

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 663 829**

51 Int. Cl.:

H04L 29/06 (2006.01)

H04W 28/06 (2009.01)

H04L 1/00 (2006.01)

H04L 1/18 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.03.2012 PCT/CN2012/072785**

87 Fecha y número de publicación internacional: **22.11.2012 WO12155664**

96 Fecha de presentación y número de la solicitud europea: **22.03.2012 E 12785215 (0)**

97 Fecha y número de publicación de la concesión europea: **03.01.2018 EP 2720488**

54 Título: **Procedimiento, compresor y sistema de proceso de compresión robusta de cabecera**

30 Prioridad:

27.07.2011 CN 201110211930

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

17.04.2018

73 Titular/es:

**ZTE CORPORATION (100.0%)
ZTE Plaza, Keji Road South, Hi-Tech Industrial
Park, Nanshan District
Shenzhen, Guangdong 518057, CN**

72 Inventor/es:

**WU, YONG;
SHI, XUEHONG y
DONG, SHUSONG**

74 Agente/Representante:

DURAN-CORRETJER, S.L.P

ES 2 663 829 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

Procedimiento, compresor y sistema de proceso de compresión robusta de cabecera

5 Sector técnico

El presente documento se refiere al sector técnico de las comunicaciones móviles y, en particular, a un procedimiento, un compresor y un sistema de proceso de compresión robusta de cabecera.

10 Antecedentes de la técnica relacionada

Debido a la limitación de las condiciones físicas, en el sistema de comunicación móvil, una conexión inalámbrica tiene una velocidad de transmisión baja y una tasa de error de bits alta en comparación con una conexión por cable. Con el fin de utilizar de manera eficaz los recursos limitados del ancho de banda inalámbrico del canal, se introduce la compresión robusta de cabecera (en adelante ROHC, para abreviar). La base de la ROHC es utilizar la redundancia de información entre paquetes de flujo de tráfico para comprimir y descomprimir información de manera transparente en la cabecera de paquete entre nodos que están conectados directamente. La tecnología ROHC se describe mediante un documento RFC 3095 del Grupo de Trabajo de Ingeniería de Internet (IETF, Internet Engineering Task Force), pero en el documento, no hay ninguna definición de compresión y descompresión de cabecera IP y, por tanto, en junio de 2004, el grupo de trabajo ROHC tiene que definir un marco solamente para la compresión y descompresión de cabecera IP en el RFC 3843 y, en febrero de 2007, el IETF revisó el documento relacionado con la ROHC, y el documento revisado es el RFC 4815.

En la compresión de cabecera ROHC, se utilizan diferentes Perfiles para identificar diferentes flujos de paquetes de datos, de modo que se aplica un marco de compresión y descompresión de datos correspondiente en el flujo de paquetes de datos. Por ejemplo, la identidad del Perfil 0 se utiliza para transmitir flujos de paquetes de datos IP mediante descompresión, la identidad del Perfil 1 se utiliza para realizar un proceso de compresión en un flujo de paquetes de datos de RTP/UDP/IP, la identidad del Perfil 2 se utiliza para realizar un proceso de compresión en el flujo de paquetes de datos de UDP/IP, la identidad del Perfil 3 se utiliza para realizar un proceso de compresión en un flujo de paquetes de datos de ESP/IP y la identidad del Perfil 4 se utiliza para realizar un proceso de compresión en el flujo de paquetes de datos IP.

La retroalimentación es la información de estado relacionada que se refiere a una parte de descompresión relacionada cuando se realiza una descompresión en el paquete de datos, que es transmitida por la parte de descompresión a una parte de compresión según el resultado de descompresión después de recibir los paquetes de datos comprimidos y realizar la descompresión en los paquetes de datos. La parte de compresión realiza un proceso correspondiente según los datos de retroalimentación recibidos, para comprimir los paquetes de datos mejor, de manera precisa y eficiente.

En la compresión de cabecera ROHC, la retroalimentación se divide en una retroalimentación ACK (un paquete de datos transmitido por el descompresor al compresor cuando el descompresor descomprime satisfactoriamente los paquetes comprimidos) y una retroalimentación NACK (un paquete de datos transmitido por el descompresor al compresor cuando el descompresor descomprime erróneamente el paquete de datos comprimido). La retroalimentación que es transmitida por el descompresor al compresor puede asimismo llevar una opción de retroalimentación, en la que, la opción de retroalimentación se utiliza para notificar a la parte de compresión el motivo detallado por el que falla la descompresión de datos, de modo que la parte de compresión puede ajustar específicamente su estrategia de compresión, reaccionar lo antes posible y mejorar la robustez de la compresión y la descompresión. La opción de retroalimentación tiene una opción CRC para comprobar el paquete de retroalimentación, una opción REJECT que indica que el descompresor no tiene recursos suficientes para realizar un proceso de compresión de datos y una opción SN que indica que el número de serie que lleva el paquete de retroalimentación no está disponible, etc.

Existen muchas opciones de retroalimentación en la ROHC que son transmitidas por la parte de descompresión a la parte de compresión, y la opción CONTEXT_MEMORY es una de éstas, que representa que la parte de descompresión no tiene recursos de memoria suficientes para soportar la descompresión de datos comprimidos de algún flujo de paquetes especificado y, por tanto, la parte de descompresión lleva una opción CONTEXT_MEMORY para la parte de compresión en la retroalimentación NACK, y la parte de compresión realiza un ajuste correspondiente cuando recibe la opción de retroalimentación, de modo que la parte de descompresión puede realizar una descompresión correcta.

En el RFC 3843, se define un procedimiento para procesar la opción de retroalimentación CONTEXT_MEMORY mediante la parte de compresión, después de recibir la opción CONTEXT_MEMORY, la parte de compresión comprime los datos de paquete seleccionando paquetes comprimidos para hacer que la parte de descompresión descomprima satisfactoriamente con el espacio de memoria mínimo o pueda dejar de comprimir los datos de flujo de paquetes. Sin embargo, en el RFC 3843, no se ha proporcionado un procedimiento de compresión sobre cómo

seleccionar hacer que una parte de descompresión descomprima los paquetes de datos con el espacio de memoria mínimo, y la amenaza de los recursos de memoria de la parte de descompresión no puede continuar.

5 El documento no de patente de L-E JONSSON G PELLETIER ERICSSON: "RFC 3843. Robust Header Compression (ROHC): A Compression Profile for IP; rfc3843.txt" (Compresión robusta de cabecera (ROHC): un perfil de compresión para IP), 1 de junio de 2004, ISSN: 0000-0003, da a conocer un Perfil de compresión ROHC para IP, similar al Perfil de IP/UDP definido por RFC 3095, pero simplificado para excluir UDP, y mejorado para comprimir cadenas de cabecera IP de longitud arbitraria.

10 Características de la invención

El problema técnico que a resolver mediante el presente documento es dar a conocer un procedimiento, un compresor y un sistema de proceso de compresión robusta de cabecera, para resolver el problema de que la técnica anterior no contempla cómo seleccionar un modo de compresión.

15 Las características del procedimiento de proceso de compresión robusta de cabecera y el compresor robusto de cabecera según la presente invención se definen en las reivindicaciones independientes y las características preferentes según la presente invención se definen en las reivindicaciones dependientes.

20 Con el fin de resolver el problema técnico anterior, el presente documento da a conocer además un procedimiento de proceso de compresión robusta de cabecera según la reivindicación 1. Para resolver el problema técnico anterior, el presente documento da a conocer además un compresor robusto de cabecera según la reivindicación 8. Con el fin de resolver el problema técnico anterior, el presente documento da a conocer además un sistema de proceso de compresión robusta de cabecera según la reivindicación 15.

25 En las reivindicaciones dependientes se definen realizaciones preferentes. En el procedimiento de proceso de compresión robusta de cabecera, el sistema y el compresor robusto de cabecera de las realizaciones del presente documento, una parte de compresión selecciona un Perfil0 de formato de paquete de compresión después de recibir una opción de retroalimentación CONTEXT_MEMORY e intenta restablecer una compresión y descompresión normales transcurrido un periodo de tiempo determinado después de recibir una opción de retroalimentación CONTEXT_MEMORY, mejorando, por tanto, la eficiencia y la precisión de la compresión y la descompresión, así como la utilización del ancho de banda inalámbrico.

