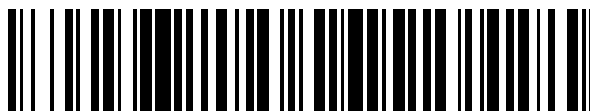


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 656 470**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/88 (2014.01)

H04N 21/44 (2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **28.03.2014 PCT/US2014/032243**

87 Fecha y número de publicación internacional: **02.10.2014 WO14160974**

96 Fecha de presentación y número de la solicitud europea: **28.03.2014 E 14722918 (1)**

97 Fecha y número de publicación de la concesión europea: **25.10.2017 EP 2979455**

54 Título: **Mejora de los diseños de formato de la carga útil de RTP**

30 Prioridad:

29.03.2013 US 201361806705 P
27.03.2014 US 201414228102

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
27.02.2018

73 Titular/es:

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

72 Inventor/es:

WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 656 470 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

Mejora de los diseños de formato de la carga útil de RTP

5 **[0001]** Esta solicitud reivindica el beneficio de:

la solicitud provisional de los Estados Unidos 61/806,705 presentada el 29 de marzo de 2013,
cuyo contenido completo se incorpora en el presente documento por referencia.

10

CAMPO TÉCNICO

[0002] Esta divulgación se refiere al procesamiento de datos de vídeo.

15 **ANTECEDENTES**

[0003] Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digitales implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo y las extensiones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de compresión de vídeo.

20

25

[0004] Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intra-imagen) y/o la predicción temporal (inter-imagen) para reducir o eliminar la redundancia intrínseca de las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un fragmento intracodificado (I) de una imagen se codifican mediante la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a las muestras de referencia de los bloques vecinos de la misma imagen, o la predicción temporal con respecto a las muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

30

35

[0005] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con una modalidad de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de los píxeles al dominio de las transformadas, dando como resultado unos coeficientes de transformación residual, que posteriormente se pueden cuantificar. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación de entropía para lograr aún más compresión.

40

45

50

RESUMEN

[0006] En general, esta divulgación describe técnicas para procesar datos de vídeo. En particular, esta divulgación describe diseños mejorados de formatos de carga útil del protocolo de transporte en tiempo real (RTP).

55

[0007] En un ejemplo, un procedimiento de procesamiento de datos de vídeo incluye recibir una primera unidad de fragmentación que comprende un subconjunto de una unidad de capa de abstracción de red (NAL) fragmentada; analizar un bit de inicio de la unidad de fragmentación para determinar si la primera unidad de fragmentación comprende un inicio de la unidad NAL fragmentada; en respuesta a la primera unidad de fragmentación que comprende el inicio de la unidad NAL fragmentada y uno o ambos modos de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor, analizar un segundo parámetro para determinar un orden de decodificación para la unidad NAL fragmentada; y decodificar la unidad NAL fragmentada basándose en el orden de decodificación determinado.

60

65

[0008] En otro ejemplo, un dispositivo para procesar datos de vídeo incluye una memoria; un receptor configurado

para paquetes de protocolo de transporte en tiempo real (RTP); y uno o más procesadores configurados para recibir una primera unidad de fragmentación que comprende un subconjunto de una unidad de capa de abstracción de red (NAL) fragmentada; analizar un bit de inicio de la unidad de fragmentación para determinar si la primera unidad de fragmentación comprende un inicio de la unidad NAL fragmentada; en respuesta a la primera unidad de fragmentación que comprende el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor, analizar un segundo parámetro para determinar un orden de decodificación para la unidad NAL fragmentada; decodificar la unidad NAL fragmentada basándose en el orden de decodificación determinado.

[0009] Un medio de almacenamiento legible por ordenador almacena instrucciones que cuando son ejecutadas por uno o más procesadores provocan que uno o más procesadores reciban una primera unidad de fragmentación que comprende un subconjunto de una unidad de capa de abstracción de red (NAL) fragmentada; analicen un bit de inicio de la unidad de fragmentación para determinar si la primera unidad de fragmentación comprende un inicio de la unidad NAL fragmentada; en respuesta a la primera unidad de fragmentación que comprende el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor, analicen un segundo parámetro para determinar un orden de decodificación para la unidad NAL fragmentada; y decodifiquen la unidad NAL fragmentada basándose en el orden de decodificación determinado.

[0010] En otro ejemplo, un aparato para procesar datos de vídeo incluye medios para recibir una primera unidad de fragmentación que comprende un subconjunto de una unidad de capa de abstracción de red (NAL) fragmentada; medios para analizar un bit de inicio de la unidad de fragmentación para determinar si la primera unidad de fragmentación comprende un inicio de la unidad NAL fragmentada; medios para analizar un segundo parámetro para determinar un orden de decodificación para la unidad NAL fragmentada en respuesta a la primera unidad de fragmentación que comprende el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor; y medios para decodificar la unidad NAL fragmentada en base al orden de decodificación determinado.

[0011] En otro ejemplo, un procedimiento de procesamiento de datos de vídeo incluye generar una primera unidad de fragmentación que comprende un subconjunto de una unidad de capa de abstracción de red (NAL) fragmentada, en el que la primera unidad de fragmentación comprende un inicio de la unidad NAL fragmentada; establecer un bit de inicio de la unidad de fragmentación para indicar que la primera unidad de fragmentación comprende el inicio de la unidad NAL fragmentada; en respuesta a la primera unidad de fragmentación que comprende el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor, establecer un segundo parámetro para indicar un orden de decodificación para la unidad NAL fragmentada; y transmitir la unidad NAL fragmentada.

[0012] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0013]

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

La FIG. 2 muestra una representación visual de una estructura de paquetes de agregación.

La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 5 es un diagrama de bloques que ilustra un conjunto de dispositivos de ejemplo que forman parte de una red.

La FIG. 6 muestra un procedimiento de ejemplo de desempaquetado de unidades NAL de acuerdo con las técnicas de esta divulgación.

La FIG. 7 muestra un procedimiento de ejemplo de empaquetado de unidades NAL de acuerdo con las técnicas

de esta divulgación.

La FIG. 8 muestra un procedimiento de ejemplo de desempaqueado de unidades NAL de acuerdo con las técnicas de esta divulgación.

5 La FIG. 9 muestra un procedimiento de ejemplo de empaquetado de unidades NAL de acuerdo con las técnicas de esta divulgación.

10 La FIG. 10 muestra un procedimiento de ejemplo de desempaqueado de unidades NAL de acuerdo con las técnicas de esta divulgación.

La FIG. 11 muestra un procedimiento de ejemplo de empaquetado de unidades NAL de acuerdo con las técnicas de esta divulgación.

15 DESCRIPCIÓN DETALLADA

[0014] Esta divulgación introduce diversas técnicas para diseños mejorados de formatos de carga útil de protocolo de transporte en tiempo real (RTP) para el transporte de datos de vídeo codificados a través de RTP. RTP es un protocolo de transporte, como se especifica en IETF RFC 3550, que a partir del 29 de marzo de 2013 está disponible en <http://www.ietf.org/rfc/rfc3550.txt>, y que se incorpora en el presente documento como referencia en su totalidad. De acuerdo con IETF RFC 3550, RTP fue desarrollado con la intención de proporcionar servicios de entrega de extremo a extremo para datos con características en tiempo real, como audio y vídeo interactivos. Los datos transportados de acuerdo con RTP se empaquetan en paquetes RTP. Los paquetes RTP son paquetes de datos que incluyen datos de cabecera y de carga útil de RTP. Los datos de la carga útil de un paquete RTP pueden ser datos de vídeo codificados. Los datos de vídeo codificados pueden, por ejemplo, tener la forma de una o más unidades de capa de abstracción de red (NAL).

[0015] Para el transporte de datos de vídeo codificados de acuerdo con un códec de vídeo a través de RTP, puede necesitar especificarse un formato de carga útil de RTP para el códec de vídeo. Por ejemplo, RFC 6184 (que a partir del 29 de marzo de 2013 está disponible en <http://www.ietf.org/rfc/rfc6184.txt>) especifica el formato de carga útil de RTP para vídeo H.264 y RFC 6190 (que a partir del 29 marzo de 2013 está disponible en <http://www.ietf.org/rfc/rfc6190.txt>) especifica el formato de carga útil de RTP para vídeo SVC, ambos incorporados en su totalidad por referencia en el presente documento. Un borrador reciente del formato de carga útil de RTP para vídeo HEVC está disponible, desde el 29 de marzo de 2013, en <http://tools.ietf.org/html/draft-schierl-payload-rtp-h265-01>, y que se incorpora por referencia en su totalidad en el presente documento. Estas diversas normas describen cómo se empaquetan los datos de vídeo codificados (por ejemplo, unidades NAL codificadas) en paquetes RTP.

[0016] De acuerdo con la especificación HEVC, una unidad NAL se define como una estructura sintáctica que contiene una indicación del tipo de datos que siguen y bytes que contienen dichos datos en forma de una carga útil de secuencia de bytes sin procesar (RBSP) intercalados según sea necesario con bytes de prevención de emulación. Una unidad NAL VCL incluye datos de la capa de codificación de vídeo, mientras que una unidad NAL no VCL puede incluir algunos otros datos sobre los datos de la capa de codificación de vídeo. De acuerdo con HEVC, una unidad de acceso se define como un conjunto de unidades NAL, que están asociadas entre sí de acuerdo con una regla de clasificación especificada, son consecutivas en el orden de decodificación y contienen exactamente una imagen codificada. Además de contener las unidades NAL VCL de la imagen codificada, una unidad de acceso también puede contener unidades NAL no VCL. La decodificación de una unidad de acceso siempre da como resultado una imagen decodificada. Los paquetes RTP son paquetes utilizados para transportar la unidad NAL.

[0017] Los diseños de los formatos de carga útil de RTP en el RFC 6184 y el RFC 6190, y el borrador existente del formato de carga útil de RTP para HEVC están asociados con varios posibles problemas o deficiencias. Como un ejemplo, se especifican varios modos de empaquetado y se especifican muchos tipos de paquetes, lo que hace que sea potencialmente difícil elegir el modo de empaquetado y los tipos de paquetes que se utilizarán. Como otro ejemplo, el entrelazado de unidades de la capa de abstracción de red (NAL) de una unidad de acceso solo es posible utilizando los Paquetes de Agregación de Múltiples Tiempos (MTAP) como se define en RFC 6184 y RFC 6190. Sin embargo, cuando las unidades NAL de una sola unidad de acceso se agregan en un paquete RTP, todas tienen la misma marca de tiempo. Por lo tanto, basta con basarse en la marca de tiempo de RTP del paquete RTP, pero el envío de información de tiempo adicional según lo requerido por RFC 6184 y RFC 6190 potencialmente desperdicia ancho de banda. El entrelazado de unidades NAL de una unidad de acceso permite el transporte de fragmentos codificados entrelazados de una imagen en diferentes paquetes, por lo que cuando se pierde un paquete, los fragmentos vecinos recibidos pueden utilizarse para una mejor ocultación.

[0018] Para hacer frente a los problemas potenciales y deficiencias introducidas anteriormente, esta divulgación presenta varias técnicas para diseños mejorados del formato de la carga útil de RTP. De acuerdo con una técnica, el modo de empaquetado no es diferenciado, de modo que son posibles tanto el empaquetado no entrelazado como el entrelazado, son posibles tanto la transmisión de una sola sesión como la transmisión de múltiples sesiones y se

especifica un proceso de desempaqueado unificado basado en los valores numéricos del orden de decodificación absoluto de las unidades NAL, que pueden obtenerse a partir de la información opcional señalada en las cargas útiles del paquete.

5 **[0019]** De acuerdo con otra técnica, el diseño de los paquetes de agregación permite el entrelazado de unidades NAL de una unidad de acceso sin requerir el envío de información de tiempo redundante. Los paquetes de agregación, como se describe en esta divulgación, pueden mejorar la codificación de vídeo cuando se transportan múltiples fragmentos pequeños. Permitir el entrelazado de unidades NAL de acuerdo con las técnicas de esta divulgación puede mejorar la calidad global de la imagen reconstruida. Por ejemplo, si los paquetes de agregación incluyen unidades NAL entrelazadas y se pierde un paquete de agregación, entonces las unidades NAL entrelazadas probablemente correspondan a un grupo disperso de bloques de vídeo en lugar de bloques de vídeo adyacentes. Las técnicas de ocultación de errores suelen ser más eficaces para áreas de pérdida más pequeñas, y por lo tanto, pueden ser más eficaces para ocultar la pérdida de un grupo de bloques de vídeo dispersos en comparación con ocultar la pérdida de un grupo de bloques de vídeo adyacentes.

15 **[0020]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de procesamiento de vídeo 10 a modo de ejemplo que puede utilizarse junto con las técnicas descritas en esta divulgación. El sistema 10 puede, por ejemplo, generar, procesar y transmitir datos de vídeo usando las técnicas de RTP descritas en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados, que un dispositivo de destino 14 va a decodificar en un momento posterior. Los datos de vídeo codificados pueden encaminarse desde el dispositivo de origen 12 hasta el dispositivo de destino 14 mediante un elemento de red 29 con reconocimiento de medios (MANE). El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, notebook (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos inalámbricos tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de transmisión de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

30 **[0021]** El sistema 10 puede operar de acuerdo con diferentes normas de codificación de vídeo, una norma o técnica patentada, o cualquier otra forma de codificación multivista. Por ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con una norma de compresión de vídeo, tal como ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC). El último borrador conjunto de la extensión de MVC disponible al público se describe en "Codificación de vídeo avanzada para servicios audiovisuales genéricos", Recomendación ITU-T H.264, marzo de 2010. Un borrador conjunto más reciente de la extensión de MVC, disponible al público, se describe en "Codificación de vídeo avanzada para servicios audiovisuales genéricos", Recomendación ITU-T H.264, junio de 2011. A partir de enero de 2012 se aprobó un borrador conjunto actual de la extensión de MVC.

