



19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 266 273**

51 Int. Cl.:  
**H04N 7/24** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Número de solicitud europea: **01977284 .7**

86 Fecha de presentación : **28.09.2001**

87 Número de publicación de la solicitud: **1415474**

87 Fecha de publicación de la solicitud: **06.05.2004**

54 Título: **Método y compresor para la compresión de información de indicación de tiempo de paquetes.**

30 Prioridad: **28.09.2000 US 236120 P**  
**12.10.2000 US 239703 P**

45 Fecha de publicación de la mención BOPI:  
**01.03.2007**

45 Fecha de la publicación del folleto de la patente:  
**01.03.2007**

73 Titular/es: **Nokia Corporation**  
**Keilalahdentie 4**  
**02150 Espoo, FI**

72 Inventor/es: **Leon, David y**  
**Khiem, Le**

74 Agente: **Curell Suñol, Marcelino**

ES 2 266 273 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**

Método y compresor para la compresión de información de indicación de tiempo de paquetes.

**5 Campo de la invención**

La presente invención se refiere a la compresión robusta de encabezamientos. El esquema de la compresión robusta de encabezamientos del estado de la técnica comprime los encabezamientos RTP/UDP/IP a típicamente un byte para flujos continuos de audio y dos bytes para flujos continuos de vídeo. Este nuevo esquema comprime los encabezamientos RTP/UDP/IP de algunos flujos continuos de vídeo a un byte.

**Antecedentes de la invención**

El problema actual de la compresión de encabezamientos es que, para algunos medios tales como el vídeo, tanto el número de secuencia (SN) comprimido como la indicación de tiempo (TS) comprimida se deben enviar en la mayor parte, cuando no en la totalidad, de los paquetes. La invención permite que se envíe solamente el SN, y obtener la TS a partir del SN. La invención queda definida por las reivindicaciones adjuntas.

**Sumario de la invención**

El nuevo esquema se aprovecha del patrón observado en el flujo continuo de medios para permitir que el compresor entre más frecuentemente en el estado de compresión más alto (estado SO o de segundo orden).

A medida que la invención se entienda mejor haciendo referencia a la siguiente descripción detallada y a los dibujos adjuntos, se apreciarán fácilmente otras ventajas.

**Breve descripción de los dibujos**

La Fig. 1 muestra casos típicos de la TS en función del SN para audio;  
 la Fig. 2 muestra casos típicos de la TS en función del SN para vídeo;  
 la Fig. 3A muestra la TS, patrón SO, en función del SN;  
 la Fig. 3B muestra la ID-IP en función del SN;  
 la Fig. 3C muestra el bit M en función del SN; y  
 la Fig. 4 es un diagrama lógico que muestra los estados lógicos del compresor.

**Descripción detallada de la invención***Marco conceptual de la compresión de encabezamientos*

Para obtener una visión general del marco conceptual de la compresión robusta, consúltese el proyecto Compresión robusta de encabezamientos (ROHC), draft-ietf-rohc-rtp-02, Solicitud De Comentarios (RFC) de Internet, el cual se incorpora al presente documento en forma de apéndice. Al mismo se le hará referencia en lo sucesivo como esquema actual o ROHC. La compresión puede estar entre tres estados diferentes: inicialización, primer orden, segundo orden. En el estado de inicialización, el compresor envía paquetes de encabezamiento completos (sin compresión). En el estado de primer orden, envía paquetes FO los cuales contienen solamente campos codificados que varían dinámicamente. Un tamaño típico de los encabezamientos FO es dos bytes o más. En el estado SO, envía solamente un SN codificado. El estado SO se define con respecto a un patrón que sigue una serie de campos del encabezamiento. Existen diferentes modos en los cuales puede funcionar el esquema de compresión. En el modo fiable, el compresor se asegura, a través de acuses de recibo del descompresor, de que el descompresor está sincronizado antes de entrar en un estado de compresión más alto. En el modo unidireccional u optimista, si el compresor estima que con toda probabilidad el descompresor está sincronizado, en ese caso entra en un estado de compresión más alto. El modo optimista y unidireccional requiere que el descompresor pueda comprobar, por ejemplo, a través del uso de una suma de comprobación, que el compresor y el descompresor están realmente sincronizados.

**60 Patrón SO**

El patrón actual se define de la siguiente manera:

- la diferencia de segundo orden (con respecto al SN) de la TS es nula
- la diferencia de segundo orden de ID-IP es nula
- La totalidad de campos restantes son constantes.

## ES 2 266 273 T3

Este esquema ha demostrado ser altamente adecuado para el audio. Para un flujo continuo típico de audio, el patrón debería ser verificado en relación con los paquetes generados durante una secuencia hablada (es decir, cuando la voz en cuestión es detectada y codificada por el codificador). Como durante el silencio habitualmente no se generan o se generan muy pocos paquetes (ruido de confort), esta situación da como resultado el envío de los primeros pocos paquetes de una secuencia hablada como paquetes FO y el envío del resto de los paquetes como SO. No obstante, este patrón no resulta adecuado para el vídeo.

Esta situación puede observarse en las figuras 1 y 2, las cuales muestran casos típicos de la TS en función del SN respectivamente para el audio y el vídeo. En la figura 1, es evidente que la diferencia de segundo orden es cero durante periodos de tiempo prolongados. No obstante, en la figura 2, los saltos de indicación de tiempo son mucho más frecuentes (en cada límite de cuadro) y la diferencia de segundo orden es nula solamente durante periodos de tiempo cortos (paquetes que pertenecen a una trama determinada los cuales presentan todos ellos las mismas indicaciones de tiempo).

Como consecuencia, podría ocurrir que en el caso del vídeo, el estado de compresión se quedase atascado en FO. En el modo fiable, si el enlace de TIEMPO DE IDA Y VUELTA (RTT) (tiempo de ida y vuelta) es mayor que el salto de cuadro (es decir, el tiempo entre dos cuadros codificados), el compresor no puede entrar de forma fiable en el estado SO (segundo orden). Incluso en el caso de un RTT muy rápido o en el modo optimista, el estado de compresión volvería al FO en cada límite de cuadro.

Tal como puede observarse a partir de la figura 2, la TS es una función escalonada del SN. Esta función escalonada podría ser un patrón usado para el SO. No obstante, para clasificarse como patrón, esta función debe ser totalmente conocida. En otras palabras, la longitud de un escalón y el salto entre escalones debería ser constante durante un periodo de tiempo suficientemente prolongado, es decir, varias veces el RTT. En la presente memoria se argumenta que esta puede ser la situación por lo menos cuando no se usan cuadros B. Los cuadros B se consideran posteriormente en la presente memoria. En efecto, los formatos de carga útil para la codificación de vídeo son tales que un paquete RTP determinado no puede contener datos de vídeo de dos cuadros diferentes. Dos paquetes RTP consecutivos (así generados por la aplicación emisora) bien presentarán la misma indicación de tiempo RTP si los mismos transportan datos de la misma imagen o bien su diferencia de indicación de tiempo reflejará el intervalo de tiempo entre sus instantes de muestreo de cuadro respectivos. Por esta razón, para que sean predecibles, el número de paquetes por cuadro y el salto de cuadro deberían ser constantes. Posteriormente se muestra en la presente memoria que en muchos casos un codificador de vídeo puede mantener constantes estos parámetros.

Con el fin de una mayor robustez, las aplicaciones de vídeo pueden escoger la paquetización de un cuadro de vídeo de manera que contenga un número constante de macrobloques o en otras palabras que cada cuadro de vídeo se trocee de forma idéntica. Ésta es una situación muy parecida a la de un paquete de audio que abarca una longitud determinada de audio. La única buena razón (por lo menos que se les haya ocurrido a los presentes inventores) por la que la aplicación emisora escogería realizar dicha operación de otra manera es si la misma deseara limitar el tamaño máximo de los paquetes. Esto implicaría que algunos cuadros pudieran tener más paquetes que el número habitual de paquetes por cuadro. No obstante, con frecuencia el número de dichos cuadros será limitado. Por ejemplo, se podría enviar un intracadro en un número mayor de paquetes. Esta situación implicaría que la compresión entrase en FO cuando se comprimesen estos paquetes.

A un codificador se le proporciona una frecuencia de cuadro objetivo. Siempre que el codificador se ajuste a su frecuencia de cuadro objetivo, el salto de la indicación de tiempo será constante. No obstante, habitualmente las implementaciones de los codificadores solamente se ajustan a una frecuencia de cuadro objetivo media y el salto de cuadro puede ser variable. Sin embargo, por lo menos en el caso del flujo continuo, un codificador puede usar una memoria intermedia mayor (un retardo mayor) la cual proporciona una mayor flexibilidad para el control de la frecuencia y para mantener la frecuencia de cuadro objetivo durante toda la conexión (o por lo menos durante un periodo de tiempo prolongado). Se espera que la invención resulte útil para los codificadores que se han diseñado de esta manera.

Adicionalmente, el bit marcador RTP se debería activar únicamente para el último paquete de un cuadro. Así, este también se puede obtener a partir del patrón.

Por esta razón se define un patrón de la manera siguiente:

- TS es cierta función  $f$  de SN, en la que  $f$  es más general que simplemente una extrapolación lineal. Por ejemplo,  $f$  puede ser una función escalonada de SN la cual presente escalones de longitud constante y saltos constantes entre escalones.
- El bit marcador es también cierta función  $g$  del SN. Por ejemplo, el bit marcador se activa únicamente para el último paquete de cada escalón.
- La diferencia de segundo orden ID-IP es nula (igual que para el audio).

Usando el ejemplo del vídeo, este patrón se muestra en las Figs. 3a, 3b, 3c para una serie de 40 paquetes en la que la frecuencia de cuadro es 10 (cuadros por segundo) fps y el número de paquetes por cuadro es 9.

## ES 2 266 273 T3

### *Lógica del compresor y el descompresor*

La única modificación en el esquema actual introducida por el patrón nuevo está en relación con cómo se realizan las transiciones del compresor y el descompresor entre el estado FO y SO. El compresor es el que toma la decisión sobre en qué estado funcionar. El descompresor sigue las decisiones del compresor. Funciona en el estado FO cuando recibe paquetes FO y funciona en el estado SO cuando recibe paquetes SO. En la figura 4 se muestra la lógica del compresor. En este caso el estado FO se divide conceptualmente en dos subestados FO\_1 y FO\_2.

El compresor entra en FO por FO\_1 y se mueve a FO\_2 si se detecta un patrón. El compresor va desde FO\_2 a SO cuando el descompresor está sincronizado, es decir, cuando el descompresor podría descomprimir paquetes SO.

De este modo, los aspectos característicos del esquema nuevo son:

- Cómo descomprime el descompresor paquetes en el estado SO
- Cómo adquiere el patrón el compresor
- Cómo sabe el compresor que el descompresor está sincronizado para entrar en el estado SO

A continuación se examina cada uno de estos aspectos.

### *Descompresión de paquetes SO*

En el estado SO, el descompresor debe conocer las funciones patrón f y g, para descomprimir paquetes SO. Nuevamente usando el ejemplo del vídeo, el descompresor debe conocer n, el número de paquetes por cuadro y TS\_inc el incremento de la indicación de tiempo entre dos cuadros. Además, debe haber descomprimido previamente de forma satisfactoria un paquete el cual era un cuadro de un primer paquete.

Denominemos SN\_0, TS\_0 al número de secuencia y a la indicación de tiempo de dicho paquete.

Para cualquier paquete SO entrante, el descompresor descomprime SN tal como en el esquema actual. Si n y m son el cociente y el módulo de  $SN - SN_0$  por q, es decir,  $SN - SN_0 = q * n + m$ , el descompresor calcula la TS del paquete según  $TS = TS_0 + q * TS_{inc}$ . El bit marcador M se activa solamente si  $m = n - 1$ . La totalidad del resto de campos se obtiene tal como en el esquema actual.

### *Detección del patrón*

El compresor puede determinar la función mediante observación del flujo continuo/aprendizaje o API o algunos otros medios. Nuevamente usando el ejemplo del vídeo, y considerando la observación del flujo continuo, el patrón nuevo requiere la obtención de suficientes paquetes del emisor antes de que se pueda tomar la decisión. Esto a su vez requiere almacenar temporalmente una copia de un cierto número de paquetes anteriores.

El patrón se podría detectar buscando en esta memoria intermedia tres paquetes cuyos (SN, TS, M, ID-IP) sean tales que las diferencias ID-IP de segundo orden sean nulas y (SN\_1, TS\_1, M\_1) (SN\_2, TS\_2, M\_2) (SN\_3, TS\_3, M\_3) sean tales que:

$$SN_2 = SN_1 + 1$$

$$TS_3 = TS_2$$

$$M_1 = 1$$

$$M_2 = 0$$

$$M_3 = 1$$

Esto implica que SN\_2 es un cuadro de un primer paquete y  $TS_{inc} = TS_2 - TS_1$  y el número de paquetes por cuadro es  $n = SN_3 - SN_2$ .

Existe un caso específico en el que el patrón se puede detectar por medio de dos paquetes (SN\_1, TS\_1, M\_1) (SN\_2, TS\_2, M\_2) de tal manera que:

$$SN_2 = SN_1 + 1$$

$$M_1 = 1$$

$$M_2 = 1$$

## ES 2 266 273 T3

En ese caso, existe solamente un paquete por cuadro.

### *Sincronización del descompresor*

5 En el esquema actual, el compresor únicamente necesita conseguir dos ACK del descompresor para asegurarse de que este último está sincronizado. El descompresor obtiene, a partir de los últimos dos paquetes recibidos, las diferencias de primer orden requeridas para descomprimir paquetes SO.

10 No obstante, con el patrón nuevo, esto no es suficiente. Existen varias formas según las cuales el compresor se puede asegurar de que el descompresor está sincronizado. En el presente documento se sugieren tres de ellas.

15  Después de detectar el patrón, el compresor puede enviar explícitamente las funciones patrón f y g hacia el descompresor usando la señalización dentro de la banda. Nuevamente usando el ejemplo del vídeo, el compresor envía n y TS\_inc, junto con una indicación de que el bit marcador se activa únicamente para el último paquete de cada escalón. A continuación, el compresor únicamente debe asegurarse de que el descompresor ha recibido un paquete con un bit marcador activado (primer paquete de trama). A continuación, el mismo sabe que el descompresor posee toda la información necesaria para descomprimir el paquete. Después de recibir un encabezamiento FO que transporta la descripción del patrón, el receptor debería intentar acusar el recibo de paquetes con el bit marcador activado de manera que el compresor pueda comenzar a enviar paquetes SO lo antes posible. Pros: la lógica del descompresor se mantiene a un nivel bajo (no es necesario realizar una detección del patrón). El compresor no necesita saber de antemano si el descompresor es capaz de interpretar la señalización dentro de la banda; dicha situación se puede señalar por medio de un RECHAZO proveniente del descompresor, con causa "No reconocido". Contras: tara adicional, cambios en el formato actual de los paquetes.