35 Breve descripción de los dibujos

La figura 1 es un diagrama de la realización uno de un procedimiento de proceso de compresión robusta de cabecera, según una realización del presente documento;

40 la figura 2 es un diagrama de flujo de un proceso cuando un compresor recibe una opción De retroalimentación CONTEXT_MEMORY, según una realización del presente documento;

la figura 3 es un diagrama de flujo de una estrategia de procesamiento de datos cuando un compresor recibe un flujo de paquetes de datos, según una realización del presente documento;

45 la figura 4 es un diagrama de flujo de un proceso de un compresor que recibe una retroalimentación ACK después de recibir una opción de retroalimentación CONTEXT_MEMORY, según una realización del presente documento;

50 la figura 5 es un proceso realizado por un descompresor sobre paquetes comprimidos recibidos después de transmitir una opción de retroalimentación CONTEXT_MEMORY, según una realización del presente documento;

la figura 6 es un procedimiento de compresión y descompresión para una opción de retroalimentación CONTEXT_MEMORY de datos entre un compresor y un descompresor, según una realización del presente documento; y

55 las figuras 7 a 10B son diagramas de estructura de módulos de las realizaciones uno a cuatro de un compresor robusto de cabecera según el presente documento.

Realizaciones preferentes de la invención

60 La idea principal del procedimiento de proceso de compresión robusta de cabecera, el sistema y el compresor robusto de cabecera del presente documento es que, después de recibir una opción de retroalimentación CONTEXT_MEMORY, una parte de compresión selecciona un formato de paquete de compresión mediante el que la parte de descompresión puede consumir un espacio de memoria mínimo para descomprimir correctamente e intenta restablecer una compresión y descompresión normal transcurrido un periodo de tiempo determinado después de recibir una opción de retroalimentación CONTEXT_MEMORY, mejorando, por tanto, la eficiencia y la precisión de la compresión y la descompresión, así como la utilización del ancho de banda inalámbrico.

Las realizaciones del presente documento se describirán en detalle a continuación junto con los dibujos adjuntos. Se debe mostrar que las realizaciones de la presente solicitud y las características de las realizaciones pueden combinarse aleatoriamente entre sí sin generar conflicto.

5 Realización uno

La realización uno de un procedimiento de proceso de compresión robusta de cabecera según el presente documento comprende las siguientes etapas.

10 En la etapa -101-, después de recibir una opción CONTEXT_MEMORY, un compresor selecciona utilizar el Perfil0 para llevar a cabo compresión de datos en paquetes de datos a comprimir (paquetes a comprimir en el presente documento, para abreviar);

15 el compresor inicia un primer contador o un primer temporizador después de recibir una opción CONTEXT_MEMORY, el primer contador cuenta el número de utilizaciones del Perfil0 o el primer temporizador cuenta el tiempo para utilizar el Perfil0, y el compresor determina si se alcanza el primer umbral según el primer contador o el primer temporizador.

20 La identidad de Perfil0 se utiliza para transmitir los flujos de paquetes de datos IP mediante descompresión.

En la etapa -102-, después de determinar que se alcanza un primer umbral utilizando el Perfil0, el compresor realiza compresión de datos utilizando un tipo de Perfil al que pertenecen los paquetes a comprimir.

25 Preferentemente, el compresor también inicia un segundo contador o un segundo temporizador después de recibir una opción CONTEXT_MEMORY que se refiere a un flujo de paquetes actual por primera vez, el segundo contador cuenta el número de recepciones de la opción CONTEXT_MEMORY (es decir, inicializa (o reinicia) un número del primer contador o del temporizador) o el segundo temporizador cuenta el tiempo para utilizar el Perfil0, preferentemente, el segundo umbral del segundo contador o del segundo temporizador se establece para ser mayor que el primer umbral, y después de determinar que se alcanza el segundo umbral, el compresor utiliza continuamente el Perfil0 para realizar compresión de datos sobre paquetes a comprimir y no intenta utilizar un tipo de Perfil al que pertenecen los paquetes de datos para llevar a cabo de datos.

30 Preferentemente, el compresor establece una identidad de retroalimentación de opción como una primera identidad después de recibir la opción CONTEXT_MEMORY, establece la identidad de retroalimentación de opción como una segunda identidad después de recibir una retroalimentación (ACK) que representa una descompresión satisfactoria y cuando el primer contador o el primer temporizador alcanza el primer umbral, y el primer contador o el primer temporizador se reinicia después de alcanzar el primer umbral.

40 Preferentemente, el primer contador o el primer temporizador cuenta el número o cuenta el tiempo cuando la identidad de retroalimentación de opción es la primera identidad, y deja de contar el número o contar el tiempo cuando la identidad de retroalimentación de opción es la segunda identidad.

45 El procedimiento de proceso de compresión según el presente documento es un procedimiento para procesar paquetes a comprimir para un flujo de paquetes.

Realización dos

50 En la realización dos, el proceso mediante una parte de compresión en una opción de retroalimentación CONTEXT_MEMORY se implementa mediante una parte de compresión que establece una identidad y que suma dos contadores; la parte de descompresión establece una identidad para recodificar si la opción de retroalimentación CONTEXT_MEMORY se transmite, ayudando a que la parte de descompresión elimine oportunamente un efecto generado por la opción de retroalimentación CONTEXT_MEMORY cuando los paquetes comprimidos se descomprimen, posteriormente, satisfactoriamente. El contador controla el compresor para comprimir los datos de paquete de un flujo de paquetes especificado usando un Perfil 0 en un periodo de tiempo determinado, y reselectiona un tipo de Perfil al que pertenece el flujo de paquetes para comprimir el flujo de paquetes de datos, con el fin de garantizar la eficiencia de la compresión y, si el compresor continúa recibiendo la opción de retroalimentación CONTEXT_MEMORY después de la reselection, el número de recepciones de la retroalimentación se acumula, y cuando el número de recepciones de la opción de retroalimentación CONTEXT_MEMORY alcanza un valor determinado, el Perfil 0 se utiliza todo el tiempo para llevar a cabo compresión sobre el flujo de paquetes de datos, que implementa un procesamiento eficaz para la opción de retroalimentación CONTEXT_MEMORY.

65 El esquema de implementación de la parte de compresión que procesa la opción de retroalimentación CONTEXT_MEMORY implicada en el presente documento se describirá adicionalmente en detalle a continuación junto con dibujos adjuntos.

Tal como se muestra en la figura 2, el flujo de proceso después de que un compresor recibe paquetes de datos a comprimir incluye las siguientes etapas.

5 En la etapa -S201-, el compresor recibe paquetes de datos a comprimir;
 en la etapa -S202-, se determina si el flujo de paquetes existe; si el flujo de paquetes no existe, se avanza a la etapa -S203-, de lo contrario, se avanza a la etapa -S205-;

10 en la etapa -S203-, se inicializa CtxMemFlag y un contador RecCtxMemCounter (número de retroalimentaciones de opción) y un PackCounter (cómputo de paquetes).

El CtxMemFlag se denomina también una identidad de retroalimentación de opción, la identidad CtxMemFlag inicializada es Falso, y los contadores RecCtxMemCounter y PackCounter iniciados se ajustan a 0.

15 En la etapa -S204-, el proceso de compresión de datos se realiza según el tipo de perfil al que pertenece el flujo de paquetes de datos, y se avanza a la etapa -S212-;

20 En la etapa -S205-, el Proceso se selecciona según la identidad CtxMemFlag, si la identidad CtxMemFlag es Verdadero, se avanza a la etapa -S206-, y si la identidad CtxMemFlag es Falso, se avanza a la etapa -S204-.

En la etapa -S206-, se determina si el contador RecCtxMemCounter alcanza un umbral, y si el umbral se alcanza, se avanza a la etapa -S211-; de lo contrario, se avanza a la etapa -S207-.

25 En la etapa -S207-, se suma 1 al contador PackCounter;

En la etapa -S208-, se determina si el contador PackCounter alcanza un umbral, y si el umbral se alcanza, se avanza a la etapa -S209-; de lo contrario, se avanza a la etapa -S211-.

30 En la etapa -S209-, la identidad CtxMemFlag se ajusta a Falso;

en la etapa -S210-, el contador PackCounter se ajusta a 0;

35 en la etapa -S211-, el Perfil0 se utiliza para realizar compresión en el flujo de paquetes de datos;

en la etapa -S212-, termina el flujo de proceso después de que el compresor reciba los paquetes de datos.

A partir del flujo anterior se puede ver que se realizan diferentes procesos en diferentes casos:

40 A1) el compresor recibe paquetes a comprimir, si los paquetes son un nuevo flujo de paquetes, se establecen contadores PackCounter y RecCtxMemCouter para el flujo de paquetes y se inicializan a 0, y CtxMemFlag se ajusta a 0 (Falso), lo que representa que el proceso de compresión se lleva a cabo sobre el flujo de paquetes utilizando un tipo de Perfil al que pertenece el flujo de paquetes, y se establece un contexto correspondiente para el flujo de paquetes.