45 **[0022]** Además, existe una nueva norma de codificación de vídeo, concretamente la norma de Codificación de Vídeo de Alta Eficiencia (HEVC) que está siendo desarrollada actualmente por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Un Borrador de Trabajo (BT) reciente de la HEVC, denominado HEVC WD9, está disponible desde el 15 de marzo de 2013 en http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v10.zip. Para los fines de la descripción, el codificador de vídeo 20 y el decodificador de vídeo 30 se describen en el contexto del HEVC o la norma H.264 y las extensiones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263. Las técnicas de codificación patentadas, tales como las denominadas On2 VP6/VP7/VP8, también pueden implementar una o más de las técnicas descritas en el presente documento. Las técnicas de esta divulgación son potencialmente aplicables a varias normas de codificación de vídeo, incluyendo HEVC y otras.

55 **[0023]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a decodificar, mediante un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir los datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14. El enlace 16 puede incluir uno o más

MANE, tales como el MANE 27, que encamina los datos de vídeo desde el dispositivo de origen 12 al dispositivo 14 de destino.

[0024] De forma alternativa, los datos codificados pueden transmitirse desde la interfaz de salida 22 a un dispositivo de almacenamiento 25. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 25 mediante una interfaz de entrada. El dispositivo de almacenamiento 25 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, discos DVD, discos CD-ROM, una memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento 25 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados del dispositivo de almacenamiento 25 a través de transmisión en continuo o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los ejemplos de servidores de archivos incluyen un servidor de la red (por ejemplo, para una página web), un servidor de FTP, dispositivos de almacenamiento anexos a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 25 puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas. Los datos de vídeo recuperados del dispositivo de almacenamiento 25 pueden encaminarse al dispositivo de destino 14 utilizando uno o más MANE, tales como el MANE 27.

[0025] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera de una diversidad de aplicaciones de multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para admitir la transmisión de vídeo unidireccional o bidireccional a fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0026] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un empaquetador 21 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de tales fuentes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o alámbricas.

[0027] El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o de forma alternativa) pueden almacenarse en el dispositivo de almacenamiento 25 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para su decodificación y/o reproducción.

[0028] El dispositivo de destino 14 incluye una interfaz de entrada 28, un desempaquetador 29, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados por el enlace 16. Los datos de vídeo codificados, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 25, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Tales elementos sintácticos pueden incluirse con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos.

[0029] El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo decodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de

visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0030] Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados, cada uno de ellos, con un codificador y un decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos diferentes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0031] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio, legible por ordenador, y ejecutar las instrucciones en hardware mediante uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

[0032] El equipo JCT-VC está trabajando en el desarrollo de la norma HEVC. Los esfuerzos de normalización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modalidades de codificación de intra-predicción, el HM puede proporcionar hasta treinta y tres modalidades de codificación de intra-predicción.

[0033] En general, el modelo de funcionamiento del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques de árbol o unidades de codificación de máximo tamaño (LCU), que incluyen muestras tanto de bloques de luminancia como de croma. Un bloque de árbol tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un cierto número de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque de árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol, como un nodo raíz del árbol cuaternario, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, como nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque de árbol, y también pueden definir un tamaño mínimo de los nodos de codificación. La división de un bloque de árbol puede ocurrir en el dominio de luminancia, y puede imitarse en los dominios de croma, posiblemente con un submuestreo adicional de los nodos hojas.

[0034] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Las modalidades de división pueden diferir en función de si la CU está codificada en modalidad de salto o directa, codificada en modalidad de intra-predicción o codificada en modalidad de inter-predicción. Las PU pueden dividirse para no tener forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener forma cuadrada o no cuadrada.

[0035] La norma HEVC admite unas transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU presentan típicamente el mismo tamaño o un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformación, que pueden cuantificarse.

[0036] En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está codificada en modalidad intra, la PU puede incluir datos que describen una modalidad de intra-predicción para la PU. Como otro ejemplo, cuando la PU está codificada en la modalidad inter, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de

píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

5 **[0037]** En general, se usa una TU para los procesos de transformación y de cuantificación. Una CU dada que presenta una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo 20 puede calcular los valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que se pueden transformar en coeficientes de transformación, cuantificar y escanear mediante las TU, para generar coeficientes de transformación en serie para la codificación de entropía. Esta divulgación usa típicamente el término “bloque de vídeo” para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término “bloque de vídeo” para referirse a un bloque de árbol, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

15 **[0038]** Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un cierto número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

25 **[0039]** Como un ejemplo, el HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el HM admite la intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$ y la inter-predicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la inter-predicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en el 25% y el 75%. La parte de la CU correspondiente a la división del 25% está indicada por una “n” seguida de una indicación de “arriba”, “abajo”, “izquierda” o “derecha”. Así, por ejemplo, “ $2N \times nU$ ” se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

35 **[0040]** En esta divulgación, “ $N \times N$ ” y “N por N” pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de tamaño 16×16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ presenta, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar ordenados en filas y columnas. Además, no es necesario que los bloques presenten necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

40 **[0041]** Tras la codificación de intra-predicción o inter-predicción mediante las PU de una CU, el codificador de vídeo 20 puede calcular los datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de las transformadas tras la aplicación de una transformación, por ejemplo, una transformada de coseno discreta (DCT), una transformación de enteros, una transformada wavelet o una transformada similar desde un punto de vista conceptual a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y los valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU, y a continuación transformar las TU para generar coeficientes de transformación para la CU.

45 **[0042]** Tras cualquier transformación para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformación. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. Por ejemplo, un valor de n bits puede redondearse a la baja a un valor de m bits durante la cuantificación, donde n es mayor que m .

50 **[0043]** En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de escaneado predefinido para escanear los coeficientes de transformación cuantificados, para generar un vector en serie que pueda ser codificado de entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneado adaptativo. Después de escanear los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación de entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa de acuerdo con el contexto (CAVLC), la codificación aritmética binaria adaptativa según el contexto (CABAC), la codificación aritmética binaria adaptativa según el contexto basada en la sintaxis (SBAC), la codificación de entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos de codificación de entropía. El codificador de vídeo 20 también puede realizar la codificación de entropía de los

elementos sintácticos asociados a los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

[0044] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

[0045] Las técnicas descritas en la presente divulgación se pueden aplicar independientemente o conjuntamente. Aspectos de estas técnicas pueden ser realizados por el empaquetador 21 y el desempaquetador 29. En algunos casos, el empaquetador 21 puede denominarse emisor de RTP, o simplemente emisor, mientras que el desempaquetador 29 puede denominarse receptor de RTP o simplemente receptor. Los aspectos de estas técnicas se resumen de la siguiente manera:

- Señalización de la primera dirección de la primera unidad de árbol de codificación (CTU) de un elemento codificado transportado en una Unidad de Fragmentos (FU)

- La ID del elemento (o un delta de dos valores de ID de elemento) se señala en la estructura de FU, antes de la carga útil de FU. Esta señalización especifica o indica la dirección de la CTU en el escaneado del elemento (así como también la dirección en el barrido de trama) de la primera CTU en el elemento.

- Alternativamente, la dirección de la CTU en el escaneado del elemento (o un delta de dos de tales valores) de la primera CTU en un elemento codificado que se transporta en una FU se señala en la estructura de FU, antes de la carga útil de FU.

- Alternativamente, la dirección de la CTU en el barrido de trama (o un delta de dos de tales valores) de la primera CTU en un elemento codificado que se transporta en una FU se señala en la estructura de FU, antes de la carga útil de FU.

- Alternativamente, la señalización (en cualquiera de las formas anteriores) está presente solo cuando hay una indicación (por ejemplo, un parámetro de tipo de medio) que indica la presencia de la señalización. Tal parámetro de tipo de medio puede simplemente indicar la presencia de la señalización anterior, o indicar el uso de elementos (y si no se indican los elementos, entonces la señalización anterior no está presente).

- Cuando un elemento se transporta en múltiples FU:

- Usar/agregar un indicador S en la cabecera de FU para indicar el inicio de un elemento fragmentado.

- Con esto, la presencia de cualquier señalización mencionada anteriormente para obtener la dirección de la CTU de la primera CTU en el elemento está (además) condicionada a que el indicador S sea igual a 0.

- Usar/agregar un indicador E en la cabecera de FU para indicar el final de un elemento fragmentado.

- Usar/agregar un indicador en la cabecera de la carga útil del paquete RTP para indicar si todas las unidades NAL del paquete contienen segmentos de fragmento de división dependientes.

- Alternativamente, dos bits en la cabecera del paquete RTP para indicar uno de los siguientes

- Todas las unidades NAL del paquete son segmentos de fragmento dependientes.

- Al menos una de las unidades NAL del paquete es un segmento de fragmento dependiente para el cual el segmento de fragmento independiente correspondiente no está en el mismo paquete

- Al menos una de las unidades NAL del paquete es un segmento de fragmento independiente.

- Todas las unidades NAL del paquete son segmentos de fragmento independientes.

5 • En un paquete que contiene solo una unidad NAL, solo se necesita un bit para indicar si la unidad NAL contiene un segmento de fragmento dependiente.

10 • Alternativamente, la señalización (en cualquiera de las formas anteriores) está presente solo cuando hay una indicación (por ejemplo, un parámetro de tipo de medio) que indica la presencia de la señalización. Tal parámetro de tipo de medio puede simplemente indicar la presencia de la señalización anterior, o indicar el uso de segmentos de fragmento dependientes (y si los segmentos de fragmento dependientes no están indicados, entonces la señalización anterior no está presente).

15 **[0046]** A continuación se describirán aspectos de las estructuras de la carga útil. Estas estructuras de la carga útil pueden ser generadas por el empaquetador 21 y ser analizadas por el desempaquetador 29. Los primeros dos bytes de la carga útil de un paquete RTP pueden definir la cabecera de la carga útil. La cabecera de la carga útil puede consistir en los mismos campos que la cabecera de la unidad NAL de HEVC (F, Tipo, ID de capa y TID, que corresponden a los elementos sintácticos `forbidden_zero_bit`, `nal_unit_type`, `nuh_layer_id` y `nuh_temporal_id_plus1` como se especifica en la sección 7.3.1.2 de HEVC WD 10), independientemente del tipo de estructura de la carga útil.

20 **[0047]** Se especifican tres tipos diferentes de estructuras de carga útil de paquetes RTP. Un receptor puede identificar el tipo de carga útil de un paquete RTP a través del campo Tipo en la cabecera de la carga útil. El receptor puede ser un desempaquetador de un dispositivo que incluye un decodificador de vídeo, o puede formar parte de un MANE u otra entidad de red. Las tres estructuras de carga útil diferentes son las siguientes:

25 Paquete de unidad NAL individual: Contiene una sola unidad NAL en la carga útil, y la cabecera de la unidad NAL de la unidad NAL también sirve como la cabecera de la carga útil. Los paquetes de la unidad NAL única NO DEBEN utilizarse cuando el modo tx es igual a "MST" o `sprop-depack-buf-nalus` es mayor que 0.

30 Paquete de agregación (AP): Contiene una o más unidades NAL dentro de una unidad de acceso. Véase abajo.

Unidad de fragmentación (FU): Contiene un subconjunto de una sola unidad NAL. Véase abajo.

35 **[0048]** A continuación se describirán los modos de transmisión admitidos por empaquetador 21 y el desempaquetador 29. Las técnicas de esta divulgación pueden permitir la transmisión de un flujo de bits HEVC en una única sesión de RTP o múltiples sesiones de RTP. El concepto y el principio de funcionamiento son coherentes con RFC6190 y siguen un diseño similar, pero potencialmente más simple. Si solo se utiliza una sesión de RTP para la transmisión del flujo de bits HEVC, el modo de transmisión se denomina transmisión de sesión única (SST); de lo contrario (se usa más de una sesión de RTP para la transmisión del flujo de bits HEVC), el modo de transmisión se denomina transmisión de multisesión (MST).

40 **[0049]** SE DEBERÍA utilizar SST para escenarios de unidifusión punto a punto, mientras que SE DEBERÍA utilizar MST para escenarios de multidifusión punto a multipunto donde diferentes receptores requieren diferentes puntos de operación del mismo flujo de bits HEVC, para mejorar la eficiencia de utilización del ancho de banda.

45 **[0050]** Si el modo tx es igual a "SST", SE DEBE utilizar SST. De lo contrario (el modo tx es igual a "MST"), SE DEBE utilizar MST.

50 **[0051]** A continuación se describirán aspectos del número de orden de decodificación. Para cada unidad NAL, se obtiene la variable `AbsDon`, que representa el número de orden de decodificación que indica el orden de decodificación de la unidad NAL.

55 **[0052]** Supongamos que la unidad NAL n es la unidad NAL n-ésima en el orden de transmisión dentro de una sesión RTP.

[0053] Si el modo tx es igual a "SST" y `sprop-depack-buf-nalus` es igual a 0, `AbsDon[n]`, el valor de `AbsDon` para la unidad NAL n, se obtiene como igual a n.

60 **[0054]** De lo contrario (el modo tx es igual a "MST" o `sprop-depack-buf-nalus` es mayor que 0), `AbsDon[n]` se obtiene como sigue, donde `DON[n]` es el valor de la variable `DON` para la unidad NAL n:

- Si n es igual a 0 (es decir, la unidad NAL n es la primera unidad NAL en el orden de transmisión), `AbsDon[0]` se establece igual a `DON[0]`.

65 - De lo contrario (n es mayor que 0), se aplica lo siguiente para la obtención de `AbsDon[n]`:

Si $DON[n] == DON[n-1]$,
 $AbsDon[n] = AbsDon[n-1]$
 Si $(DON[n] > DON[n-1] \text{ y } DON[n] - DON[n-1] < 32768)$,
 $AbsDon[n] = AbsDon[n-1] + DON[n] - DON[n-1]$
 Si $(DON[n] < DON[n-1] \text{ y } DON[n-1] - DON[n] \geq 32768)$,
 $AbsDon[n] = AbsDon[n-1] + 65536 - DON[n-1] + DON[n]$
 Si $(DON[n] > DON[n-1] \text{ y } DON[n] - DON[n-1] \geq 32768)$,
 $AbsDon[n] = AbsDon[n-1] - (DON[n-1] + 65536 - DON[n])$
 Si $(DON[n] < DON[n-1] \text{ y } DON[n-1] - DON[n] < 32768)$,
 $AbsDon[n] = AbsDon[n-1] - (DON[n-1] - DON[n])$

[0055] Para cualesquiera dos unidades NAL m y n, se aplica lo siguiente:

- 5 - $AbsDon[n]$ mayor que $AbsDon[m]$ indica que la unidad NAL n sigue a la unidad NAL m en el orden de decodificación de la unidad NAL.
- Cuando $AbsDon[n]$ es igual a $AbsDon[m]$, el orden de decodificación de la unidad NAL de las dos unidades NAL puede ser en cualquier orden.
- 10 - $AbsDon[n]$ menor que $AbsDon[m]$ indica que la unidad NAL n precede a la unidad NAL m en el orden de decodificación.

