



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 313 323**

51 Int. Cl.:
H04L 12/56 (2006.01)
H04L 29/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **05732319 .8**
96 Fecha de presentación : **11.04.2005**
97 Número de publicación de la solicitud: **1872536**
97 Fecha de publicación de la solicitud: **02.01.2008**

54 Título: **Técnica para controlar transmisiones de paquetes de datos de velocidad binaria variable.**

45 Fecha de publicación de la mención BOPI:
01.03.2009

45 Fecha de la publicación del folleto de la patente:
01.03.2009

73 Titular/es: **Telefonaktiebolaget LM Ericsson (publ)**
164 83 Stockholm, SE

72 Inventor/es: **Kampmann, Markus y**
Plum, Christoph

74 Agente: **Elzaburu Márquez, Alberto**

ES 2 313 323 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

Técnica para controlar transmisiones de paquetes de datos de velocidad binaria variable.

5 **Campo de la invención**

La presente invención se refiere a la comunicación de datos y, más específicamente, a una técnica para controlar transmisiones de paquetes de datos de flujo continuo (del inglés “streaming”).

10 **Antecedentes de la invención**

Cada vez se están transmitiendo mayores cantidades de datos desde servidores a clientes a través de infraestructuras de comunicación tales como redes del Protocolo de Internet (IP) basadas en paquetes. Una aplicación en particular cuya popularidad está creciendo es el flujo continuo multimedia. No obstante, se deben realizar mejoras en la provisión de flujos continuos de datos fiables antes de la adopción generalizada de dichos servicios. Por ejemplo, como las velocidades de los enlaces de transmisión de datos entre la red IP y un dispositivo cliente de un usuario tienden a fluctuar, cualquier perturbación en la entrega de datos al usuario puede dar como resultado un deterioro importante de la reproducción para el usuario final, es decir, un deterioro en la calidad de los medios observados por el usuario. En particular, es importante que exista un suministro suficiente de paquetes de datos en el dispositivo cliente para que sean recuperados por una aplicación multimedia a medida que avanza la reproducción (es decir, la presentación del archivo multimedia por parte de la aplicación o reproductor multimedia).

En muchos casos, la velocidad de transmisión de los paquetes no se puede cambiar, ya que esta velocidad depende del ancho de banda del enlace de comunicación (o por lo menos resulta poco práctico cambiar la velocidad de transmisión de los paquetes). No obstante, con frecuencia debe cambiarse la velocidad a la que se alimentan datos hacia el dispositivo de salida del usuario. Típicamente, para aplicaciones de flujo continuo, dichos ajustes se logran usando la “conmutación del flujo continuo”. Con la conmutación del flujo continuo, el mismo contenido de medios, por ejemplo, una secuencia de vídeo particular, se codifica previamente a velocidades binarias diferentes y se almacena en el servidor. De este modo, hay disponibles versiones diferentes del mismo flujo continuo. Durante la transmisión, el servidor selecciona la versión particular que tiene una velocidad binaria de datos más apropiada basándose en el ancho de banda actual disponible en la red y basándose en el estado de la memoria intermedia del cliente. Los módulos lógicos de conmutación utilizados por el servidor deciden si se debe conmutar a otra versión del flujo continuo y cuándo debe realizarse dicha conmutación. En el caso de la denominada “conmutación descendente”, el flujo continuo se conmuta a una versión con una velocidad binaria codificada menor. En el caso de una “conmutación ascendente”, la conmutación se realiza hacia una versión con una velocidad binaria codificada mayor. En muchas implementaciones, los criterios para la conmutación utilizan umbrales predefinidos que se han definido con respecto al estado de la memoria intermedia del cliente. En un ejemplo, los umbrales se basan en un nivel de llenado de la memoria intermedia, que representa la cantidad de datos dentro de la memoria intermedia del cliente en bytes. En otro ejemplo, los umbrales se basan en una longitud de reproducción (PT) de medios almacenados en la memoria intermedia del cliente, la cual representa la cantidad de tiempo en segundos que se tardará en reproducir para el usuario los datos que ya están dentro de la memoria intermedia del cliente. En este caso, se describen ejemplos que involucran la longitud de reproducción, aunque en su lugar se puede usar el nivel de llenado de la memoria intermedia u otros parámetros apropiados.

Algunas técnicas convencionales para determinar el estado de la memoria intermedia del cliente utilizan información incluida en informes del receptor (RRs) del Protocolo de Control de Transporte de Tiempo Real (RTCP). El RR contiene información referente al siguiente número de secuencia (NSN) o al número de secuencia más antiguo almacenado en memoria intermedia (OBSN) dentro de la memoria intermedia del cliente y al número de secuencia más alto recibido (HRSN) dentro de la memoria del cliente, y dicha información se usa para determinar el espacio consumido de la memoria intermedia ya que se conoce el tamaño de cada paquete dentro del intervalo desde el HRSN al NSN/OBSN. Si el espacio libre dentro de la memoria intermedia del cliente está por debajo de un nivel de llenado preferido de la memoria intermedia del cliente, entonces se selecciona una versión diferente del flujo continuo. Por ejemplo, si la longitud de reproducción (PT) de la memoria intermedia cae por debajo de un umbral mínimo predeterminado (PT_{DESC}), entonces se produce un riesgo de vaciado de la memoria intermedia, es decir, la memoria intermedia del cliente se vacía de tal manera que no hay datos para ser enviados por flujo continuo al usuario. Esto da como resultado una congelación de la reproducción, en la que la última imagen presentada al usuario típicamente se congela hasta que se ha añadido a la memoria intermedia del cliente una cantidad suficiente de datos adicionales para reiniciar el flujo continuo hacia el dispositivo de salida utilizado por el usuario, es decir, se requiere un “realmacenamiento” en memoria intermedia del cliente. El realmacenamiento en memoria intermedia puede resultar extremadamente molesto desde el punto de vista del usuario.

Para evitar un posible realmacenamiento en memoria intermedia debido a un vaciado de la memoria intermedia del cliente, el servidor detecta cuándo la longitud de reproducción (PT) dentro de la memoria intermedia del cliente cae por debajo del umbral PT_{DESC} , y entonces ajusta la velocidad binaria (es decir, selecciona una versión del flujo continuo que tiene una velocidad binaria diferente) en un intento de evitar que la memoria intermedia del cliente llegue a vaciarse completamente. Más específicamente, el servidor realiza una conmutación descendente, es decir, una conmutación a un flujo continuo de velocidad binaria menor. La razón por la que se realiza una conmutación descendente, en lugar de una conmutación ascendente, es que la razón más probable de que la memoria intermedia del cliente se esté vaciando es que la velocidad del enlace entre el servidor y la memoria intermedia del cliente sea

menor que la prevista, es decir, el ancho de banda efectivo sea menor que el necesario para la velocidad binaria que se está usando en ese momento. Como consecuencia, los datos no están siendo recibidos por la memoria intermedia del cliente a la misma velocidad a la que la memoria intermedia del cliente está alimentando datos hacia el dispositivo de salida del usuario. Por lo tanto, la memoria intermedia del cliente, que debe permanecer bastante bien poblada de datos, se vacía. Al conmutar hacia la velocidad binaria inferior, la memoria intermedia del cliente alimenta datos hacia la unidad de presentación a una velocidad menor, permitiendo de este modo más tiempo para la recepción de datos desde el servidor, y evitando así que la memoria intermedia del cliente llegue a vaciarse completamente. Desde el punto de vista del usuario, la calidad del flujo continuo de medios se degrada debido a la conmutación descendente, por ejemplo, el tamaño de la imagen presentada del flujo continuo de vídeo se hace más pequeño, la resolución de la imagen se reduce, o se observan distorsiones mayores en la imagen. Aún así, esto es preferible a la congelación de la reproducción antes mencionada que se produce durante el realmacenamiento en memoria intermedia.

Por otro lado, si la longitud de reproducción (PT) de la memoria intermedia supera un umbral máximo predeterminado (PT_{ASC}), entonces se produce un riesgo de desbordamiento de la memoria intermedia, es decir, la memoria intermedia del cliente se llena de modo que no queda espacio para paquetes adicionales. Típicamente, cualquier paquete recibido por la memoria intermedia del cliente, aunque no almacenado en la misma, no es vuelto a enviar por el servidor, y por lo tanto los datos de esos paquetes simplemente no se reenvían hacia el dispositivo de salida del usuario. Una vez que la memoria intermedia del cliente es capaz nuevamente de almacenar paquetes, el flujo continuo de datos se reanuda con los paquetes nuevos. De este modo, desde el punto de vista del usuario, se produce una pérdida repentina de contenido cuando el flujo continuo simplemente salta hacia delante. En el caso de un film o película, se puede perder diálogo, interfiriendo de este modo con la capacidad del usuario para seguir la historia. En el caso de música, la canción simplemente salta hacia delante. Tal como se apreciará, esto puede resultar bastante molesto también desde el punto de vista del usuario.

Para evitar una interrupción del flujo continuo debida a un desbordamiento de la memoria intermedia del cliente, el servidor detecta cuándo la longitud de reproducción (PT) dentro de la memoria intermedia del cliente supera el umbral PT_{ASC} y entonces realiza una conmutación ascendente, es decir, una conmutación a un flujo continuo de velocidad binaria mayor. La razón por la que se realiza una conmutación ascendente, en lugar de una conmutación descendente, es que la razón más probable de que la memoria del cliente se esté llenando es que la velocidad del enlace entre el servidor y la memoria intermedia del cliente es mayor que la prevista, es decir, el ancho de banda efectivo es mayor que el necesario para la velocidad binaria que se está usando de ese momento. Como consecuencia, los datos son recibidos por la memoria intermedia del cliente a una velocidad mayor que la velocidad a la que la memoria intermedia del cliente alimenta los datos hacia el dispositivo de salida del usuario. Por lo tanto, la memoria intermedia del cliente se desborda. Al conmutar a la velocidad binaria mayor, la memoria intermedia del cliente alimenta datos hacia el dispositivo de salida a la velocidad mayor, evitando de este modo el desbordamiento de la memoria intermedia del cliente. Desde el punto de vista del usuario, la calidad del flujo continuo de medios se mejora debido a la conmutación ascendente, por ejemplo, el tamaño de la imagen presentada del flujo continuo de vídeo se hace mayor o la resolución de la imagen aumenta. De este modo, la conmutación ascendente ayuda a evitar la interrupción del flujo continuo y mejora la calidad de los medios, beneficiando ambos aspectos al usuario.