45 En el que la identidad del Perfil 1 se utiliza para realizar un proceso de compresión en un flujo de paquetes de datos de RTP/UDP/IP, la identidad del Perfil 2 se utiliza para realizar un proceso de compresión en el flujo de paquetes de datos de UDP/IP, la identidad del Perfil 3 se utiliza para realizar un proceso de compresión en un flujo de paquetes de datos de ESP/IP y la identidad del Perfil 4 se utiliza para realizar un proceso de compresión en el flujo de paquetes de datos IP.

50 A2) el compresor recibe paquetes a comprimir, si el flujo de paquetes ya existe, el CtxMemFlag es verdadero y el contador RecCtxMemCouter no alcanza el umbral, se suma 1 al contador PackCounter, y el contador PackCounter no alcanza el umbral, y se realiza compresión de datos en el flujo de paquetes utilizando un tipo de Perfil 0.

55 A3) el compresor recibe paquetes a comprimir, si el flujo de paquetes ya existe, el CtxMemFlag es verdadero y el contador RecCtxMemCouter no alcanza el umbral, se suma 1 al contador PackCounter, el contador PackCounter alcanza el umbral, la identidad CtxMemFlag se ajusta a Falso, el contador PackCounter se ajusta a 0 y se realiza compresión de datos en el flujo de paquetes utilizando un tipo de Perfil 0.

60 A4) el compresor recibe paquetes a comprimir, si el flujo de paquetes ya existe y el CtxMemFlag es Falso, el tipo de Perfil al que pertenece el flujo de paquetes se utiliza para llevar a cabo compresión sobre el flujo de paquetes y el contador PackCounter permanece en 0, es decir, el cómputo se detiene.

65

El contador PackCounter se utiliza para contar el número de utilizations de Perfil0, alternativamente, el contador puede sustituirse por un temporizador para contar el tiempo para utilizar el Perfil0, ambos tienen el mismo efecto.

5 Un valor del umbral del contador PackCounter es fijo o variable, preferentemente, el umbral del contador PackCounter puede ajustarse según un valor actual del contador RecCtxMemCounter multiplicado por un coeficiente fijo, es decir, el umbral del contador PackCounter es proporcional al valor actual del contador RecCtxMemCounter, por tanto, con el aumento del número de las opciones de retroalimentación CONTEXT_MEMORY recibidas, el valor de cómputo del contador RecCtxMemCounter también aumenta gradualmente, y cada vez que el contador RecCtxMemCounter aumenta, el umbral del contador PackCounter aumenta correspondientemente en uno, y al mismo tiempo, el tiempo para utilizar el tipo de Perfil0 para comprimir los paquetes de datos aumenta con el mismo.

15 En las etapas de implementación anteriores, los umbrales del contador RecCtxMemCounter y el contador PackCounter se determinan según los casos reales, en los que, en la etapa -S108- (figura 2), el umbral del contador PackCounter es preferentemente un valor de contador del contador RecCtxMemCounter multiplicado por un valor fijo y, de esta manera, el compresor selecciona de manera eficaz el tipo para comprimir los paquetes cuando recibe la opción de retroalimentación CONTEXT_MEMORY.

20 Tal como se muestra en la figura 3, un flujo de proceso después de que un compresor reciba una opción de retroalimentación CONTEXT_MEMORY incluye las siguientes etapas.

En la etapa -S301-, el compresor recibe una retroalimentación NACK que lleva una opción de retroalimentación CONTEXT_MEMORY

25 en la etapa -S302-, la identidad CtxMemFlag se ajusta a Verdadero;

en la etapa -S303-, se suma 1 al contador RecCtxMemCounter;

en la etapa -S304-, termina el flujo.

30 Se puede saber, junto con los procedimientos anteriores de las figuras 2 y 3, que el contador RecCtxMemCounter es un contador que cuenta el número de recepciones de opción de retroalimentación CONTEXT_MEMORY con respecto al flujo de paquetes actual. Alternativamente, el contador RecCtxMemCounter puede sustituirse por un temporizador, que cuenta el tiempo para utilizar el Perfil0, ambos tienen el mismo efecto.

35 Cuando se recibe la retroalimentación ACK, muestra que la parte de descompresión tiene espacio de memoria suficiente para realizar un proceso de compresión en los datos, todo proceso original en la opción CONTEXT_MEMORY puede reiniciarse, incluyendo ajustar el contador RecCtxMemCounter a 0.

40 Por tanto, el valor del contador RecCtxMemCounter se pone a cero después de que el valor de cómputo alcance un valor determinado (después de recibir la retroalimentación ACK), y es 0 todo el tiempo (no recibe la opción de retroalimentación CONTEXT_MEMORY).

45 Además, excepto en el caso de que el compresor no soporte el tipo de Perfil al que pertenece el flujo de paquetes, después de recibir la opción CONTEXT_MEMORY, el compresor selecciona utilizar Perfil0 para llevar a cabo compresión de datos sobre los paquetes de datos a comprimir.

Puede entenderse que, cuando el compresor utiliza el Perfil0 para realizar compresión de datos, no se recibirá la retroalimentación NACK que lleva la opción de retroalimentación CONTEXT_MEMORY.

50 Tal como se muestra en la figura 4, el procedimiento después de que el compresor reciba la retroalimentación ACK portada incluye las siguientes etapas.

En la etapa -S401-, el compresor recibe la retroalimentación ACK;

55 Cuando la identidad CtxMemFlag es Verdadero o el RecCtxMemCounter no es cero, se lleva a cabo el procedimiento.

En la etapa -S402-, la identidad CtxMemFlag se ajusta a Falso;

60 En la etapa -403-, el contador RecCtxMemCounter se ajusta a 0.

En la etapa -S404-, termina el procedimiento.

65 Tal como se muestra en la figura 5, el Flujo de proceso después de que el descompresor reciba los paquetes a descomprimir incluye las siguientes etapas.

- En la etapa -S501-, el descompresor recibe los paquetes de datos a descomprimir;
- 5 en la etapa -S502-, se determina si el flujo de paquetes existe, y si el flujo de paquetes no existe, se avanza a la etapa -S503-, de lo contrario, se avanza a la etapa -S505-;
- en la etapa -S503-, se establece el contexto de descompresión al que pertenece el flujo de paquetes;
- en la etapa -S504-, la identidad DecmpCtxMemFlag (flujo de paquetes de descompresión) se inicializa como Falso;
- 10 en la etapa -S505-, la descompresión se realiza según el tipo de Perfil al que pertenece el flujo de paquetes comprimido;
- en la etapa -S506-, se determina si la descompresión para los paquetes comprimidos es satisfactoria; si la descompresión es satisfactoria, se avanza a la etapa -S510-; de lo contrario, se avanza a la etapa -S507-;
- 15 en la etapa -S507-, se determina si el motivo para el fallo de la descompresión para los paquetes comprimidos es que no hay espacio de memoria suficiente para descomprimir los datos comprimidos, en tal caso, se avanza a la etapa -S508-; de lo contrario, se avanza a la etapa -S511-;
- 20 en la etapa -S508-, se transmite una retroalimentación NACK que lleva una opción de retroalimentación CONTEXT_MEMORY;
- en la etapa -S509-, la identidad DecmpCtxMemFlag se ajusta a Verdadero;
- 25 en la etapa -S510-, se determina si la identidad DecmpCtxMemFlag es Verdadero, y si la identidad DecmpCtxMemFlag es Verdadero, se avanza a la etapa -S512-; de lo contrario, se avanza a la etapa -S511-;
- en la etapa -S511-, se realiza el proceso de retroalimentación según el procedimiento normal;
- 30 en la etapa -S512-, se transmite la retroalimentación ACK;
- en la etapa -S513-, la identidad DecmpCtxMemFlag se ajusta a Falso;
- en la etapa -S514-, termina el flujo de proceso después de que el descompresor reciba los datos comprimidos;
- 35 se puede saber, según el procedimiento anterior, que el descompresor realiza diferentes procesos en diferentes casos:
- 40 B1) el descompresor recibe los paquetes de datos a descomprimir, si el flujo de paquetes es un flujo de paquetes nuevo, se establece una identidad DecmpCtxMemFlag para el flujo de paquetes y la identidad DecmpCtxMemFlag se ajusta a 0 (Falso), el flujo de paquetes comprimido se descomprime, si la descompresión para los paquetes comprimidos falla debido a que no hay espacio de memoria suficiente para descomprimir los paquetes de datos comprimidos, se transmite una retroalimentación NACK que lleva la opción de retroalimentación CONTEXT_MEMORY. La identidad DecmpCtxMemFlag se ajusta a Verdadero. Si la descompresión para los paquetes comprimidos falla debido a otros motivos, el proceso se realiza según el procedimiento existente.
- 45 B2) el descompresor recibe los paquetes de datos a descomprimir, si el flujo de paquetes es un flujo de paquetes nuevo, se establece una identidad DecmpCtxMemFlag para el flujo de paquetes y la identidad DecmpCtxMemFlag se ajusta a 0 (Falso), el flujo de paquetes comprimido se descomprime, si la descompresión para los paquetes comprimidos es satisfactoria, el proceso se realiza según el procedimiento existente.
- 50 B3) el descompresor recibe los paquetes de datos a descomprimir, si el flujo de paquetes ya existe y la descompresión para los paquetes comprimidos es satisfactoria, el proceso se realiza según el procedimiento existente.
- 55 B4) el descompresor recibe los paquetes de datos a descomprimir, si el flujo de paquetes ya existe y la descompresión para los paquetes comprimidos es satisfactoria y la identidad DecmpCtxMemFlag es Verdadero, se transmite una retroalimentación ACK y la identidad DecmpCtxMemFlag se ajusta a Falso.
- 60 B5) el descompresor recibe los paquetes de datos a descomprimir, si el flujo de paquetes ya existe y la descompresión para los paquetes comprimidos falla, si la descompresión para los paquetes comprimidos falla debido a que no hay memoria suficiente para descomprimir los paquetes de datos comprimidos, se transmite una retroalimentación NACK que lleva la opción de retroalimentación CONTEXT_MEMORY. La
- 65