[0056] Cuando dos unidades NAL consecutivas en el orden de decodificación de la unidad NAL tienen diferentes valores de $AbsDon$, el valor de $AbsDon$ para la segunda unidad NAL en el orden de decodificación DEBE ser mayor que el valor de $AbsDon$ para la primera unidad NAL, y la diferencia absoluta entre los dos valores $AbsDon$ PUEDE ser mayor o igual a 1.

[0057] Ahora se describirán los paquetes de agregación (AP). La FIG. 2 muestra una representación visual de una estructura de paquetes de agregación. El paquete de agregación 120 incluye una cabecera 122 de la carga útil (indicada como PayloadHdr) seguida por los datos de la carga útil 124. Los datos de la carga útil incluyen una o más unidades de agregación, que se muestran como paquetes de agregación 0 a N en la FIG. 2. Cada unidad de agregación puede incluir unidades NAL. Por ejemplo, la unidad de agregación 0 incluye la unidad NAL 0, la unidad de agregación 1 incluye la unidad NAL 1, y la unidad de agregación N incluye la unidad NAL N. La FIG. 2 también muestra los primeros 16 bits de la cabecera de la carga útil, que incluye un bit F, un campo TYPE, un campo R (también denominado a veces campo LayerId) y un campo TID.

[0058] Se introdujeron los AP para permitir la reducción de la sobrecarga de empaquetado para las unidades NAL pequeñas, como la mayoría de las unidades NAL no VCL, que son a menudo solo unos pocos octetos de tamaño. Un AP agrega unidades NAL dentro de una unidad de acceso. Cada unidad NAL que se transportará en un AP está encapsulada en una unidad de agregación. Las unidades NAL agregadas en un AP están en orden de decodificación de las unidades NAL. Un AP puede consistir en una cabecera de la carga útil (denotada como PayloadHdr) seguida de una o más unidades de agregación.

[0059] Los campos de la cabecera de la carga útil se establecen como sigue. El bit F DEBE ser igual a 0 si el bit F de cada unidad NAL agregada es igual a cero; de lo contrario, DEBE ser igual a 1. El campo Tipo DEBE ser igual a 48. El valor de LayerId DEBE ser igual al valor más bajo de LayerId de todas las unidades NAL agregadas. El valor de TID DEBE ser el valor más bajo de TID de todas las unidades NAL agregadas.

[0060] Un AP puede llevar a tantas unidades de agregación como sea necesario; sin embargo, la cantidad total de datos en un AP obviamente DEBE caber en un paquete IP, y el tamaño DEBE ser elegido para que el paquete IP resultante sea más pequeño que el tamaño de MTU para evitar la fragmentación de la capa IP. Un AP NO DEBE contener FU. Los AP NO DEBEN estar anidados; es decir, un AP NO DEBE contener otro AP.

[0061] La primera unidad de agregación en un AP puede consistir en un campo DONL opcional de 16 bits (en el orden de bytes de red) seguido de una información sin signo de 16 bits (en el orden de bytes de red) que indica el tamaño de la unidad NAL en bytes (excluyendo estos dos octetos, pero incluyendo la cabecera de la unidad NAL), seguidos por la propia unidad NAL, incluida su cabecera de unidad NAL.

[0062] El campo DONL, cuando está presente, especifica el valor de los 16 bits menos significativos del número de orden de decodificación de la unidad NAL agregada.

[0063] Si el modo tx es igual a "MST" o sprop-depack-buf-nalus es mayor que 0, el campo DONL DEBE estar presente en una unidad de agregación que es la primera unidad de agregación en un AP, y la variable DON para la unidad NAL agregada se obtiene como igual al valor del campo DONL. De lo contrario (el modo tx es igual a "SST" y sprop-depack-buf-nalus es igual a 0), el campo DONL NO DEBE estar presente en una unidad de agregación que es la primera unidad de agregación en un AP.

- 5 **[0064]** Una unidad de agregación que no es la primera unidad de agregación en un AP puede consistir en un campo DOND opcional de 8 bits seguido de una información sin signo de 16 bits (en el orden de bytes de red) que indica el tamaño de la unidad NAL en bytes (excluyendo estos dos octetos, pero incluyendo la cabecera de la unidad NAL), seguidos por la propia unidad NAL, incluida su cabecera de unidad NAL.
- 10 **[0065]** Cuando está presente, el campo DOND más 1 puede especificar la diferencia entre los valores de número de orden de decodificación de la unidad NAL agregada actual y la unidad NAL agregada precedente en el mismo AP. A diferencia de las estructuras de la carga útil que requieren que las unidades NAL se decodifiquen en el orden en que aparecen en los paquetes RTP, el uso de los parámetros DOND y DONL descritos en esta divulgación puede permitir que se especifique un orden de decodificación específico.
- 15 **[0066]** Si el modo tx es igual a "MST" o sprop-depack-buf-nalus es mayor que 0, el campo DOND DEBE estar presente en una unidad de agregación que no es la primera unidad de agregación en un AP, y la variable DON para la unidad NAL agregada se obtiene como igual al DON de la unidad NAL agregada precedente en el mismo AP más el valor del campo DOND más 1 módulo 65536. De lo contrario (el modo tx es igual a "SST" y sprop-depack-buf-nalus es igual a 0), el campo DOND NO DEBE estar presente en una unidad de agregación que no sea la primera unidad de agregación en un AP.
- 20 **[0067]** En una alternativa, el campo DOND puede ser de una longitud diferente, por ejemplo, 4 bits. En otra alternativa, dos unidades de agregación no primeras comparten un campo de 8 bits, 4 bits para que cada unidad de agregación señalice el valor DOND. En otra alternativa más, la longitud del campo DOND se señala mediante un parámetro de tipo de medio, y el valor de ese parámetro igual a 0 significa que el campo DOND no está presente.
- 25 **[0068]** A continuación se describirán las Unidades de Fragmentación (UF). Las unidades de fragmentación (FU) se introducen para permitir la fragmentación de una única unidad NAL en múltiples paquetes RTP. Un fragmento de una unidad NAL puede consistir en un número entero de octetos consecutivos de esa unidad NAL. Los fragmentos de la misma unidad NAL DEBEN enviarse en orden consecutivo con números de secuencia RTP ascendentes (sin que se envíen otros paquetes RTP dentro del mismo flujo de paquetes RTP entre el primer y el último fragmento).
- 30 **[0069]** Cuando una unidad NAL es fragmentada y transportada dentro de una UF, que se conoce como una unidad NAL fragmentada. Los AP NO DEBEN estar fragmentados. Las FU NO DEBEN estar anidadas; es decir, una FU NO DEBE contener otra FU.
- 35 **[0070]** La marca de tiempo de RTP de un paquete RTP que lleva una FU se establece en el tiempo NALU de la unidad NAL fragmentada.
- 40 **[0071]** Una FU puede consistir en una cabecera de la carga útil (denotada como PayloadHdr), una cabecera de la FU de un octeto, un campo DONL de 16 bits opcional (en el orden de bytes de red), y una carga útil de una FU.
- 45 **[0072]** Los campos de la cabecera de la carga útil se establecen como sigue. El campo Tipo DEBE ser igual a 49. Los campos F, LayerId y TID DEBEN ser iguales a los campos F, LayerId y TID, respectivamente, de la unidad NAL fragmentada.
- 50 **[0073]** La cabecera de FU puede consistir en un bit S, un bit E, y un campo Tipo de 6 bits.
- [0074]** En este ejemplo, la semántica de los campos de la cabecera de FU son los siguientes:
- S: 1 bit
- Cuando se establece a uno, el bit S indica el inicio de una unidad NAL fragmentada, es decir, el primer byte de la carga útil de FU es también el primer byte de la carga útil de la unidad NAL fragmentada. Cuando la carga útil de FU no es el inicio de la carga útil de la unidad NAL fragmentada, el bit S DEBE establecerse a cero.
- 55 E: 1 bit
- Cuando se establece a uno, el bit E indica el final de una unidad NAL fragmentada, es decir, el último byte de la carga útil es también el último byte de la unidad NAL fragmentada. Cuando la carga útil de FU no es el último fragmento de una unidad NAL fragmentada, el bit E DEBE establecerse a cero.
- 60 Tipo: 6 bits
- El campo Tipo DEBE ser igual al campo Tipo de la unidad NAL fragmentada.
- 65 **[0075]** El campo DONL, cuando está presente, puede especificar el valor de los 16 bits menos significativos del número de orden de decodificación de la unidad NAL fragmentada.

5 **[0076]** Si el modo tx es igual a "MST" o sprop-depack-buf-nalus es mayor que 0, y el bit S es igual a 1, el campo DONL DEBE estar presente en la FU, y la variable DON para la unidad NAL fragmentada se obtiene como igual al valor del campo DONL. De lo contrario (el modo tx es igual a "SST" y sprop-depack-buf-nalus es igual a 0, o el bit S es igual a 0), el campo DONL NO DEBE estar presente en la FU.

[0077] Una unidad NAL no fragmentada NO DEBE ser transmitida en una FU; es decir, el bit de inicio y el bit final NO DEBEN establecerse a uno en la misma cabecera de FU.

10 **[0078]** La carga útil de FU puede consistir en fragmentos de la carga útil de la unidad NAL fragmentada de manera que si las cargas útiles de FU de FU consecutivas, que comienzan con una FU con el bit S igual a 1 y terminan con una FU con el bit E igual a 1, se concatenan secuencialmente, la carga útil de la unidad NAL fragmentada se puede reconstruir. La cabecera de la unidad NAL de la unidad NAL fragmentada no se incluye como tal en la carga útil de FU, sino que la información de la cabecera de la unidad NAL de la unidad NAL fragmentada se transmite en los
15 campos F, LayerId y TID de las cabeceras de la carga útil FU de las FU y el campo Tipo de la cabecera de FU de las FU. Una carga útil de FU PUEDE tener cualquier cantidad de octetos y PUEDE estar vacía.

20 **[0079]** Si se pierde una FU, el receptor DEBE descartar todas las siguientes unidades de fragmentación en orden de transmisión correspondientes a la misma unidad NAL fragmentada, a menos que el decodificador del receptor se caracterice por estar preparado para manejar sin problemas unidades NAL incompletas.

25 **[0080]** Un receptor en un punto final o en un MANE PODRÁ agregar los primeros n-1 fragmentos de una unidad NAL a una unidad NAL (incompleta), incluso si no se recibe el fragmento n de esa unidad NAL. En este caso, el forbidden_zero_bit de la unidad NAL DEBE establecerse a uno para indicar una violación de sintaxis.

[0081] Ahora se expondrán las reglas de empaquetado. Se aplican las siguientes reglas de empaquetado:

30 - Si el modo tx es igual a "MST" o sprop-depack-buf-nalus es mayor que 0 para una sesión de RTP, el orden de transmisión de las unidades NAL transportadas en la sesión de RTP PUEDE ser diferente del orden de decodificación de la unidad NAL. De lo contrario (el modo tx es igual a "SST" y sprop-depack-buf-nalus es igual a 0 para una sesión de RTP), el orden de transmisión de las unidades NAL transportadas en la sesión de RTP DEBE ser el mismo que el orden de decodificación de las unidades NAL.

35 - Cuando el modo tx es igual a "MST" o sprop-depack-buf-nalus es mayor que 0, no se pueden usar paquetes de unidades NAL individuales. En este caso, un AP PUEDE ser utilizado para encapsular una sola unidad NAL en un paquete RTP.

40 - Una unidad NAL de pequeño tamaño DEBERÍA estar encapsulada en un paquete de agregación junto con una o más unidades NAL adicionales para evitar la sobrecarga innecesaria de paquetes para unidades NAL pequeñas. Por ejemplo, las unidades NAL no VCL, como los delimitadores de la unidad de acceso, los conjuntos de parámetros o las unidades NAL SEI son típicamente pequeñas.

45 - Cada unidad NAL no VCL DEBERÍA estar encapsulada en un paquete de agregación junto con su unidad NAL VCL asociada, ya que típicamente una unidad NAL no VCL no tendría sentido sin que la unidad NAL VCL asociada esté disponible.

50 - El valor TID indica la importancia relativa de un paquete RTP. Un valor inferior de TID indica una mayor importancia. Las unidades NAL más importantes PUEDEN estar mejor protegidas contra las pérdidas de transmisión que las unidades NAL menos importantes.

[0082] A continuación se describirá un proceso de desempaqueado. El concepto general detrás del desempaqueado es sacar las unidades NAL de los paquetes RTP en una sesión de RTP y todas las sesiones de RTP dependientes, si las hay, y pasarlas al decodificador en el orden de decodificación de las unidades NAL.

55 **[0083]** El proceso de desempaqueado es dependiente de la implementación. Por lo tanto, la siguiente descripción debe verse como un ejemplo de una implementación adecuada. Se pueden usar otros esquemas, siempre y cuando la salida para la misma entrada sea la misma que el proceso descrito a continuación. El resultado es el mismo, lo que significa que el conjunto de unidades NAL y su orden son idénticos. Optimizaciones relativas a los algoritmos descritos son posibles.

60 **[0084]** Se aplican todos los mecanismos de RTP normales relacionados con la gestión de la memoria intermedia. En particular, se eliminan los paquetes RTP duplicados u obsoletos (como se indica mediante el número de secuencias de RTP y la marca de tiempo de RTP). Para determinar el tiempo exacto de decodificación, se deben tener en cuenta factores tales como un posible retraso intencional para permitir una sincronización entre secuencias
65 adecuadas.