25  De forma alternativa, el compresor puede observar los acks recibidos desde el descompresor para determinar si el descompresor ha adquirido las funciones patrón f y g. El descompresor está también realizando la detección del patrón sobre los paquetes descomprimidos en el modo FO. Cuando se adquiere el patrón, dicha situación se señalará en los ACK subsiguientes enviados al compresor. Cuando el compresor consigue suficientes ACK para indicar que el patrón ha sido detectado, puede comenzar a enviar paquetes SO. Pros: la tara se mantiene a un nivel mínimo. No es necesario normalizar el formato de la señalización dentro de la banda. Contras: el descompresor debe realizar una detección del patrón lo cual conlleva una complejidad mayor y retardos más elevados en los casos en los que el enlace pierda paquetes usados para la detección; las funciones patrón se deben normalizar. El compresor también debe saber si el descompresor es capaz de detectar la función f (solamente la recepción de los ACK no garantiza que el descompresor haya adquirido la función): esta opción se podría realizar mediante cierto intercambio o negociación de capacidad.

30  El descompresor realiza la detección del patrón y el descompresor también realiza la detección del patrón para los paquetes cuyo recibo ha sido acusado por el descompresor. En otras palabras, el compresor intenta averiguar si el patrón se puede detectar usando solamente los paquetes que es seguro han sido recibidos por el descompresor. A continuación, el descompresor debería escoger acusar el recibo de paquetes de los que se sabe que son suficientes para detectar el patrón, por ejemplo, la tripleta mostrada anteriormente. Pros: no es necesaria ninguna modificación en el formato de los paquetes. Se pueden usar los mismos formatos de paquetes para el patrón de audio y el patrón de vídeo. Contras: complejidad adicional, retardo antes de entrar en el SO, se reduce la libertad de decodificador para escoger si realizar o no una ACK a un paquete.

### *Cuadros B y PB*

50 Se ha presentado el patrón como un patrón típico para vídeo. No obstante, en el caso de que se usen cuadros B, no se sigue este patrón. Los cuadros B no se consideran como un caso típico por las siguientes razones:

Los cuadros B son usados únicamente por un número limitado de códecs. Los mismos no se usan en el perfil simple del MPEG 4.

55  Para una videoconferencia de baja velocidad binaria, los cuadros B no se consideran adecuados.

Esto es debido a que el cuadro P requiere ser recibido antes de que se pueda decodificar el cuadro B. Para una frecuencia de cuadro típica de 10 fps, dicha situación significaría unos 100 ms adicionales en el retardo de extremo-a-extremo.

60 En el caso en el que se usen cuadros B, todavía podría disponerse de un patrón típico si el codificador escoge codificar un número fijo de cuadros B por número de cuadro codificado, por ejemplo, una de cada dos cuadros es un cuadro B.

### *Conclusión*

65 Este esquema podría resultar beneficioso para muchas aplicaciones. No existe ninguna penalización si una aplicación no sigue el patrón. Adicionalmente, si dicho esquema estuviera normalizado, podría ser un incentivo para las

## ES 2 266 273 T3

aplicaciones el escoger una estrategia de paquetización que fuera atractiva, desde el punto de vista funcional, para la compresión de encabezamientos, es decir, la cual siguiera el patrón SO. En particular, los diseñadores de la futura aplicación de vídeo para terminales móviles de 3G podrían tener en cuenta esta opción.

5 APÉNDICE - PROYECTO RFC Compresión Robusta de Encabezamientos

draft-ietf-rohc-rtp-02.txt

10 Compresión Robusta de Encabezamientos (ROHC) <draft-ietf-rohc-rtp-02.txt>

### Estado de la presente memoria

15 El presente documento es un Proyecto de Internet y está en total conformidad con todas las disposiciones de la Sección 10 del RFC2026.

Los Proyectos de Internet son documentos de trabajo del Grupo de Trabajo de Ingeniería de Internet (IETF), sus áreas, y sus grupos de trabajo. Obsérvese que otros grupos también pueden distribuir documentos de trabajo como Proyectos de Internet.

20 Los Proyectos de Internet son documentos de borrador válidos durante un máximo de seis meses y pueden ser actualizados, sustituidos, o quedar obsoletos por otros documentos en cualquier momento. Resulta inadecuado usar los Proyectos de Internet como material de referencia o citarlos de otra manera que no sea “trabajo en curso”.

25 Se puede acceder a la lista de Proyectos de Internet actuales en <http://www.ietf.org/ietf/lid-abstracts.txt>.

Se puede acceder a la lista de Directorios Ocultos de los Proyectos de Internet en <http://www.ietf.org/shadow.html>.

30 Este documento es un producto de IETF ROHC WG. Los comentarios deberían ir dirigidos a su lista de correo, [rohc@cdt.luth.se](mailto:rohc@cdt.luth.se).

### Resumen

35 Los esquemas existentes de compresión de encabezamientos no funcionan bien cuando se usan a través de enlaces con tasas de errores significativas, especialmente cuando el tiempo de ida y vuelta del enlace es largo. Para muchos enlaces con ancho de banda limitado en los que la compresión de los encabezamientos es esencial, dichas características son comunes.

40 En el presente documento se introduce un marco conceptual de compresión de encabezamientos y un esquema de compresión de encabezamientos altamente robusto y eficaz, adaptable a las características del enlace sobre el cual se usa y también a las propiedades de los flujos de paquetes que comprime.

### Historial de Revisión

- 45 - 02: Cambios principales después del 48° IETF
- 01: Cambios editoriales poco significativos para el 48° IETF
- 00: Documento creado a partir de presentaciones ROHC

50

### TABLA DE CONTENIDO

Estado de la presente memoria	1
55 Resumen	2
Historial de Revisión	2
Tabla de contenido	3
0. Planificación temporal interna de corto plazo del ROHC WG	7
60 1. Introducción	8
2. Terminología	10
3. Antecedentes	14
3.1. Principios básicos de la compresión de encabezamientos	14
3.2. Esquemas existentes de la compresión de encabezamientos	14
65 3.3. Requisitos de un esquema nuevo de compresión de encabezamientos	16
3.4. Clasificación de campos de encabezamientos	16

# ES 2 266 273 T3

(Continuación)

	4. Marco conceptual de la compresión de encabezamientos	18
5	4.1. Suposiciones de funcionamiento	18
	4.2. Dinamicidad	19
	4.3. Estados de compresión y de descompresión	20
	4.3.1. Estados del compresor	20
10	4.3.1.1. Estado de Inicio y Restauración (IR)	21
	4.3.1.2. Estado de Primer Orden (FO)	21
	4.3.1.3. Estado de Segundo Orden (SO)	21
	4.3.2. Estados del descompresor	22
	4.4. Modos de funcionamiento	22
15	4.4.1. Modo unidireccional - modo U	23
	4.4.2. Modo optimista bidireccional - modo O	24
	4.4.3. Modo fiable bidireccional - modo R	24
	4.5. Métodos de codificación	24
20	4.5.1. Codificación de Bits Menos Significativos (LSB)	24
	4.5.2. Codificación LSB basada en ventanas (codificación W-LSB)	26
	4.5.3. Codificación Escalonada de las Indicaciones de Tiempo RTP	27
	4.5.4. Compresión de las Indicaciones de Tiempo RTP, Basada en Temporizadores	29
25	4.5.5. Codificación de ID-IP con desviación	31
	4.5.6. Valores independientes de longitud variable	32
	4.5.7. Valores codificados a través de varios campos en encabezamientos comprimidos	32
	5. El protocolo	34
30	5.1. Estructuras de datos	34
	5.1.1. Parámetros por canal	34
	5.1.2. Parámetros por CID, perfiles	34
	5.1.3. Contextos	34
35	5.2. Tipos de paquetes	34
	5.2.1. Formatos de paquetes desde el compresor al descompresor	35
	5.2.2. Paquetes de realimentación desde el descompresor al compresor	36
	5.2.3. Parámetros necesarios para la transición del modo	36
40	5.3. Funcionamiento en el modo unidireccional	37
	5.3.1. Estados y lógica del compresor (modo U)	37
	5.3.1.1. Lógica de transición de estados (modo U)	37
	5.3.1.1.1. Planteamiento optimista, transición en sentido ascendente	37
	5.3.1.1.2. Temporizaciones, transición en sentido descendente	38
45	5.3.1.1.3. Necesidad de actualizaciones, transición en sentido descendente	38
	5.3.1.2. Lógica de compresión y paquetes usados (modo U)	38
	5.3.1.3. Realimentación en el modo unidireccional	38
	5.3.2. Estados y lógica del descompresor (modo U)	39
50	5.3.2.1. Lógica de transición de estados (modo U)	39
	5.3.2.2. Lógica de descompresión (modo U)	39
	5.3.2.2.1. Decidir si está permitida la descompresión	39
	5.3.2.2.2. Reconstruir y verificar el encabezamiento	40
55	5.3.2.2.3. Causas de las discordancias de la CRC	40
	5.3.2.2.4. Detección de daños en el contexto	41
	5.3.2.2.5. Reparación de reinicios cíclicos del SN	41
	5.3.2.2.6. Reparación de actualizaciones incorrectas del SN	42
60	5.3.2.3. Realimentación en el modo unidireccional	43
	5.4. Funcionamiento en el modo optimista bidireccional	44
	5.4.1. Estados y lógica del compresor (modo O)	44
	5.4.1.1. Lógica de transición de estados	44
65	5.4.1.1.1. Solicitudes de contexto, acuses de recibo negativos (acuses NACK)	44
	5.4.1.1.2. Acuses de recibo opcionales	44
	5.4.1.2. Lógica de compresión y paquetes usados	44
	5.4.2. Estados y lógica del descompresor	45

# ES 2 266 273 T3

(Continuación)

	5.4.2.1. Lógica de descompresión	45
5	5.4.2.1.1. Descompresión de indicaciones de tiempo basada en temporizadores	45
	5.4.2.2. Lógica de realimentación	45
	5.5. Funcionamiento en el modo fiable bidireccional	46
	5.5.1. Estados y lógica del compresor	46
	5.5.2. Estados y lógica del descompresor	46
10	5.5.2.1. Lógica de descompresión	46
	5.5.2.2. Lógica de realimentación	46
	5.6. Transiciones del modo	48
	5.6.1. Compresión y descompresión durante transiciones del modo	48
15	5.6.2. Transición desde el modo Unidireccional al Optimista	49
	5.6.3. Desde el modo Optimista al Fiable	50
	5.6.4. Desde el modo Unidireccional al Fiable	50
	5.6.5. Desde el modo Fiable al Optimista	50
20	5.6.6. Transición al modo Unidireccional	51
	5.7. Formatos de los paquetes	53
	5.7.1. Tipo de paquete 0: UO-0, R-0, R-0-CRC	53
	5.7.2. Tipo de paquete 1 (modo R): R-1, R-1-TS, R-1-ID	53
25	5.7.3. Tipo de paquete 1 (modos UO): UO-1, UO-1-ID, UO-1-TS	55
	5.7.4. Tipo de paquete 2: UOR-2	56
	5.7.5. Formatos de extensión	57
	5.7.6. Paquetes de realimentación: REALIMENTACIÓN-3 y REALIMENTACIÓN-4	60
30	5.7.7. Paquetes IR e IR-DYN	62
	5.7.7.1. Estructura básica del paquete IR	62
	5.7.7.2. Estructura básica del paquete IR-DYN	63
	5.7.7.3. Inicialización del Encabezamiento IPv6 [IPv6]	64
35	5.7.7.4. Inicialización del Encabezamiento IPv4 [IPv4, sección 3.1]	65
	5.7.7.5. Inicialización del Encabezamiento UDP [RFC-768]	65
	5.7.7.6. Inicialización del Encabezamiento RTP [RTP]	66
	5.7.7.7. Encabezamiento de Encapsulación Mínima [RFC-2004, sección 3.1]	67
40	5.8. Compresión Basada en Listas	68
	5.8.1. Compresión CSRC	68
	5.8.1.1. Clasificación de la Transformación para la Lista CSRC	68
	5.8.1.2. Esquemas de Codificación	68
	5.8.1.3. Formato de la lista CSRC comprimida	69
45	5.8.1.3.1. Esquema de Solamente Inserción	69
	5.8.1.3.1.1. Modo R	69
	5.8.1.3.1.2. Modos UO	70
	5.8.1.3.2. Esquema de Solamente Eliminación	71
50	5.8.1.3.2.1. Modo R	71
	5.8.1.3.2.2. Modo UO	71
	5.8.1.3.3. Esquema Genérico	72
	5.8.1.3.3.1. Modo R	72
55	5.8.1.3.3.2. Modo UO	73
	5.8.2. Compresión de Encabezamientos para Encabezamientos de Extensión IPv6	73
	5.8.2.1. Terminología	74
	5.8.2.2. Clasificación de la Transformación y Esquemas de Codificación	74
60	5.8.2.2.1. Clasificación de la Transformación	74
	5.8.2.2.2. Esquemas de Codificación	75
	5.8.2.2.3. Tratamiento Especial	75
	5.8.2.2.3.1. Tratamiento Especial del AH	75
	5.8.2.2.3.2. Encabezamiento de Carga Útil de Seguridad de Encapsulación	76
65	5.8.2.2.3.3. Tratamiento Especial del Campo del Siguiendo Encabezamiento	76
	5.8.2.3. Formato de los Paquetes	78
	5.8.2.3.1. Formato en el encabezamiento “11” de extensión	78

# ES 2 266 273 T3

(Continuación)