Una secuencia lógica sencilla para realizar conmutaciones ascendentes y conmutaciones descendentes se puede representar de la manera siguiente:

Si $PT > PT_{ASC}$ entonces

Realizar conmutación ascendente

por otra parte, si $PT < PT_{DESC}$

Realizar conmutación descendente

fin del si.

La selección apropiada de estos umbrales es crítica para la impresión global de los medios por parte de los usuarios. En el caso de que la conmutación descendente se realice demasiado tarde, se producirá un acontecimiento de realmacenamiento en memoria intermedia. En el caso de una conmutación ascendente que se realice demasiado tarde, el usuario recibe unos medios de menor calidad que los, por otro lado, necesarios y, como se ha indicado, se puede producir una interrupción en el flujo continuo de datos como consecuencia de un desbordamiento de la memoria intermedia. De forma similar, si se realiza una conmutación descendente antes de lo necesario, el usuario recibe unos medios de menor calidad que los, por otro lado, necesarios. Si se realiza una conmutación ascendente antes de lo necesario, puede que entonces se requiera en poco tiempo una conmutación descendente, dando como resultado fluctuaciones molestas en la calidad de los medios. Para evitar estos problemas, se pueden usar potencialmente múltiples umbrales de conmutación descendente y múltiples umbrales de conmutación ascendente. Como la longitud de reproducción se reduce hacia el vaciado de la memoria intermedia, varios de los umbrales de conmutación descendente se cruzan, activando cada uno de ellos una conmutación descendente. A la inversa, como la longitud de reproducción aumenta hacia el desbordamiento de la memoria intermedia, se cruzan varios de los umbrales de conmutación ascendente, activando cada uno de ellos una conmutación ascendente.

No obstante, después de que se haya producido una conmutación y se haya transmitido un flujo continuo con la velocidad binaria nueva, transcurre cierto tiempo antes de que la conmutación tenga algún efecto sobre la longitud de reproducción de la memoria intermedia del cliente. En primer lugar, se produce un retardo de la transmisión hasta que un primer paquete que contiene datos codificados con la velocidad nueva alcanza la memoria intermedia del cliente.

5 Durante este periodo de tiempo, la longitud de reproducción de los medios almacenados en la memoria intermedia del cliente no se ve afectada por la velocidad nueva. Por lo tanto, si la longitud de reproducción estaba aumentando hacia un posible desbordamiento de la memoria intermedia, la misma probablemente continuará aumentando. A la inversa, si la longitud de reproducción se estaba reduciendo hacia un posible vaciado de la memoria intermedia, la misma probablemente continuará reduciéndose. Además, incluso después de la llegada del primer paquete con la velocidad binaria nueva, la longitud de reproducción puede cambiar sólo lentamente al principio. Por ejemplo, puede haber todavía algunos paquetes enviados con datos con la velocidad binaria anterior que no habían sido recibidos todavía por la memoria intermedia del cliente. Por lo tanto, las condiciones de conmutación normalmente son todavía válidas y, entonces, después de una primera conmutación siguen varias conmutaciones, las cuales son con frecuencia innecesarias. En el caso de una primera conmutación descendente, se pueden realizar varias conmutaciones descendentes adicionales, dando como resultado una velocidad binaria del flujo continuo que es mucho menor que la necesaria. Frecuentemente, las conmutaciones descendentes no se detienen hasta que se ha seleccionado la velocidad binaria más baja del flujo. Este comportamiento da como resultado una calidad del flujo continuo de medios innecesariamente baja para el usuario. En el caso de una conmutación ascendente, se pueden producir varias conmutaciones ascendentes adicionales, dando como resultado una velocidad binaria del flujo continuo que es demasiado alta, frecuentemente hasta la velocidad más alta posible. Esto da como resultado una velocidad binaria del flujo que es realmente demasiado alta en comparación con el ancho de banda de la red disponible en ese momento, activándose una serie de conmutaciones descendentes.

Como consecuencia, el usuario observa variaciones frecuentes y molestas en la calidad del flujo continuo. Por otra parte, si se ha seleccionado una velocidad binaria que es realmente demasiado alta, con frecuencia las conmutaciones descendentes subsiguientes no se pueden ejecutar de una forma suficientemente rápida, dando como resultado acontecimientos molestos de realmacenamiento en memoria intermedia y congelación de la reproducción. De modo similar, si se ha seleccionado una velocidad binaria que es realmente baja, frecuentemente las conmutaciones ascendentes subsiguientes no se pueden ejecutar de una forma suficientemente rápida, dando como resultado desbordamientos molestos de la memoria intermedia y una pérdida asociada de datos. Incluso, con solamente un único umbral de conmutación ascendente y un único umbral de conmutación descendente, pueden surgir estos tipos de problemas, particularmente si los umbrales se fijan demasiado próximos entre ellos.

Pueden surgir todavía más problemas cuando se transmite un contenido de medios que tiene una velocidad binaria variable. Convencionalmente, cada versión precodificada del flujo continuo multimedia tiene una única velocidad binaria, y por lo tanto la velocidad binaria de un flujo continuo únicamente cambia si el servidor conmuta a un flujo continuo diferente que tiene una velocidad mayor o menor, como ya se ha descrito. No obstante, en algunos casos, resulta apropiado proporcionar a los flujos continuos una velocidad binaria variable, particularmente para adaptarse al almacenamiento y la transmisión de archivos de medios grandes. En otras palabras, cada versión de un flujo continuo puede tener partes con una velocidad binaria y otras partes con otra velocidad. Preferentemente, la velocidad binaria para secciones individuales de una versión particular de un flujo continuo se escoge basándose en el contenido de la sección individual. Por ejemplo, una parte del flujo continuo puede ser bastante estática, permitiendo una baja velocidad binaria para capturar adecuadamente el contenido. Seguidamente, puede ser necesaria una velocidad binaria mayor para capturar adecuadamente un contenido más dinámico. Mediante la fijación de la velocidad binaria de cada parte de un flujo continuo multimedia basándose en el contenido dinámico de esa parte del flujo continuo, se puede reducir el tamaño global del archivo aunque transportando todavía adecuadamente el contenido.

Cuando se aplican técnicas convencionales de conmutación de flujos continuos a flujos continuos de velocidad binaria variable, pueden surgir varios problemas. En particular, las velocidades binarias cambiantes del flujo continuo pueden agravar los problemas antes mencionados, dando como resultado conmutaciones todavía más frecuentes e innecesarias, lo cual provoca molestias adicionales para el usuario y, frecuentemente, un despilfarro del ancho de banda.

El documento US2004/098748 A1 da a conocer un sistema de flujos continuos que comprende un servidor y un cliente y un mecanismo de control que ajustan la transmisión de velocidad binaria de datos de un servidor según la cantidad de datos dentro de la memoria intermedia del cliente.

Por consiguiente, existe una necesidad de una técnica mejorada para controlar la conmutación de flujos continuos de datos de velocidad binaria variable con el fin de proporcionar un contenido más estable y fiable al usuario, y es en este propósito en el que se centra principalmente la invención.

60 Sumario de la invención

La invención se puede materializar en un método para controlar transmisiones de paquetes de datos de velocidad binaria variable desde un servidor a un cliente que tiene una memoria intermedia de cliente en donde el servidor conmuta entre diferentes versiones de un flujo continuo de datos de velocidad binaria variable que se están transmitiendo basándose en un estado de la memoria intermedia del cliente. Según el método, se selecciona una versión inicial de un flujo continuo de datos de velocidad binaria variable para la transmisión y se realiza un seguimiento de un valor (PT) representativo de una cantidad de datos dentro de la memoria intermedia del cliente. Se determina una velocidad binaria (BR) de una parte de datos de velocidad binaria variable todavía por transmitir dentro de la versión seleccionada

del flujo continuo. Entonces, el servidor controla conmutaciones a versiones diferentes del flujo continuo, que tienen velocidades binarias medias diferentes, basándose en el valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente en combinación con la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir.

En un ejemplo, se determina la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte secuencial de datos a transmitir. Si un $BR_{SIGUIENTE}$ supera una velocidad binaria media (BR_{MED}), entonces se realiza la etapa de control de la conmutación para retardar cualquier conmutación hacia una versión del flujo continuo que tenga una velocidad binaria en general mayor, es decir, se retardan las conmutaciones ascendentes.

En otro ejemplo, si $BR_{SIGUIENTE}$ supera la velocidad binaria (BR_{MED}), se realiza la etapa de control de la conmutación para acelerar cualquier conmutación hacia una versión del flujo continuo que tenga una velocidad binaria en general menor, es decir, se aceleran las conmutaciones descendentes.

Todavía en otro ejemplo, si $BR_{SIGUIENTE}$ está por debajo de la velocidad binaria (BR_{MED}), se realiza la etapa de control de conmutación para acelerar cualquier conmutación hacia una versión del flujo continuo que tenga una velocidad binaria en general mayor, es decir, se aceleran las conmutaciones ascendentes.

En una implementación preferida, el servidor controla conmutaciones a versiones diferentes del flujo continuo basándose en un estado actual de la memoria intermedia del cliente mediante la aplicación de uno o más umbrales a datos ya en la memoria intermedia del cliente. La etapa de controlar la conmutación se realiza ajustando dinámicamente el umbral o umbrales basándose en las velocidades binarias (BR) de la parte de datos de velocidad binaria variable a transmitir y, a continuación, determinando si conmutar a una versión diferente del flujo continuo mediante la aplicación del umbral o umbrales ajustables al valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente.