[0085] Solo las unidades NAL con valores de tipo unidad NAL en el intervalo de 0 a 47, inclusive PUEDEN pasarse al decodificador. Las estructuras tipo unidad NAL con valores de tipo de unidad NAL en el intervalo de 48 a 63, inclusive, NO DEBEN pasarse al decodificador.

5 **[0086]** El receptor incluye una memoria intermedia de receptor, que se utiliza para compensar la fluctuación del retardo de transmisión, para reordenar las unidades NAL a partir del orden de transmisión al orden de decodificación de unidades NAL, y para la recuperación del orden de decodificación de las unidades NAL en MST, cuando sea aplicable. En esta sección, la operación del receptor se describe bajo la suposición de que no hay fluctuación del retardo de transmisión. Para marcar la diferencia con respecto a una memoria intermedia de recepción práctica que también se usa para la compensación de la fluctuación del retardo de transmisión, la memoria intermedia de recepción se llama en esta memoria la memoria intermedia de desempaquetado en esta sección. Los receptores DEBERÍAN prepararse también para la fluctuación del retardo de transmisión; es decir, bien reservar memorias intermedias separadas para el almacenamiento intermedio de la fluctuación del retardo de la transmisión y el almacenamiento intermedio del desempaquetado, o bien utilizar una memoria intermedia de recepción tanto para la fluctuación del retardo de transmisión como para el desempaquetado. Además, los receptores DEBERÍAN tener en cuenta la fluctuación del retardo de transmisión en la operación de almacenamiento intermedio; por ejemplo, mediante almacenamiento intermedio inicial adicional antes del inicio de la decodificación y la reproducción.

10 **[0087]** Hay dos estados de almacenamiento intermedio en el receptor: el almacenamiento temporal inicial y el almacenamiento temporal durante la reproducción. El almacenamiento intermedio inicial comienza cuando se inicializa la recepción. Después del almacenamiento intermedio inicial, se inician la decodificación y la reproducción, y se utiliza el modo de almacenamiento intermedio mientras se reproduce.

15 **[0088]** El receptor almacena paquetes entrantes en el orden de recepción en la memoria intermedia de recepción y pasa las unidades NAL en los paquetes RTP de cada sesión en el orden de número de secuencia de RTP para la memoria intermedia de re-multiplexación. Se calcula y almacena el valor CS-DON para cada unidad NAL en la memoria intermedia de remultiplexación.

20 **[0089]** Con independencia del estado de la memoria intermedia, el receptor almacena las unidades NAL entrantes, en orden de recepción, en la memoria intermedia de desempaquetado. Las unidades NAL transportadas en paquetes de unidades NAL individuales, los AP y las FU se almacenan individualmente en la memoria intermedia de desempaquetado, y se calcula y almacena el valor de AbsDon para cada unidad NAL.

25 **[0090]** El almacenamiento intermedio inicial dura hasta que la condición A (el número de unidades NAL en la memoria intermedia de desempaquetado es mayor que el valor de sprop-depack-buf-nalus de la sesión de RTP más alta) es verdadera. Después del almacenamiento intermedio inicial, siempre que la condición A sea verdadera, la siguiente operación se aplica repetidamente hasta que la condición A se convierte en falsa: La unidad NAL en la memoria intermedia de desempaquetado con el menor valor de AbsDon se elimina de la memoria intermedia de desempaquetado y se pasa al decodificador.

30 **[0091]** Cuando ya no fluyen más unidades NAL a la memoria intermedia de desempaquetado, todas las unidades NAL que quedan en la memoria intermedia de desempaquetado son eliminadas de la memoria intermedia y se pasan al decodificador en el orden de los valores crecientes de AbsDon.

35 **[0092]** A continuación se describirá el registro del tipo de medio. El subtipo de medio para el códec HEVC se asigna desde el árbol IETF.

[0093] El receptor DEBE ignorar cualquier parámetro no especificado.

40 Nombre del tipo de medio: vídeo

Nombre del subtipo de medio: H265

Parámetros requeridos: ninguno

45 Parámetros OPCIONALES:

Modo tx:

50 Este parámetro indica si el modo de transmisión es SST o MST. Este parámetro puede ser un parámetro de tipo de medio que se aplica a todos los paquetes en una sesión particular. En otras palabras, el valor puede ser fijo para todos los paquetes de la sesión.

55 El valor de modo tx DEBE ser igual a "MST" o "SST". Si no está presente, se deduce que el valor del modo tx es igual a "SST".

Si el valor es igual a "MST", MST DEBE estar en uso. De lo contrario (el valor es igual a "SST"), SST DEBE estar en uso.

El valor de modo tx DEBE ser igual a "MST" para todas las sesiones de RTP en un MST.

5

sprop-depack-buf-nalus:

Este parámetro puede especificar el número máximo de unidades NAL que preceden a una unidad NAL en la memoria intermedia de desempaqueado en el orden de recepción y siguen a la unidad NAL en el orden de decodificación. Este parámetro puede ser un parámetro de tipo de medio que se aplica a todos los paquetes en una sesión particular. En otras palabras, el valor puede ser fijo para todos los paquetes de la sesión.

10

El valor de sprop-depack-buf-nalus DEBE ser un entero en el intervalo de 0 a 32767, inclusive.

15

Cuando no está presente, el valor de sprop-depack-buf-nalus se deduce que es igual a 0.

Cuando la sesión de RTP depende de una o más sesiones de RTP (en este caso el modo tx DEBE ser igual a "MST"), el valor de sprop-depack-buf-nalus DEBE ser mayor que 0.

20

sprop-depack-buf-bytes:

Este parámetro indica el tamaño requerido de la memoria intermedia de desempaqueado en unidades de bytes. El valor del parámetro DEBE ser mayor o igual que la ocupación máxima de la memoria intermedia (en unidades de bytes) de la memoria intermedia de desempaqueado como se especifica en la sección 6.

25

El valor de sprop-depack-buf-bytes DEBE ser un entero en el intervalo de 0 a 4294967295, inclusive.

depack-buf-cap:

30

Este parámetro señala las capacidades de una implementación de receptor e indica la cantidad de espacio de la memoria intermedia de desempaqueado en unidades de bytes que el receptor tiene disponible para reconstruir el orden de decodificación de la unidad NAL. Un receptor puede manejar cualquier flujo para el cual el valor del parámetro sprop-depack-buf-bytes sea menor o igual que este parámetro.

35

Si no está presente, se deduce que el valor de depack-buf-req es igual a 0. El valor de depack-buf-cap DEBE ser un entero en el intervalo de 0 a 4294967295, inclusive.

40

[0094] La FIG. 3 es un diagrama de bloques que ilustra un codificador de vídeo 20 a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación. El codificador de vídeo 20 puede realizar la codificación intra y la codificación inter de bloques de vídeo dentro de fragmentos de vídeo. La codificación intra se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La codificación inter se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la bi-predicción (modo B), pueden referirse a cualquiera de varios modos de compresión de base temporal.

45

50

[0095] En el ejemplo de la FIG. 3, el codificador de vídeo 20 incluye una unidad de división 35, una unidad de procesamiento de predicción 41, una unidad de filtro 63, una memoria de imágenes 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54 y una unidad de codificación de entropía 56. La unidad de procesamiento de predicción 41 incluye la unidad de estimación de movimiento 42, la unidad de compensación de movimiento 44 y la unidad de procesamiento de intra-predicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de procesamiento de transformación inversa 60 y un sumador 62. La unidad de filtro 63 está destinada a representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento de muestras adaptativo (SAO). Aunque la unidad de filtro 63 se muestra en la FIG. 3 como un filtro de bucle de entrada, en otras configuraciones, la unidad de filtro 63 puede implementarse como un filtro de bucle posterior. La FIG. 3 también muestra el dispositivo de procesamiento posterior 57 que puede realizar un procesamiento adicional en los datos de vídeo codificados generados por el codificador de vídeo 20. Las técnicas de esta divulgación pueden implementarse en algunos casos mediante el codificador 20 de vídeo. En otros casos, sin embargo, las técnicas de esta divulgación pueden implementarse mediante el dispositivo de procesamiento posterior 57. Por ejemplo, las técnicas descritas con respecto al empaquetador 21 de la FIG. 1 pueden, en algunos casos, ser realizadas por un empaquetador del dispositivo de procesamiento posterior 57.

55

60

65

[0096] Como se muestra en la FIG. 3, el codificador de vídeo 20 recibe datos de vídeo y la unidad de división 35 divide los datos en bloques de vídeo. Esta división también puede incluir la división en fragmentos, elementos u otras unidades mayores, así como la división de bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de unas LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo de un fragmento de vídeo que se va a codificar. El fragmento puede dividirse en varios bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados elementos). La unidad de procesamiento de predicción 41 puede seleccionar una entre una pluralidad de posibles modos de codificación, tal como una entre una pluralidad de modos de codificación intra, o una de una pluralidad de modos de codificación inter, para el bloque de vídeo actual, basándose en resultados de errores (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloques residuales, y al sumador 62 para reconstruir el bloque codificado para su uso como una imagen de referencia.

[0097] La unidad de procesamiento de intra-predicción 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación predictiva intra del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque actual que va a codificarse, para proporcionar una compresión espacial. La unidad de estimación del movimiento 42 y la unidad de compensación del movimiento 44 de la unidad de procesamiento de predicción 41 realizan la codificación de inter-predicción del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia, para proporcionar una compresión temporal.

[0098] La unidad de estimación del movimiento 42 puede estar configurada para determinar el modo de inter-predicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo de la secuencia como fragmentos P, fragmentos B o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo de una trama o imagen de vídeo actual con respecto a un bloque predictivo de una imagen de referencia.

[0099] Un bloque predictivo es un bloque que se descubre que se corresponde estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, que puede determinarse mediante la suma de diferencias absolutas (SAD), la suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular los valores para las posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y emitir un vector de movimiento con una precisión de píxel fraccionaria.

[0100] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de entre una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación de entropía 56 y a la unidad de compensación de movimiento 44.

[0101] La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar obtener o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento, posiblemente realizando interpolaciones hasta la precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, generando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia de luminancia y croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

[0102] La unidad de procesamiento de intra-predicción 46 puede intra-prededir un bloque actual, de forma alternativa a la inter-predicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intra-predicción 46 puede determinar un modo de intra-predicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad

de procesamiento de intra-predicción 46 puede codificar un bloque actual usando varias modalidades de intra-predicción, por ejemplo, durante diferentes pases de codificación, y la unidad de procesamiento de intra-predicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar un modo adecuado de intra-predicción para utilizar a partir de los modos probados. Por ejemplo, la unidad de procesamiento de intra-predicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intra-predicción probados, y seleccionar el modo de intra-predicción que tenga las mejores características de velocidad-distorsión entre los modos probados. El análisis de velocidad-distorsión determina, en general, una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para generar el bloque codificado, así como una velocidad de bits (es decir, un número de bits) utilizada para generar el bloque codificado. La unidad de procesamiento de intra-predicción 46 puede calcular proporciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intra-predicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0103] En cualquier caso, tras seleccionar un modo de intra-predicción para un bloque, la unidad de procesamiento de intra-predicción 46 puede proporcionar información que indica el modo de intra-predicción seleccionado para el bloque, a la unidad de codificación de entropía 56. La unidad de codificación de entropía 56 puede codificar la información que indica el modo de intra-predicción seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir datos de configuración en el flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modos de intra-predicción y una pluralidad de tablas de índices de modos de intra-predicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones del modo de intra-predicción más probable, una tabla de índices de modos de intra-predicción y una tabla de índices de modos de intra-predicción modificados a utilizar para cada uno de los contextos.

[0104] Después de que la unidad de procesamiento de predicción 41 genera el bloque predictivo para el bloque de vídeo actual, ya sea mediante la inter-predicción o la intra-predicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más TU y aplicarse a la unidad de procesamiento de transformación 52. La unidad de procesamiento de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales mediante una transformación, tal como una transformación de coseno discreta (DCT) o una transformación similar desde un punto de vista conceptual. La unidad de procesamiento de transformación 52 puede convertir los datos de vídeo residuales de un dominio de píxeles a un dominio de las transformadas, tal como un dominio de frecuencia.

[0105] La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o a la totalidad de los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar, a continuación, un escaneado de la matriz que incluye los coeficientes de transformación cuantificados. De forma alternativa, la unidad de codificación de entropía 56 puede llevar a cabo el escaneado.

[0106] Tras la cuantificación, la unidad de codificación de entropía 56 puede realizar la codificación de entropía de los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación de entropía 56 puede realizar una codificación de longitud variable adaptativa según el contexto (CAVLC), una codificación aritmética binaria adaptativa según el contexto (CABAC), una codificación aritmética binaria adaptativa según el contexto basándose en la sintaxis (SBAC), una codificación de entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos o técnicas de codificación de entropía. Tras la codificación de entropía realizada por la unidad de codificación de entropía 56, el flujo de bits codificado puede transmitirse al decodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación por el decodificador de vídeo 30. La unidad de codificación de entropía 56 también puede realizar la codificación de entropía de los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

[0107] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de los píxeles, para su posterior uso como bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las imágenes de referencia de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios y usarlos en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque predictivo con compensación de movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes 64. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar el bloque de referencia como un bloque de referencia para realizar la inter-predicción de un bloque en una imagen o trama de vídeo subsiguiente.

[0108] La FIG. 4 es un diagrama de bloques que ilustra una entidad de red 79 y un decodificador de vídeo 30 a modo de ejemplo que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 4, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantificación inversa 86, una unidad de procesamiento de transformación inversa 88, un sumador 90, una unidad de filtro 91 y una memoria de imágenes 92. La unidad de procesamiento de predicción 81 incluye la unidad de compensación de movimiento 82 y la unidad de procesamiento de intra-predicción 84. En algunos ejemplos, el decodificador de vídeo 30 puede realizar un pase de decodificación en general recíproca al pase de codificación descrito con respecto al codificador de vídeo 20 de la FIG. 3.