identidad DecmpCtxMemFlag se ajusta a Verdadero. Si la descompresión para los paquetes comprimidos falla debido a otros motivos, el proceso se realiza según el procedimiento existente.

Realización tres

5 Tal como se muestra en la figura 6, un procedimiento de realización tres de un procedimiento de proceso de compresión robusta de cabecera incluye las siguientes etapas.

10 en la etapa -S601-, el compresor transmite paquetes de datos comprimidos según un tipo de Perfil al que pertenece el flujo de paquetes de datos;

en la etapa -S602-, si no hay recursos suficientes para descomprimir el flujo de paquetes de datos, el descompresor transmite una retroalimentación NACK que lleva la CONTEXT_MEMORY;

15 en la etapa -S603-, el compresor transmite un número determinado de paquetes de datos comprimidos de Perfil0 según la retroalimentación NACK recibida anteriormente;

20 en la etapa -S604-, el compresor transmite paquetes de datos comprimidos según un tipo de Perfil al que pertenece el flujo de paquetes de datos;

en la etapa -S605-, si se realiza la descompresión satisfactoriamente, el descompresor transmite una ACK y, si no hay recursos suficientes para descomprimir el flujo de paquetes de datos, el descompresor transmite una retroalimentación NACK que lleva la CONTEXT_MEMORY;

25 en la etapa -S606-, el compresor transmite datos comprimidos de un tipo de Perfil al que pertenece el flujo de paquetes de datos si recibe una retroalimentación ACK, y utiliza Perfil0 para realizar compresión en el flujo de paquetes de datos si recibe la retroalimentación NACK que lleva CONTEXT_MEMORY;

30 En la etapa -S607-, si el número de opciones de retroalimentación CONTEXT_MEMORY recibidas de manera continuada por el compresor alcanza un valor umbral, se utiliza Perfil0 todo el tiempo para realizar compresión en el flujo de paquetes de datos, de lo contrario, se debe volver a la etapa -S603-.

35 En el procedimiento anterior, el descompresor sólo transmite una retroalimentación (ACK) que representa descompresión satisfactoria al compresor robusto de cabecera cuando la descompresión para paquetes de datos de un tipo de Perfil al que pertenece un flujo de paquetes es satisfactoria por primera vez después de transmitir una retroalimentación (NACK) de una descompresión fallida que lleva una opción CONTEXT_MEMORY, que es un esquema preferible con una modificación mínima del procedimiento existente, naturalmente, el descompresor puede transmitir también la retroalimentación (ACK) que representa una descompresión satisfactoria al compresor robusto de cabecera después de que la descompresión sea satisfactoria.

40 En correspondencia con el procedimiento descrito anteriormente, el presente documento da a conocer además una realización uno de un compresor robusto de cabecera, tal como se muestra en la figura 7, particularmente con respecto al presente documento, el compresor robusto de cabecera incluye:

45 un módulo de recepción de retroalimentación, configurado para recibir una retroalimentación transmitida por un descompresor, que incluye recibir una retroalimentación (NACK) de una descompresión fallida que lleva una opción CONTEXT_MEMORY;

50 un módulo de compresión de Perfil0, configurado para utilizar Perfil0 para realizar compresión de datos en paquetes a comprimir después de recibir una opción CONTEXT_MEMORY; y

un módulo de compresión de tipo de Perfil, configurado para realizar compresión de datos utilizando un tipo de Perfil al que pertenecen los paquetes a comprimir después de que se alcance un primer umbral utilizando Perfil0.

55 En la figura 8 se muestra la realización dos del compresor robusto de cabecera, y la diferencia entre la realización dos y la realización mostrada en la figura 7 es que el compresor robusto de cabecera mostrado en la figura 8 comprende además:

60 Un módulo de inicialización, configurado para inicializar un primer contador o un primer temporizador después de recibir una opción CONTEXT_MEMORY;

el primer contador o el primer temporizador, configurado para contar el número de las utilizaciones del Perfil0 o contar el tiempo para utilizar el Perfil0;

65 un módulo de determinación de umbral, configurado para determinar si el primer contador o el primer temporizador alcanzan el primer umbral;

el módulo de compresión de tipo de Perfil determina que se alcanza un primer umbral utilizando el Perfil0 cuando el primer contador o el primer temporizador alcanza el primer umbral.

5 En la figura 9 se muestra la realización tres del compresor robusto de cabecera, y la diferencia entre la realización tres y la realización dos del compresor robusto de cabecera es que el compresor robusto de cabecera de la realización tres comprende además un segundo contador o un segundo temporizador; donde,

10 el módulo de inicialización se utiliza además para inicializar el segundo contador o el segundo temporizador después de recibir una opción CONTEXT_MEMORY de un flujo de paquetes actual por primera vez;

el segundo contador o el segundo temporizador se utiliza para contar el número de recepciones de la opción CONTEXT_MEMORY o contar el tiempo para utilizar el Perfil0;

15 el módulo de determinación de umbral se utiliza además para determinar si el segundo contador o el segundo temporizador alcanzan el segundo umbral; y

el módulo de compresión de Perfil0 se utiliza además para utilizar el Perfil0 de manera continuada para realizar compresión de datos en paquetes a comprimir después de que se alcance el segundo umbral.

20 En las figuras 10A y 10B se muestra la realización cuatro del compresor robusto de cabecera, y la diferencia entre la realización cuatro y la realización dos o tres es que la realización cuatro del compresor comprende además un módulo de mantenimiento de identidad de retroalimentación de opción, que se usa para: establecer una identidad de retroalimentación de opción como la primera identidad después de recibir una opción CONTEXT_MEMORY, y se utiliza además para establecer la identidad de retroalimentación de opción como la segunda identidad después de recibir una retroalimentación (ACK) que representa una descompresión satisfactoria y cuando el primer contador o el primer temporizador alcanza el primer umbral, y el primer contador o el primer temporizador se utiliza además para reinicio después de alcanzar el primer umbral.

30 Preferentemente, en la realización cuatro anterior, el primer contador o el primer temporizador cuenta el número o cuenta el tiempo cuando la identidad de retroalimentación de opción es la primera identidad, y deja de contar el número o de contar el tiempo cuando la identidad de retroalimentación de opción es la segunda identidad.

35 Además, en las realizaciones tres y cuatro anteriores, el segundo contador o el segundo temporizador se utiliza además para ponerlo a cero después de recibir la retroalimentación ACK. Preferentemente, el primer umbral es proporcional al valor actual del segundo contador o el segundo temporizador actuales.

40 Además, el presente documento da a conocer además un sistema de proceso de compresión robusta de cabecera, que comprende el compresor o descompresor robusto de cabecera que se ha descrito anteriormente, en el que el compresor se utiliza para transmitir una retroalimentación (ACK) de una descompresión satisfactoria al compresor robusto de cabecera cuando la descompresión es satisfactoria, o sólo transmite una retroalimentación (ACK) de una descompresión satisfactoria al compresor robusto de cabecera cuando la descompresión para paquetes de datos de un tipo de Perfil al que pertenece un flujo de paquetes es satisfactoria por primera vez después de transmitir una retroalimentación (NACK) de una descompresión fallida que lleva una opción CONTEXT_MEMORY.