En un ejemplo de la implementación preferida, el umbral o umbrales incluyen un umbral de conmutación ascendente (PT_{ASC}) y un umbral de conmutación descendente (PT_{DESC}). La etapa de ajustar dinámicamente el umbral o umbrales se realiza determinando una velocidad binaria media (BR_{MED}) de los datos de velocidad binaria variable y a continuación ajustando selectivamente los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} y PT_{DESC}) basándose en una comparación de la velocidad binaria ($BR_{SIGUIENTE}$) de una parte siguiente de datos de velocidad binaria variable a transmitir con la velocidad binaria media (BR_{MED}) de los datos de velocidad binaria variable. La etapa de ajustar selectivamente los umbrales de conmutación ascendente y conmutación descendente incluyen la etapa de aumentar los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} y PT_{DESC}) si la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de datos de velocidad binaria variable supera la velocidad binaria media (BR_{MED}). La etapa de ajustar selectivamente los umbrales de conmutación ascendente y de conmutación descendente incluye también la etapa de reducir el umbral de conmutación ascendente (PT_{ASC}) si la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de los datos de velocidad binaria variable están por debajo de la velocidad binaria media (BR_{MED}) y si el umbral de conmutación ascendente (PT_{ASC}) supera el umbral de conmutación descendente (PT_{DESC}). En otras palabras, PT_{ASC} no se ajusta en sentido descendente si en ese caso cayera por debajo de PT_{DESC} .

En la implementación preferida, la etapa de aumentar los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} y PT_{DESC}) si la velocidad binaria (BR) de la siguiente parte de datos de velocidad binaria variable supera la velocidad binaria media (BR_{MED}) se realiza determinando la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de datos de velocidad binaria variable y a continuación calculando una relación (F) de la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de datos de velocidad binaria variable con respecto a la velocidad binaria media (BR_{MED}). Entonces, los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} y PT_{DESC}) se multiplican por la relación (F) para aumentar de este modo los umbrales. La etapa de reducir el umbral de conmutación ascendente (PT_{ASC}) se realiza también determinando la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de datos de velocidad binaria variable y a continuación calculando una relación (F) de la velocidad binaria ($BR_{SIGUIENTE}$) de la siguiente parte de datos de velocidad binaria variable con respecto a la velocidad binaria media (BR_{MED}). Entonces, el umbral de conmutación ascendente (PT_{ASC}) se multiplica por la relación (F) para reducir de este modo el umbral. Entonces, se selecciona el mayor del umbral de conmutación ascendente reducido ($PT_{ASC} * F$) y el umbral de conmutación descendente (PT_{DESC}) para ser usado como nuevo umbral de conmutación ascendente.

En varias implementaciones, los valores representativos de cantidades de datos dentro de la memoria intermedia del cliente son representativos de longitudes de reproducción (PT) de los datos o niveles de llenado de la memoria intermedia. La velocidad binaria media (BR_{MED}) puede ser representativa de una velocidad binaria media durante la versión completa del flujo continuo que se esté transmitiendo o puede ser representativa, en cambio, de una velocidad binaria media de datos que ya han sido transmitidos dentro de la versión del flujo que se esté transmitiendo.

Dependiendo de la implementación, el cliente puede ser un terminal móvil de comunicaciones tal como un teléfono móvil, y de forma adicional, o alternativa, el servidor puede estar integrado en un terminal móvil de comunicaciones de manera que el enlace entre el servidor y la red puede ser inalámbrico. Adicionalmente, el método según la presente invención puede ser realizado por uno o más nodos de red intermediarios (tales como proxies) dispuestos entre el servidor y el cliente. El método también se puede utilizar en arquitecturas que tengan una pluralidad de flujos continuos de datos que estén siendo almacenados en memoria intermedia por la memoria intermedia del cliente (o múltiples memorias intermedias de cliente dependiendo de la configuración de este último).

La invención también se puede materializar en un producto de programa de ordenador, el cual se puede almacenar en un soporte de grabación legible por ordenador, que comprende partes de código de programa para realizar cualquiera de las etapas de los métodos anteriores cuando el producto de programa de ordenador se ejecuta en un sistema de ordenador.

La invención puede comprender además un aparato que comprenda un procesador de ordenador y una memoria acoplada al procesador, en donde la memoria esté codificada con uno o más programas que pueden realizar cualquiera de las etapas de los métodos anteriores.

Todavía en otra realización, la invención se refiere a un aparato para controlar transmisiones de paquetes de datos de velocidad binaria variable desde un servidor a un cliente que tiene una memoria intermedia de cliente en donde el servidor conmuta entre versiones diferentes de un flujo continuo de datos de velocidad binaria variable que estén siendo transmitidos basándose en un estado de la memoria intermedia del cliente. El aparato comprende: un controlador anticipatorio de la transmisión del flujo continuo de velocidad binaria variable para seleccionar una versión inicial de un flujo continuo de datos de velocidad binaria variable para la transmisión; un monitor de la memoria intermedia del cliente para realizar un seguimiento de un valor representativo de una cantidad de datos dentro de la memoria intermedia de cliente; una unidad de determinación de la velocidad binaria variable para determinar una velocidad binaria (BR) de una parte de datos de velocidad binaria variable todavía por transmitir dentro de la versión seleccionada del flujo continuo; y en donde el controlador anticipatorio de la transmisión del flujo continuo de velocidad binaria variable controla entonces las conmutaciones a versiones diferentes del flujo continuo, que tienen velocidades binarias medias diferentes, basándose en el valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente en combinación con la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir.

El aparato puede estar configurado como un componente fijo o móvil de una red, tal como un servidor de red y/o un terminal inalámbrico. Adicionalmente, el aparato puede estar constituido por un nodo de red intermediario, tal como un proxy.

Breve descripción de los dibujos

A continuación se describirá la invención en referencia a realizaciones a título de ejemplo ilustradas en las figuras, en las que:

la Fig. 1 es un diagrama esquemático de un sistema de comunicaciones útil para entender y ejecutar la invención;

la Fig. 2 es un diagrama de flujo de proceso que proporciona una visión general de una realización del método de la invención;

la Fig. 3 es una gráfica que ilustra varias versiones de un flujo continuo de datos de velocidad binaria variable, ilustrando particularmente diferentes velocidades binarias medias de las diversas versiones del flujo continuo;

la Fig. 4 es un diagrama de bloques de un flujo continuo individual de datos de velocidad binaria variable, ilustrando particularmente velocidades binarias de varias partes del flujo continuo;

la Fig. 5 es un diagrama de flujo de proceso que ilustra una ejecución ilustrativa de la secuencia lógica de velocidad binaria variable de la invención;

la Fig. 6 es un diagrama de bloques de una memoria intermedia de cliente, que ilustra particularmente varios umbrales utilizados por la invención;

la Fig. 7 es un diagrama de bloques que ilustra una ejecución de aparato ilustrativa de un componente de servidor de la invención;

Descripción detallada de las realizaciones preferidas

En la siguiente descripción, con fines explicativos y no limitativos, se exponen detalles específicos, tales como secuencias particulares de etapas y varias configuraciones, con el fin de proporcionar una comprensión completa de la invención. Se pondrá de manifiesto para los expertos en la materia que la invención se puede llevar a la práctica en otras realizaciones que se desvíen con respecto a estos detalles específicos. Además, aquellos expertos en la materia apreciarán que las funciones explicadas en el presente documento a continuación se pueden ejecutar usando software que funcione conjuntamente con un microprocesador programado u ordenador de propósito general, y/o usando un circuito integrado de aplicación específica (ASIC). Se apreciará también que aunque la invención se describe principalmente en forma de un método, también se puede materializar en un producto de programa de ordenador, así como un sistema o aparato que comprenda un procesador de ordenador y una memoria acoplada al procesador, en donde la memoria esté codificada con uno o más programas que puedan realizar los métodos dados a conocer en el presente documento.

La Fig. 1 ilustra una arquitectura de muestra 100 que se puede usar en relación con la invención, que incluye un servidor 105 que puede estar acoplado a un cliente 115 a través de una vía de comunicaciones tal como una red IP 110. El servidor incluye un módulo 120 de contenido de medios que accede a y transmite ciertos contenidos de medios (por ejemplo, archivos de datos multimedia) a través de un módulo RTP/UDP 125 usando una normativa de flujos continuos tal como el RTP (Protocolo de Transporte en Tiempo Real) a través de UDP u otro protocolo de transporte de datos para gestionar la transmisión en tiempo real de datos multimedia (con un protocolo de capa de transporte tal como el Protocolo de Datagrama de Usuario - UDP). Los paquetes se transmiten a una red pública 130 (por ejemplo, Internet, aunque no es necesaria una red pública externa cuando el servidor está acoplado directamente a la red 135 del operador) que entrega los paquetes a una red 135 del operador, tal como una red “alámbrica” del operador de comunicaciones de móviles, acoplada al mismo.

La red 135 del operador incluye una red central 140 que proporciona enlaces de comunicación entre el servidor 105 y el cliente 115. La red central 140, que opcionalmente puede tener una memoria intermedia, proporciona los paquetes recibidos desde el módulo RTP/UDP 125 para su almacenamiento en una memoria intermedia dentro de una red de acceso de radiocomunicaciones (RAN) 145 (tal como una memoria intermedia en el SGSN ó en el RNC) antes de su transmisión por parte de un transmisor inalámbrico 150. Las memorias intermedias de la red central 140 (si se utiliza almacenamiento en memoria intermedia) y la RAN 145 están dispuestas en serie y constituyen una memoria intermedia de red.

El cliente 115 recibe los paquetes transmitidos por el transmisor inalámbrico 150 en una memoria intermedia 155 de cliente. Los paquetes se transfieren desde la memoria intermedia 155 de cliente hacia un módulo RTP/UDP 160 para su entrega a y su uso por parte del módulo 165 de aplicación de medios (o reproductor multimedia). A efectos de la presente solicitud, la expresión “velocidad de transmisión de paquetes” se referirá a la velocidad de transmisión de paquetes desde el servidor 105 a la red IP 110, la expresión “velocidad del enlace” se referirá a la velocidad de transmisión de paquetes desde la red IP 110 al cliente 115, y la expresión “velocidad del contenido” se referirá a la velocidad con la que se transfieren los datos desde la memoria intermedia 115 de cliente hacia el módulo 165 de aplicación de medios para su reproducción por parte del módulo 165 de aplicación de medios.

