. El método de la reivindicación 9, en el que el fichero delta de MPD comprende adicionalmente un comando de cambio y una localización en la que se aplica el comando de cambio.
12. El método de la reivindicación 11, en el que el comando de cambio comprende uno de:
  - 35 un comando añadir;
  - un comando borrar; o
  - un comando sustituir.
13. El método de la reivindicación 9, en el que el elemento de XML comprende un segundo atributo de XML indicativo de una duración de actualización mínima.
- 40 14. El método de la reivindicación 9, en el que el elemento de XML comprende un segundo atributo de XML indicativo de una disponibilidad del fichero delta de MPD.
15. El método de la reivindicación 9, que comprende adicionalmente recibir una segunda solicitud HTTP GET para un segundo fichero delta de MPD.

16. Un servidor (120) configurado para realizar las etapas de cualquiera de las reivindicaciones de método 9-15.

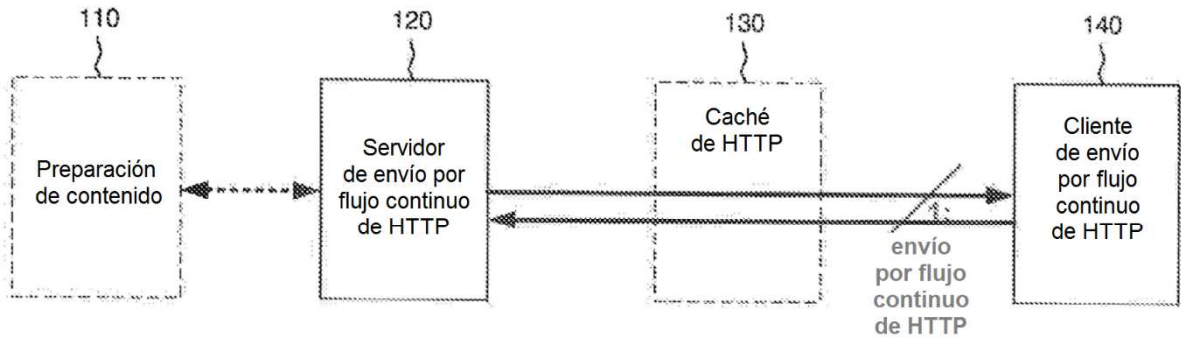


Figura 1

```

<?xml version="1.0" encoding="UTF-8" ?>
- <MPD type="Live" availabilityStartTime="2010-07-01T05:00:00Z"
  availabilityEndTime="2010-07-08T05:00:00Z" mediaPresentationDuration="PT2H"
  minimumUpdatePeriodMPD="PT10S" minBufferTime="PT10S"
  timeShiftBufferDepth="PT30M" baseUrl="http://www.example.com/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009
  3GPP-MPD-r1.xsd"
  xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009">
- <ProgramInformation moreInformationURL="http://www.example.com">
  <Title>Example</Title>
  <Source>Example</Source>
  <Copyright>Example</Copyright>
  </ProgramInformation>
- <Period start="PT0S" segmentAlignmentFlag="true" bitstreamSwitchingFlag="true">
- <Representation bandwidth="239000" width="320" height="240" lang="en"
  mimeType="video/3gpp; codecs=avc1.42E00b, mp4a.40.2">
- <SegmentInfo duration="PT10S">
  <InitialisationSegmentURL sourceURL="p1rep1.3gp" range="0-985" />
  <Url sourceURL="p1rep1.3gp" range="985-293761" />
  <Url sourceURL="p1rep1.3gp" range="293761-592501" />
  * * *
  <Url sourceURL="p1rep1.3gp" range="17304004-17600064" />
  <Url sourceURL="p1rep1.3gp" range="17600064-17894640" />
  </SegmentInfo>
  </Representation>
- <Representation bandwidth="892000" width="480" height="240" lang="en"
  mimeType="video/3gpp; codecs=avc1.42E015, mp4a.40.2">
- <SegmentInfo duration="PT10S">
  <InitialisationSegmentURL sourceURL="p3rep3.3gp" range="0-985" />
  <Url sourceURL="p3rep3.3gp" range="985-1126190" />
  <Url sourceURL="p3rep3.3gp" range="1126190-2228575" />
  * * *
  <Url sourceURL="p3rep3.3gp" range="64712374-65844317" />
  <Url sourceURL="p3rep3.3gp" range="65844317-66966044" />
  </SegmentInfo>
  </Representation>
  </Period>
</MPD>