45 El presente documento supera el problema en la técnica anterior de que la parte de descompresión no puede descomprimir correctamente debido a que la parte de compresión no puede seleccionar un formato de paquete de compresión con un consumo de memoria menor para la descompresión cuando procesa la opción de retroalimentación CONTEXT_MEMORY, y mejora la eficiencia del algoritmo de compresión intentando restablecer una compresión y descompresión normales transcurrido un periodo de tiempo después de recibir la opción de retroalimentación CONTEXT_MEMORY, y las ventajas de utilizar este procedimiento son las siguientes.

50 Puede seleccionarse un formato de paquete de compresión adecuado para comprimir los datos a comprimir después de recibir la opción de retroalimentación CONTEXT_MEMORY, lo que garantiza que existe memoria suficiente en la parte de descompresión para descomprimir los paquetes comprimidos generados por la parte de compresión;

55 Se sondea si se ha liberado la limitación en los recursos de memoria en la parte de descompresión, utilizando periódicamente un contexto de compresión original para comprimir los paquetes de datos, entonces el contexto original puede utilizarse para continuar comprimiendo y descomprimiendo el flujo de paquetes, mejorando por tanto la eficiencia de la compresión y la descompresión;

60 Se permite que la parte de descompresión reciba los paquetes comprimidos del presente flujo de paquetes para intentar descomprimir después de transmitir la opción de retroalimentación CONTEXT_MEMORY, si la descompresión es satisfactoria, los datos descomprimidos se entregan a una aplicación de capa superior, de lo contrario, la opción de retroalimentación CONTEXT_MEMORY continúa transmitiéndose a la parte de compresión, para avisar a la parte de compresión de que ajuste la estrategia de compresión.

Después de recibir la opción de retroalimentación CONTEXT_MEMORY, el Perfil0 se selecciona para utilizarse para realizar compresión de datos, de modo que la parte de descompresión puede tener espacio de memoria suficiente para descomprimir los datos comprimidos.

5 Un experto en la materia puede comprender que la totalidad o parte de las etapas en el procedimiento anterior pueden implementarse mediante programas que envían órdenes a un hardware relacionado, y los programas pueden almacenarse en un medio de almacenamiento legible por ordenador, tal como una memoria de sólo lectura, disco o CD-ROM, etc. Alternativamente, la totalidad o parte de las etapas en las realizaciones anteriores pueden
10 implementarse asimismo mediante uno o más circuitos integrados. Por consiguiente, cada módulo/unidad en las realizaciones anteriores puede implementarse en forma de hardware, y puede implementarse también en forma de módulo funcional de software. El presente documento no está limitado a una combinación de ninguna forma particular de hardware y software.

15 La descripción anterior consiste solamente en las realizaciones preferentes del presente documento y no pretende limitar el presente documento. Para los expertos en la materia, la invención puede tener diversas modificaciones y variaciones.

20 Aplicabilidad industrial

En el procedimiento de proceso de compresión robusta de cabecera, el sistema y el compresor robusto de cabecera de las realizaciones del presente documento, una parte de compresión selecciona un Perfil0 de formato de paquete de compresión después de recibir una opción de retroalimentación CONTEXT_MEMORY e intenta restablecer una compresión y descompresión normales transcurrido un periodo de tiempo determinado después de recibir una
25 opción de retroalimentación CONTEXT_MEMORY, mejorando, por tanto, la eficiencia y la precisión de la compresión y la descompresión así como la utilización del ancho de banda inalámbrico.

REIVINDICACIONES

1. Procedimiento de proceso de compresión robusta de cabecera, que comprende:

5 recibir, mediante un compresor, una retroalimentación transmitida por un descompresor, que incluye una retroalimentación NACK de una descompresión fallida que lleva una opción CONTEXT_MEMORY;

después de recibir la opción CONTEXT_MEMORY transmitida por el descompresor, seleccionar por parte del compresor utilizar Perfil0 para realizar compresión de datos en paquetes a comprimir de un flujo de paquetes; y

10 después de determinar que un número de las utilizaciones del Perfil0 para paquetes del flujo de paquetes o el tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes alcanza un primer umbral, realizar por parte del compresor compresión de datos utilizando un tipo de Perfil al que pertenecen los paquetes a comprimir,

15 2. Procedimiento, según la reivindicación 1, en el que el compresor inicializa un primer contador (PackCounter) o un primer temporizador después de recibir la opción CONTEXT_MEMORY, el primer contador cuenta dicho número de las utilizaciones del Perfil0 para paquetes del flujo de paquetes o el primer temporizador cuenta dicho tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes, y el compresor determina si el primer umbral se alcanza según el primer contador o el primer temporizador.

20 3. Procedimiento, según la reivindicación 2, en el que el compresor inicializa además un segundo contador (RecCtxMemCounter) o un segundo temporizador después de recibir una opción CONTEXT_MEMORY que se refiere al flujo de paquetes por primera vez, el segundo contador cuenta el número de recepciones de la opción CONTEXT_MEMORY o el segundo temporizador cuenta el tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes, y después de que el compresor determina que se alcanza un segundo umbral por el segundo contador o el segundo temporizador, el compresor utiliza el Perfil0 de manera continuada para realizar compresión de datos sobre los paquetes a comprimir del flujo de paquetes.

30 4. Procedimiento, según la reivindicación 1 o 2, en el que el compresor establece un indicador de retroalimentación de opción (CtxMemFlag) como un primer indicador después de recibir la opción CONTEXT_MEMORY; establece el indicador de retroalimentación de opción (CtxMemFlag) como un segundo indicador después de recibir una retroalimentación ACK que representa una descompresión satisfactoria o cuando el primer contador (PackCounter) o el primer temporizador alcanza el primer umbral; y en el que el primer contador (PackCounter) o el primer temporizador se reinicia después de alcanzar el primer umbral.

35 5. Procedimiento, según la reivindicación 4, en el que el primer contador (PackCounter) o el primer temporizador cuenta cuando el indicador de retroalimentación de opción (CtxMemFlag) es el primer indicador, y deja de contar cuando el indicador de retroalimentación de opción (CtxMemFlag) es el segundo indicador.

40 6. Procedimiento, según la reivindicación 3, en el que, el segundo contador (RecCtxMemCounter) o el segundo temporizador se pone a cero después de que el compresor reciba una retroalimentación ACK.

45 7. Procedimiento, según la reivindicación 3, en el que el primer umbral es proporcional al valor actual del segundo contador actual (RecCtxMemCounter) o el segundo temporizador.

8. Compresor robusto de cabecera, que comprende:

50 un módulo de recepción de retroalimentación, configurado para recibir una retroalimentación transmitida por un descompresor, que incluye recibir una retroalimentación NACK de una descompresión fallida que lleva una opción CONTEXT_MEMORY;

un módulo de compresión de Perfil0, configurado para utilizar Perfil0 para realizar compresión de datos en paquetes a comprimir de un flujo de paquetes después de recibir la opción CONTEXT_MEMORY; y

55 un módulo de compresión de tipo de Perfil, configurado para realizar compresión de datos utilizando un tipo de Perfil al que pertenecen los paquetes a comprimir después de que el número de utilizaciones del Perfil0 para paquetes del flujo de paquetes o el tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes alcance un primer umbral.

60 9. Compresor robusto de cabecera, según la reivindicación 8, en el que el compresor robusto de cabecera comprende además:

un módulo de inicialización, configurado para inicializar un primer contador (PackCounter) o un primer temporizador después de recibir la opción CONTEXT_MEMORY;

el primer contador o el primer temporizador, configurado para contar dicho número de las utilidades del Perfil0 para paquetes del flujo de paquetes o para contar dicho tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes;

5 un módulo de determinación de umbral, configurado para determinar si el primer contador o el primer temporizador alcanzan el primer umbral.

10 10. Compresor robusto de cabecera, según la reivindicación 9, en el que el compresor robusto de cabecera comprende además un segundo contador (RecCtxMemCounter) o un segundo temporizador; el módulo de inicialización está configurado además para inicializar el segundo contador o el segundo temporizador después de recibir una opción CONTEXT_MEMORY del flujo de paquetes por primera vez; el segundo contador o el segundo temporizador está configurado para contar el número de repeticiones de la opción CONTEXT_MEMORY o cuenta el tiempo para utilizar el Perfil0 para paquetes del flujo de paquetes; el módulo de determinación de umbral está configurado además para determinar si el segundo contador o el segundo temporizador alcanzan un segundo umbral; y

20 el módulo de compresión de Perfil0 está configurado además para utilizar el Perfil0 de manera continuada para realizar compresión de datos en paquetes a comprimir del flujo de paquetes después de que se alcance el segundo umbral.