[0109] Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados del codificador de vídeo 20. El decodificador de vídeo 30 puede recibir el flujo de bits de vídeo codificados desde una entidad de red 79. La entidad de red 79 puede ser, por ejemplo, un servidor, un MANE, un editor/empalmador de vídeo u otro dispositivo similar configurado para implementar una o más de las técnicas descritas anteriormente. La entidad de red 79 puede incluir o no el codificador de vídeo 20. Como se describió anteriormente, algunas de las técnicas descritas en esta divulgación pueden ser implementadas por la entidad de red 79 antes de que la entidad de red 79 transmita el flujo de bits de vídeo codificado al decodificador de vídeo 30. En algunos sistemas de decodificación de vídeo, la entidad de red 79 y el decodificador de vídeo 30 pueden ser partes de dispositivos separados, mientras que en otros casos, la funcionalidad descrita con respecto a la entidad de red 79 puede ser realizada por el mismo dispositivo que comprende el decodificador de vídeo 30. Aunque la FIG. 1 muestra el desempaquetador 29 como parte del dispositivo de destino 14, las técnicas descritas anteriormente con respecto al desempaquetador 29 también pueden ser realizadas por un desempaquetador dentro de la entidad de red 79.

[0110] Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa los bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados del codificador de vídeo 20. Los bloques de vídeo pueden, por ejemplo, encaminarse desde el codificador de vídeo 20 al decodificador de vídeo 30 a través de uno o más MANE, como el MANE 27 de la FIG. 1 o la entidad de red 79 de la FIG. 4. La unidad de decodificación de entropía 80 del decodificador de vídeo 30 realiza la decodificación de entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de decodificación de entropía 80 remite los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.

[0111] Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de procesamiento de intra-predicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de intra-predicción señalizada y datos de los bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo es codificada como un fragmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación de entropía 80. Los bloques predictivos pueden ser generados a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por defecto, basándose en las imágenes de referencia almacenadas en la memoria de imágenes 92.

[0112] La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que está siendo decodificado. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intra-predicción o inter-predicción), usada para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de inter-predicción (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de inter-predicción para cada bloque de vídeo inter-codificado del fragmento y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

[0113] La unidad de compensación de movimiento 82 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

[0114] La unidad de cuantificación inversa 86 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformación cuantificados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado

5 por el codificador de vídeo 20 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 88 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada inversa entera o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada, con el fin de generar bloques residuales en el dominio de píxeles.

10 **[0115]** Después de que la unidad de compensación de movimiento 82 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformación inversa 88 a los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también pueden utilizarse filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. La unidad de filtro 91 está destinada a representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF) y un filtro de desplazamiento de muestras adaptativo (SAO). Aunque la unidad de filtro 91 se muestra en la FIG. 4 como un filtro de bucle de entrada, en otras configuraciones, la unidad de filtro 91 puede implementarse como un filtro de bucle posterior. Los bloques de vídeo decodificados en una trama o imagen dada son a continuación almacenados en la memoria de imágenes 92, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes 92 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

25 **[0116]** La FIG. 5 es un diagrama de bloques que ilustra un conjunto de dispositivos a modo de ejemplo que forman parte de la red 150. En este ejemplo, la red 150 incluye los dispositivos de encaminamiento 154A, 154B (dispositivos de encaminamiento 154) y el dispositivo de transcodificación 156. Los dispositivos de encaminamiento 154 y el dispositivo de transcodificación 156 están concebidos para representar un pequeño número de dispositivos que pueden formar parte de la red 150. Otros dispositivos de red, tales como conmutadores, concentradores, pasarelas, cortafuegos, puentes y otros dispositivos de ese tipo también pueden estar incluidos dentro de la red 150. Además, pueden proporcionarse dispositivos de red adicionales a lo largo de un trayecto de red entre el dispositivo servidor 152 y el dispositivo cliente 158. El dispositivo servidor 152 puede corresponder al dispositivo de origen 12 (FIG. 1), mientras que el dispositivo cliente 158 puede corresponder al dispositivo de destino 14 (FIG. 1), en algunos ejemplos. Los dispositivos de encaminamiento 154 pueden, por ejemplo, estar configurados para MANE para desviar datos de medios.

35 **[0117]** En general, los dispositivos de encaminamiento 154 implementan uno o más protocolos de encaminamiento para intercambiar datos de red a través de la red 150. En general, los dispositivos de encaminamiento 154 ejecutan protocolos de encaminamiento para descubrir rutas a través de la red 150. Al ejecutar tales protocolos de encaminamiento, el dispositivo de encaminamiento 154B puede descubrir una ruta de red desde sí mismo hasta el dispositivo servidor 152, mediante el dispositivo de encaminamiento 154A. Los diversos dispositivos de la FIG. 5 representan ejemplos de dispositivos que pueden implementar las técnicas de esta divulgación y pueden configurarse para procesar datos de RTP de acuerdo con las técnicas de esta divulgación.

45 **[0118]** La FIG. 6 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 6 pueden, por ejemplo, ser realizadas por un dispositivo tal como el dispositivo de destino 14, y más particularmente, pueden ser realizadas por el desempaquetador 29 del dispositivo de destino 14. El desempaquetador 29 recibe el primer paquete de agregación de acuerdo con un protocolo de RTP (160). El primer paquete de agregación puede, por ejemplo, incluir una cabecera de carga útil y una o más unidades de agregación. El desempaquetador 29 puede analizar una primera unidad de agregación para determinar un valor para un primer parámetro (162). El primer parámetro puede, por ejemplo, corresponder al parámetro DONL descrito anteriormente y puede especificar un número de orden de decodificación. El desempaquetador 29 puede analizar una segunda unidad de agregación para determinar un valor para un segundo parámetro (164). La segunda unidad de agregación puede seguir a la primera unidad de agregación, y el segundo parámetro puede, por ejemplo, corresponder al parámetro DOND descrito anteriormente. En base al primer parámetro y al segundo parámetro, el desempaquetador 29 determina un orden de decodificación para la segunda unidad de agregación.

55 **[0119]** La FIG. 7 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 7 pueden, por ejemplo, realizarse mediante un dispositivo tal como el dispositivo de origen 12, y más particularmente, pueden realizarse mediante el empaquetador 21 del dispositivo de origen 12. El empaquetador 21 recibe una o más unidades NAL y empaqueta la una o más unidades NAL en un primer paquete de agregación de acuerdo con un protocolo RTP (170). El primer paquete de agregación puede, por ejemplo, incluir una cabecera de carga útil y una o más unidades de agregación. El empaquetador 21 establece un valor para un primer parámetro de una primera unidad de agregación basándose en un número de orden de decodificación para la unidad NAL incluida en la primera unidad de agregación (172). El primer parámetro puede, por ejemplo, corresponder al parámetro DONL descrito anteriormente y puede especificar un número de orden de decodificación. El primer parámetro puede, por ejemplo, especificar un valor de un número de bits menos significativos del número de orden de decodificación. Basándose en una diferencia entre un orden de decodificación

para la unidad NAL incluido en la segunda unidad de agregación y el número de orden de decodificación para la unidad NAL incluido en la primera unidad de agregación, el empaquetador 21 puede establecer un valor para un segundo parámetro de una segunda unidad de agregación (174). La segunda unidad de agregación puede seguir a la primera unidad de agregación, y el segundo parámetro puede, por ejemplo, corresponder al parámetro DOND descrito anteriormente. El segundo parámetro puede, por ejemplo, identificar una diferencia entre el primer parámetro y el número de orden de decodificación.

[0120] La FIG. 8 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 8 pueden, por ejemplo, realizarse mediante un dispositivo tal como el dispositivo de destino 14, y más particularmente, pueden realizarse mediante el desempaketador 29 del dispositivo de destino 14. El desempaketador 29 recibe una primera unidad de fragmentación que incluye un subconjunto de una unidad NAL fragmentada (180). El desempaketador 29 analiza un bit de inicio de la unidad de fragmentación para determinar si la primera unidad de fragmentación incluye un inicio de la unidad NAL fragmentada (182). El bit de inicio puede ser, por ejemplo, un bit S como se describió anteriormente. En respuesta a la primera unidad de fragmentación que incluye el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor, el desempaketador 29 analiza un segundo parámetro para determinar un orden de decodificación para la unidad NAL fragmentada. El primer parámetro puede ser, por ejemplo, un parámetro sprop-depack-buf-nalus como se describió anteriormente, y el primer valor puede ser cero. El segundo parámetro puede ser, por ejemplo, un parámetro DONL como se describió anteriormente. El dispositivo de destino 14 puede decodificar la unidad NAL fragmentada basándose en el orden de decodificación determinado (186).

[0121] La FIG. 9 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 9 pueden realizarse, por ejemplo, mediante un dispositivo tal como el dispositivo de origen 12, y más particularmente, pueden realizarse mediante el empaquetador 21 del dispositivo de origen 12. El empaquetador 21 genera la primera unidad de fragmentación que comprende un subconjunto de una unidad NAL fragmentada (190). La primera unidad de fragmentación, por ejemplo, incluye un inicio de la unidad NAL fragmentada. El empaquetador 21 establece un bit de inicio de la unidad de fragmentación para indicar que la primera unidad de fragmentación incluye el inicio de la unidad NAL fragmentada (192). El bit de inicio puede ser, por ejemplo, un bit S como se describió anteriormente. En respuesta a la primera unidad de fragmentación que incluye el inicio de la unidad NAL fragmentada y uno o ambos de un modo de transmisión para la primera unidad de fragmentación que es un modo de transmisión multisesión y un primer parámetro que es mayor que un primer valor. El empaquetador 21 establece un segundo parámetro para indicar un orden de decodificación para la unidad NAL fragmentada. El primer parámetro puede ser, por ejemplo, un parámetro sprop-depack-buf-nalus como se describió anteriormente, y el primer valor puede ser cero. El segundo parámetro puede ser, por ejemplo, un parámetro DONL como se describió anteriormente. El empaquetador 21 puede transmitir la unidad NAL fragmentada (196). El primer parámetro puede, por ejemplo, especificar un número máximo de unidades NAL que preceden a la primera unidad NAL en una memoria intermedia de desempaketado en orden de recepción y sigue la primera unidad NAL en un orden de decodificación, y el segundo parámetro puede especificar un valor de un número de bits menos significativos del número de orden de decodificación.

[0122] La FIG. 10 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 8 pueden, por ejemplo, realizarse mediante un dispositivo tal como el dispositivo de destino 14, y más particularmente, pueden realizarse mediante el desempaketador 29 del dispositivo de destino 14. El desempaketador 29 recibe un primer paquete RTP que comprende una primera unidad NA (200). En respuesta a un modo de transmisión para el primer paquete RTP que es un modo de transmisión de sesión única y un primer parámetro igual a un primer valor, el desempaketador 29 determina un número de orden de decodificación para la primera unidad NAL basándose en un orden de transmisión de la primera unidad NAL (202). El primer parámetro puede ser, por ejemplo, un parámetro sprop-depack-buf-nalus como se describió anteriormente, y el valor puede ser igual a cero.

[0123] La FIG. 11 muestra un ejemplo de un procedimiento de procesamiento de datos de vídeo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 9 pueden realizarse, por ejemplo, mediante un dispositivo tal como el dispositivo de origen 12, y más particularmente, pueden realizarse mediante el empaquetador 21 del dispositivo de origen 12. El empaquetador 21 que genera una unidad RT (paquete que comprende una primera NA) (210). En respuesta a un modo de transmisión para el primer paquete RTP siendo un modo de transmisión de sesión única y un primer parámetro igual a un primer valor, estableciendo un orden de transmisión para la primera unidad NAL basada en un orden de decodificación para la primera unidad NAL (212). El primer parámetro puede ser, por ejemplo, un parámetro sprop-depack-buf-nalus como se describió anteriormente.

[0124] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse, como una o más instrucciones o código, en un medio legible por ordenador o transmitirse a través de este, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia

de un programa informático desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento tangibles legibles por ordenador que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0125] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender memoria RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del alcance de los medios legibles por ordenador.

[0126] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados de aplicación específica (ASIC), matrices de puertas programables in situ (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

[0127] Las técnicas de la esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

[0128] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de recepción de datos de vídeo, comprendiendo el procedimiento:

5 recibir (180) una primera unidad de fragmentación (FU) que comprende un subconjunto de una primera unidad de capa de abstracción de red fragmentada (NAL);

10 analizar (182) un bit de inicio de la primera FU para determinar si la primera FU comprende un inicio de la primera unidad NAL fragmentada;

determinar (184) si el modo de transmisión para la primera FU es un modo de transmisión multisesión que utiliza más de una sesión de protocolo de transporte en tiempo real (RTP);

15 en respuesta a determinar que la primera FU comprende el inicio de la primera unidad NAL fragmentada y el modo de transmisión para la primera FU es un modo de transmisión multisesión, determinar que un parámetro de orden de decodificación está presente en la primera FU, analizar (184) el parámetro de número de orden de decodificación para determinar un orden de decodificación para la primera unidad NAL fragmentada, y decodificar la primera unidad NAL fragmentada basándose en el orden de decodificación determinado;

20 en respuesta a determinar que la primera FU no comprende el inicio de la primera unidad NAL fragmentada y/o que el modo de transmisión para la primera FU no es un modo de transmisión multisesión, determinar que un parámetro de orden de decodificación no está presente en la primera FU y no analizar el parámetro de número de orden de decodificación para determinar un orden de decodificación para la primera unidad NAL fragmentada.

2. El procedimiento según la reivindicación 1, en el que el parámetro del orden de decodificación especifica un valor de un número de bits menos significativos del número de orden de decodificación.

30 3. El procedimiento según la reivindicación 2, en el que el número es 16.

4. El procedimiento según la reivindicación 1, que comprende además:

35 recibir una segunda FU que comprende un segundo subconjunto de la primera unidad NAL fragmentada;

analizar un bit final de la segunda FU para determinar si la segunda FU comprende un extremo de la primera unidad NAL fragmentada; y

40 en respuesta a la segunda unidad de fragmentación que comprende el extremo de la primera unidad NAL fragmentada, determinar que un parámetro de número de orden de decodificación no está presente en la segunda FU.

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

45 basándose en la primera FU y la segunda FU, reconstruir la primera unidad NAL fragmentada.