```

Figura 2

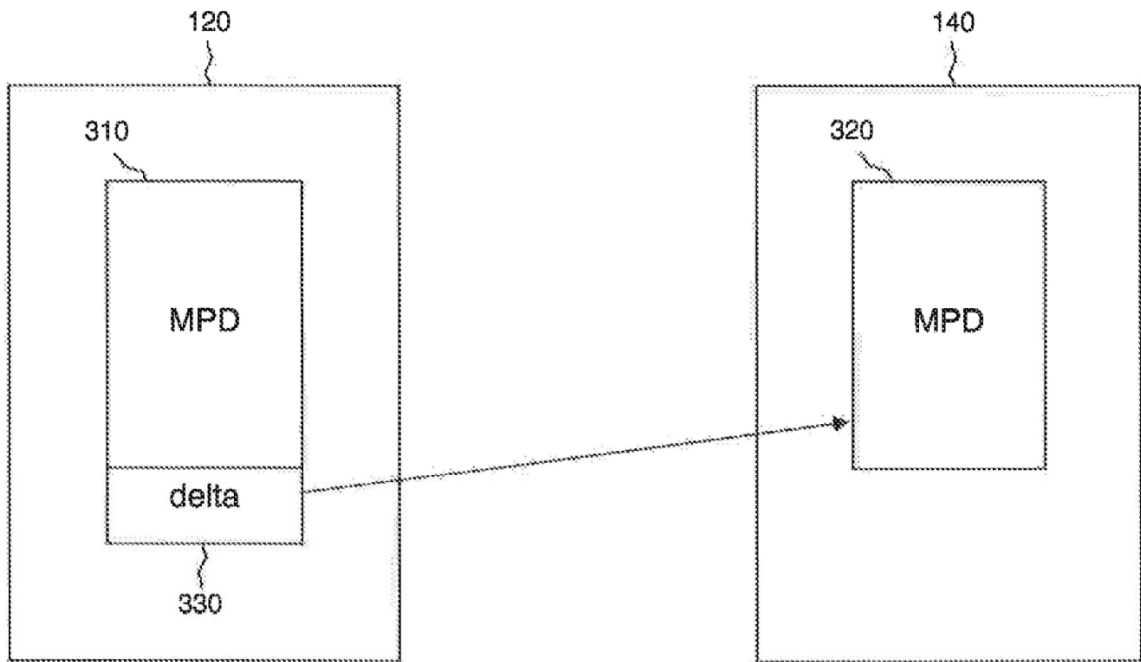


Figura 3

```

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- parche-ops -->

  <xs:simpleType name="pos">
    <xs:restriction base="xs:string">
      <xs:enumeration value="before"/>
      <xs:enumeration value="after"/>
      <xs:enumeration value="prepend"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="add">
    <xs:complexContent mixed="true">
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:any processContents="lax" namespace="##any"
            minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="sel" type="xs:string"
          use="required"/>
        <xs:attribute name="pos" type="pos"/>
        <xs:attribute name="type" type="xs:string"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="replace">
    <xs:complexContent mixed="true">
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:any processContents="lax" namespace="##any"
            minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="sel" type="xs:string" use="required"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:simpleType name="ws">
    <xs:restriction base="xs:string">
      <xs:enumeration value="before"/>
      <xs:enumeration value="after"/>
      <xs:enumeration value="both"/>
    </xs:restriction>
  </xs:simpleType>