25 11. Compresor robusto de cabecera, según la reivindicación 8 o 9, en el que el compresor comprende además un módulo de mantenimiento de indicador de retroalimentación de opción, configurado para: establecer un indicador de retroalimentación de opción (CtxMemFlag) como un primer indicador después de recibir la opción CONTEXT_MEMORY; establecer el indicador de retroalimentación de opción (CtxMemFlag) como un segundo indicador después de recibir una retroalimentación ACK que representa una descompresión satisfactoria o cuando el primer contador (PackCounter) o el primer temporizador alcanzan el primer umbral; y el compresor robusto de cabecera está configurado además para reiniciar el primer contador (PackCounter) o el primer temporizador después de alcanzar el primer umbral.

30 12. Compresor robusto de cabecera, según la reivindicación 11, en el que el primer contador (PackCounter) o el primer temporizador está configurado además para contar cuando el indicador de retroalimentación de opción (CtxMemFlag) es el primer indicador, y para dejar de contar cuando el indicador de retroalimentación de opción (CtxMemFlag) es el segundo indicador.

35 13. Compresor robusto de cabecera, según la reivindicación 10, configurado además para poner a cero el segundo contador (RecCtxMemCounter) o el segundo temporizador después de recibir una retroalimentación ACK.

40 14. Compresor robusto de cabecera, según la reivindicación 10, en el que el primer umbral es proporcional al valor actual del segundo contador actual (RecCtxMemCounter) o del segundo temporizador.

45 15. Sistema de proceso de compresión robusta de cabecera, que comprende el compresor robusto de cabecera, según cualquiera de las reivindicaciones 8 a 14 y un descompresor, en el que el descompresor está configurado sólo para transmitir una retroalimentación ACK que representa una descompresión satisfactoria al compresor robusto de cabecera cuando la descompresión para paquetes de datos de un tipo de Perfil al que pertenece el flujo de paquetes es satisfactoria por primera vez después de transmitir una retroalimentación NACK de una descompresión fallida que lleva una opción CONTEXT_MEMORY.