6. El procedimiento según la reivindicación 1, que comprende además:

50 recibir una tercera FU que comprende un subconjunto de una segunda unidad NAL fragmentada; y en respuesta a un modo de transmisión para que la tercera FU sea un modo de transmisión de sesión única, determinar que un parámetro de orden de decodificación no está presente en la segunda FU.

7. Un procedimiento de transmisión de datos de vídeo, comprendiendo el procedimiento:

55 generar (190) una primera unidad de fragmentación (FU) que comprende un subconjunto de una unidad de capa de abstracción de red fragmentada (NAL), en la que si la primera FU comprende un inicio de la unidad NAL fragmentada:

60 establecer (192) un bit de inicio de la primera FU para indicar que la primera FU comprende el inicio de la unidad NAL fragmentada;

65 si la primera FU que comprende el inicio de la unidad NAL fragmentada y si un modo de transmisión para la primera FU es un modo de transmisión multisesión, utilizando más de una sesión de protocolo de transporte en tiempo real, establecer (194) un parámetro de orden de decodificación para indicar un orden de decodificación para la unidad NAL fragmentada;

si la primera FU no comprende el inicio de la unidad NAL fragmentada y/o si un modo de transmisión para la primera FU no es un modo de transmisión multisesión, no establecer un parámetro de orden de decodificación para indicar un orden de decodificación para la unidad NAL fragmentada; y transmitir (196) la unidad NAL fragmentada.

- 5
8. Un dispositivo para recibir los datos de vídeo, comprendiendo el aparato:
- una memoria configurada para almacenar datos de vídeo;
- 10 un receptor configurado para recibir paquetes de protocolo de transporte en tiempo real (RTP); y
- uno o más procesadores configurados para:
- 15 recibir una primera unidad de fragmentación (FU) que comprende un subconjunto de una primera unidad de capa de abstracción de red (NAL) fragmentada;
- analizar un bit de inicio de la primera FU para determinar si la primera FU comprende un inicio de la primera unidad NAL fragmentada;
- 20 determinar si el modo de transmisión para la primera FU es un modo de transmisión multisesión utilizando más de una sesión de protocolo de transporte en tiempo real (RTP);
- en respuesta a determinar que la primera FU comprende el inicio de la primera unidad NAL fragmentada y el modo de transmisión para la primera FU es un modo de transmisión multisesión, determinar que un parámetro de orden de decodificación está presente en la primera FU, analizar el parámetro de número de orden de decodificación para determinar un orden de decodificación para la primera unidad NAL fragmentada y decodificar la primera unidad NAL fragmentada basándose en el orden de decodificación determinado;
- 25
- en respuesta a determinar que la primera FU no comprende el inicio de la primera unidad NAL fragmentada y/o que el modo de transmisión para la primera FU no es un modo de transmisión multisesión, determinar que un parámetro de orden de decodificación no está presente en la primera FU y no analizar el parámetro de número de orden de decodificación para determinar un orden de decodificación para la primera unidad NAL fragmentada.
- 30
- 35
9. El dispositivo según la reivindicación 8, en el que el parámetro de orden de decodificación especifica un valor de un número de bits menos significativos del número de orden de decodificación.
- 40
10. El dispositivo según la reivindicación 9, en el que el número es 16.
11. El dispositivo según la reivindicación 8, en el que el uno o más procesadores están configurados además para:
- 45 recibir una segunda FU analizar un bit final de la segunda FU para determinar si la segunda FU comprende un extremo de la primera unidad NAL fragmentada; y
- en respuesta a la segunda unidad de fragmentación que comprende el extremo de la primera unidad NAL fragmentada, determinar que un parámetro de número de orden de decodificación no está presente en la segunda FU.
- 50
12. El dispositivo según la reivindicación 11, en el que el uno o más procesadores están configurados además para:
- basándose en la primera FU y la segunda FU, reconstruir la primera unidad NAL fragmentada.
- 55
13. El dispositivo según la reivindicación 8, en el que el uno o más procesadores están configurados además para:
- 60 recibir una tercera FU que comprende un subconjunto de una segunda unidad NAL fragmentada; y
- en respuesta a un modo de transmisión para la tercera FU que es un modo de transmisión de sesión única, determinar que un parámetro de orden de decodificación no está presente en la segunda FU.
- 65
14. El dispositivo, según la reivindicación 8, en el que el dispositivo comprende al menos uno de:
- un circuito integrado;

un microprocesador; y

un dispositivo de comunicación inalámbrica que comprende un codificador de vídeo.

- 5
- 15.** Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando son ejecutadas por uno o más procesadores, hacen que el uno o más procesadores lleven a cabo el procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 7.