```

Figura 4a

```

    xs:complexType name="remove">
      <xs:attribute name="sel" type="xs:string" use="required"/>
      <xs:attribute name="ws" type="ws"/>
    </xs:complexType>
  <!-- fin de parche-ops -->

  <!-- elemento raíz -->
  <xs:element name="delta">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element name="add">
            <xs:complexType mixed="true">
              <xs:complexContent>
                <xs:extension base="add">
                  <xs:anyAttribute processContents="lax"/>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="remove">
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="remove">
                  <xs:anyAttribute processContents="lax"/>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="replace">
            <xs:complexType mixed="true">
              <xs:complexContent>
                <xs:extension base="replace">
                  <xs:anyAttribute processContents="lax"/>
                </xs:extension>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:any namespace="##other" processContents="lax"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="sequenceNumber" type="xs:positiveInteger" use="required"/>
      <xs:attribute name="xpathVersion" type="xs:string" default="1.0"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figura 4b

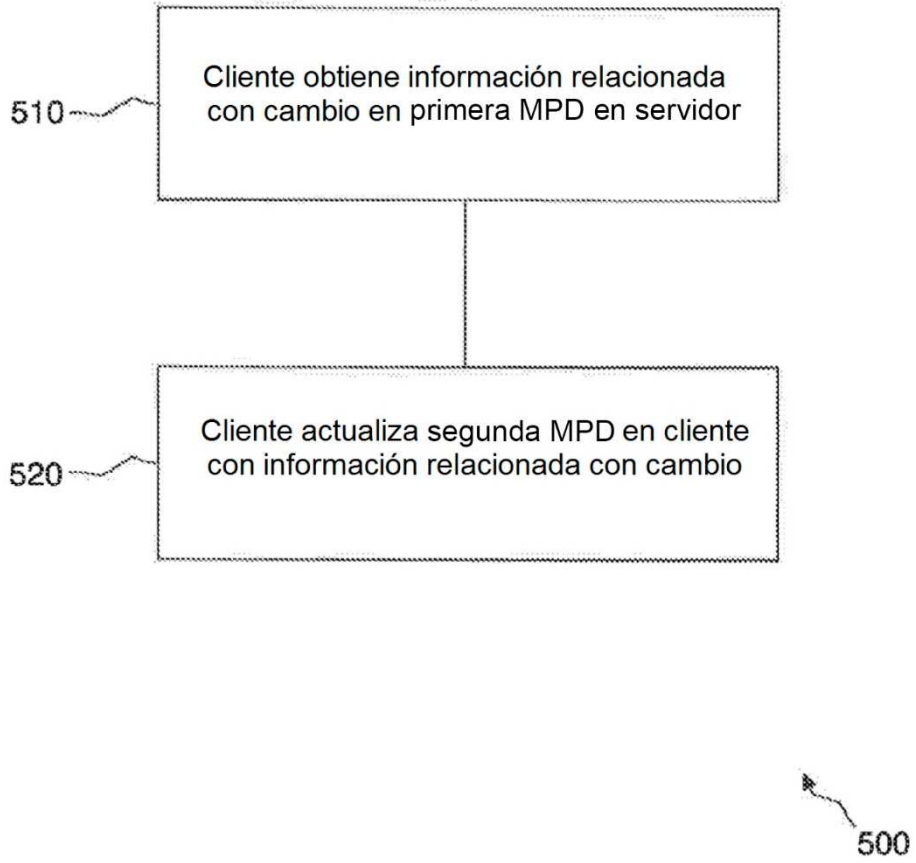


Figura 5



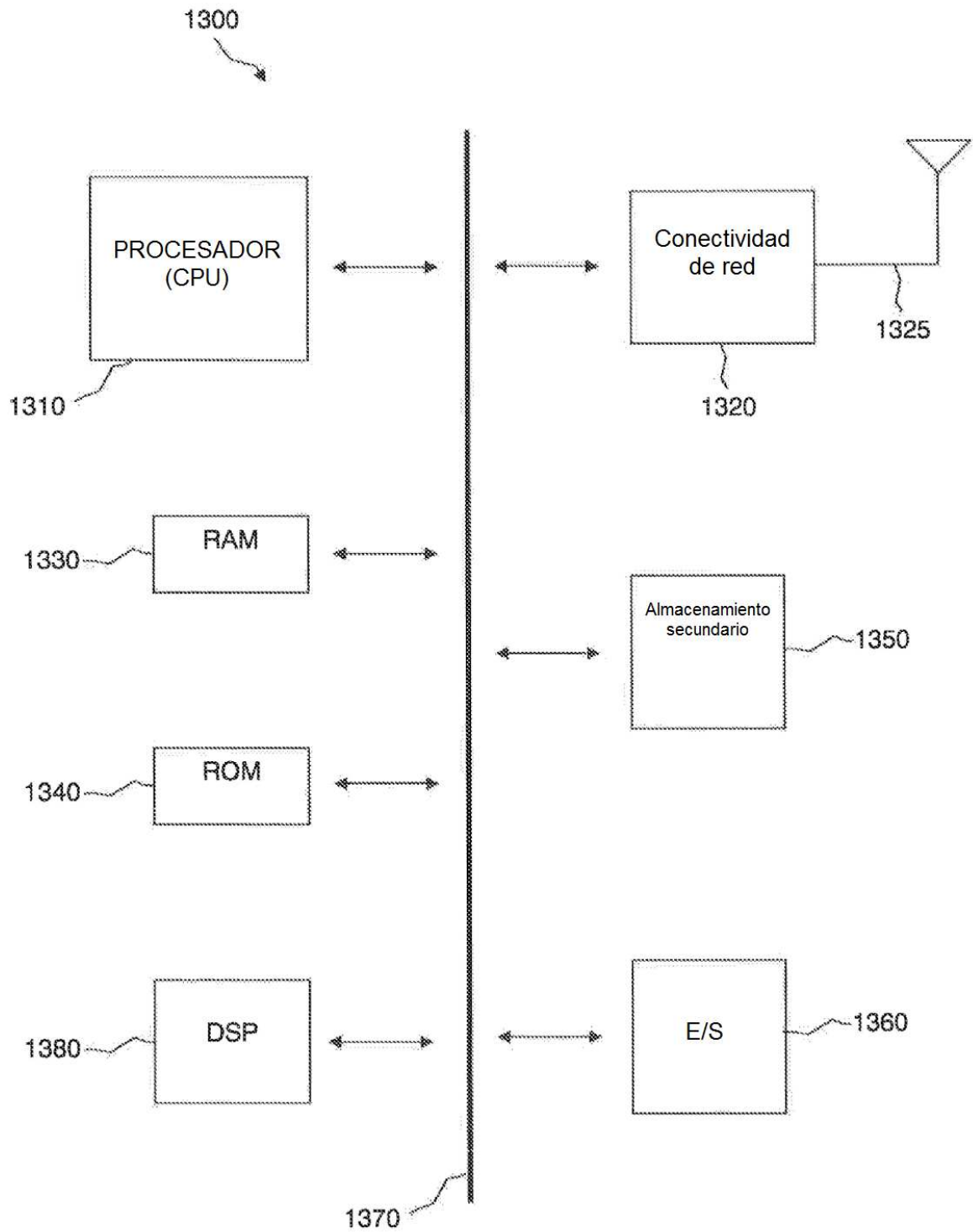


Figura 6