	5.8.2.3.2. Formato del encabezamiento de extensión IPv6 comprimido	79
5	5.8.2.3.2.1. Esquema de Solamente Inserción	79
	5.8.2.3.2.1.1. Modo R	79
	5.8.2.3.2.2. Esquema de Solamente Eliminación	80
	5.8.2.3.2.2.1. Modo R	80
10	5.8.2.3.2.3. Esquema de Solamente Cambio del Contenido	81
	5.8.2.3.2.3.1. Modo R	81
	5.8.2.3.2.3.2. Modo UO	82
	5.8.2.3.2.4. Esquema sin Compresión (común para el modo R y los modos UO)	82
15	5.9. Comprobaciones CRC de compresión de encabezamientos, alcance y polinomios	83
	5.9.1. Comprobaciones CRC de los paquetes IR e IR-DYN	83
	5.9.2. Comprobaciones CRC en paquetes comprimidos	83
	6. Aspectos de la implementación	84
20	6.1. Descompresión inversa	84
	6.2. RTCP	85
	7. Tareas adicionales	86
	7.3. Tunelización	86
	7.3.1. Compresión de Encabezamientos para el Encabezamiento de Tunelización IPv4	86
25	7.3.1.1. Tipo de Campos de los Encabezamientos de Tunelización IPv4 Móvil	86
	7.3.1.2. Compresión de Encabezamientos de Tunelización en MIPv4	87
	7.3.1.2.1. Encapsulación IP en IP en el IPv4	87
	7.3.1.2.2. Encapsulación Mínima en el IPv4	87
30	7.3.1.2.3. Encapsulación Genérica de Encaminamiento en el IPv4	87
	7.4. Tráfico UDP no RTP	88
	8. La sección 8 ha sido eliminada	90
	9. Consideraciones sobre la seguridad	90
35	10. Acuses de recibo	90
	11. Consideraciones sobre la propiedad intelectual	91
	12. Referencias	92
	13. Direcciones de los autores	93
40	Apéndice A. Clasificación detallada de los campos de los encabezamientos	94
	A.1. Clasificación general	94
	A.1.1. Campos de los encabezamientos IPv6	95
	A.1.2. Campos de los encabezamientos IPv4	96
45	A.1.3. Campos de los encabezamientos UDP	98
	A.1.4. Campos de los encabezamientos RTP	99
	A.1.5. Resumen para IP/UDP/RTP	100
	A.2. Análisis de patrones de cambio de los campos de los encabezamientos	100
50	A.2.1. Identificación IPv4	102
	A.2.2. Clase de Tráfico / Tipo De Servicio IP	103
	A.2.3. Límite de Saltos / Tiempo De Vida IP	104
	A.2.4. Suma de Comprobación UDP	104
55	A.2.5. Contador CSRC RTP	104
	A.2.6. Marcador RTP	104
	A.2.7. Tipo de Carga Útil RTP	104
	A.2.8. Número de Secuencia RTP	104
	A.2.9. Indicación de Tiempo RTP	104
60	A.2.10. Fuentes Contribuyentes (CSRC) RTP	105
	A.3. Estrategias de compresión de los encabezamientos	105
	A.3.1. No enviar en absoluto	105
	A.3.2. Transmitir sólo inicialmente	106
65	A.3.3. Transmitir inicialmente, pero estar preparado para actualizar	106
	A.3.4. Estar preparado para actualizar o para enviar tales como estén frecuentemente	106
	A.3.5. Garantizar robustez continua	106
	A.3.6. Transmitir tal como están en todos los paquetes	107

## ES 2 266 273 T3

(Continuación)

	A.3.7. Establecer y estar preparado para actualizar la diferencia delta	107
5	Apéndice E - Ejemplos de Codificación	108
	E.1. VLE Básica	108
	E.2. VLE Basada en Temporizadores	109

(Nota del editor: El TOC no ha sido actualizado necesariamente).

Se ha marcado el texto considerado discutible poniéndolo en cursiva, y el texto que simplemente se cree que debería ser eliminado tachándolo).

### 0. Planificación temporal interna de corto plazo del ROHC WG

Este documento plasma el estado de la especificación ROHC RTP tal como estaba el 18 de septiembre de 2000. A título de información, la planificación temporal interna de corto plazo ROHC WG es la siguiente:

20	18 de septiembre	ROHC-02 completado (presente documento)
		Revisión y descripción del proyecto
		Contribuciones complementarias creadas
25	29 de septiembre	Límite para la revisión y descripción del proyecto
	2 de octubre	ROHC-03 completado
	4 de octubre	Decisión sobre si el proyecto está preparado para la última llamada

Si se decide que no está preparado para la última llamada WG en este momento:

35	4 de octubre	Identificar los aspectos que faltan/incorrectos
		Descripción y contribuciones adicionales
	13 de octubre	Límite para las “segundas” revisión y descripción del proyecto
40	16 de octubre	ROHC-04 Completado

Última llamada WG al finalizar el ROHC-03 o ROHC-04, respectivamente (dependiendo de la evolución tal y como se ha indicado anteriormente).

### 1. Introducción

Durante los últimos cinco años, el público en general ha llegado que se debe utilizar de forma habitual en particular dos tecnologías de comunicaciones: la telefonía celular e Internet. La telefonía celular ha proporcionado a sus usuarios la posibilidad revolucionaria de estar siempre accesibles con una calidad de servicios razonable con independencia del sitio en el que se encuentre. El servicio principal proporcionado por los terminales dedicados ha sido la voz. Por otro lado, desde su inicio, Internet ha estado diseñada para múltiples servicios y su flexibilidad para todo tipos de uso ha sido uno de sus valores. Habitualmente, los terminales de Internet han sido polivalentes y se han conectado a través de conexiones fijas. En ocasiones, la calidad experimentada de algunos servicios (tales como la telefonía por Internet) ha sido baja.

Actualmente, la telefonía IP está cobrando fuerza gracias a soluciones técnicas mejoradas. Parece razonable pensar que en los próximos años, el IP se convertirá en una forma usada habitualmente para implementar la telefonía. Algunos enlaces futuros de telefonía celular también podrían basarse en el IP y la telefonía IP. Los teléfonos celulares se han convertido en dispositivos más polivalentes, y pueden disponer de pilas de protocolos IP que soporten no solamente el audio y el vídeo, sino también la navegación web, el correo electrónico, juegos, etcétera.

En ese caso, uno de los escenarios que se prevén podría ser el de la Figura 1.1, en el que dos terminales móviles se están comunicando entre sí. Ambos están conectados a estaciones base a través de enlaces celulares, y las estaciones base están conectadas entre sí a través de una red por cable (o posiblemente inalámbrica). En lugar de dos terminales móviles, también podría haber evidentemente un móvil y un terminal por cable, aunque el caso de los dos enlaces celulares es técnicamente más exigente.

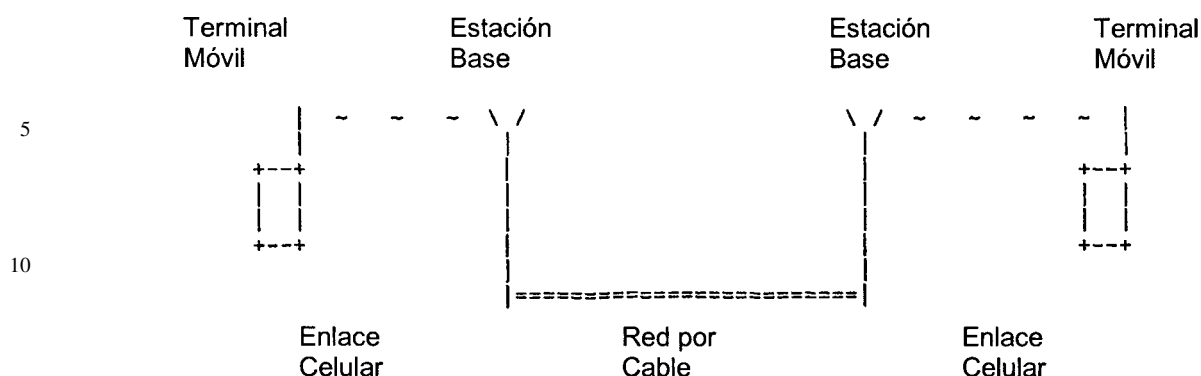


Figura 1.1: Escenario para la telefonía IP respecto a enlaces celulares

Es evidente que la red por cable se puede basar en el IP. Con los enlaces celulares, la situación es menos clara. El IP podría terminar en la red fija, y se podrían implementar soluciones especiales para cada servicio soportado a través del enlace celular. No obstante, esta situación limitaría la flexibilidad de los servicios soportados. Si fuera viable técnica y económicamente, una solución con un IP puro en su totalidad de terminal a terminal presentaría ciertas ventajas. No obstante, para conseguir que esta situación sea una alternativa viable, se debe hacer frente a una serie de problemas, en particular problemas referentes a la eficacia del ancho de banda.

Para sistemas telefónicos celulares, es de vital importancia el uso de los escasos recursos de radiocomunicaciones de una forma eficaz. Es crucial disponer de un número suficiente de usuarios por célula, de otro modo los costes del despliegue serán prohibitivos [CELL]. La calidad del servicio de voz también debería ser tan buena como en los sistemas celulares actuales. Es probable que incluso con soporte para servicios nuevos, la calidad inferior del servicio de voz sea aceptable únicamente si los costes se reducen significativamente.

Uno de los problemas con los enlaces celulares sobre IP cuando se usan para conversaciones de voz interactivas es la gran tara de los encabezamientos. Lo más probable es que los datos de voz para la telefonía IP sean transportados mediante RTP [RTP]. En ese caso, un paquete, además de la estructuración en tramas de la capa de enlace, tendrá un encabezamiento IP [IPv4] (20 octetos), un encabezamiento UDP [UDP] (8 octetos), y un encabezamiento RTP (12 octetos) para un total de 40 octetos. Con el IPv6 [IPv6], el encabezamiento IP es de 40 octetos para un total de 60 octetos. El tamaño de la carga útil depende de la codificación de la voz y de los tamaños de las tramas que se estén usando y puede tener un valor tan bajo como entre 15 y 20 octetos.

A partir de estos números, es evidente la necesidad de reducir los tamaños de los encabezamientos por razones de eficacia. No obstante, los enlaces celulares presentan características que provocan que la compresión de los encabezamientos según se define en [IPHC, CRTP, PPPHC] se comporte a un nivel inferior al satisfactorio. La característica más importante es el comportamiento disipativo de los enlaces celulares, en los que se debe aceptar un índice de errores de bit (BER) de un valor tan alto como  $1e-3$  para mantener una utilización eficaz de los recursos de radiocomunicaciones [CELL]. En situaciones de funcionamiento rigurosas, el BER puede tener un valor tan alto como  $1e-2$ . La otra característica problemática es el gran tiempo de ida y vuelta (RTT) del enlace celular, el cual puede presentar un valor tan alto como entre 100 y 200 milisegundos [CELL]. Un problema adicional es que el BER residual no es insignificante, es decir, las capas inferiores pueden entregar ocasionalmente tramas que contengan errores no detectados. Un esquema viable de compresión de encabezamientos para enlaces celulares debe poder gestionar las pérdidas en el enlace entre el punto de compresión y de descompresión así como las pérdidas antes del punto de compresión.

El ancho de banda es el recurso más costoso en los enlaces celulares. El poder de procesado es muy económico en comparación con este último. Por esta razón, la simplicidad de la implementación o computacional de un esquema de compresión de encabezamientos es menos importante que su relación de compresión y robustez.

## 2. Terminología

Las expresiones clave “DEBE”, “NO DEBE”, “REQUERIDO”, “DEBERÍA”, “NO DEBERÍA”, “RECOMENDADO”, “PUEDE”, y “OPCIONAL” en el presente documento deben interpretarse tal como se describen en el RFC 2119.

### BER

Índice de Errores de Bits. Los enlaces celulares de radiocomunicaciones pueden tener un BER bastante alto. En el presente documento, el BER se proporciona habitualmente en forma de una probabilidad, aunque también es necesario considerar la distribución de errores ya que los errores de bits no son independientes.

## ES 2 266 273 T3

### *Enlaces celulares*

Enlaces inalámbricos entre terminales móviles y estaciones base. El BER y el RTT son bastante altos para obtener un sistema eficaz en conjunto.

### *Eficacia de la compresión*

El rendimiento de un esquema de compresión de encabezamientos se puede describir con tres parámetros, eficacia de la compresión, robustez y transparencia de la compresión. La eficacia de la compresión se determina por la magnitud en la que se reducen los tamaños de los encabezamientos por parte del esquema de compresión.

### *Transparencia de la compresión*

El rendimiento de un esquema de compresión de encabezamientos se puede describir con tres parámetros, eficacia de la compresión, robustez y transparencia de la compresión. La transparencia de la compresión es una medida de la bondad del esquema a la hora de garantizar que los encabezamientos descomprimidos son semánticamente idénticos a los encabezamientos originales. Si todos los encabezamientos descomprimidos son semánticamente idénticos a los encabezamientos originales correspondientes, la transparencia es del 100 por cien. La transparencia de la compresión es alta cuando la propagación de daños es baja.

### *Contexto*

El contexto es el estado que usa el compresor para comprimir un encabezamiento y que usa el descompresor para descomprimir un encabezamiento. El contexto contiene básicamente la versión no comprimida del último encabezamiento enviada (compresor) o recibido (descompresor) a través del enlace, excepto para los campos del encabezamiento que se incluyan "tal como están" en encabezamientos comprimidos o que se puedan deducir a partir de, por ejemplo, el tamaño de la trama a nivel del enlace. El contexto también puede contener datos adicionales.

Cuando el contexto del descompresor no está en concordancia con el contexto del compresor, puede que la descompresión del encabezamiento no consiga reproducir el encabezamiento original. Esta situación se puede producir cuando el contexto del descompresor no se ha inicializado adecuadamente o cuando se han perdido o dañado paquetes entre el compresor y el descompresor. Se dice que los paquetes para los cuales el descompresor detecta que no se pueden descomprimir debido a contextos discordantes se han perdido debido a daños en el contexto.

### *Mecanismo de reparación del contexto*

Para evitar daños excesivos en el contexto, es necesario un mecanismo de reparación de contextos. Los mecanismos de reparación de contextos se pueden basar en solicitudes explícitas de actualizaciones de contexto, actualizaciones periódicas enviadas por el compresor, o métodos para una reparación local en el lado del descompresor.

### *Propagación de daños*

Generación de encabezamientos descomprimidos incorrectos debido a daños en un(os) paquete(es) previo(s).

### *Propagación de pérdidas*

No se consigue descomprimir encabezamientos debido a pérdidas de una(s) trama(s) previa(s).

### *Detección de errores*

Detección de errores. Si la detección de errores no es perfecta, se producirán errores residuales.

### *Propagación de errores*

Propagación de daños o propagación de pérdidas.

### *FLR*

Tasa de Pérdida de Tramas, proporcionada en forma de una probabilidad de que se pierda una trama sobre el canal entre el compresor y el descompresor. (En contraposición, las tramas perdidas debido a daños en el contexto contribuyen con la tasa de pérdidas de paquetes).

### *Trama*

Paquete emitido por el compresor/recibido por el descompresor. Obsérvese que, en el presente documento, no existe ninguna relación con otros conceptos de trama (por ejemplo, capa física) tales como tramas de radiocomunicaciones. Marco conceptual de la compresión introducido en el presente documento. El concepto de perfil hace uso de identificadores de perfil para separar los diferentes perfiles que se usan cuando se establece el esquema de compresión.

Todas las variaciones y parámetros del esquema de compresión de encabezamientos que no son parte del estado de contexto son gestionados por identificadores de perfil diferentes.

5 (Nota del editor: el concepto de perfil no se ha finalizado todavía - este texto se basa en el estado actual del presente documento).

#### *Paquete*

10 En general, una unidad de transmisión y recepción (unidad de datos de protocolo). Específicamente, cuando se contrapone con “trama”, el paquete comprimido y a continuación descomprimido mediante ROHC. También denominado “paquete no comprimido”.

#### *Enlaces Pre-HC*

15 Los enlaces Pre-HC son todos los enlaces que ha atravesado un paquete antes del punto de compresión de encabezamientos. Si se considera un camino con enlaces celulares como primer y último saltos, los enlaces Pre-HC para el compresor en el último enlace son el primer enlace celular más los enlaces por cable situados en medio.