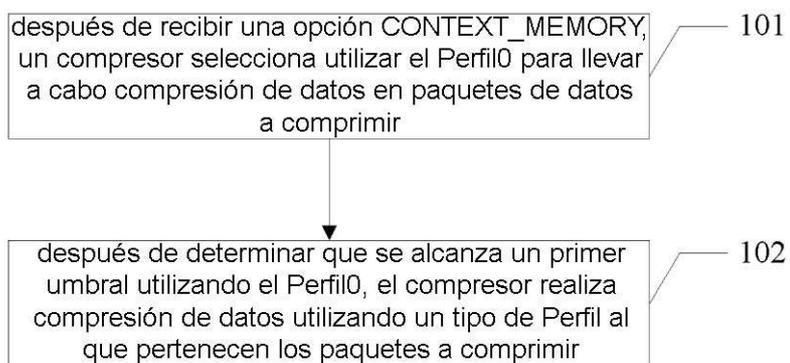


FIG. 1

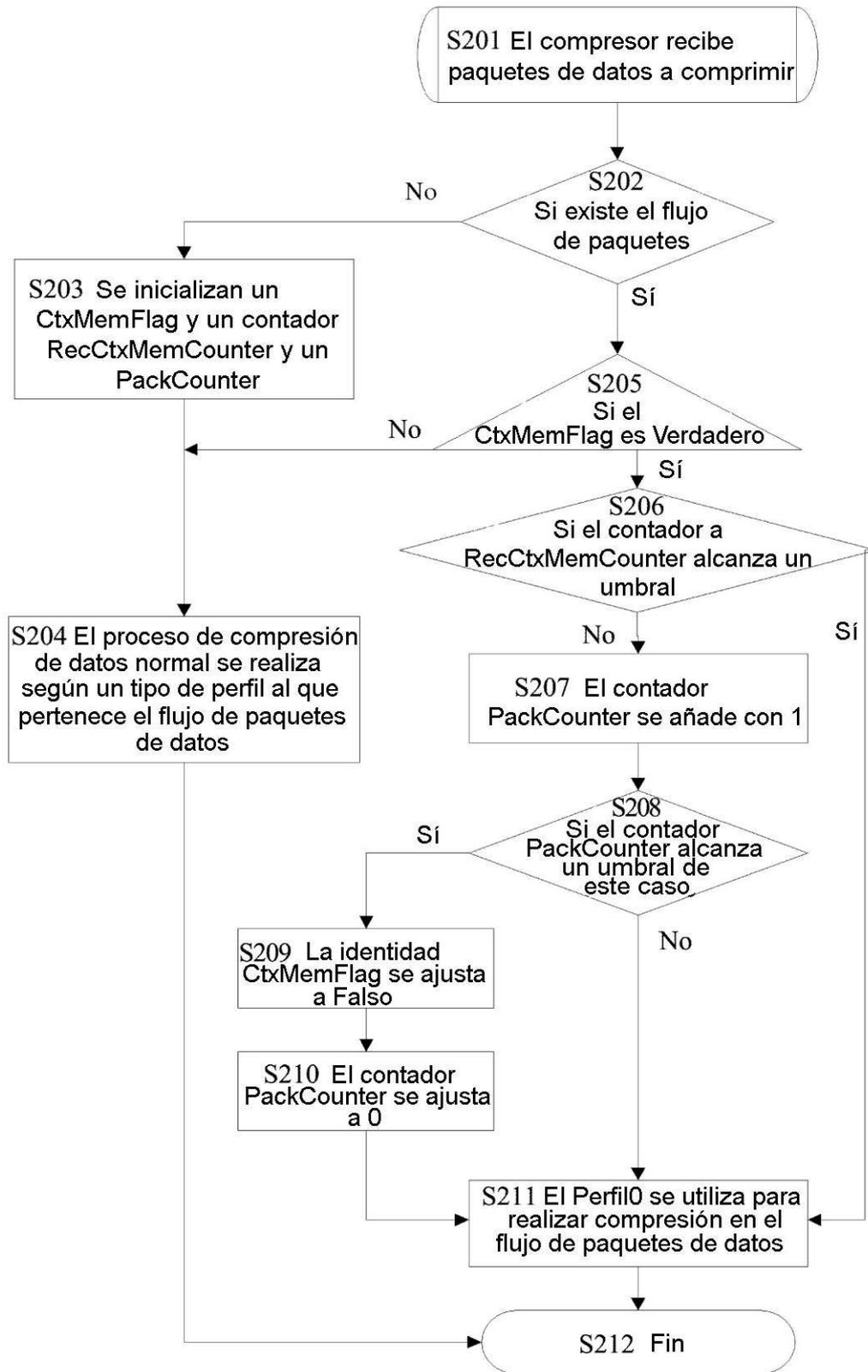


FIG. 2

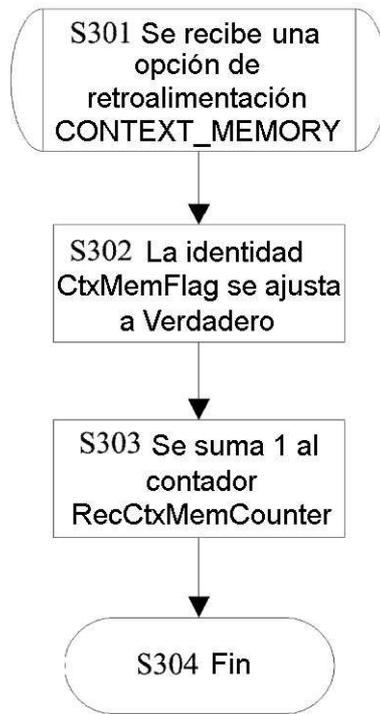


FIG. 3

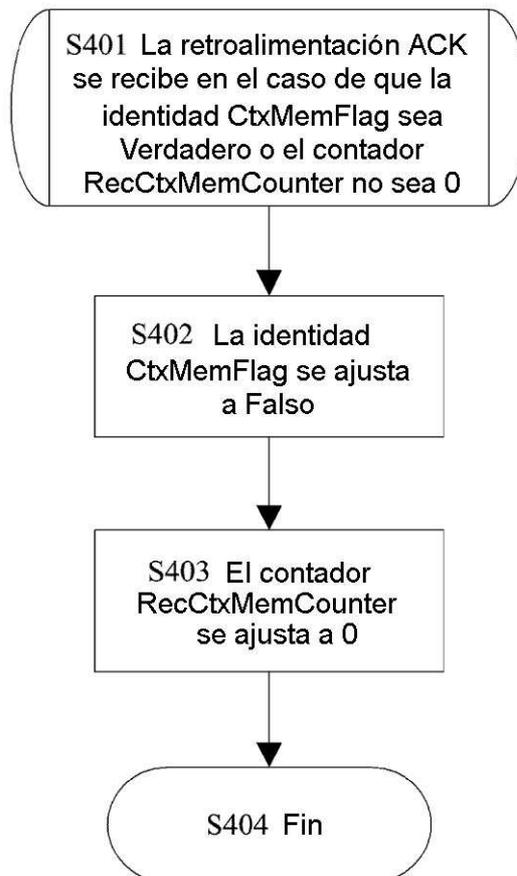


FIG. 4

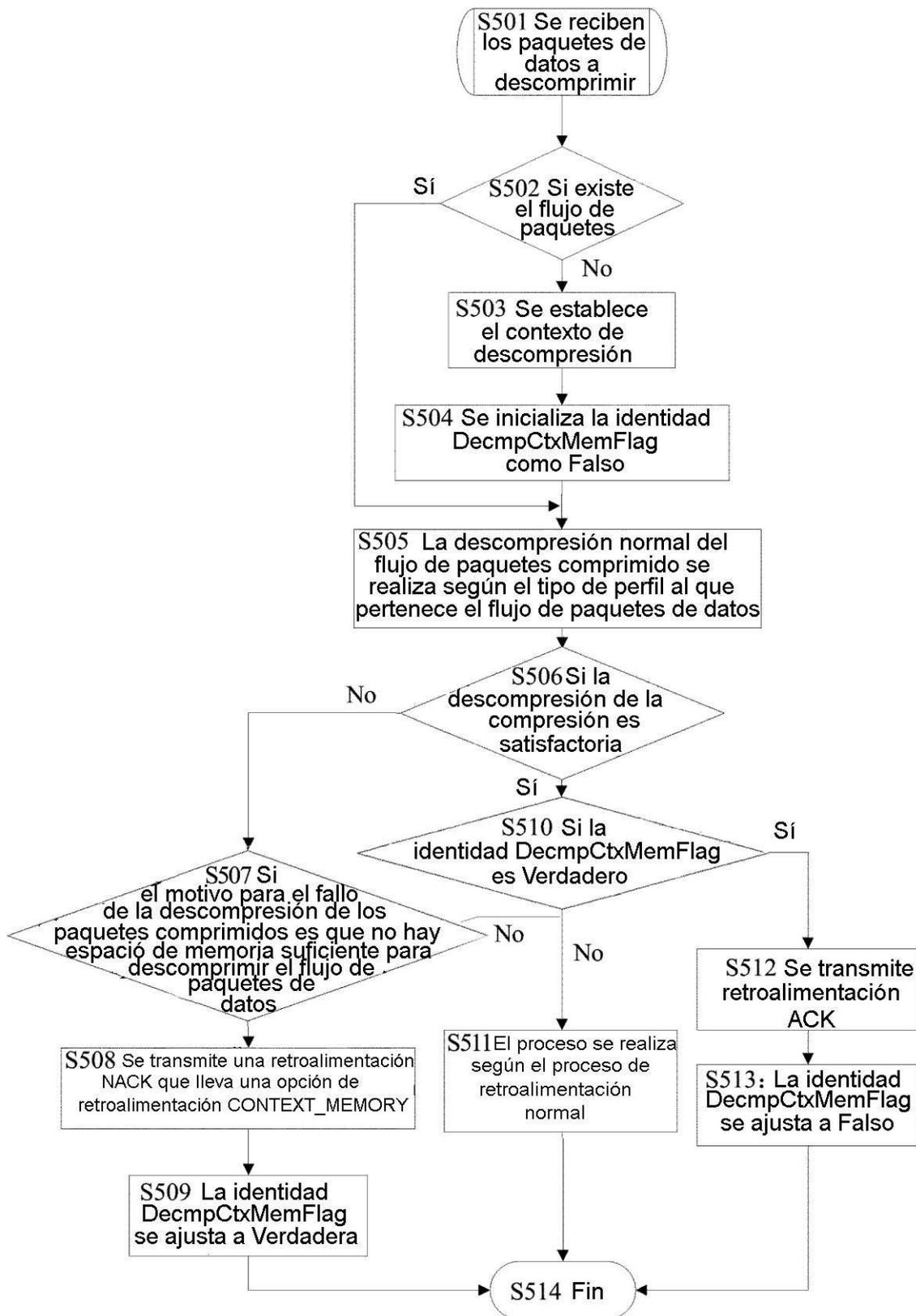


FIG. 5

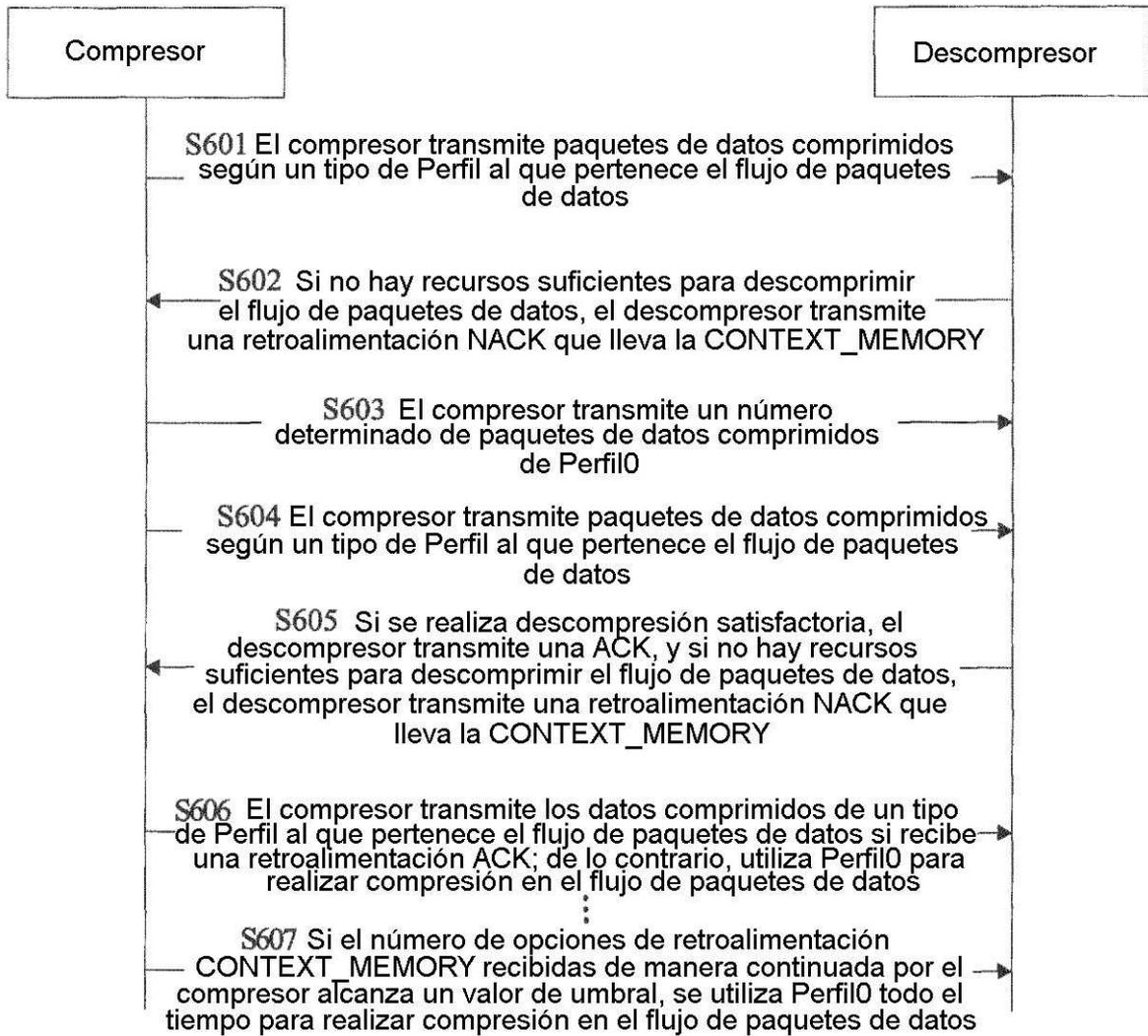