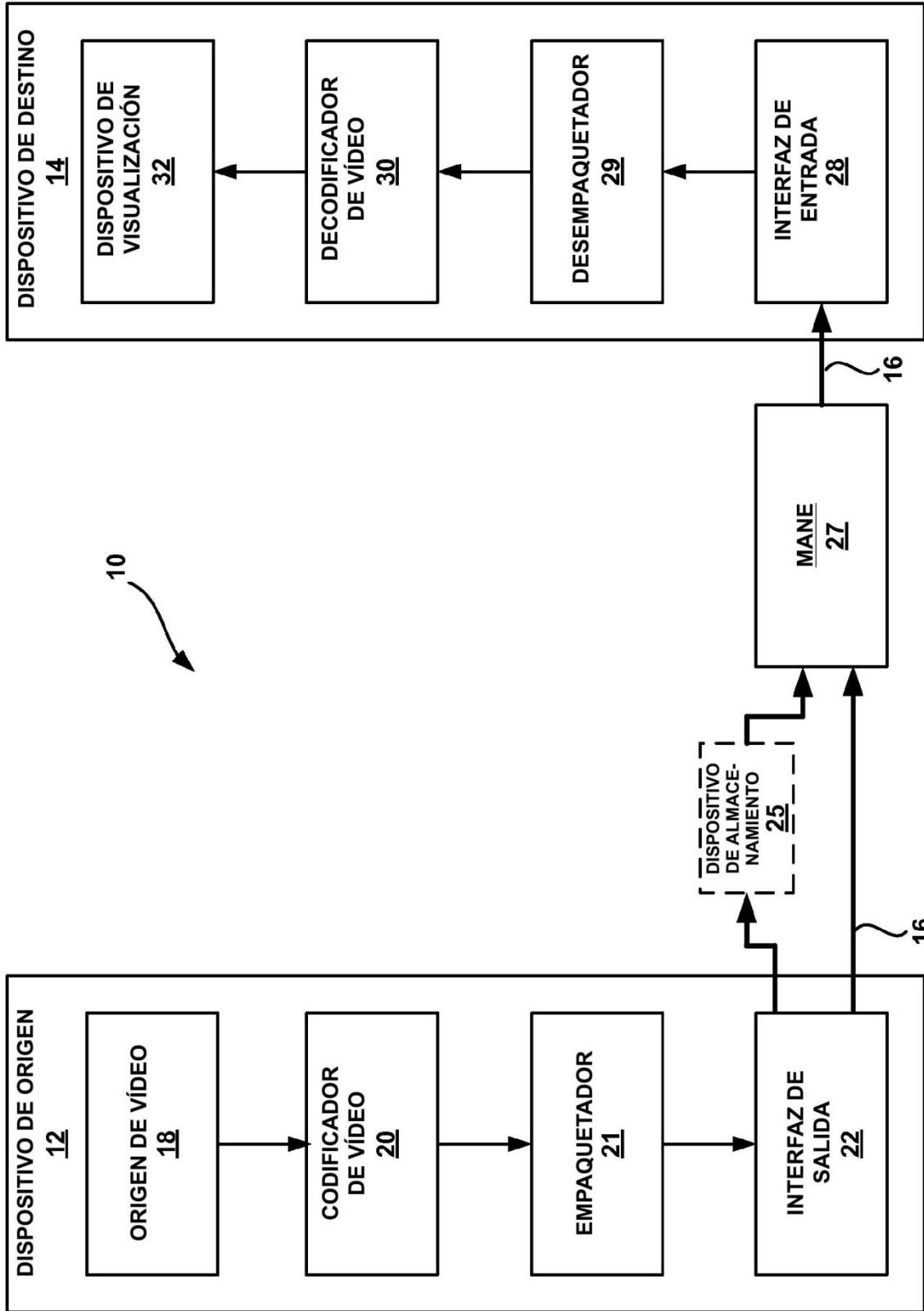


FIG. 1

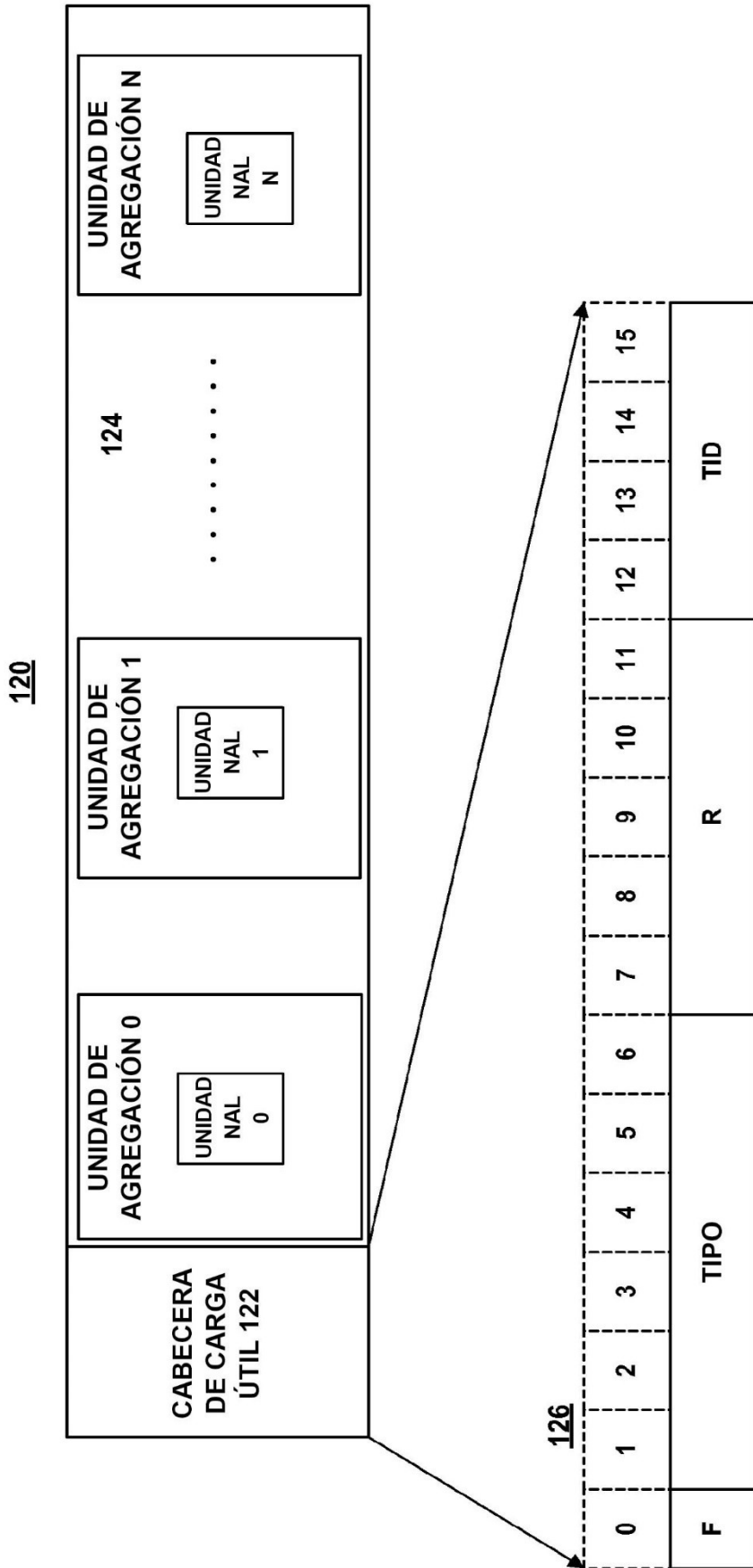


FIG. 2

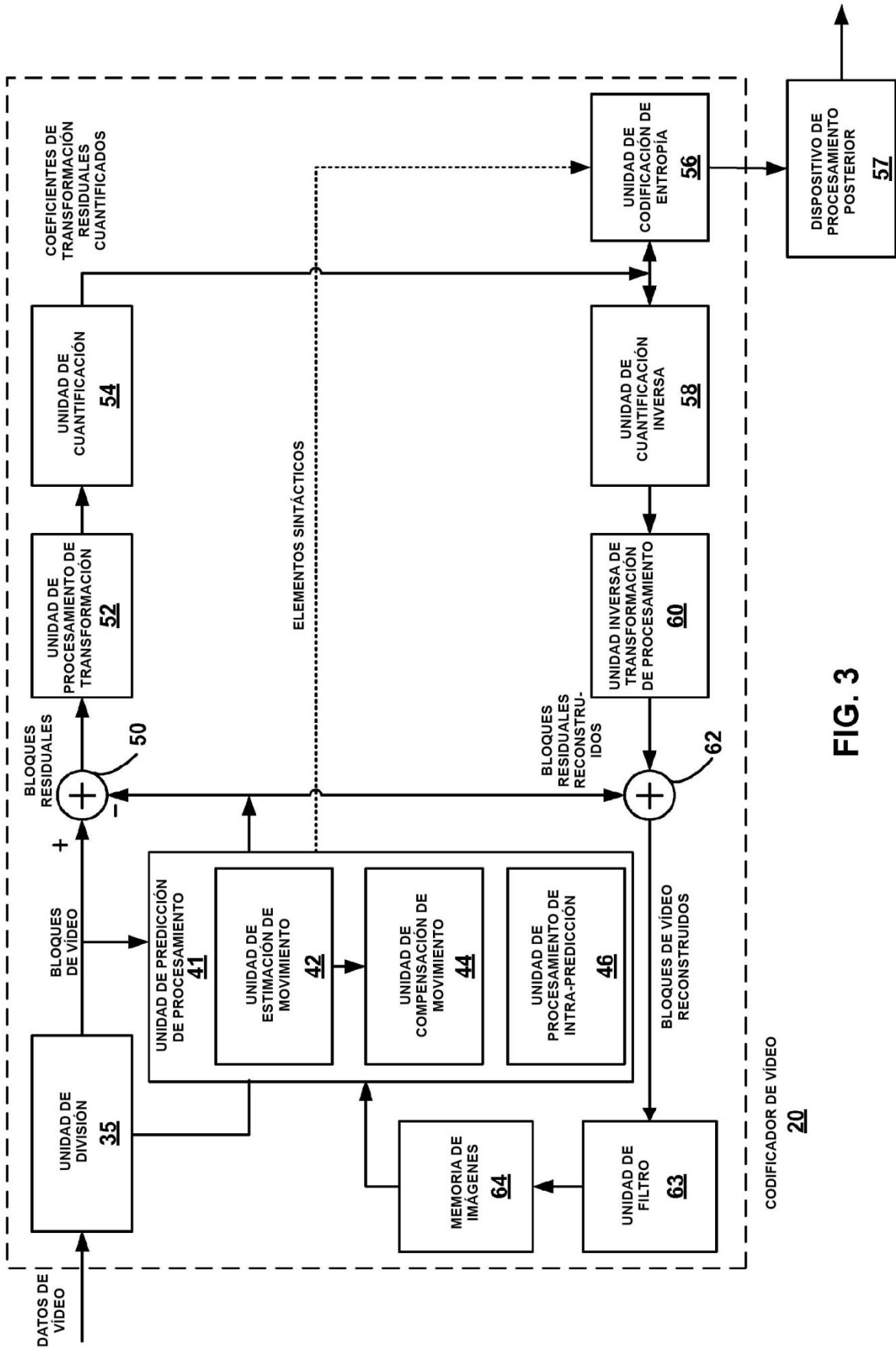


FIG. 3

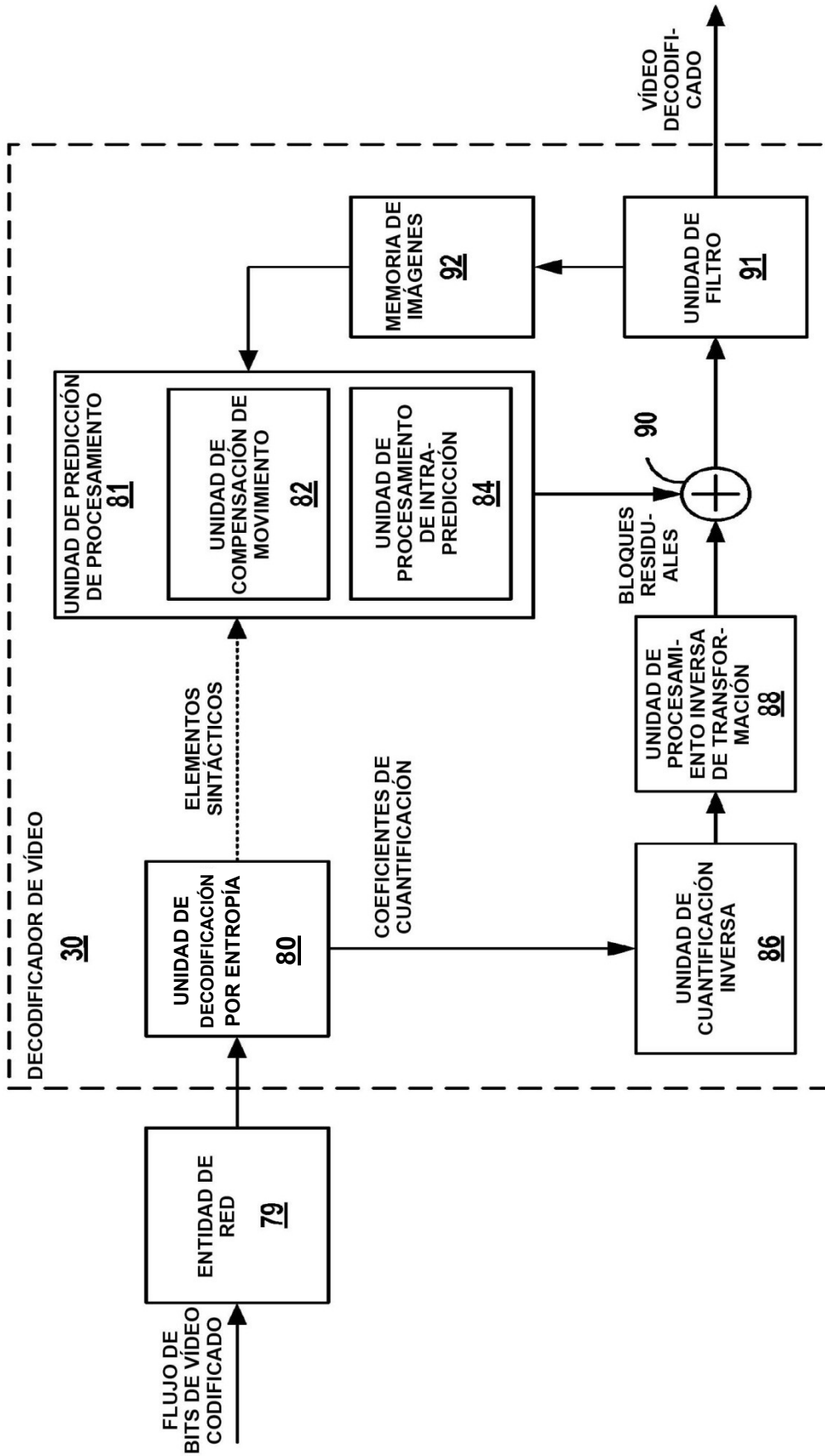


FIG. 4

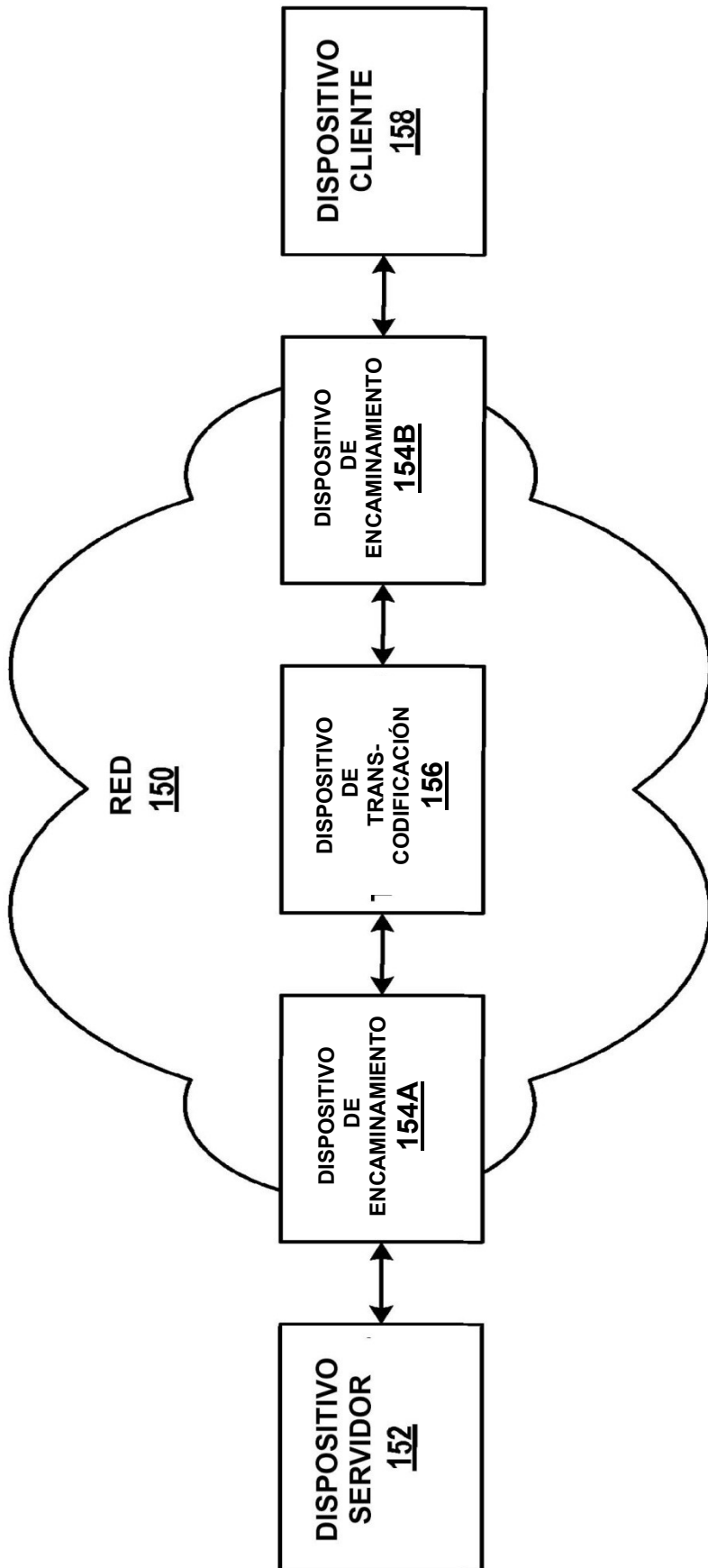


FIG. 5

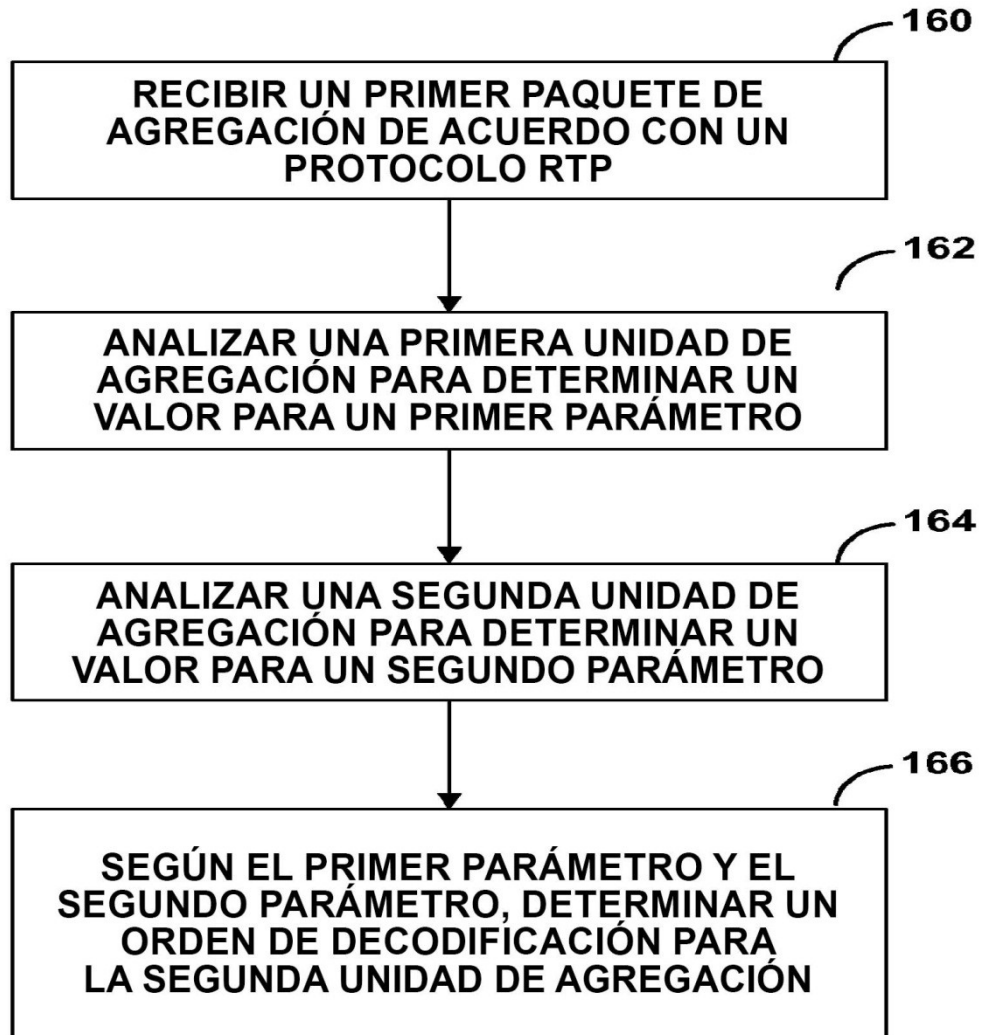