#### *Error residual*

20 Error introducido durante la transmisión y no detectado por esquemas de detección de errores de capas inferiores.

#### *Robustez*

25 El rendimiento de un esquema de compresión de encabezamientos se puede describir con tres parámetros, eficacia de la compresión, robustez y transparencia de la compresión. Un esquema robusto tolera errores sobre el enlace a través del cual tiene lugar la compresión de encabezamientos (incluyendo tanto pérdidas de tramas como errores de bits residuales) sin perder paquetes adicionales, introducir errores adicionales, o usar más ancho de banda.

#### *RTT*

Tiempo de ida y vuelta - El tiempo que se tarde en enviar un paquete desde el compresor al descompresor y de vuelta nuevamente desde el descompresor al compresor.

#### *Eficiencia espectral*

35 Los recursos de radiocomunicaciones son limitados y caros. Por esta razón, se deben usar de forma eficaz para conseguir que el sistema sea económicamente viable. En los sistemas celulares, esta situación se alcanza maximizando el número de usuarios a los que se presta servicio dentro de cada célula, mientras que la calidad de los servicios proporcionados se mantiene a un nivel aceptable. Una de las consecuencias de un uso eficaz del espectro es una elevada tasa de errores (pérdida de tramas y errores de bits residuales), incluso después de la codificación de los canales con corrección de errores.

#### *Paso de la indicación de tiempo*

45 El paso de la indicación de tiempo (PASO TS) es el aumento del valor de la indicación de tiempo entre dos paquetes consecutivos.

### *3. Antecedentes*

50 Este capítulo proporciona unos antecedentes para el tema de la compresión de encabezamientos. Las ideas fundamentales se describen junto con descripciones de esquemas existentes de compresión de encabezamientos, sus inconvenientes y requisitos y la motivación para nuevas soluciones de compresión de encabezamientos.

#### *3.1. Principios básicos de la compresión de encabezamientos*

60 La razón principal por la que tan siquiera puede realizarse la compresión de encabezamientos es el hecho de que existe una redundancia significativa entre los campos de los encabezamientos, tanto dentro del encabezamiento del mismo paquete como, en particular, entre paquetes consecutivos pertenecientes al mismo flujo continuo de paquetes. Mediante el envío de la información de los campos estáticos sólo inicialmente y la utilización de dependencias y predictibilidad para otros campos, el tamaño del encabezamiento se puede reducir significativamente para la mayor parte de paquetes.

65 En general, los métodos de compresión de encabezamientos mantienen un contexto, el cual es esencialmente la versión no comprimida del último encabezamiento enviado a través del enlace, más cierta información adicional, tanto en el compresor como en el descompresor. La compresión y descompresión se realizan en relación con el contexto. Cuando los encabezamientos comprimidos acarrearán diferencias con respecto al encabezamiento anterior, cada encabezamiento comprimido actualizará el contexto del descompresor. En este caso, cuando se pierde un paquete entre

el compresor y el descompresor, el contexto del descompresor quedará desincronizado ya que no se ha actualizado correctamente. Un método de compresión de encabezamientos debe disponer de una forma de reparar el contexto, es decir, sincronizarlo después de dichos acontecimientos.

### 5 3.2. Esquemas existentes de compresión de encabezamientos

El esquema original de compresión de encabezamientos, CTCP [VJHC], fue inventado por Van Jacobson. El CTCP comprime el encabezamiento IP+TCP de 40 octetos a 4 octetos. El compresor CTCP detecta retransmisiones a nivel del transporte y envía un encabezamiento que actualiza el contexto completamente cuando las mismas se producen.  
10 Este mecanismo de reparación no requiere ninguna señalización explícita entre el compresor y el descompresor.

Un esquema general de compresión de encabezamientos IP, la compresión de encabezamientos IP [IPHC], ofrece ciertas mejoras con respecto al CTCP y puede comprimir encabezamientos IP, TCP, y UDP arbitrarios. Cuando se comprimen encabezamientos que no son TCP, el IPHC no usa la codificación delta y es robusto. Cuando se comprime el TCP, el mecanismo de reparación del CTCP se complementa con un esquema de acuses de recibo negativos a nivel del enlace el cual acelera la reparación. El IPHC no comprime encabezamientos RTP.

El CRTP [CRTP, IPHC] de Casner y Jacobson es un esquema de compresión de encabezamientos que comprime 40 octetos de encabezamientos IPv4/UDP/RTP a un mínimo de 2 octetos cuando no hay presente ninguna suma de comprobación UDP. Si está presente la suma de comprobación UDP, el encabezamiento CRTP mínimo es 4 octetos. El CRTP no puede usar el mismo mecanismo de reparación que el CTCP ya que el UDP/RTP no retransmite. En cambio, el CRTP usa mensajes de señalización explícita desde el descompresor al compresor, denominados mensajes ESTADO\_CONTEXTO, para indicar que el contexto está desincronizado. De este modo, el tiempo de ida y vuelta del enlace limitará la velocidad de este mecanismo de reparación de contextos.

En enlaces disipativos con tiempos de ida y vuelta largos, tales como la mayoría de enlaces celulares, el CRTP no presenta un buen comportamiento. Cada paquete perdido a través del enlace provoca la pérdida de varios paquetes subsiguientes ya que el contexto está desincronizado durante por lo menos un periodo de tiempo de ida y vuelta del enlace. Este comportamiento está documentado en el [CRTPC]. Para conversaciones de voz, dichos acontecimientos prolongados de pérdidas deteriorarán la calidad de la voz. Por otra parte, el ancho de banda es malgastado por los grandes encabezamientos enviados por el CRTP cuando se actualiza el contexto. El [CRTPC] observó que el CRTP no presentaba un comportamiento suficientemente bueno para un enlace celular disipativo. Es evidente que solamente el CRTP no es un esquema viable de compresión de encabezamientos para la telefonía IP sobre enlaces celulares.

Para evitar la pérdida de paquetes debido a la desincronización del contexto, los descompresores CRTP pueden intentar reparar el contexto localmente usando un mecanismo conocido como TWICE. Cada paquete CRTP contiene un contador el cual se incrementa en uno por cada paquete enviado hacia fuera por el compresor CRTP. Si el contador se incrementa en más de uno, se perdió por lo menos un paquete a través del enlace. A continuación, el descompresor intenta reparar el contexto suponiendo cómo lo habría actualizado el(los) paquete(s) perdido(s). A continuación, la suposición se verifica descomprimiendo el paquete y comprobando la suma de comprobación UDP - si la misma tiene éxito, la reparación se considera satisfactoria y el paquete se puede reenviar o entregar. El TWICE obtiene su nombre a partir de la observación de que cuando el flujo continuo de paquetes comprimidos es regular, la suposición correcta consiste en aplicar la actualización en el paquete actual dos veces. El [CRTPC] observó que incluso son el TWICE, el CRTP doblaba el número de paquetes perdidos. El TWICE mejora el rendimiento del CRTP significativamente. No obstante, existen varios problemas relacionados con el uso del TWICE:

#### 1) Se hace obligatorio el uso de la suma de comprobación UDP:

- el tamaño mínimo del encabezamiento comprimido se incrementa en un 100% a 4 octetos.
- la mayoría de códecs de voz desarrollados para enlaces celulares toleran errores en los datos codificados. Dichos códecs no desearán habilitar la suma de comprobación UDP, ya que los mismos no desean entregar paquetes dañados.
- los errores en la carga útil harán que la suma de comprobación UDP no resulte satisfactoria cuando la suposición sea correcta (y podrían hacer que la misma resultase satisfactoria cuando fuera errónea).

2) La pérdida en un flujo continuo RTP que se produce antes del punto de compresión hará que las actualizaciones en los encabezamientos CRTP sean menos regulares. En ese caso, las versiones sencillas del TWICE presentarán un mal comportamiento. Las versiones más sofisticadas necesitarían más intentos de reparación para resultar satisfactorias.

### 3.3. Requisitos de un esquema nuevo de compresión de encabezamientos

El problema principal con el CRTP es que no es suficientemente robusto contra paquetes que han resultado dañados entre el compresor y el descompresor. Un esquema viable de compresión de encabezamientos debe ser menos frágil. Este aumento de la robustez se debe obtener sin incrementar el tamaño del encabezamiento comprimido; un encabezamiento mayor haría que la telefonía IP sobre enlaces celulares resultase poco atractiva económicamente.

Una de las causas principales del mal comportamiento del CRTP sobre enlaces celulares es el tiempo de ida y vuelta prolongado del enlace, durante el cual se pierden muchos paquetes cuando el contexto queda desincronizado. Este problema se puede atacar directamente encontrando formas de reducir el tiempo de ida y vuelta del enlace. De hecho, las futuras generaciones de tecnologías celulares puede que alcancen tiempos de ida y vuelta menores del enlace. No obstante, los mismos serán siempre probablemente bastante elevados [CELL]. Las ventajas en términos de unas exigencias de una pérdida menor y un ancho de banda más pequeño si el contexto se puede reparar localmente estarán presentes incluso si se reduce el tiempo de ida y vuelta del enlace. A continuación es necesaria una forma fiable de detectar una reparación satisfactoria del contexto.

Se podría argumentar que una forma mejor de resolver el problema consiste en mejorar el enlace celular de manera que haya menos probabilidad de que se produzca la pérdida de paquetes. No obstante, dichas modificaciones no se obtienen a cambio de nada. Si los enlaces se realizaran (casi) exentos de errores, el sistema podría no ser capaz de soportar un número suficientemente grande de usuarios por célula y de este modo podría resultar económicamente inviable [CELL].

También se podría argumentar que los códecs de voz deberían poder gestionar el tipo de pérdida de paquetes inducido por el CRTP, en particular debido a que probablemente los códecs de voz deben poder gestionar de todos modos las pérdidas de paquetes si el flujo continuo de RTP cruza por Internet. Aunque esta última opción es cierta, la gestión del tipo de pérdida inducido por el CRTP es dificultosa. Habitualmente no es posible ocultar completamente un acontecimiento de pérdida en el que se pierde completamente sonido de una duración bien por encima de 100 ms. Si dicha pérdida se produce frecuentemente en ambos extremos del camino de extremo-a-extremo, la calidad de la voz sufrirá las consecuencias.

En [REQ] se puede encontrar una descripción detallada de los requisitos especificados para la ROHC.

### 3.4. Clasificación de los campos de los encabezamientos

Tal como se ha mencionado anteriormente, la compresión de encabezamientos es posible debido al hecho de que existe una gran redundancia entre los valores de los campos de los encabezamientos dentro de los paquetes, aunque especialmente entre paquetes consecutivos. Para utilizar estas propiedades para la compresión de encabezamientos, es importante entender los patrones de cambio de los diversos campos de los encabezamientos.

Todos los campos de los encabezamientos se han clasificado de forma detallada en el apéndice A. Los campos se clasifican en primer lugar en un nivel alto y a continuación algunos de ellos se estudian más detalladamente. Finalmente, el apéndice concluye con recomendaciones sobre cómo deberían ser tratados los diversos campos por los algoritmos de compresión de encabezamientos. La principal conclusión que se puede extraer es que la mayoría de los campos de encabezamientos se pueden comprimir fácilmente ya que los mismos nunca o rara vez cambian. Únicamente 5 campos, con un tamaño combinado de aproximadamente 10 octetos, necesitan mecanismos más sofisticados. Dichos campos son:

- Identificación IPv4 (16 bits) - ID-IP
- Suma de Comprobación UDP (16 bits)
- Marcador RTP (1 bit) - bit M
- Número de Secuencia RTP (16 bits) - SN
- Indicación de Tiempo RTP (32 bits) - TS

El análisis del Apéndice A revela que los valores de los campos TS e ID-IP se pueden predecir habitualmente a partir del Número de Secuencia RTP, el cual se incrementa en uno por cada paquete emitido por una fuente RTP. El bit M también es habitualmente el mismo, aunque necesita ser comunicado de forma explícita ocasionalmente. La suma de comprobación UDP no debería predecirse y se envía tal como está cuando se habilita.

En este caso, la forma en la que funciona la compresión RTP ROHC, consiste en primer lugar en establecer funciones desde el SN a los otros campos, y a continuación comunicar de forma fiable el SN. Siempre que cambie una función del SN a otro campo, es decir, la función, existente proporciones un resultado el cual sea diferente con respecto al campo del encabezamiento que se debe comprimir, se envía información adicional para actualizar los parámetros de dicha función.

## 4. Marco conceptual de la compresión de encabezamientos

### 4.1. Suposiciones de funcionamiento

Los enlaces celulares, los cuales constituyen un objetivo principal para la ROHC, presentan una serie de características que se describen brevemente en el presente documento. La ROHC requiere funcionalidad de capas más bajas

## ES 2 266 273 T3

lo cual se expresa en líneas generales en el presente documento y se describe más minuciosamente en el documento de los requisitos de las capas inferiores [LLG].

### Canales

5 Los paquetes con encabezamientos comprimidos por ROHC fluyen sobre canales. A diferencia de la mayoría de enlaces fijos, algunos enlaces celulares de radiocomunicaciones pueden tener varios canales que conecten el mismo par de nodos. Cada canal puede tener diferentes características en términos de tasa de errores, ancho de banda, etcétera.

### 10 Identificadores de contexto

En algunos canales, se requiere la capacidad de transportar múltiple flujos continuos de paquetes. También puede resultar factible disponer de canales dedicados a flujos continuos individuales de paquetes. De este modo, la ROHC usa un espacio identificador de contexto distinto por canal y elimina los identificadores de contexto completamente en los canales en los que se comprime solamente un único flujo continuo de paquetes.

### Indicación de tipo de paquete

20 La indicación de tipo de paquete se realiza en el propio esquema de compresión de encabezamientos. A no ser que el enlace disponga de una forma de indicar tipos de paquetes que pueden ser usados, tales como el PPP, esta opción proporciona encabezamientos comprimidos más pequeños en conjunto. También puede ser que resulte menos difícil asignar un único tipo de paquete, en lugar de muchos, para ejecutar la ROHC sobre enlaces tales el PPP.

### Reordenación

25 Se considera que el canal entre el compresor y el descompresor no reordena paquetes, es decir, el descompresor recibe paquetes en el mismo orden que los envía el compresor, no obstante, se aborda la reordenación antes del punto de compresión.

### 30 Longitud de los paquetes

La ROHC está diseñada bajo la suposición de que las capas inferiores pueden proporcionar la longitud de un paquete comprimido. Los paquetes ROHC no contienen información de longitud para la carga útil.

### 35 Alineación de las tramas

La capa del enlace debe proporcionar alineación de las tramas, lo cual hace que resulte posible distinguir límites de tramas y tramas individuales.

### 40 Detección de/protección contra errores

45 El esquema ROHC se ha diseñado para hacer frente a los errores residuales en los encabezamientos entregados al descompresor. Se usan comprobaciones CRC y comprobaciones de cordura (*sanity checks*) para evitar o reducir la propagación de daños. No obstante, se RECOMIENDA que las capas inferiores desplieguen una detección de errores para los encabezamientos ROHC que no entreguen encabezamientos ROHC con una tasa elevada de errores residuales.