FIG. 6

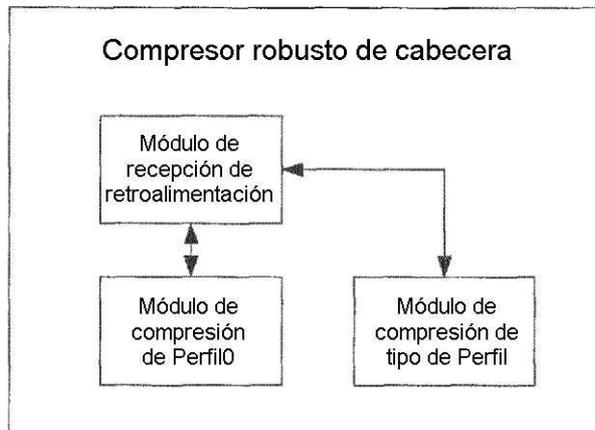


FIG. 7

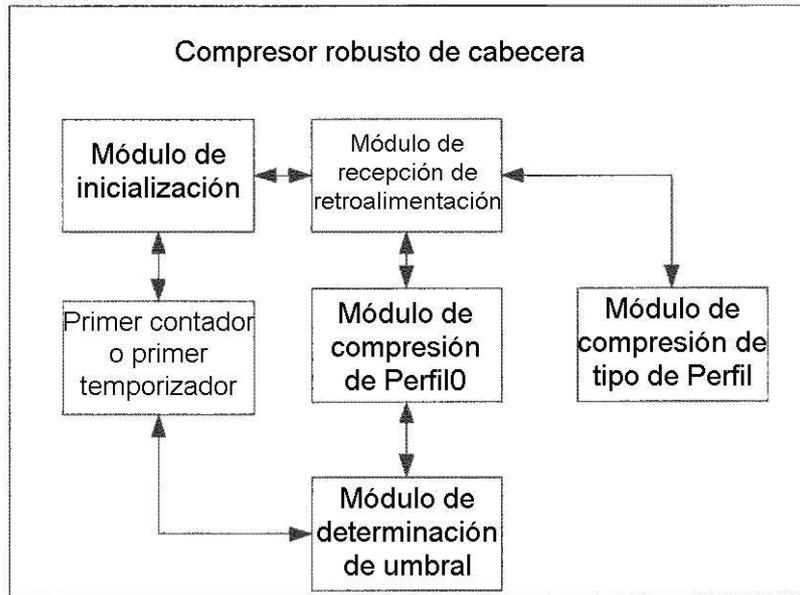


FIG. 8

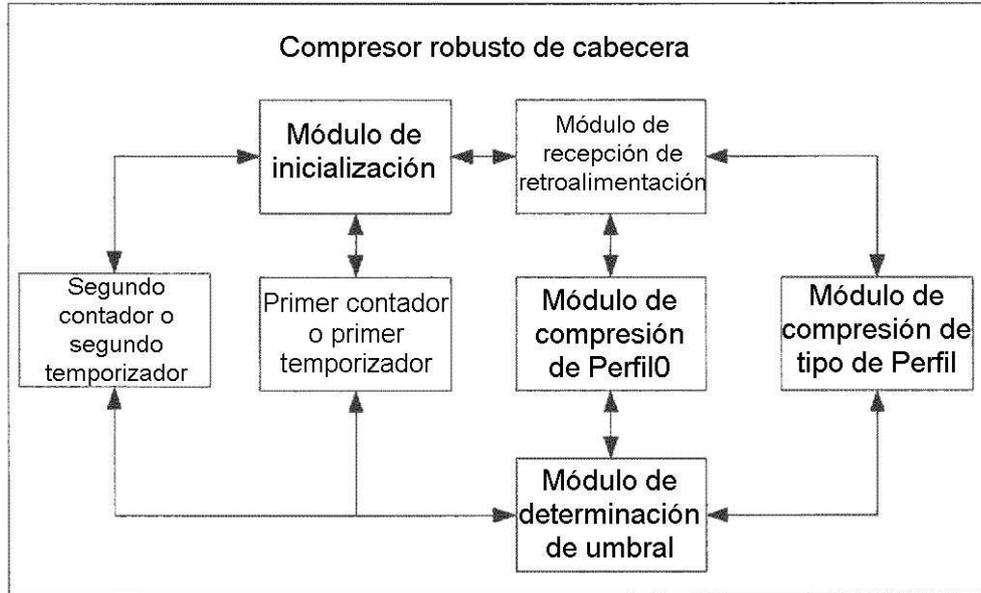


FIG. 9

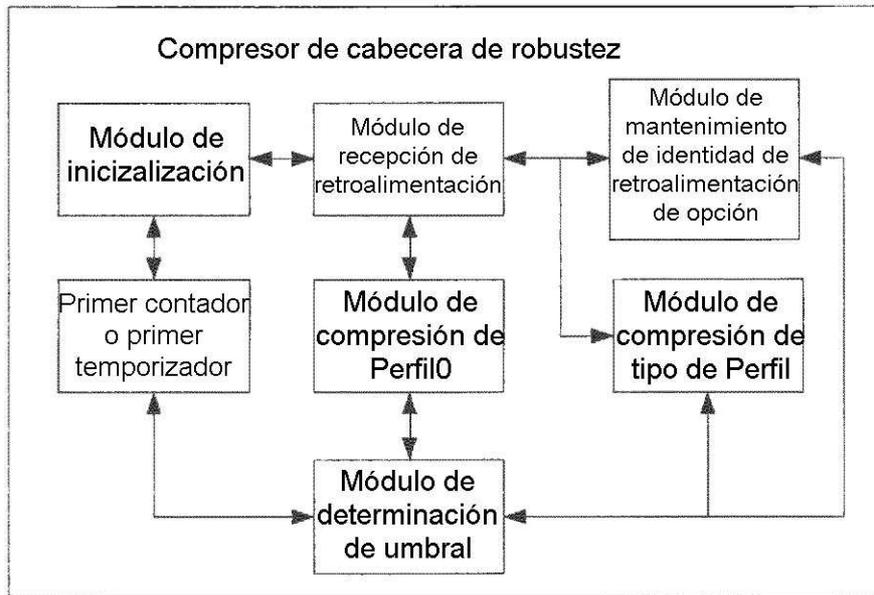


FIG. 10A

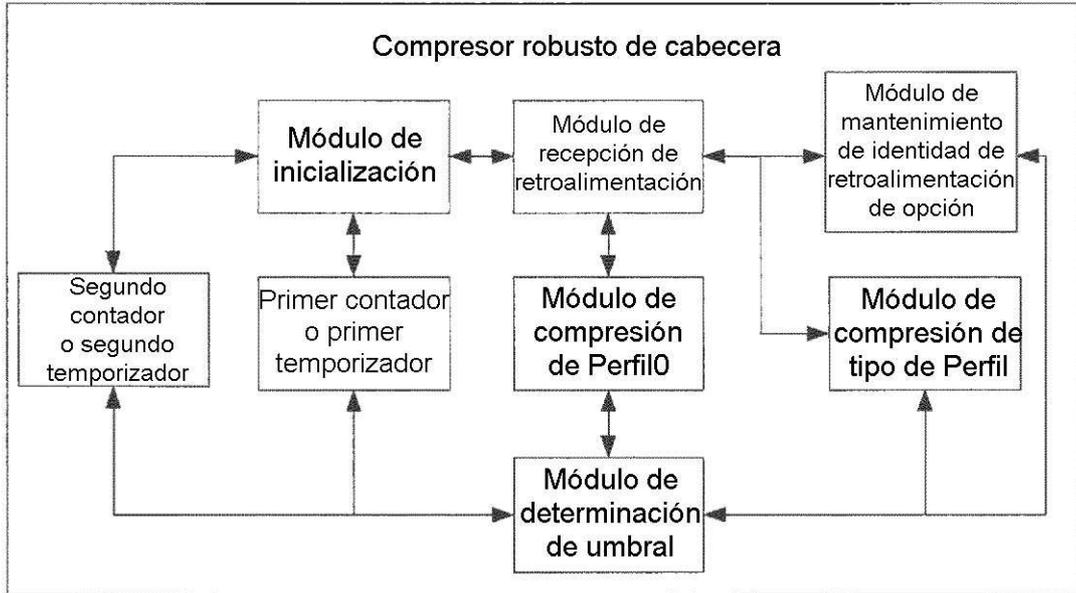


FIG. 10B