FIG. 6

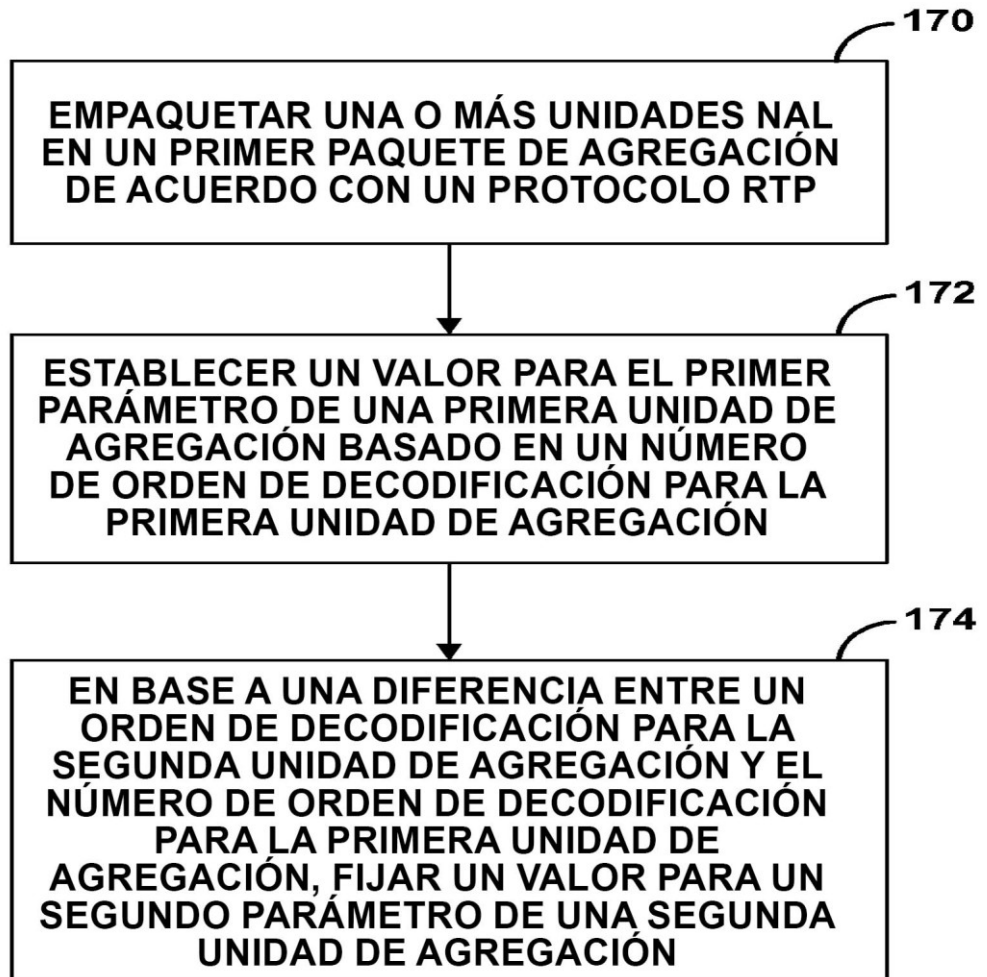


FIG. 7

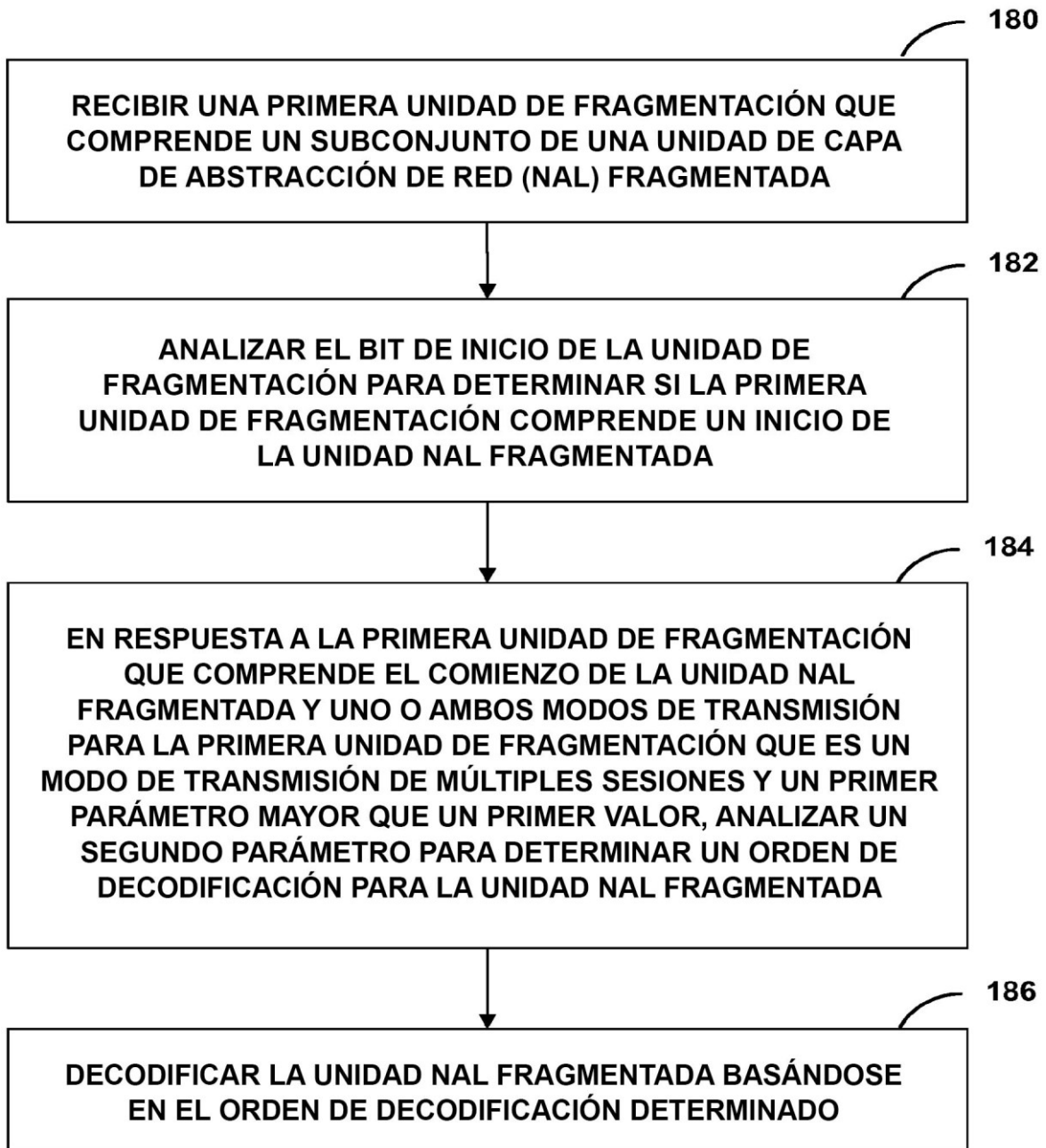


FIG. 8

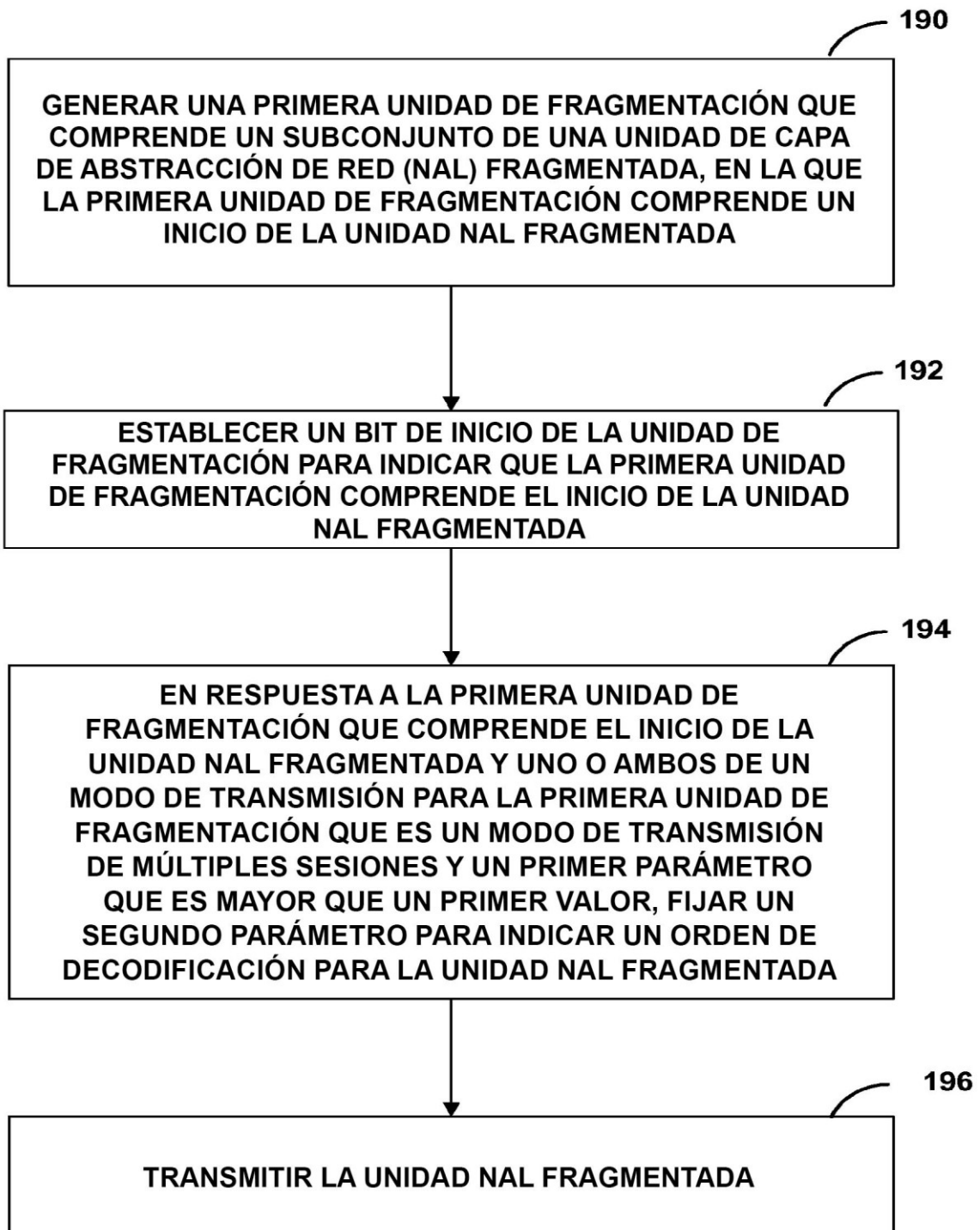


FIG. 9

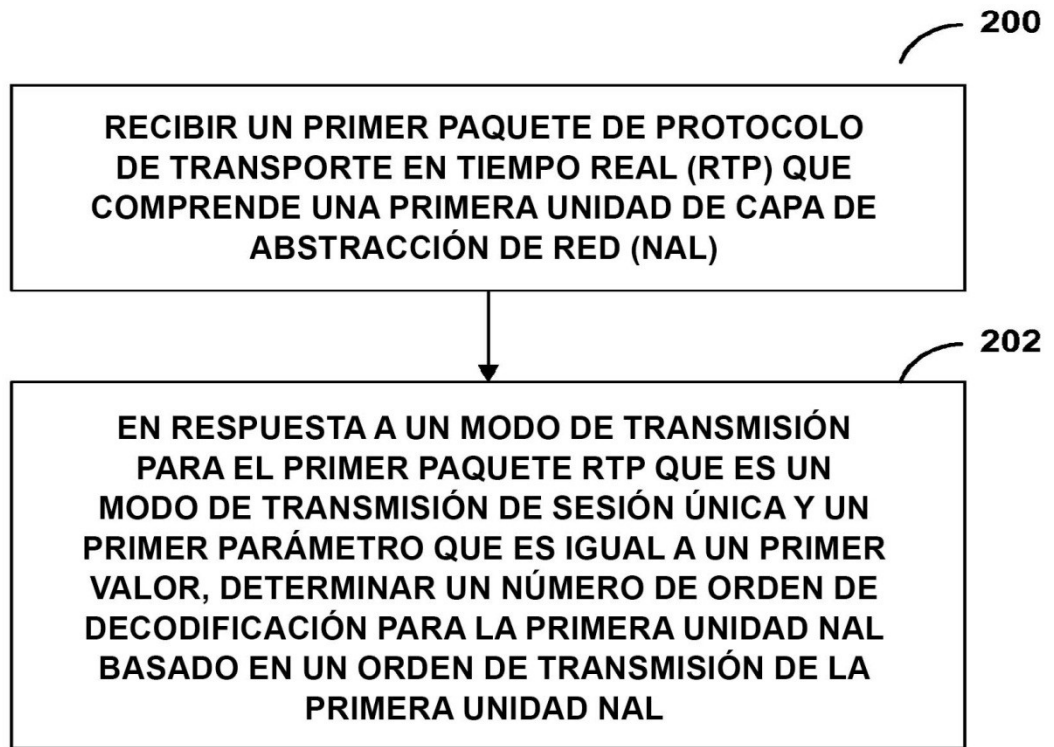


FIG. 10

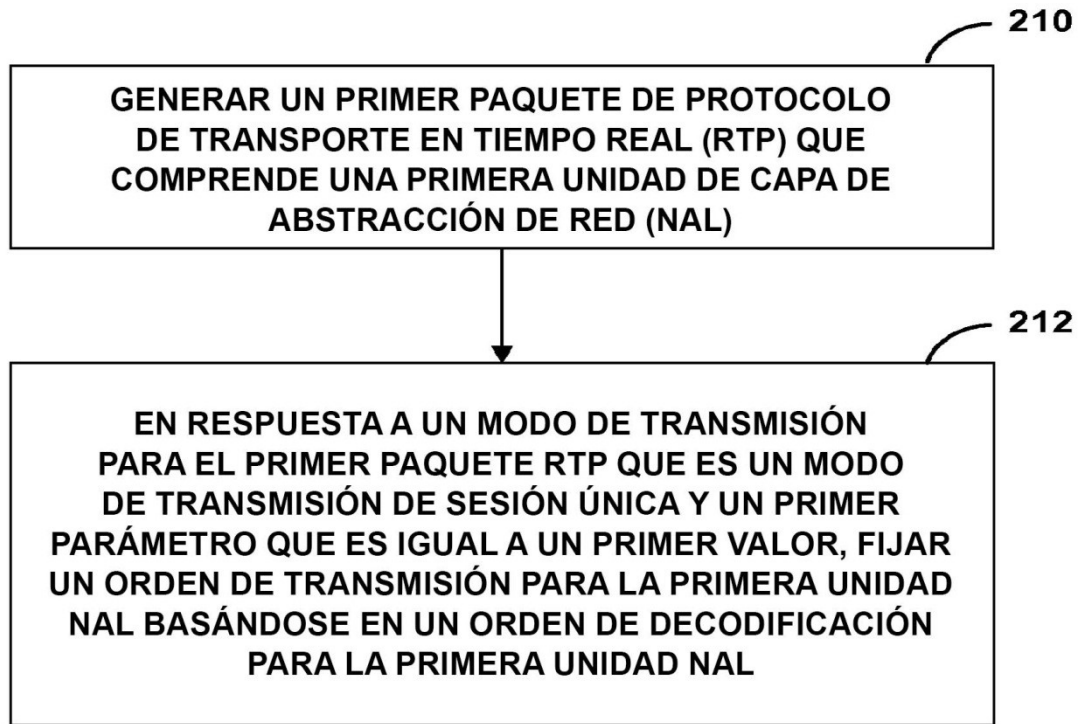


FIG. 11