50 Sin proporcionar un límite estricto sobre la tasa de errores residuales aceptables para la ROHC, se hace observar que para una tasa de errores de bits residuales de como mucho  $1E-5$ , el esquema ROHC se ha diseñado para no incrementar el número de encabezamientos dañados, es decir, el número de encabezamientos dañados debido a la propagación de daños está diseñado para que sea inferior al número de encabezamientos dañados capturado por el esquema de detección de errores ROHC.

### Negociación

55 Además de los anteriores mecanismos de tratamiento de paquetes, la capa de enlace DEBE proporcionar una forma de negociar los parámetros de compresión de encabezamientos. (Para enlaces unidireccionales, esta negociación se puede realizar fuera de banda o incluso *a priori*).

### 60 4.2. Dinamicidad

65 [[Esta sección contendrá texto introductorio para la información de estados dinámicos mantenida en las entidades de protocolo que ejecutan el protocolo ROHC, por ejemplo, qué parámetros se establecen en la negociación por canal, cuáles constituyen el flujo continuo (es decir, el cambio de los mismos crea un flujo continuo diferente), y cuáles forman parte del contexto por flujo continuo y por lo tanto se pueden cambiar durante la vida útil del flujo continuo]].

## ES 2 266 273 T3

Establecimiento del canal.

(configurado o negociado en el momento del establecimiento del canal o antes de dicho momento):

5 Conjunto de perfiles aceptables.

MAX-HEADER.  
Tamaño del espacio CID.

10 Establecimiento del flujo continuo:

(negociado en el momento del establecimiento del flujo continuo).  
Perfil. El cambio del perfil implica un flujo continuo nuevo.

15 Contexto por flujo continuo:

(se puede cambiar durante la vida útil de un flujo continuo)

Contexto dentro:

20 Valor actual de todos los campos del encabezamiento (a partir de esto se puede deducir  
encabezamiento IPv4 presente,  
suma de comprobación UDP habilitada)

25 Contexto adicional que no está en el encabezamiento:

PASO TS  
¿ID-IP en el orden de los bytes de la red?  
30 ¿ID-IP es aleatoria?  
Un número de encabezamientos de referencia antiguos.

Estado del compresor y del descompresor.

35 4.3. *Estados de compresión y descompresión*

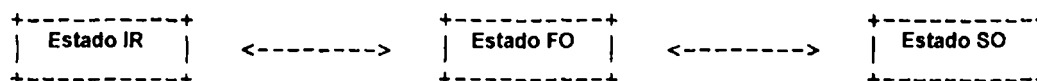
La compresión de encabezamientos con la ROHC se puede caracterizar como una interacción entre dos máquinas de estados, una máquina del compresor y una máquina del descompresor. El compresor y el descompresor presentan tres estados cada uno de ellos, los cuales están relacionados de muchas maneras entre ellos incluso si el significado de los estados es ligeramente diferente para las dos partes. Ambas máquinas comienzan en el estado de compresión más bajo y realizan transiciones gradualmente a estados superiores. No es necesario que las transiciones estén sincronizadas entre las dos máquinas y normalmente el compresor es el único que temporalmente puede realizar transiciones de vuelta a estados inferiores.

45 Las secciones subsiguientes presentan una visión general de las máquinas de estado y sus estados correspondientes respectivamente, comenzando con el compresor.

### 4.3.1. *Estados del compresor*

50 Para la compresión ROHC, los tres estados del compresor son los estados de Inicio y Restauración (IR), de Primer Orden (FO), y de Segundo Orden (SO). El compresor comienza en el estado de compresión más bajo (IR) y realiza una transición gradualmente a estados de compresión superiores. El compresor siempre funcionará en el estado de compresión más alto posible, bajo la limitación de que el compresor esté suficientemente seguro de que el descompresor descubre la información necesaria para descomprimir un encabezamiento comprimido según dicho estado.

55



60

El compresor toma decisiones sobre las transiciones entre los diversos estados de compresión basándose en:

- 65 - variaciones de los encabezamientos de los paquetes.
- realimentación positiva desde el descompresor (Acuses de Recibo - Acuses ACK)

## ES 2 266 273 T3

- realimentación negativa desde el descompresor (ACK Negativos - Acuses NACK)
- temporizaciones periódicas (cuando no está presente la realimentación)

5 En el capítulo 5 se explica detalladamente cómo se realizan las transiciones para cada modo de funcionamiento.

### 4.3.1.1. Estado de Inicio y Restauración (IR)

10 La finalidad del estado IR es inicializar las partes estáticas del contexto en el descompresor o recuperar después de un fallo. En este estado, el compresor envía la información completa del encabezamiento. La misma incluye todos los campos estáticos y no estáticos en un formato no comprimido más alguna información adicional. Las restauraciones también se pueden realizar únicamente sobre información no estática.

15 El compresor debería permanecer en el estado IR hasta que el mismo esté bastante seguro de que el descompresor ha recibido la información correctamente.

### 4.3.1.2. Estado de Primer Orden (FO)

20 La finalidad del estado FO es comunicar eficazmente irregularidades en el flujo continuo de paquetes. Cuando está funcionando en este estado, el compresor rara vez envía información completa y la información enviada está habitualmente comprimida por lo menos de forma parcial. Por esta razón, la diferencia entre IR y FO debería quedar clara.

25 El compresor entra en este estado cuando los encabezamientos del flujo continuo de paquetes no se ajustan a su patrón anterior, y permanecen en el mismo hasta que está seguro de que el descompresor ha adquirido todos los parámetros del patrón nuevo. Los campos que son siempre irregulares no requieren el FO ya que deben ser comunicados en todos los paquetes y por lo tanto forman parte de lo que se denomina patrón uniforme.

30 Como los paquetes enviados en el estado FO habitualmente transportan información de actualización de contextos, la transmisión satisfactoria de la información FO puede ser de vital importancia para realizar una descompresión satisfactoria de paquetes subsiguientes. El proceso de descompresión es sensible a la pérdida de, o a los daños de, los paquetes FO que están en tránsito.

### 4.3.1.3. Estado de Segundo Orden (SO)

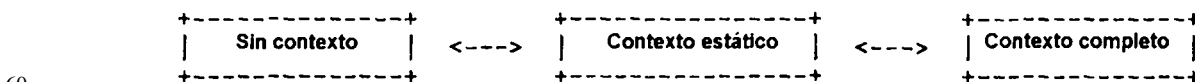
35 Este es el estado en el que la compresión es óptima. El compresor entra en el estado SO cuando el encabezamiento que se debe comprimir es completamente predecible dado el SN, y el compresor está suficientemente seguro de que el descompresor ha adquirido todos los parámetros de las funciones desde el SN a otros campos. Los paquetes enviados en el estado SO son casi mutuamente independientes y por lo tanto la sensibilidad a los errores es baja. No obstante, una descompresión satisfactoria de paquetes enviados en el estado SO requiere que la información enviada en las operaciones anteriores del estado FO haya sido recibida satisfactoriamente por el descompresor.

45 Habitualmente, la cantidad de información enviada en el estado SO es fija. No obstante, puede variar a largo plazo ya que el descompresor puede decidir incrementar la cantidad de información a enviar en el estado SO usando un formato de encabezamiento mayor, diferente. No obstante, el requisito básico es que los paquetes enviados en el estado SO deben ser casi mutuamente independientes para la descompresión.

50 El compresor abandona este estado y vuelve al estado FO cuando el encabezamiento ya no se ajusta al patrón uniforme y no se puede comprimir de forma independiente basándose en la información de contexto anterior.

### 4.3.2. Estados del descompresor

55 El descompresor comienza en su estado de compresión más bajo, "Sin Contexto" y realiza una transición gradualmente a estados superiores. Normalmente la máquina de estados del descompresor nunca abandona el estado de "Contexto Completo" una vez que ya ha comenzado a funcionar en dicho estado.



65 Cuando está funcionando inicialmente en el estado "Sin Contexto", el descompresor no ha descomprimido nunca satisfactoriamente nunca ningún paquete. Cuando se ha descomprimido correctamente una vez un paquete (al recibir un paquete de inicialización con formación estática y dinámica, por ejemplo), el descompresor puede realizar una transición completa al estado de "Contexto Completo", y solamente si se producen fallos repetidos realizará una transición de vuelta a estados inferiores. No obstante, cuando se produce dicha situación, en prime lugar realiza una transición de vuelta al estado de "Contexto Estático". En dicho estado, la recepción de un paquete FO es normalmente

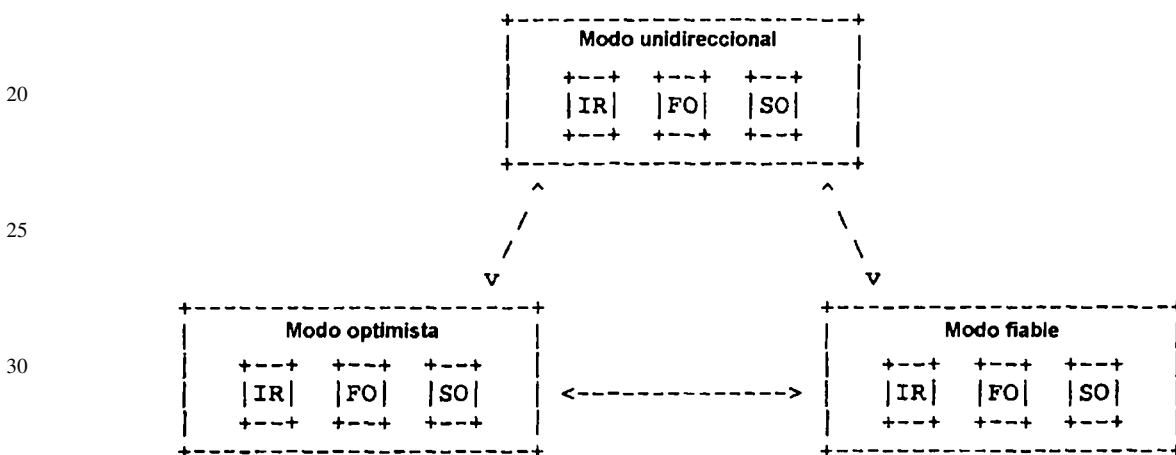
suficiente para permitir la transición nuevamente al estado de “Contexto Completo”. Únicamente cuando falle la descompresión de varios paquetes FO en el estado de “Contexto Estático”, el descompresor volverá directamente al estado de “Sin Contexto”.

5 En el capítulo 5 se explica detalladamente cuándo se realizan transiciones de estado.

4.4. *Modos de funcionamiento*

10 El esquema ROHC tiene tres modos de funcionamiento, denominados modo Unidireccional, Optimista Bidireccional, y Fiable Bidireccional.

15 Es importante entender la diferencia entre los estados, tal como se ha descrito en el capítulo anterior, y los modos, ya que los mismos son abstracciones ortogonales entre sí. La abstracción de los estados es la misma para todos los modos de funcionamiento, aunque cada modo tiene sus formas específicas de funcionar en los diversos estados y de tomar decisiones sobre las transiciones entre estados.



35 El modo en el de funcionar depende del entorno en el que se realice la compresión considerando parámetros de los enlaces tales como capacidades de realimentación, probabilidades y distribuciones de errores, efectos de la variación del tamaño de los encabezamientos. Todas las implementaciones ROHC DEBEN implementar y soportar la totalidad de los tres modos de funcionamiento. Los tres modos se describen brevemente en las siguientes subsecciones.

40 En el capítulo 5 se proporcionan descripciones detalladas de los tres modos de funcionamiento en relación con la lógica de compresión y descompresión. En el capítulo 5 se describen también los mecanismos de transición de modos.

4.4.1. *Modo Unidireccional - modo U*

45 Cuando se encuentra en el modo unidireccional de funcionamiento, los paquetes se envían solamente en una dirección: desde el compresor al descompresor. Por esta razón, este modo consigue que la ROHC sea utilizable sobre enlaces en los que no esté disponible o no sea deseable un camino de retorno desde el descompresor al compresor.

50 En el modo U, las transiciones entre los estados del compresor se realizan basándose solamente en temporizaciones periódicas e irregularidades de los patrones de cambio de los campos de los encabezamientos en el flujo continuo de paquetes comprimidos. Debido a las restauraciones periódicas y a la falta de realimentación para el inicio de la recuperación de errores, la compresión en el modo unidireccional será menos eficaz y presentará una probabilidad ligeramente mayor de propagación de pérdidas en comparación con cualquiera de los modos bidireccionales.

55 La compresión con la ROHC DEBE comenzar en el modo unidireccional. La transición a cualquiera de los modos bidireccionales se puede realizar en cuanto un paquete haya alcanzado al descompresor y el mismo haya respondido con un paquete de realimentación indicando que se desea una transición de modo (ver capítulo 5).

60 4.4.2. *Modo optimista bidireccional - modo O*

65 El modo optimista bidireccional es similar al modo unidireccional. La diferencia es que se usa un canal de realimentación para enviar solicitudes de recuperación de errores y (opcionalmente) acuses de recibo de actualizaciones de contexto significativas desde el descompresor al compresor (no solamente para actualizaciones de números de secuencia). En el modo optimista bidireccional no se usan restauraciones periódicas.

El modo O está diseñado para una eficacia de compresión óptima y un uso escaso del canal de retorno aunque manteniendo una robustez razonable. La pérdida de sincronización compresor-descompresor y la introducción de

propagación de pérdidas son poco frecuentes incluso con tasas de errores elevadas. Cuando se produce la propagación de pérdidas, la magnitud es bastante insignificante. Sin embargo, este modo no es completamente robusto y con tasas de errores extremadamente elevadas puede producirse la propagación de pérdidas.

5 4.4.3. *Modo fiable bidireccional - modo R*

El modo fiable bidireccional es diferente en muchos aspectos con respecto a los dos anteriores. Las diferencias más importantes en relación con la implementación y la funcionalidad son un uso más intensivo del canal de realimentación y una lógica más estricta tanto en el compresor como en el descompresor. La realimentación se envía para acusar el recibo de todas las actualizaciones de contexto, incluyendo actualizaciones del campo número de secuencia. No obstante, no todos los paquetes actualizan el contexto en un modo fiable.

La ventaja principal del modo R es una robustez casi total contra la pérdida de paquetes entre el compresor y el descompresor. La propagación de pérdidas no se puede producir nunca debido a la compresión de encabezamientos cuando se funciona en este modo. El precio es una tara ligeramente mayor en algunos casos y el tráfico de realimentación adicional introducido.

4.5. *Métodos de codificación*

20 Este capítulo describe los métodos de codificación que se usan para diferentes campos de los encabezamientos. En el capítulo de formato de los paquetes se especifica cómo se aplican los métodos a cada campo (por ejemplo, valores de parámetros asociados).