En referencia a la Fig. 2, se ilustra una realización 200 del método de la invención que se puede realizar, por ejemplo, por medio del sistema de la Fig. 1. El método es implementado preferentemente por el servidor, tal como un servidor 105 de la Fig. 1, aunque puede ser implementado por cualquier otro componente apropiado de la red. En las siguientes descripciones, se supondrá que un servidor implementa las etapas del método. Comenzando en la etapa 202, el servidor selecciona una versión inicial de un flujo continuo de datos de velocidad binaria variable para su transmisión. Se selecciona la versión inicial del flujo continuo, según técnicas por otro lado convencionales, de entre un grupo de versiones diferentes del flujo continuo que tienen velocidades de transmisión precodificadas en general diferentes. Por ejemplo, un primer flujo continuo puede comenzar con una velocidad binaria de 1 megabit (Mbit)/segundo, mientras que un segundo flujo continuo puede comenzar con una velocidad binaria de 2 Mbits/segundo. Como el propio flujo continuo tiene una velocidad binaria variable, cada versión del flujo continuo tiene de forma similar una velocidad binaria variable. Por ello, sea cual sea la versión del flujo continuo que se seleccione, la velocidad binaria de datos codificada dentro del flujo continuo cambiará. Por ejemplo, la primera versión del flujo continuo puede conmutar a una velocidad binaria de 2 Mbits/segundo después de diez segundos, mientras que la segunda versión del flujo continuo puede conmutar a una velocidad binaria de 3 megabits Mbits/segundo después de diez segundos, es decir, ambas versiones del flujo continuo tienen velocidades binarias precodificadas que aumentan en 1 Mbit/segundo después de diez segundos.

En la Fig. 3 se muestran tres versiones a título de ejemplo de un flujo continuo de bits individual. Cada versión del flujo continuo tiene una velocidad binaria que varía con el tiempo. Las versiones están sincronizadas todas ellas entre sí, es decir, todas las velocidades binarias aumentan al mismo tiempo o todas se reducen al mismo tiempo. No obstante, las velocidades binarias medias de las diversas versiones del flujo continuo difieren entre sí. El servidor realiza conmutaciones ascendentes y conmutaciones descendentes conmutando entre las diferentes versiones de flujo continuo. En el ejemplo de la Fig. 3, se muestran únicamente tres versiones del flujo continuo. Típicamente, se almacenan previamente más versiones. Además, en el ejemplo, la velocidad binaria más baja de la versión 205 de velocidad binaria más alta del flujo continuo está a una velocidad mayor que la velocidad binaria más alta de la siguiente versión 207 del flujo continuo. De modo similar, la velocidad binaria más baja de la versión 205 del flujo continuo está a una velocidad mayor que la velocidad binaria más alta de la versión 209. No obstante, no es necesario que sea así. Frecuentemente existe por lo menos cierto solapamiento en las velocidades binarias. Además, en el ejemplo, los cambios de velocidad se muestran como uniformes, es decir, la velocidad aumenta uniformemente hasta un pico y a continuación se reduce uniformemente. Tampoco es necesario que eso sea así. En muchos ejemplos, las velocidades binarias cambian de forma discontinua, es decir, la velocidad salta de un nivel (tal como 1 Mbit/segundo) a un nivel diferente (tal como 2 Mbit/segundo).

Volviendo a la Fig. 2, en la etapa 202 se transmite la versión seleccionada del flujo continuo desde el servidor al cliente tomando datos precodificados de la versión del flujo continuo y colocando esos datos precodificados en paquetes de datos, que se transmiten hacia el cliente a una velocidad de transmisión de paquetes predeterminada. Obsérvese que cualquier cambio en la velocidad binaria de los datos de velocidad binaria variable dentro de un flujo continuo que se esté transmitiendo no requiere típicamente un cambio en la velocidad de transmisión de paquetes, que se determina basándose en consideraciones del ancho de banda. De modo similar, la elección de una versión del flujo continuo por encima de otra versión no conlleva típicamente ningún cambio en la velocidad de transmisión de paquetes.

En la etapa 204, el servidor realiza un seguimiento de un valor representativo de la cantidad de datos que están actualmente dentro de la memoria intermedia del cliente. Esto se puede realizar también según técnicas por otro lado convencionales. En un ejemplo, si la memoria intermedia del cliente se implementa según protocolos RTCP, el servidor recibe una realimentación desde la memoria intermedia del cliente, que incluye los campos de datos NSN/OBSN y HRSN a partir de los cuales se determina la cantidad de datos en la memoria intermedia del cliente. Como se indica, la cantidad de datos en la memoria intermedia del cliente se puede representar, por ejemplo, en términos de una longitud de reproducción (PT), un nivel de llenado de la memoria de cliente, u otro valor apropiado. En las siguientes descripciones se describirán ejemplos usando la longitud de reproducción (PT).

En la etapa 206, el servidor determina la velocidad binaria (BR) de una parte de los datos de velocidad binaria variable todavía por transmitir. Preferentemente, es la siguiente parte secuencial de datos a transmitir la que examina el servidor, es decir, el servidor determina un valor ($BR_{SIGUIENTE}$) representativo de la velocidad binaria de la siguiente parte de datos a transmitir. La siguiente parte de datos se puede definir, por ejemplo, en términos de un número predeterminado de segundos equivalentes a datos a transmitir o se puede definir como aquella parte de datos a partir de los datos que se están transmitiendo en ese momento hasta que se produzca un cambio preestablecido sucesivo en los datos. Por ejemplo, si el flujo continuo precodificado de datos de velocidad binaria variable va a mantener su velocidad binaria actual durante los siguientes 12 segundos antes de conmutar a otra velocidad binaria, entonces la siguiente parte de datos comprende los siguientes 12 segundos de datos.

La Figura 4 ilustra una versión 208 a título de ejemplo de un flujo continuo de datos de velocidad binaria variable que están siendo transmitidos. El flujo continuo incluye partes de velocidades binarias diferentes, indicadas $BR_{\#1}$, $BR_{\#2}$, $BR_{\#3}$, $BR_{\#4}$, etcétera. Tanto $BR_{\#1}$ como $BR_{\#3}$ pueden ser, por ejemplo 1 Mbit/segundo, mientras que tanto $BR_{\#2}$ como $BR_{\#4}$ pueden ser, por ejemplo, 2 Mbits/segundo. Como puede observarse, las partes pueden ser de longitudes, es decir, duraciones diferentes. En el ejemplo, la flecha 210 indica el punto dentro del flujo continuo correspondiente a datos que se están transmitiendo en ese momento. Por lo tanto, la siguiente parte de datos es datos que comienzan en el punto 210. La velocidad de esos datos es $BR_{\#2}$. Si la siguiente parte de datos se define en términos de esa parte de datos a partir de los datos que se están transmitiendo en ese momento hasta que se produzca un cambio preestablecido sucesivo en los datos, entonces $BR_{SIGUIENTE}$ es simplemente $BR_{\#2}$. Por el contrario, si la siguiente parte de datos se define en términos de un número predeterminado de segundos equivalentes a datos a transmitir, entonces la siguiente parte de datos puede abarcar dos o más velocidades binarias diferentes. En ese caso, el servidor se puede configurar simplemente para seleccionar la primera de esas velocidades binarias para ser usada como $BR_{SIGUIENTE}$ ó, en cambio, se puede configurar para calcular la velocidad binaria media BR_{MED} durante ese periodo de tiempo predeterminado, para ser usada como $BR_{SIGUIENTE}$. En cualquier caso, en la etapa 206 de la Fig. 2 se determina un valor representativo de la velocidad binaria de alguna parte de datos todavía por transmitir.

A continuación, en la etapa 212 de la Fig. 2, el servidor controla conmutaciones a versiones diferentes del flujo continuo, que presentan las diferentes velocidades binarias medias, basándose en el valor (PT) representativo de la cantidad de datos actualmente dentro de la memoria intermedia del cliente y basándose en la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir, por ejemplo, el servidor controla conmutaciones ascendentes y conmutaciones descendentes basándose tanto en el PT como en el $BR_{SIGUIENTE}$. Teniendo en cuenta la velocidad binaria de datos todavía por transmitir, además del PT, el servidor es capaz de evitar muchas de las conmutaciones de velocidad innecesarias que se producen en sistemas convencionales que utilizan únicamente umbrales de conmutación de velocidad fijos.

Preferentemente, si $BR_{SIGUIENTE}$ supera a BR_{MED} , cualquier conmutación ascendente debida a un cambio en el estado de la memoria intermedia del cliente se retarda. En otras palabras, si cualquier aumento de la longitud de reproducción dentro de la memoria intermedia del cliente hubiera activado en otras circunstancias otra conmutación a una versión diferente del flujo continuo que presenta una velocidad de transmisión en general mayor, esa conmutación se retarda. En cambio, si el servidor realizara una conmutación ascendente antes de un punto en el flujo continuo en el que se esperaba que la velocidad binaria variable aumentase de todos modos, la conmutación ascendente resultaría probablemente contraproducente y podría necesitar una conmutación descendente compensatoria. En relación con esto, al realizar una conmutación ascendente antes de un punto en el flujo continuo en el que aumenta la velocidad binaria del flujo continuo de velocidad binaria variable, se producen por lo tanto dos aumentos de la velocidad binaria, uno tras otro. Como consecuencia, la velocidad binaria resulta entonces probablemente mayor que la necesaria activando probablemente una conmutación descendente compensatoria para evitar un posible vaciado de la memoria intermedia. En cambio, al retardar una conmutación ascendente en circunstancias en las que se espera que la velocidad binaria de datos de velocidad binaria variable aumente de todos modos, el servidor probablemente evita tanto una conmutación ascendente innecesaria como una subsiguiente conmutación descendente compensatoria, proporcionando de este modo un nivel más consistente de calidad de los medios para el usuario final.

También preferentemente, si $BR_{SIGUIENTE}$ supera a BR_{MED} , se acelera la conmutación descendente debido a un cambio en el estado de la memoria intermedia del cliente. En otras palabras, si una reducción en la longitud de reproducción dentro de la memoria intermedia del cliente hubiera activado en otras circunstancias una conmutación a una versión diferente del flujo continuo que presentase una velocidad de transmisión en general menor, esa conmutación se acelera si se espera que la velocidad binaria variable aumente. Al acelerar la conmutación descendente en circunstancias en las que se espera que aumente la velocidad binaria de los datos que se están transmitiendo, el servidor ayuda de este modo a evitar un posible acontecimiento de realmacenamiento en memoria intermedia. Por otro lado, si el servidor no previese el aumento de la velocidad binaria dentro de los datos de velocidad binaria variable, una conmutación

descendente activada por una caída de la longitud de reproducción por debajo del umbral de conmutación descendente (PT_{DESC}) fijo convencional podría producirse demasiado tarde para evitar el realmacenamiento en memoria intermedia, dado que la velocidad binaria creciente del propio flujo continuo acelerará el vaciado de la memoria intermedia.