25 4.5.1. *Codificación de Bits Menos Significativos (LSB)*

La codificación de Bits Menos Significativos (LSB) se usa para campos de encabezamientos cuyos valores están sujetos habitualmente a cambios pequeños. Con la codificación LSB, en lugar del valor de campo original se transmiten los k bits menos significativos del valor del campo, en los que k es un entero positivo. Después de recibir los k LSB, el descompresor obtiene el valor original usando como referencia un valor recibido anteriormente (v ref).

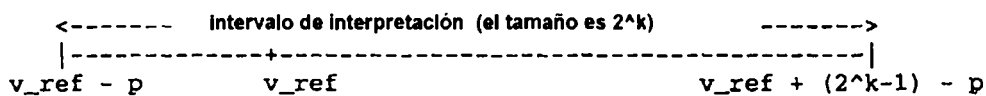
Se garantiza que el esquema es correcto si el compresor y el descompresor llegan a un acuerdo sobre un intervalo de interpretación

- 1) en el cual reside el valor original, y
- 2) en el cual el valor original es el único valor que presenta exactamente los mismos k LSB que los transmitidos.

El intervalo de interpretación se puede describir como una función f (v\_ref, k). Siendo

$$f(v\_ref, k) = [v\_ref - p, v\_ref + (2^k - 1) - p]$$

en la que p es un entero.



La función f presenta la siguiente propiedad: para cualquier valor k, k bits LSB identificarán de forma exclusiva un valor en f (v ref, k). Esto cumple la anterior condición 2).

55 El parámetro p se introduce de manera que el intervalo de interpretación se pueda desplazar con respecto a v\_ref. La selección de un valor adecuado para p producirá una codificación más eficaz para campos con ciertas características. Por ejemplo, si un valor de campo observado por el compresor aumenta siempre, p se debería fijar a 0, y el intervalo se convierte en [v\_ref, v\_ref + 2^k - 1]. Como la codificación LSB en la ROHC se aplicará a campos de encabezamientos cuyos valores aumentan excepto en situaciones inhabituales (tales como un desorden de los paquetes o la TS RTP para el vídeo), a p se le asignarán dos valores: p1 = 0 ó p2 = 2^(k-2) - 1. Esta última proporciona el intervalo de interpretación [v\_ref - (2^(k-2) - 1), v\_ref + 3\*2^(k-2)], el cual puede tratar cambios negativos pequeños aunque usando 3/4 del intervalo para cambios positivos. Consultar la sección 5.7 para obtener más detalles.

65 El siguiente es un procedimiento simplificado para la compresión y descompresión LSB, se modifica en aras de la robustez y protección contra la propagación de daños en la siguiente subsección:

- 1) El compresor (descompresor) usa siempre v\_ref\_c (v ref d), el último valor que ha sido comprimido (descomprimido), como v ref;

## ES 2 266 273 T3

2) Cuando se comprime un valor  $v$ , el compresor calcula el valor mínimo (en aras de la eficacia) aunque suficiente (en aras de la exactitud) de  $k$  tal que  $v$  quede incluido en el intervalo  $f(v_{\text{ref\_c}}, k)$ . Representemos este procedimiento como una función  $k = g(v, v_{\text{ref}})$ . Obsérvese que se pueden enviar más de  $k$  bits LSB para adaptarse al formato del paquete (ver la sección de formato de los paquetes);

3) Cuando se reciben  $m$  bits LSB, el descompresor usa el intervalo de interpretación  $f(v_{\text{ref\_d}}, m)$ , denominado intervalo  $d$ . Escoge como valor descomprimido el valor del intervalo  $d$  cuyos bits LSB coinciden con los  $m$  bits recibidos.

El esquema se ve complicado por dos factores: pérdida de paquetes entre el compresor y del descompresor, y errores de transmisión no detectados por la capa inferior. En el primer caso, el compresor y el descompresor perderán la sincronización de  $v_{\text{ref}}$ , y por lo tanto también del intervalo de interpretación. Si  $v$  queda todavía cubierto por la intersección (intervalo  $c$ , intervalo  $d$ ), la descompresión será correcta. En cualquier otro caso, se producirá una descompresión incorrecta. La siguiente sección tratará adicionalmente esta cuestión.

En el caso de errores de transmisión no detectados, los LSB alterados proporcionarán un valor descomprimido de forma incorrecta que posteriormente se usará como  $v_{\text{ref\_d}}$ , el cual a su vez es probable que desemboque en propagación de daños. Este problema se afronta usando una referencia segura, es decir, un valor de referencia cuya exactitud se verifica por medio de una CRC de protección. Consecuentemente, el anterior procedimiento 1) se modifica de la forma siguiente:

1) a) el compresor usa siempre como  $v_{\text{ref\_c}}$  el último valor que ha sido comprimido y enviado con una CRC de protección.

b) el descompresor usa siempre como  $v_{\text{ref\_d}}$  el último valor correcto, verificado por una CRC satisfactoria.

Obsérvese que en el modo U/0, el punto 1) b) se modifica de manera que si la descompresión no consigue usar el último valor correcto, se realiza otro intento de descompresión usando el último valor que sea correcto. Este procedimiento mitiga la propagación de daños cuando una CRC pequeña no consigue detectar un valor dañado.

### 4.5.2. Codificación LSB basada en ventanas (codificación W-LSB)

Esta sección describe cómo modificar el algoritmo simplificado en 4.5.1 para alcanzar la robustez.

Puede que el compresor no sea capaz de determinar el valor exacto de  $v_{\text{ref\_d}}$  que será usado por el descompresor para un valor específico  $v$ , ya que algunos candidatos para  $v_{\text{ref\_d}}$  pueden haberse perdido o dañado. No obstante, usando la realimentación o realizando consideraciones razonables el compresor puede limitar el conjunto de candidatos. A continuación, el compresor calcula  $k$  de tal manera que con independencia de qué  $v_{\text{ref\_d}}$  del conjunto de candidatos use el descompresor,  $v$  quede cubierto por el intervalo  $d$  resultante.

Como el descompresor usa siempre el último valor recibido en el que la CRC resultó satisfactoria, como referencia, el compresor mantiene una ventana deslizante (VSW) que contiene los candidatos para  $v_{\text{ref\_d}}$ . VSW está inicialmente vacía. El compresor realiza las siguientes operaciones sobre VSW:

1) Después de enviar un valor  $v$  (comprimido o no comprimido) protegido por una CRC, el compresor añade  $v$  a la VSW;

2) Para cada valor  $v$  que esté siendo comprimido, el compresor selecciona  $k = \max(g(v, v_{\text{min}}), g(v, v_{\text{max}}))$ , que  $v_{\text{min}}$  y  $v_{\text{max}}$  son el valor mínimo y máximo en VSW, y  $g$  es la función definida en la sección anterior;

3) Cuando el compresor está suficientemente seguro de que un cierto valor  $v$  no será usado como referencia por el descompresor, la ventana se hace avanzar eliminando  $v$  y todos los valores más antiguos que  $v$ . La seguridad se puede obtener por varios medios. En el modo R una ACK del descompresor implica que de la VSW se pueden eliminar valores más antiguos que el correspondiente del que se ha acusado el recibo. En el modo U/O existe siempre una CRC para verificar la descompresión correcta, y se usa una VSW con una anchura máxima limitada. La anchura de la ventana es un parámetro de optimización determinado durante la implementación.

Obsérvese que el descompresor sigue el procedimiento descrito en la sección anterior, excepto que en el modo R DEBE ACK cada valor recibido con una CRC (consúltese la sección de lógica de compresión/descompresión para obtener detalles).

### 4.5.3 Codificación escalonada de las Indicaciones de Tiempo RTP

La Indicación de Tiempo RTP (TS) no se incrementará habitualmente en un número arbitrario de un paquete a otro. En cambio, el incremento es normalmente un múltiplo entero de cierta unidad (PASO\_TS). Por ejemplo, en el caso del audio, la frecuencia de muestreo es normalmente 8KHz y una trama de voz puede abarcar 20 ms. Además,

## ES 2 266 273 T3

cada trama de voz se transporta habitualmente en un paquete RTP. En este caso, el incremento RTP es siempre  $n * 160$  ( $= 8.000 * 0,02$ ), para cierto entero  $n$ . Obsérvese que los periodos de silencio no tienen ninguna influencia sobre esta situación ya que el reloj de muestreo en la fuente normalmente se mantiene funcionando sin cambiar ni la frecuencia de las tramas ni los límites de las tramas.

Para el caso del vídeo, también existe habitualmente un PASO\_TS cuando se considera el nivel de los cuadros de vídeo. La frecuencia de muestreo para la mayoría de códecs de vídeo es 90Khz. Si la frecuencia de cuadro del vídeo es fija, digamos 30 cuadros/segundo, la TS se incrementará en  $n * 3000$  ( $= n * 90.000 / 30$ ) entre los cuadros de vídeo. Obsérvese que un cuadro de vídeo se divide normalmente en varios paquetes RTP para alcanzar la robustez contra la pérdida de paquetes. En este caso, varios paquetes RTP transportarán la misma TS.

Cuando se usa la codificación escalonada de indicaciones de tiempo RTP, la TS se reduce a escala en un factor de PASO\_TS antes de la compresión. Esto ahorra  $\text{floor}(\log_2(\text{PASO\_TS}))$  bits para cada TS comprimida. Entre la TS y la TS ESCALADA se cumple la siguiente igualdad:

$$TS = TS \text{ ESCALADA} * \text{PASO\_TS} + \text{DESVIACIÓN\_TS}$$

Al descompresor se le comunican PASO\_TS explícitamente, y la DESVIACIÓN\_TS implícitamente. Se usa el siguiente algoritmo:

1. Inicialización: El compresor envía al descompresor el valor PASO\_TS (por ejemplo, a través de una señalización dentro de la banda, ver sección de formato de paquetes) y el valor absoluto una o varias TS. Estas últimas son usadas por el descompresor para inicializar DESVIACIÓN\_TS a (valor absoluto) módulo PASO\_TS. Obsérvese que DESVIACIÓN\_TS es la misma con independencia de qué valor absoluto se use, siempre que el valor de TS no escalado no realice un reinicio cíclico, ver el punto 4) más abajo.
2. Compresión: Después de la inicialización, el compresor ya no comprime los valores de TS originales. En su lugar, comprime los valores reducidos a escala:  $TS\_ESCALADA = TS / \text{PASO\_TS}$ . El método de compresión podría ser bien la codificación W-LSB o bien la codificación basada en temporizadores que se describe en la siguiente sección.
3. Descompresión: Cuando se recibe el valor comprimido de TS\_ESCALADA, en primer lugar el descompresor obtiene el valor de la TS\_ESCALADA original. A continuación, se calcula la TS RTP original como  $TS = TS\_ESCALADA * \text{PASO\_TS} + \text{DESVIACIÓN\_TS}$ .
4. Reinicio cíclico: El reinicio cíclico de la TS de 32 bits no escalada invalidará el valor actual de DESVIACIÓN\_TS usado en la ecuación anterior. Por ejemplo, supóngase  $\text{PASO\_TS} = 160 = 0xA0$  y la TS actual =  $0xFFFFFFFF0$ . En ese caso DESVIACIÓN\_TS es  $0x50 = 80$ . A continuación si la siguiente TS RTP =  $0x00000130$  (es decir, el incremento es  $160 * 2 = 320$ ), la nueva DESVIACIÓN\_TS debería ser  $0x00000130$  módulo  $0xA0 = 0x90 = 144$ . No es necesario que el compresor reinicialice la DESVIACIÓN\_TS en el reinicio cíclico. En cambio, el descompresor DEBE detectar el reinicio cíclico de la TS no escalada (lo cual es trivial) y actualizar la DESVIACIÓN\_TS a

$$\text{DESVIACIÓN\_TS} = (\text{TS no escalada con reinicio cíclico}) \text{ módulo PASO\_TS}$$

Este método de escalado se puede aplicar a muchos códecs basados en tramas. No obstante, el valor de PASO\_TS podría cambiar durante una sesión, por ejemplo debido a estrategias de adaptación. Si se produce esa situación, la TS no escalada se comprime hasta que se ha completado la reinicialización de las nuevas PASO\_TS y DESVIACIÓN\_TS.

### 4.5.4 Compresión de la Indicación de Tiempo RTP, Basada en Temporizadores

La indicación de tiempo RTP [RFC 1889] se define para identificar el número de la primera muestra usada para generar la carga útil. Cuando los paquetes RTP transportan cargas útiles correspondientes a un intervalo de muestreo fijado, el muestreo se realiza a una frecuencia constante, y los paquetes se generan al unísono con el muestreo, la indicación de tiempo seguirá de forma ajustada a un patrón lineal en función de la hora del día. Este es el caso correspondiente a la voz interactiva. La relación lineal se determina por medio de la frecuencia de muestreo de la fuente. El patrón lineal puede verse complicado por la paquetización (por ejemplo, en el caso del vídeo en el que un cuadro de vídeo se corresponde habitualmente con varios paquetes RTP) o la reorganización de las tramas (por ejemplo, los cuadros B MPEG son enviados desordenados por algunos códecs de vídeo).

Con una frecuencia de muestreo fijada de 8 KHz, 20 ms en el dominio del tiempo es equivalente a un incremento de 160 en el dominio de la TS no escalada, y a un incremento de 1 en el dominio de la TS escalada con  $\text{PASO\_TS} = 160$ .

## ES 2 266 273 T3

Como consecuencia, la TS (escalada) de encabezamientos que llegan al descompresor seguirán un patrón lineal en función de la hora del día, con alguna desviación debido a la fluctuación del retardo entre la fuente y el descompresor. Durante un funcionamiento normal, es decir, sin averías o fallos, la fluctuación del retardo estará destinada a cumplir los requisitos del tráfico conversacional de tiempo real. Por ello, con el uso de un reloj local el descompresor puede obtener una aproximación de la TS (escalada) en el encabezamiento que se debe descomprimir considerando su tiempo de llegada. A continuación, la aproximación se puede precisar con los k bits LSB de la TS (escalada) transportada en el encabezamiento. El valor requerido de k para garantizar una descompresión correcta es una función de la fluctuación entre la fuente y el descompresor.

Si el compresor conoce la fluctuación potencial introducida entre el compresor y el descompresor, puede determinar k usando un reloj local para realizar una estimación de la fluctuación en los tiempos de llegada de los paquetes, o alternativamente puede usar una k fija y rechazar los paquetes que lleguen demasiado fuera de tiempo.

Las ventajas de este esquema incluyen:

- a) El tamaño de la TS comprimida es constante y pequeño. En particular, NO depende de la longitud de los intervalos de silencio. Esta opción está en contraposición con otras técnicas de compresión de TS, las cuales en el comienzo de una secuencia hablada requieren el envío de una serie de bits en dependencia de la duración del intervalo de silencio precedente.
- b) No se requiere ninguna sincronización entre el reloj local para el compresor y el reloj local para el descompresor.

Obsérvese que aunque este esquema se puede hacer funcionar usando una TS tanto escalada como no escalada, en la práctica es preferible combinarlo con una codificación TS escalonada debido al requisito menos exigente sobre la resolución del reloj, por ejemplo, 20 ms en lugar de 1/8 ms. Por esta razón, el algoritmo que se describe a continuación supone que el esquema de codificación basado en reloj actúa sobre la TS escalada. El caso de la TS no escalada será similar, con cambios en los factores de escala.

Compresor: su función principal es determinar el valor de k. En este momento su ventana deslizante, TSW, contiene no solamente valores de referencia potenciales para la TS, sino también sus tiempos de llegada al compresor.

- 1) El compresor mantiene una ventana deslizante  $TSW = \{(T_j, a_j)\}$ , para cada encabezamiento j que se puede usar como referencia, en la que  $T_j$  es la TS escalada para el encabezamiento j, y  $a_j$  es el tiempo de llegada del encabezamiento j. La TSW desempeña la misma función que la VSW de la sección 4.5.2.
- 2) Cuando llega un nuevo encabezamiento n con  $T_n$  como TS escalada, el compresor anota el tiempo de llegada  $a_n$ . A continuación calcula

$$Fluct\_Max\_AC = \max \{ |(T_n - T_j) - ((a_n - a_j) / PASO\_TIEMPO)|, \text{ para todos los encabezamientos } j \text{ en } TSW \},$$

en la que PASO\_TIEMPO es el intervalo de tiempo equivalente a una PASO\_TS, por ejemplo, 20 ms. Fluct\_Max\_AC es la fluctuación máxima observada antes del compresor, en unidades de PASO\_TS, para los encabezamientos en TSW.

- 3) k se calcula como:  $k = \text{ceiling}(\log_2(2 * J + 1))$ , en la que  $J = Fluct\_Max\_AC + Fluct\_Max\_CD + 2$ .

Fluct\_Max\_CD es el límite superior de fluctuación esperada sobre el canal de comunicación entre el compresor y el descompresor (CD-CC). Depende únicamente de las características de CD-CC.

El factor 2 se explica por el error de cuantificación introducido por los relojes en el compresor y el descompresor, el cual puede ser +/-1.

Obsérvese que el cálculo de k sigue el algoritmo de compresión descrito en la sección 4.5.1, con  $p = 2^{(k-1)} - 1$ .

- 4) TSW está sometida a las mismas operaciones de ventana que en la sección 4.5.2, 1) y 3), excepto que los valores añadidos y eliminados están emparejados con sus tiempos de llegada.

Descompresor:

- 1) El descompresor siempre usa como encabezamiento de referencia el último encabezamiento descomprimido correctamente (verificado por la CRC). Mantiene el par  $(T_{ref}, a_{ref})$ , en el que  $T_{ref}$  es la TS escalada del encabezamiento de referencia, y  $a_{ref}$  es el tiempo de llegada del encabezamiento de referencia.
- 2) Cuando se recibe un encabezamiento comprimido n en el tiempo  $a_n$ , la aproximación de la TS escalada original se calcula como:

## ES 2 266 273 T3

$$T_{\text{aprox}} = T_{\text{ref}} + (a_n - a_{\text{ref}}) / \text{PASO\_TIEMPO}.$$

5 3) A continuación, la aproximación se precisa por medio de los k bits LSB transportados en el encabezamiento n, siguiendo al algoritmo de descompresión de la sección 4.5.1, con  $p = 2^{(k-1)} - 1$ .

10 Obsérvese que el algoritmo no supone ningún patrón específico en los paquetes que llegan al compresor, es decir, tolera una reordenación antes del compresor y un comportamiento de indicaciones de tiempo RTP no crecientes. Por otra parte, la resolución del reloj puede ser peor que PASO\_TIEMPO, en cuyo caso la diferencia, es decir, la resolución real - PASO\_TIEMPO, se trata como una fluctuación adicional en el cálculo de k.

### 4.5.5 Codificación de ID-IP con desviación

15 La presente sección supone que la pila Ipv4 en el anfitrión fuente asigna ID-IP al valor de un contador de 2 bytes el cual se incrementa en uno después de cada asignación a un paquete saliente. De este modo, el campo ID-IP de un flujo de paquetes IPv4 específico se incrementará en 1 de un paquete a otro excepto cuando la fuente haya emitido paquetes intermedios que no pertenezcan a ese flujo.

20 Para esas pilas IPv4, el SN RTP se incrementará en 1 para cada paquete y la ID-IP se incrementará por lo menos en la misma cantidad. De este modo, resulta más eficaz comprimir la desviación, es decir, (ID-IP - SN RTP), en lugar de la propia ID-IP.

25 El siguiente texto describe cómo comprimir/descomprimir la secuencia de desviaciones usando la codificación/decodificación W-LSB, con  $p = 0$  (consultar sección 4.5.1).

Compresor:

El compresor usa la codificación W-LSB para comprimir una secuencia de desviaciones

$$30 \text{ Desviación } i = \text{ID } i - \text{SN } i,$$

35 en la que ID  $i$  y SN  $i$  son los valores de la ID-IP y el SN RTP del encabezamiento  $i$ . La ventana deslizante contiene dichas desviaciones y no los valores de los campos de los encabezamientos, aunque en otros aspectos las reglas para añadir y eliminar desviaciones de la ventana siguen en la sección 4.5.2.

Descompresor:

40 El encabezamiento de referencia es el último encabezamiento descomprimido correctamente (verificado por la CRC).

45 Cuando recibe un paquete  $m$  comprimido, el descompresor calcula  $\text{Desviación}_{\text{ref}} = \text{ID}_{\text{ref}} - \text{SN}_{\text{ref}}$ , en la que ID<sub>ref</sub> y SN<sub>ref</sub> son los valores de ID-IP y SN RTP en el encabezamiento de referencia, respectivamente. A continuación se usa la decodificación W-LSB para descomprimir  $\text{Desviación}_m$ , usando los bits LSB recibidos en el paquete  $m$  y  $\text{Desviación}_{\text{ref}}$ . Obsérvese que  $m$  puede contener cero bits LSB para  $\text{Desviación}_m$ , en cuyo caso  $\text{Desviación}_m = \text{Desviación}_{\text{ref}}$ .

Finalmente, la ID-IP para el paquete  $m$  se regenera como

$$50 \text{ ID-IP para } m = \text{SN descomprimido del paquete } m + \text{Desviación}_m$$

55 Obsérvese que algunas pilas Ipv4 no transmiten el campo ID-IP en el orden de los bits de la red, sino que en su lugar envían los dos octetos invertidos. En este caso, el compresor puede comprimir el campo ID-IP después de intercambiar los bytes. Consecuentemente, el descompresor también intercambia los bytes de la ID-IP después de la descompresión para regenerar la ID-IP original. Esta opción requiere que el compresor y el descompresor se sincronicen en el orden de los bytes del campo ID-IP (ver sección de formato de encabezamientos).

### 4.5.6. Valores independientes de longitud variable

60 Los valores de PASO TS y algunos otros parámetros de compresión pueden variar ampliamente. PASO TS puede ser 160 para la voz y 90 000 para vídeo de 1 f/s. Para optimizar la transferencia de dichos valores, se usa un número variable de octetos para codificarlos. Los primeros pocos bits del valor codificado determinan su longitud.

65 1 octeto: el primer bit es cero. 7 bits transferidos. Hasta 127 decimal. Octetos codificados en hexadecimal: 00 a 7F

2 octetos: primeros bits 10. 14 bits. Hasta el 16 383 decimal. Octetos codificados en hexadecimal: 80 00 a BF FF

## ES 2 266 273 T3

3 octetos: primeros bits 110. 21 bits. Hasta el 2 097 152 decimal. Octetos codificados en hexadecimal: C0 00 00 a DF FF FF

5 4 octetos: primeros bits 111. 29 bits. Hasta el 536 870 912 decimal. Octetos codificados en hexadecimal: E0 00 00 00 a FF FF FF FF

### 4.5.7. Valores codificados a través de varios campos en encabezamientos comprimidos

10 Cuando un encabezamiento comprimido tiene una extensión, puede haber presentes trozos de un valor codificado en más de un campo. Cuando un valor codificado se divide sobre varios campos de esta manera, los bits más significativos del valor están más cerca del comienzo del encabezamiento. Si el número de bits disponibles en los campos de los encabezamientos comprimidos supera el número de bits del valor, el campo más significativo se rellena con ceros en sus bits más significativos.

15 Por ejemplo, se puede transferir un valor TS no escalado usando un encabezamiento UOR-2 (ver sección 5.7) con una extensión de tipo 3. En ese caso, el bit Tsc de la extensión no está activado y el campo TS de longitud variable de la extensión es 4 octetos (ver sección 4.5.6). El campo TS UOR-2 contendrá los tres bits más significativos de la TS no escalada, y el campo TS de 4 octetos en la extensión contiene los 29 bits restantes.

20

## 5. Protocolo

### 5.1. Estructuras de datos

25 [[Nota del editor: Esta sección será principalmente una versión a modo de especificación, más detallada, de la información introductoria del punto 4.2.]]

#### 5.1.1. Parámetros por canal

30 5.1.2. Parámetros por CID, perfiles

#### 5.1.3. Contextos

### 5.2. Tipos de paquetes

35

40 El esquema ROHC usa seis tipos de paquetes, indicados por los primeros pocos bits del encabezamiento. El formato de un tipo de paquete puede depender del modo. Por esta razón, se usa un esquema de nomenclatura de la forma <modos en los que se usa el formato>-<número de tipo de paquetes>-<alguna propiedad> para identificar de forma exclusiva el formato cuando sea necesario. Por ejemplo, UOR-2, R-1. Para obtener los formatos exactos de los tipos de paquetes, consultar la sección 5.7.

45 [Nota de Mikael Degermark - el resto de la presente sección es un intento de escribir qué tipos de paquetes se pueden usar y cuándo. Comenzó como un intento de agrupar conjuntamente los tipos de paquetes de una manera que facilitara adicionalmente la escritura de los siguientes capítulos. Parece que no he obtenido un resultado demasiado satisfactorio a la hora de encontrar un número pequeño de clases de paquetes. ¿Se puede definir esta tabla o algo similar? Resultaría muy útil para un desarrollador. Tal vez, la tabla se debería trasladar a algún otro sitio...]

50 La siguiente tabla resume qué formato se puede usar y cuándo. Si un tipo de paquete está entre paréntesis, el uso de dicho tipo de paquete reducirá la robustez. Si se indican varios tipos de paquete, debería usarse el tipo más pequeño que pueda representar el encabezamiento que se debe comprimir.

55

60

65

## ES 2 266 273 T3

El encabezamiento adhiere funciones SN del compresor al formato del encabezamiento de funciones SN

5	Modo	estado	establecido	TS/ID	otros	acción
	UO	SO	Sí Sí	Sí	Sí No	UO-0 goto FO
10		FO	Sí Sí Sí	Sí No *	Sí Sí No	goto SO UO-1, UOR-2 UO-1 Ext, UOR-2
15			No No No	Sí No *	Sí Sí No	(UO-0), UO-1, UOR-2 *UO-1, UOR-2 *UO-1, UOR-2, IR-DYN
20	R	SO	Sí Sí	Sí	Sí No	R-0, R-0-CRC goto FO
25		FO	Sí Sí No No	Sí No Sí No	Sí No Sí No	goto SO *R-1, UOR-2 (R-0-CRC), *R-1, UOR-2 *R-1, UOR-2
30	*: la robustez se verá afectada a no ser que se use un encabezamiento PT-1 el cual proporcione el resultado correcto también cuando se usen funciones SN antiguas.					

### 5.2.1. Formatos de paquetes desde el compresor al descompresor

Se usan cuatro tipos de paquetes desde el compresor al descompresor. Tres para encabezamientos comprimidos, y uno para inicialización/restauración. A estos encabezamientos les preceden identificadores de contexto de la longitud adecuada. En la sección 5.7 se encuentran formatos de encabezamientos exactos.

Tipo de paquete cero: R-0, R-0-CRC, UO-0.

Este tipo de paquete, el mínimo, se usa cuando el descompresor conoce los parámetros de todas las funciones SN, y el encabezamiento que se debe comprimir se adhiere a dichas funciones. De este modo, solamente es necesario comunicar el SN RTP codificado en W-LSB.

Modo R: En el encabezamiento se puede usar como referencia para la subsiguiente descompresión únicamente si hay presente una CRC (tipo de paquete R-0-CRC).

Modo U y modo O: Hay presente una CRC pequeña en el paquete UO-0.

Tipo de paquete 1: R-1, R-1-ID, R-1-TS, UO-1, UO-1-ID, UO-1-TS.

Este tipo de paquete se usa cuando el número de bits necesarios para el SN supera los disponibles en el tipo de paquete cero o cuando los parámetros de las funciones SN para TS RTP o ID-IP cambian.

Modo R: Los paquetes R-1-\* no se usan como referencias para la subsiguiente descompresión. Los valores para otros campos que no sean la TS RTP o la ID-IP se pueden comunicar usando una extensión, aunque los mismos no actualizan el contexto.

Modo U y modo O: Como referencias para una futura compresión solamente se pueden usar los valores de SN RTP, TS RTP e ID-IP. Se pueden proporcionar valores que no realicen actualizaciones, para otros campos usando una extensión (UO-1-ID).

Tipo de paquete 2: UOR-2, UOR-2-ID, UOR-2-TS

Este tipo de paquete se puede usar para cambiar los parámetros de cualquier función SN, excepto para los parámetros correspondientes a los campos más estáticos. Los encabezamientos de paquetes transferidos usando el tipo de paquete 2 se pueden usar como referencias para una subsiguiente descompresión.

Tipo de paquete 5: IR

Este tipo de paquete comunica la parte estática del contexto, es decir, el valor de las funciones SN constantes. Opcionalmente, también puede comunicar la parte dinámica del contexto, es decir, los parámetros de las funciones SN no constantes.

Tipo de paquete 6: IR-DYN

Este tipo de paquete comunica la parte dinámica del contexto, es decir, los parámetros de las funciones SN no constantes.

### 5.2.2. Paquetes de realimentación del descompresor al compresor

Los paquetes de realimentación transportan información desde el descompresor al compresor. Se soportan los siguientes tipos de realimentación. Los tres primeros controlan la transición de estados y modos, y el cuarto informa al compresor de que el descompresor no dispone de los suficientes recursos para descomprimir un flujo continuo de paquetes.

- ACK: Acusa el recibo de una descompresión satisfactoria de un paquete, lo cual significa que hay una probabilidad elevada de que el contexto esté actualizado.
- NACK: Indica que el contexto dinámico del descompresor está desincronizado.
- NACK ESTÁTICO: Indica que el contexto estático del descompresor no es válido o no ha sido establecido.
- RECHAZO: Indica que no se puede soportar la compresión de este flujo continuo de paquetes debido a limitaciones en el recurso.