5 Preferentemente, si $BR_{SIGUIENTE}$ está en cambio por debajo de BR_{MED} , se acelera cualquier conmutación ascendente debida a un cambio en el estado de la memoria intermedia del cliente. En otras palabras, si un aumento en la longitud de reproducción dentro de la memoria intermedia del cliente hubiera activado en otras circunstancias una conmutación a una versión diferente del flujo continuo que presentase una velocidad de transmisión en general mayor, esa conmutación se acelera si se espera que la velocidad binaria variable se reduzca. Al acelerar la conmutación ascendente en
10 circunstancias en las que se espera que se reduzca la velocidad binaria de los datos que se están transmitiendo, el servidor ayuda de este modo a evitar un posible acontecimiento de desbordamiento. Por otro lado, si el servidor no previese la reducción de la velocidad binaria dentro de los datos de velocidad binaria variable, una conmutación ascendente activada por una superación, por parte de la longitud de reproducción, del umbral de conmutación ascendente (PT_{ASC}) fijo convencional podría producirse demasiado tarde para evitar el desbordamiento de memoria intermedia dado que
15 la velocidad binaria decreciente del propio flujo continuo acelerará el desbordamiento de la memoria intermedia. Por otra parte, la conmutación ascendente acelerada proporciona de forma más inmediata una mejora de la calidad de los medios para el usuario.

No obstante, obsérvese que cuando $BR_{SIGUIENTE}$ está por debajo de BR_{MED} , preferentemente no se retarda (ni se
20 acelera) ninguna conmutación descendente debida a un cambio en el estado de la memoria de cliente. Antes bien, en ese caso, las conmutaciones descendentes se activan preferentemente basándose en el umbral de conmutación descendente (PT_{DESC}) fijo convencional. Aunque una conmutación descendente se podría retardar potencialmente considerando el hecho de que de todos modos se espera que la velocidad binaria de los datos de velocidad binaria variable se reduzca en un tiempo breve, esta opción no se realiza en la implementación preferida de la invención para evitar el riesgo de
25 vaciado de la memoria intermedia.

Volviendo ahora a la Fig. 5, se describirá a continuación una ejecución preferida de la secuencia lógica para la velocidad binaria variable destinada a ser usada en la etapa dos 212 de la Fig. 2. Comenzando en la etapa 300, el servidor determina valores para PT_{ASC} y PT_{DESC} . PT_{ASC} y PT_{DESC} pueden ser umbrales de conmutación ascendente y
30 de conmutación descendente fijos, predeterminados, y se pueden fijar según técnicas por otro lado convencionales. En la etapa 302, el servidor determina valores para $BR_{SIGUIENTE}$ y BR_{MED} . $BR_{SIGUIENTE}$ se puede determinar examinando las velocidades binarias codificadas asociadas a datos a transmitir, que están almacenados dentro del servidor o que son accesibles de otra manera por el servidor. En un ejemplo BR_{MED} se calcula examinando un registro de las velocidades binarias codificadas de datos ya transmitidos (a lo largo de las duraciones de tiempo durante las que se transmitieron
35 paquetes que contenían los datos codificados a las diversas velocidades binarias). Alternativamente, para calcular la media real se puede usar la aritmética rutinaria. En un segundo ejemplo, se puede calcular un valor de BR_{MED} para el flujo continuo completo y el mismo se puede almacenar de antemano (como los flujos continuos individuales se codifican antes de que se inicie la sesión de emisión del flujo continuo, se pueden determinar valores individuales de BR_{MED} para los flujos continuos individuales antes de la reproducción). En otras palabras, en ese segundo ejemplo,
40 BR_{MED} no es la velocidad binaria media de solamente aquella parte de datos que ya ha sido transmitida, sino que representa, en cambio, la media durante el flujo continuo completo, es decir, BR_{MED} es la velocidad binaria media ilustrada, por ejemplo, en la Fig. 3.

En la etapa 304, el servidor calcula un valor F, que es representativo de una relación de $BR_{SIGUIENTE}$ con respecto a
45 BR_{MED} , es decir, $F = BR_{SIGUIENTE}/BR_{MED}$. Por lo tanto, si $BR_{SIGUIENTE}$ supera a BR_{MED} , F es entonces mayor que 1,0. Si $BR_{SIGUIENTE}$ supera a BR_{MED} , F es entonces menor que 1,0. En los casos en los que $BR_{SIGUIENTE}$ es igual a BR_{MED} , F es entonces igual a 1,0. Si todavía no se han transmitido datos (de tal manera que no existe ningún valor actual de BR_{MED}), entonces F simplemente se fija en 1,0. Si no quedan datos por transmitir dentro de un flujo continuo actual de datos, de tal manera que no existe un valor actual para $BR_{SIGUIENTE}$, entonces F se vuelve a fijar también preferentemente en
50 1,0 para ser usado en relación con un flujo continuo sucesivo de datos a transmitir.

En la etapa 306, el servidor fija un umbral de conmutación ascendente ajustable ($PT_{ASC-AJU}$) de manera que sea igual a PT_{ASC} multiplicado por F, es decir, $PT_{ASC} * F$, y también fija un umbral de conmutación descendente ajustable ($PT_{DESC-AJU}$) de manera que sea igual a PT_{DESC} multiplicado por F, es decir, $PT_{DESC} * F$. Por lo tanto, si $BR_{SIGUIENTE}$
55 supera a BR_{MED} , entonces $PT_{ASC-AJU}$ es mayor que PT_{ASC} y $PT_{DESC-AJU}$ es también mayor que PT_{DESC} . A la inversa, si $BR_{SIGUIENTE}$ está por debajo de BR_{MED} , entonces $PT_{ASC-AJU}$ es menor que PT_{ASC} y $PT_{DESC-AJU}$ es también menor que PT_{DESC} .

Los umbrales se ilustran en la Fig. 6 que proporciona una representación en diagrama de bloques de la memoria intermedia 115 de cliente de la Fig. 1. En el ejemplo de la Fig. 6, los umbrales ajustables son mayores que los umbrales fijos correspondientes. Obsérvese que potencialmente pueden surgir circunstancias en las que $PT_{ASC-AJU}$ se pueda calcular de manera que resulte mayor que el valor máximo de la memoria intermedia de cliente (MAX). Esto puede ocurrir si $BR_{SIGUIENTE}$ es bastante mayor que BR_{MED} , generando un valor alto para F. Si esto es así, entonces $PT_{ASC-AJU}$ simplemente se fija igual a MÁX o a algún otro valor por defecto, menor. En el caso altamente improbable
65 de que $PT_{DESC-AJU}$ también se calcule de manera que resulte mayor que MÁX debido a un valor extremadamente alto para F, entonces $PT_{DESC-AJU}$ también se fija preferentemente en algún valor por defecto, que sea menor que el valor por defecto al que se fija $PT_{ASC-AJU}$, garantizando de este modo que $PT_{DESC-AJU}$ sigue siendo menor $PT_{ASC-AJU}$.

ES 2 313 323 T3

Volviendo a la Fig. 5, en la etapa 308, el servidor comienza a determinar valores para PT, es decir, la longitud de reproducción actual de datos ya contenidos dentro de la memoria intermedia de cliente, de la que se ha realizado un seguimiento en la etapa 204 de la Fig. 2. Si, en la etapa de decisión 310, PT es mayor que PT_{DESC} y PT también es mayor que $PT_{ASC-AJU}$, entonces en la etapa 312 se activa una conmutación ascendente. En caso negativo, entonces se realiza la etapa de decisión 314 en la que, si PT es menor que PT_{DESC} ó PT es menor que $PT_{DESC-AJU}$, entonces en la etapa 316 se activa una conmutación descendente.

La secuencia lógica de las etapas de decisión 310 y 316 se puede representar de la manera siguiente:

Si $PT > PT_{DESC}$ Y $PT > PT_{ASC-AJU}$ entonces

Realizar conmutación ascendente

por otra parte

si $PT < PT_{DESC}$ Ó $PT < PT_{DESC-AJU}$

Realizar conmutación descendente

fin del si

en donde, tal como se ha indicado,

$$PT_{ASC-AJU} = PT_{ASC} * F \text{ y}$$

$$PT_{DESC-AJU} = PT_{DESC} * F.$$

Por lo tanto, se activa una conmutación ascendente si la longitud de reproducción (PT) actual de la memoria intermedia del cliente supera el umbral de conmutación ascendente ajustable ($PT_{ASC-AJU}$), suponiendo que PT también supere a PT_{DESC} . Esta última condición evita conmutaciones ascendentes inadecuadas en circunstancias en las que pueda resultar más adecuada una conmutación descendente. Al activar conmutaciones ascendentes basándose en el umbral de conmutación ascendente ajustable, en lugar del umbral de conmutación ascendente fijo, el servidor tiene en cuenta de esta manera el estado actual de la memoria intermedia del cliente (según se representa mediante PT) al mismo tiempo que también prevé cambios en la velocidad binaria dentro del flujo continuo de datos de velocidad binaria variable (a través del ajuste del umbral de conmutación ascendente). Esto ayuda a evitar otras conmutaciones ascendentes inadecuadas.

Considérese un ejemplo en el que el umbral de conmutación ascendente ajustable es exactamente igual inicialmente al umbral de conmutación ascendente fijo. Por lo tanto, si la longitud de reproducción de la memoria intermedia del cliente supera ese nivel de umbral, se realiza una conmutación ascendente. Entonces, si la velocidad binaria de los datos de velocidad binaria variable se espera que aumente, se hará aumentar el umbral de conmutación ascendente ajustable de modo que sea mayor que el umbral fijo. En ese caso, únicamente se realizará una conmutación ascendente adicional si la longitud de reproducción supera el nuevo valor de umbral mayor. En otras palabras, resulta más difícil que una conmutación ascendente sea activada, ya que se debe superar un valor de umbral mayor, retardando de este modo conmutaciones ascendentes adicionales en circunstancias en las que dicho retardo está garantizado, tal como se ha descrito anteriormente en relación con la Fig. 1. Considérese ahora un ejemplo en el que el umbral de conmutación ascendente ajustable se fija de nuevo inicialmente igual al umbral de conmutación ascendente fijo, pero en el que, en cambio, se espera que la velocidad binaria de los datos de velocidad binaria variable disminuya. En tal caso, el umbral de conmutación ascendente ajustable será entonces menor que el umbral fijo. En ese caso, resulta más sencillo que una conmutación ascendente sea activada, acelerando de este modo conmutaciones ascendentes adicionales en circunstancias en las que esté garantizada una de ellas, tal como también se ha descrito anteriormente.

Volviendo ahora a las conmutaciones descendentes, se activa una conmutación descendente si la longitud de reproducción (PT) actual de la memoria intermedia del cliente cae por debajo bien del umbral fijo PT_{DESC} o bien del umbral de conmutación descendente ajustable ($PT_{DESC-AJU}$). En otras palabras, se activa una conmutación descendente si PT cae por debajo del mayor de los dos umbrales de conmutación descendente. El umbral fijo se sigue usando para activar una conmutación descendente con el fin de evitar el retardo de una conmutación descendente para ayudar a evitar acontecimientos de realmacenamiento en memoria intermedia. No obstante, se puede acelerar una conmutación descendente, lo cual se produce si PT cae por debajo del umbral de conmutación descendente ajustable. Como se ha indicado anteriormente, puede resultar que el umbral de conmutación descendente ajustable esté bien por encima o bien por debajo del umbral de conmutación descendente fijo. Si el mismo está por debajo del umbral fijo, resulta superfluo ya que el umbral fijo se usa de todos modos para activar inmediatamente una conmutación descendente. No obstante, si el umbral de conmutación descendente ajustable supera al umbral de conmutación descendente fijo, el umbral de conmutación descendente ajustable puede activar entonces una conmutación descendente acelerada, es decir, resulta

ES 2 313 323 T3

más sencillo que una conmutación descendente sea activada, acelerando de este modo conmutaciones descendentes adicionales en circunstancias en las que esto esté garantizado, tal como se ha descrito también anteriormente.

5 Si no se cumple ninguna de las condiciones de las etapas de decisión 310 y 314, entonces el procesamiento vuelve a la etapa 302, en la que se actualizan los valores de $BR_{SIGUIENTE}$ y BR_{MED} para reflejar cualquier cambio de los mismos, se ajustan los valores ajustables y se introduce el último valor correspondiente a PT para aplicarlo a los diversos valores de umbral.

10 De este modo, se ha descrito una ejecución de método a título de ejemplo correspondiente a una técnica para ajustar la velocidad de transmisión global de datos en un sistema basado en paquetes mediante conmutación entre diferentes versiones de un flujo continuo precodificado. Típicamente, la velocidad de transmisión de los paquetes no se cambia, ya que la misma depende del ancho de banda del enlace de comunicación. No obstante, en otras implementaciones, la velocidad de transmisión de los paquetes también se puede cambiar usando, por ejemplo, técnicas adaptativas.

15 Aunque la invención se ha descrito principalmente en referencia a ejecuciones del método, las ejecuciones del aparato también forman parte de la invención. La Figura 7 ilustra, a un alto nivel, una ejecución del aparato a título de ejemplo. Resumiendo, el componente 400 de red, que puede formar parte del servidor 105 de la Fig.1, incluye un controlador anticipatorio (402) de la transmisión de flujos continuos de velocidad binaria variable para seleccionar (202) una versión inicial de un flujo continuo de datos de velocidad binaria variable para su transmisión. Un monitor (404) de la memoria intermedia de cliente realiza un seguimiento de un valor representativo de una cantidad de datos dentro de la memoria intermedia de cliente. Una unidad de determinación (406) de la velocidad binaria variable determina una velocidad binaria (BR) de una parte de datos de velocidad binaria variable todavía por transmitir dentro de la versión seleccionada del flujo continuo. Entonces, el controlador anticipatorio (402) de la transmisión de flujos continuos de velocidad binaria variable controla conmutaciones a versiones diferentes del flujo continuo, que presentan velocidades binarias medias diferentes, basándose en el valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente en combinación con la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir, es decir, el controlador 402 prevé cambios en la velocidad binaria del flujo continuo de velocidad binaria variable y controla de forma correspondiente las conmutaciones ascendentes y las conmutaciones descendentes.

30 Como puede ser apreciado por aquellos con conocimientos habituales en la materia, la presente invención y las técnicas asociadas a la misma proporcionan una mejora de la experiencia percibida por el usuario final para aplicaciones tales como flujos continuos multimedia al evitar desbordamientos de la memoria intermedia del cliente. Además, los profesionales expertos apreciarán también que existen muchas técnicas diferentes que pueden ser usadas para determinar niveles de llenado de la memoria intermedia del cliente, incluyendo estimaciones basadas en datos dentro de RRs e Informes del Emisor, y que la presente invención se puede implementar en paralelo con una pluralidad de flujos continuos de paquetes de datos que se estén almacenando simultáneamente en memoria intermedia para su transmisión hacia uno o más clientes.

40 Aquellos con conocimientos habituales en la materia apreciarán además que la invención puede ser implementada en y por parte de varios tipos de componentes de red, tales como terminales de red, nodos de red, y similares. En particular, la invención se puede llevar a la práctica por terminales móviles, proxies (que podrían dividir el trayecto de transmisión) y terminales fijos.

45 Aunque la invención se ha descrito con respecto a realizaciones particulares, aquellos expertos en la materia reconocerán que la invención no se limita a las realizaciones específicas descritas e ilustradas en el presente documento. Por lo tanto, aunque la invención se ha descrito en relación con sus realizaciones preferidas, debe entenderse que esta exposición es únicamente ilustrativa. Por consiguiente, se pretende que la invención quede limitada únicamente por el alcance de las reivindicaciones adjuntas a la misma.

50

55

60

65

REIVINDICACIONES

1. Método para controlar transmisiones por paquetes de datos de velocidad binaria variable desde un servidor (105) a un cliente (115) que tiene una memoria intermedia (155) de cliente, en el que el servidor conmuta entre versiones diferentes de un flujo continuo de datos de velocidad binaria variable que se están transmitiendo basándose en un estado de la memoria intermedia del cliente, comprendiendo el método las etapas de:
 - seleccionar (202) una versión inicial de un flujo continuo de datos de velocidad binaria variable para su transmisión;
 - realizar (204) un seguimiento de un valor (PT) representativo de una cantidad de datos dentro de la memoria intermedia del cliente;
 - determinar (206) una velocidad binaria (BR) de una parte de datos de velocidad binaria variable todavía por transmitir dentro de la versión seleccionada del flujo continuo; y
 - controlar conmutaciones (212) a versiones diferentes del flujo continuo, que presentan velocidades binarias medias diferentes, basándose en el valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente en combinación con la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir.
2. Método según la reivindicación 1, en el que la etapa de controlar conmutaciones (212) incluye la etapa de:
 - si una velocidad binaria ($BR_{SIGUIENTE}$) de una parte siguiente de datos de velocidad binaria variable a transmitir supera una velocidad binaria media (BR_{MED}), entonces retardar (310) una conmutación hacia una versión del flujo continuo que presente una velocidad binaria en general mayor.
3. Método según la reivindicación 1, en el que la etapa de controlar conmutaciones (212) incluye la etapa de:
 - si una velocidad binaria ($BR_{SIGUIENTE}$) de una parte siguiente de datos de velocidad binaria variable a transmitir supera una velocidad binaria media (BR_{MED}), entonces acelerar (314) una conmutación hacia una versión del flujo continuo que presente una velocidad binaria en general menor.
4. Método según la reivindicación 1, en el que la etapa de controlar conmutaciones (212) incluye la etapa de:
 - si una velocidad binaria ($BR_{SIGUIENTE}$) de una parte siguiente de datos de velocidad binaria variable a transmitir está por debajo de una velocidad binaria media (BR_{MED}), entonces acelerar una conmutación (310) hacia una versión del flujo continuo que presente una velocidad binaria en general mayor.
5. Método según la reivindicación 1, en el que el servidor (105) controla conmutaciones a versiones diferentes del flujo continuo basándose en un estado actual de la memoria intermedia (155) del cliente mediante la aplicación de uno o más umbrales a datos ya en la memoria intermedia del cliente y en el que la etapa de controlar conmutaciones (212) comprende:
 - ajustar dinámicamente (306) el umbral o umbrales basándose en la velocidad binaria (BR) de la parte de datos de velocidad binaria variable a transmitir; y
 - determinar si se conmuta a una versión diferente del flujo continuo basándose en el estado actual de la memoria intermedia (155) del cliente mediante la aplicación de uno o más umbrales ajustables al valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia (115) del cliente.
6. Método según la reivindicación 5, en el que el umbral o umbrales incluyen por lo menos uno de entre un umbral de conmutación ascendente (PT_{ASC}) y un umbral de conmutación descendente (PT_{DESC}).
7. Método según la reivindicación 6, en el que la etapa de ajustar dinámicamente (306) el umbral o umbrales incluye las etapas de:
 - determinar una velocidad binaria media (BR_{MED}) de los datos de velocidad binaria variable; y
 - ajustar selectivamente (306) los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} , PT_{DESC}) basándose en una comparación de una velocidad binaria ($BR_{SIGUIENTE}$) de una parte siguiente de datos de velocidad binaria variable a transmitir con una velocidad binaria media (BR_{MED}) de los datos de velocidad binaria variable.
8. Método según la reivindicación 7, en el que la etapa de ajustar selectivamente (306) los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} , PT_{DESC}) incluye las etapas de:
 - aumentar (306) los umbrales de conmutación ascendente y conmutación descendente si la velocidad binaria ($BR_{SIGUIENTE}$) de la parte siguiente de datos de velocidad binaria variable supera la velocidad binaria media (BR_{MED}); y

ES 2 313 323 T3

reducir (306) el umbral de conmutación ascendente si la velocidad binaria ($BR_{SIGUIENTE}$) de la parte siguiente de los datos de velocidad binaria variable está por debajo de la velocidad binaria media (BR_{MED}) y si el umbral de conmutación ascendente supera al umbral de conmutación descendente.

- 5 9. Método según la reivindicación 8, en el que la etapa de aumentar los umbrales de conmutación ascendente y de conmutación descendente (PT_{ASC} , PT_{DESC}) si la velocidad binaria de la siguiente parte de datos de velocidad binaria variable supera la velocidad binaria media incluye las etapas de:

determinar (302) la velocidad binaria de la siguiente parte de datos de velocidad binaria variable;

- 10 calcular (304) una relación (F) de la velocidad binaria de la siguiente parte de datos de velocidad binaria variable con respecto a la velocidad binaria media; y

- 15 multiplicar (306) los umbrales de conmutación ascendente y conmutación descendente (PT_{ASC} , PT_{DESC}) por la relación para aumentar de este modo los umbrales.

10. Método según la reivindicación 8, en el que la etapa de reducir el umbral de conmutación ascendente (PT_{ASC}) incluye las etapas de:

- 20 determinar (302) la velocidad binaria de la siguiente parte de datos de velocidad binaria variable;

calcular (304) una relación (F) de la velocidad binaria de la siguiente parte de datos de velocidad binaria variable con respecto a la velocidad binaria media;

- 25 multiplicar (306) el umbral de conmutación ascendente por la relación para reducir de este modo el umbral; y

seleccionar (310) el mayor de entre el umbral de conmutación ascendente reducido y el umbral de conmutación descendente para ser usado como umbral de conmutación ascendente nuevo.

- 30 11. Método según cualquiera de las reivindicaciones 2 a 12, en el que la velocidad binaria media (BR_{MED}) es representativa de una velocidad binaria media durante toda la versión del flujo continuo que se está transmitiendo.

- 35 12. Método según cualquiera de las reivindicaciones 2 a 12, en el que la velocidad binaria media (BR_{MED}) es representativa de una velocidad binaria media de datos que ya han sido transmitidos dentro de la versión del flujo continuo que se está transmitiendo.

13. Método según cualquiera de las reivindicaciones anteriores, en el que el valor asociado a datos en la memoria intermedia del cliente es representativo de una longitud de reproducción (PT) de los datos.

- 40 14. Método según cualquiera de las reivindicaciones anteriores, en el que el valor asociado a datos en la memoria intermedia del cliente es representativo de un nivel de llenado de la memoria intermedia de los datos.

- 45 15. Producto de programa de ordenador que comprende partes de código de programa para realizar las etapas de cualquiera de las reivindicaciones anteriores cuando el producto de programa de ordenador se ejecuta en un sistema de ordenador.

16. Producto de programa de ordenador según la reivindicación 15, en el que el producto de programa de ordenador está almacenado en un soporte de grabación legible por ordenador.

- 50 17. Aparato que comprende un procesador de ordenador y una memoria acoplada al procesador, en donde la memoria está codificada con uno o más programas que pueden realizar las etapas de cualquiera de las reivindicaciones 1 a 14.

- 55 18. Aparato (400) para controlar transmisiones por paquetes de datos de velocidad binaria variable desde un servidor (105) a un cliente (115) que tiene una memoria intermedia (155) de cliente, en el que el servidor conmuta entre versiones diferentes de un flujo continuo de datos de velocidad binaria variable que se están transmitiendo basándose en un estado de la memoria intermedia del cliente, comprendiendo el aparato:

- 60 un controlador anticipatorio (402) de la transmisión de flujos continuos de velocidad binaria variable para seleccionar (202) una versión inicial de un flujo continuo de datos de velocidad binaria variable para su transmisión;

un monitor (404) de la memoria intermedia de cliente para realizar un seguimiento de un valor representativo de una cantidad de datos dentro de la memoria intermedia del cliente;

- 65 una unidad de determinación (406) de la velocidad binaria variable para determinar una velocidad binaria (BR) de una parte de datos de velocidad binaria variable todavía por transmitir dentro de la versión seleccionada del flujo continuo; y

ES 2 313 323 T3

en el que el controlador anticipatorio (402) de la transmisión de flujos continuos de velocidad binaria variable controla entonces conmutaciones a versiones diferentes del flujo continuo, que presentan velocidades binarias medias diferentes, basándose en el valor (PT) representativo de la cantidad de datos dentro de la memoria intermedia del cliente en combinación con la velocidad binaria (BR) de la parte de datos de velocidad binaria variable todavía por transmitir.

5

10

15

20

25

30

35

40

45

50

55

60

65

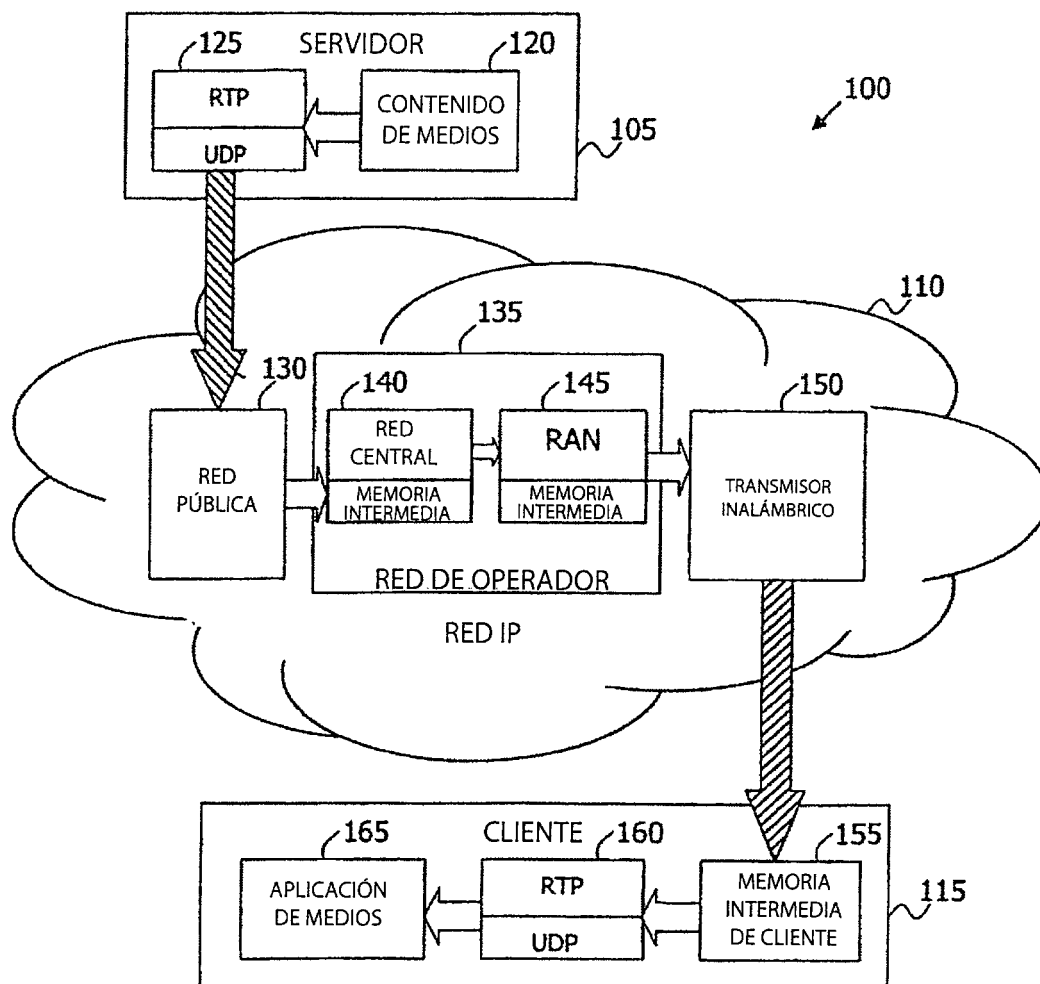


FIG. 1

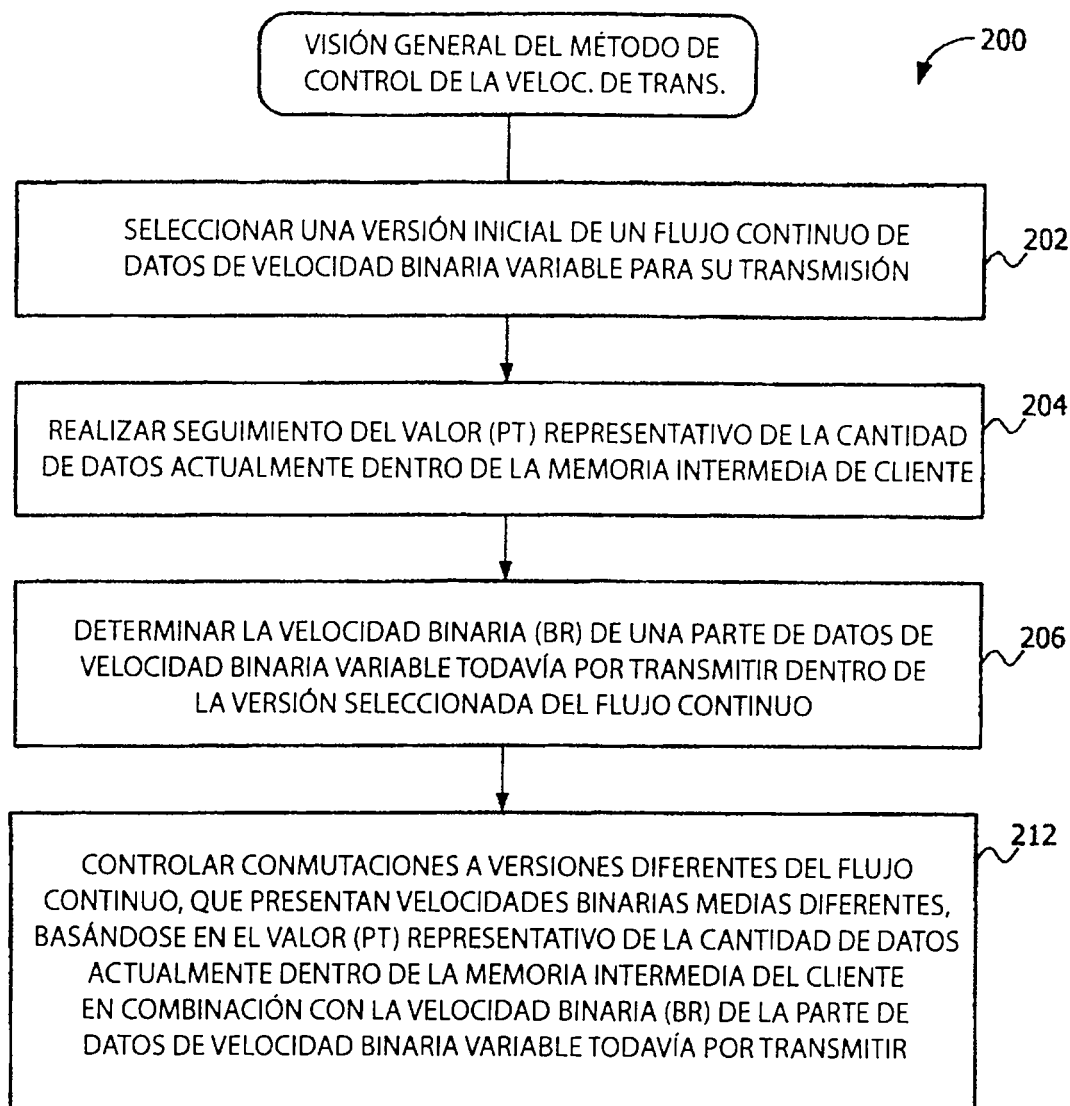


FIG. 2

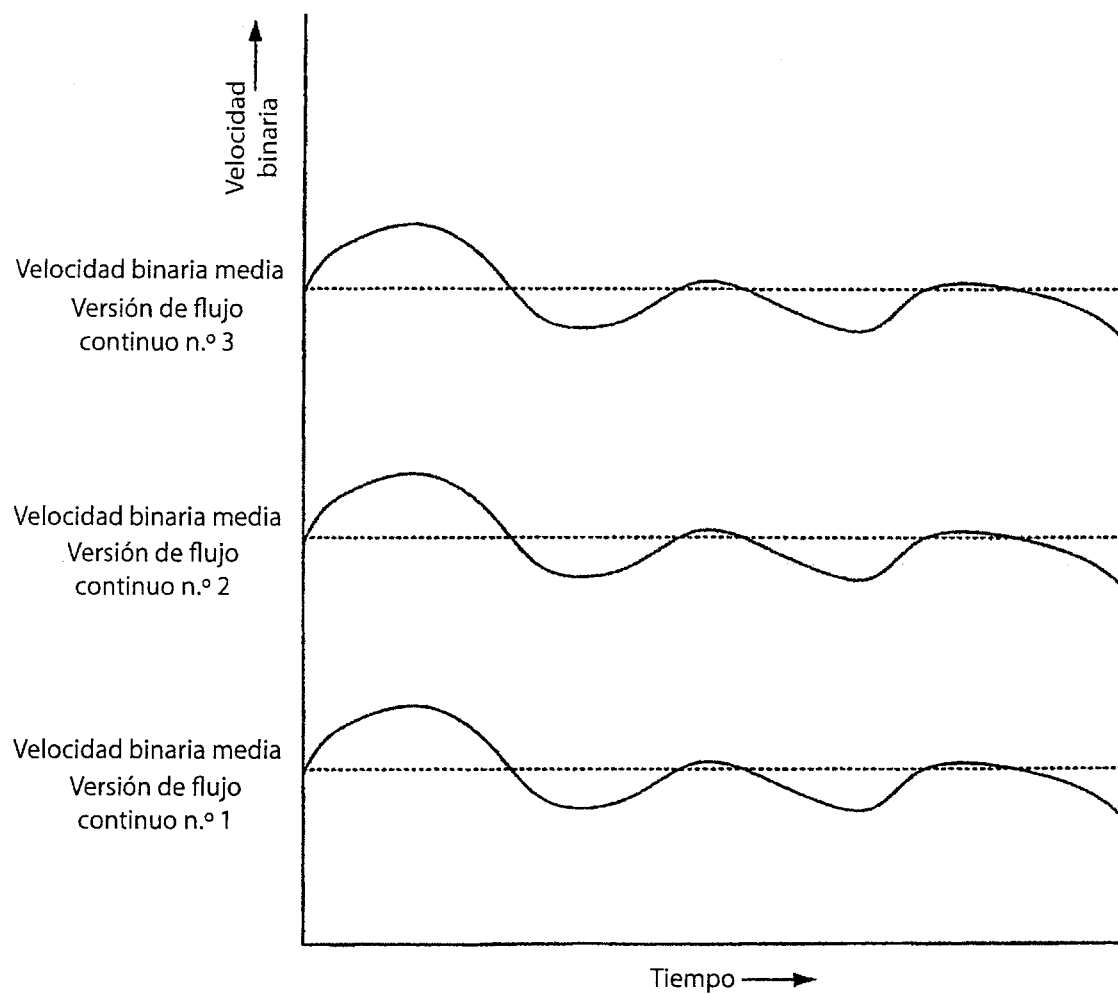


FIG. 3

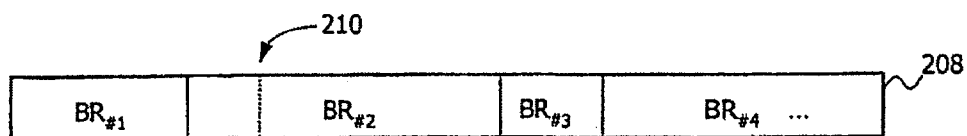


FIG. 4

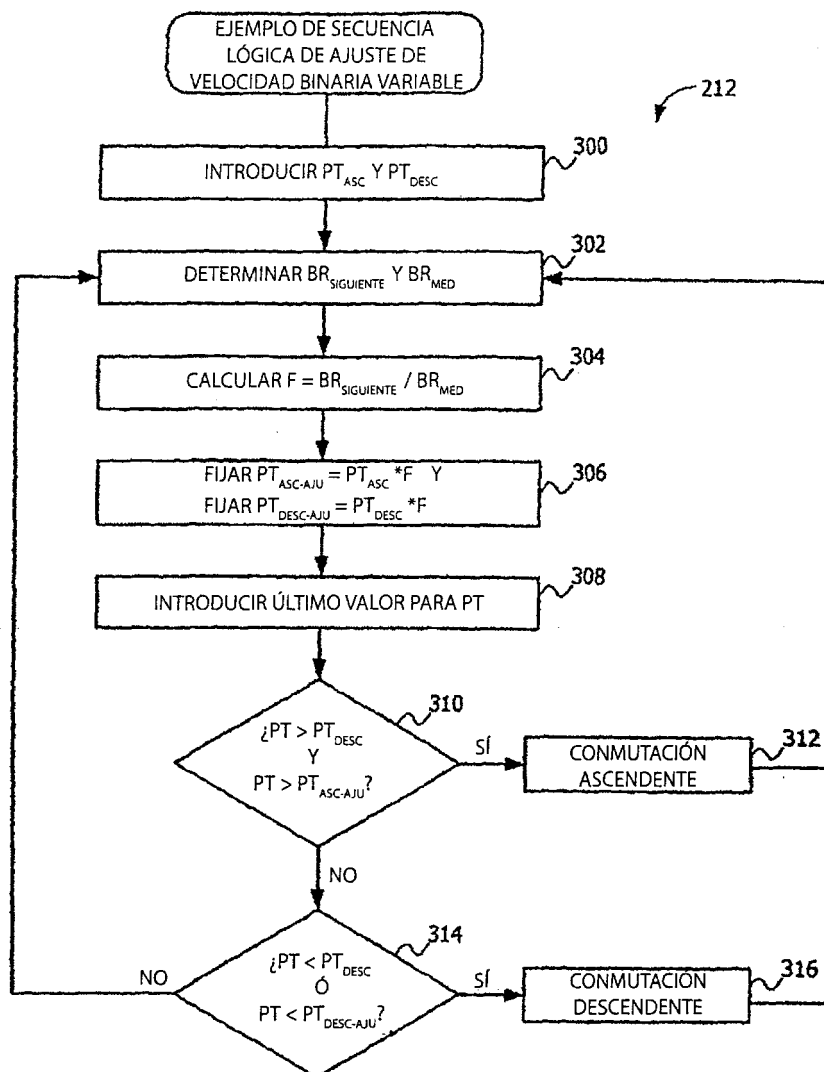


FIG. 5

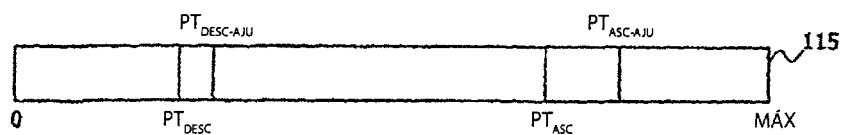


FIG. 6

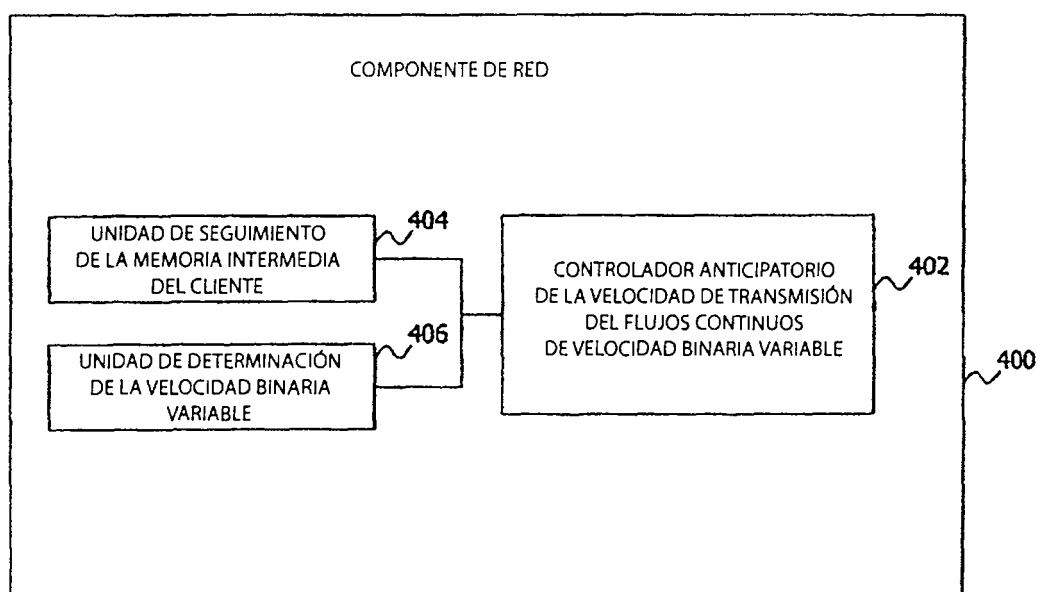


FIG. 7