### 5.2.3. Parámetros necesarios para la transición del modo

El tipo de paquete UOR-2 es común para todos los modos. Puede transportar una extensión con un parámetro de modo el cual puede adoptar los valores U = Unidireccional, O = Optimista Bidireccional, y R = Fiable Bidireccional.

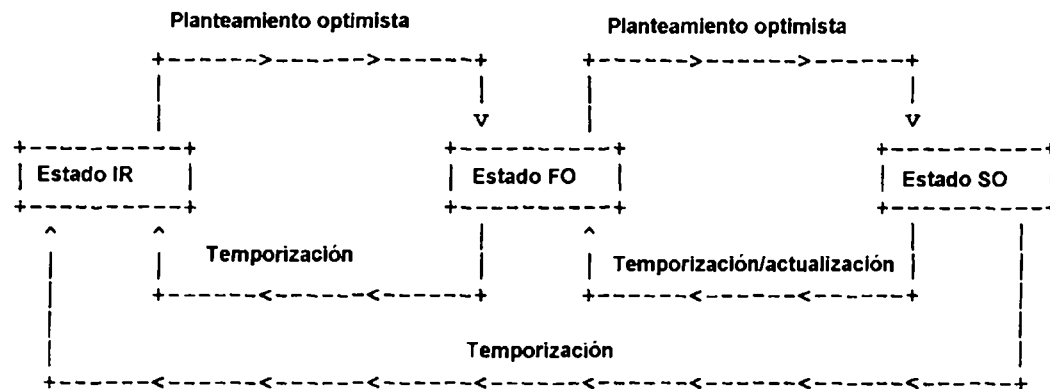
Los paquetes de realimentación de los tipos ACK, NACK, y NACK ESTÁTICO transportan números de secuencia y también pueden transportar un parámetro de modo que indique el modo de compresión deseado: U, O, o R.

De forma abreviada, la notación PAQUETE (modo) se usa para indicar qué valor de modo transporta un paquete. Por ejemplo, un ACK con el parámetro de modo R se escribe ACK(R), y un UOR-2 con el parámetro de modo O se escribe UOR-2(O).

## 5.3. Funcionamiento en el modo unidireccional

### 5.3.1. Estados y lógica del compresor (modo U)

A continuación se presenta la máquina de estados para el compresor en el modo unidireccional. Después de la figura se proporcionan detalles de las transiciones entre estados y de la lógica de compresión.



#### 5.3.1.1. Lógica de la transición de estados (modo U)

La lógica de la transición para estados de compresión en el modo unidireccional se basa en tres principios: el principio del planteamiento optimista, temporizaciones, y la necesidad de actualizaciones.

5.3.1.1.1. *Planteamiento optimista, transición en sentido ascendente*

La transición a un estado de compresión superior en el modo unidireccional se lleva a cabo según el principio del planteamiento optimista. Esto significa que el compresor realiza una transición a un estado de compresión superior cuando está bastante seguro de que el descompresor ha recibido suficiente información como para descomprimir correctamente los paquetes enviados según el estado de compresión superior.

Cuando el compresor está en el estado IR, permanecerá en el mismo hasta que considere que el descompresor ha recibido correctamente la información de contexto estática. Para la transición desde el estado FO a SO, el compresor debería estar seguro de que el descompresor tiene todos los parámetros necesarios para descomprimir según un patrón fijado.

Normalmente, el compresor obtiene la seguridad sobre el estado del descompresor enviando varios paquetes con la misma información según el estado de compresión inferior. Si el descompresor recibe cualquiera de estos paquetes, se encontrará sincronizado con el compresor. En el presente documento no se define el número de paquetes consecutivos a enviar para obtener la seguridad mencionada.

5.3.1.1.2. *Temporizaciones, transición en sentido descendente*

Mediante el uso del planteamiento optimista descrito anteriormente, existirá siempre una posibilidad de fallo ya que puede ser que el descompresor no haya recibido la suficiente información para realizar una descompresión correcta. Por esta razón, el compresor debe realizar periódicamente una transición a estados de compresión inferiores. La transición periódica al estado IR se debería llevar a cabo con menos frecuencia que la transición al estado de FO. Por esta razón, se deberían usar dos temporizaciones diferentes para estas transiciones. Para obtener un ejemplo de cómo implementar restauraciones periódicas, consultar el capítulo [IPHC] 3.3.1-3.3.2.

5.3.1.1.3. *Necesidad de actualizaciones, transición en sentido descendente*

Además de las transiciones de estado hacia atrás que se llevan a cabo debido a las temporizaciones periódicas, el compresor también debe realizar una transición inmediatamente de vuelta al estado FO cuando el encabezamiento que se debe comprimir no comprime el patrón establecido.

5.3.1.2. *Lógica de la compresión y paquetes usados (modo U)*

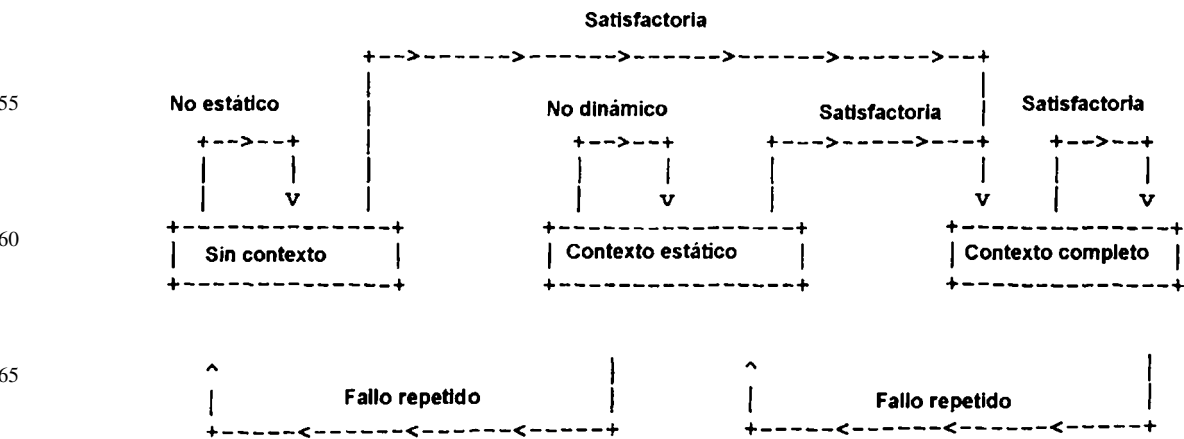
El compresor selecciona el formato de paquete más pequeño posible que puede comunicar los cambios deseados, y que tiene los bits requeridos para valores codificados W-LSB. Las ventanas deslizantes usadas en la codificación W-LSB tienen una anchura fija.

5.3.1.3. *Realimentación en el modo unidireccional*

El modo de funcionamiento unidireccional está diseñado para funcionar sobre enlaces en los que no hay disponible un canal de realimentación. No obstante, si hay disponible un canal de realimentación, el descompresor PUEDE enviar un acuse de recibo de una descompresión satisfactoria con el parámetro de modo fijado a U (enviar un ACK(U)). Cuando el compresor recibe dicho mensaje, PUEDE deshabilitar o incrementar el intervalo entre restauraciones IR periódicas.

5.3.2. *Estados y lógica del descompresor (modo U)*

A continuación se presenta la máquina de estados correspondiente al descompresor en el modo unidireccional. Después de la figura se proporcionan detalles de las transiciones entre estados y de la lógica de descompresión.



## ES 2 266 273 T3

### 5.3.2.1. *Lógica de transición de los estados (modo U)*

La lógica de transición de los estados del descompresor es mucho más sencilla que para el lado del compresor. También es común para la totalidad de los tres modos de funcionamiento. Una descompresión satisfactoria llevará siempre el descompresor al estado de Contexto Completo. Un fallo repetido de la descompresión obligará al descompresor a realizar una transición hacia atrás a un estado inferior. El descompresor no intenta descomprimir encabezamientos en absoluto en los estados Sin Contexto y Contexto Estático a no ser que en el propio paquete se incluya suficiente información.

### 5.3.2.2. *Lógica de descompresión (modo U)*

La descompresión en el modo unidireccional se lleva a cabo siguiendo tres etapas las cuales se describen en las siguientes secciones.

#### 5.3.2.2.1. *Decisión sobre si se permite la descompresión*

En el estado de Contexto Completo, se puede intentar la descompresión con independencia de qué tipo de paquetes se reciba. No obstante, para los otros estados la descompresión no está siempre permitida. En el estado Sin Contexto, solamente se pueden descomprimir paquetes IR, los cuales transportan los campos de información estática. Además, cuando se está en el estado Contexto Estático, se pueden descomprimir solamente paquetes que transporten una CRC fuerte (es decir, paquetes IR, IR-DYN, o UOR-2). Si no se puede realizar la descompresión, el paquete se descarta, a no ser que se use el mecanismo opcional de descompresión retardada, ver sección x.x.x.

#### 5.3.2.2.2. *Reconstruir y verificar el encabezamiento*

//TWB basada en el capítulo 4.5

Interpretación LSB, corrección del reinicio cíclico

Todos los encabezamientos descomprimidos se verifican con la CRC

#### 5.3.2.2.3. *Causas de las discordancias de la CRC*

Una discordancia en la CRC de 3 bits puede estar provocada por uno o más de las siguientes situaciones:

1. errores de bits residuales en el encabezamiento actual,
2. un contexto dañado debido a errores de bits residuales en los encabezamientos anteriores, o
3. pérdida de 16 o más paquetes consecutivos lo cual provoca que el SN de 4 bits realice un reinicio cíclico (lo cual es, esencialmente, otro tipo de daño del contexto).

Finalmente la CRC de 3 bits presente en los encabezamientos UO-0 y UO-1 detectará de forma fiable los daños en el contexto, ya que la probabilidad de daños no detectados en el contexto se reduce exponencialmente con cada encabezamiento nuevo procesado. No obstante, los errores de bits residuales en el encabezamiento actual solamente se detectan con una buena probabilidad, no de forma fiable.

Cuando una discordancia de la CRC está provocada por errores de bits residuales en el encabezamiento actual (caso 1 anterior), el descompresor debería permanecer en su estado actual para evitar una pérdida innecesaria de paquetes subsiguientes. Por otro lado, cuando la discordancia está provocada por un contexto dañado, el descompresor puede intentar reparar el contexto aunque si no lo consigue, debe trasladarse a un estado inferior para evitar la entrega de encabezamientos incorrectos. Cuando la discordancia está provocada por una pérdida prolongada, el descompresor podría intentar realizar intentos adicionales de descompresión.

- Detección de una pérdida prolongada (temporizador)
- Corrección de un reinicio cíclico incorrecto
- Intentos repetidos de reconstrucción en caso de una pérdida prolongada

// El siguiente texto debería condensarse y estructurarse de manera que

// trate las cuestiones enumeradas anteriormente en un orden correcto. Se

// deberían describir principios claros, sencillos y ordenados.

[Nota de Micke: ¿lo he conseguido?]

## ES 2 266 273 T3

En las siguientes secciones, se describen mecanismos que podrían detectar la causa de una discordancia de la CRC. Si estos mecanismos no consiguen encontrar la razón de la discordancia de la CRC, NO SE DEBEN realizar intentos adicionales de descompresión.

### 5 5.3.2.2.4. *Detección de daños en el contexto*

Los daños en el contexto hacen que aumente la probabilidad de una discordancia de la CRC. Con las CRC de 3 bits, la probabilidad de detectar el encabezamiento erróneo es aproximadamente 7/8. Con las CRC de 7 bits y 8 bits, la probabilidad es aproximadamente 127/128 y 255/256, respectivamente.

10 En el estado de Contexto Completo, siempre que la descompresión de 4 encabezamientos UO-0 o UO-1 se haya situado fuera de los últimos 6 encabezamientos, el descompresor se traslada al estado de Contexto Estático. En el estado de Contexto Estático. Siempre que falle la descompresión de 2 de entre los últimos 3 paquetes UOR-2 o IR-DYN, el compresor se traslada al estado Sin Contexto. Alguna de las implementaciones PUEDE trasladarse a estados inferiores antes, aunque NO DEBERÍA trasladarse a un estado inferior más tarde de los especificado en el presente documento.

[[Nota del editor: probablemente los números exactos requieren una discusión adicional]].

### 20 5.3.2.2.5. *Reparación del reinicio cíclico del SN*

Se deben perder por lo menos 16 encabezamientos para que el SN de los encabezamientos UO-0 o UO-1 realice un reinicio cíclico. El descompresor podría ser capaz de detectar esta situación usando un reloj local. Se puede usar el siguiente algoritmo:

25 a. El descompresor anota el tiempo de llegada,  $a_j$ , de cada paquete entrante  $i$ . Los tiempos de llegada de paquetes en los que la descompresión no sea satisfactoria se descartan.

30 b. Cuando la descompresión no es satisfactoria, el descompresor calcula  $INTERVALO = a_i - a_{i-1}$ , es decir, el tiempo transcurrido entre la llegada del anterior paquete descomprimido correctamente y el paquete actual.

35 c. Si se ha producido un reinicio cíclico, el INTERVALO se corresponderá con por lo menos 16 tiempos entre paquetes. Basándose en una estimación del tiempo entre llegada y llegada de paquete, obtenido por ejemplo usando una media móvil de los tiempos de llegada, PASO TS, o TIEMPO TS, el descompresor decide si el INTERVALO se puede corresponder con 16 ó más tiempos entre paquetes.

d. Si se decide que el INTERVALO es por lo menos 16 tiempos entre llegada y llegada de paquetes, el descompresor suma 16 al SN del contexto e intenta descomprimir el paquete usando el contexto nuevo.

40 e. Si esta descompresión es satisfactoria, el descompresor actualiza el contexto aunque NO DEBERÍA entregar el paquete a capas superiores. El siguiente paquete también se descomprime y actualiza el contexto si su CRC resulta satisfactoria, aunque SE DEBERÍA descartar. Si la descompresión del tercer paquete usando el contexto nuevo también es satisfactoria, se considera que la reparación del contexto es satisfactoria y este paquete y los subsiguientes descomprimidos se entregan a las capas superiores.

45 f. Si cualquiera de los tres intentos de descompresión en d. y e. no es satisfactorio, el descompresor descarta los paquetes y se traslada al estado Contexto Estático.

50 Usando este mecanismo, el descompresor puede ser capaz de reparar el contexto después de una pérdida excesiva, a costa de descartar dos paquetes.

### 5.3.2.2.6. *Reparación de actualizaciones SN incorrectas*

55 Puede que la CRC no consiga detectar errores residuales en el encabezamiento comprimido debido a su longitud limitada, es decir, puede ocurrir que el paquete descomprimido incorrectamente tenga la misma CRC que el paquete no comprimido original, provocando que el paquete descomprimido incorrectamente sea aceptado y que el contexto sea actualizado. Esto puede derivar en el uso de un SN de referencia erróneo en la decodificación W-LSB, ya que el SN de referencia se actualiza para cada encabezamiento descomprimido satisfactoriamente de ciertos tipos.

60 Si se produce esta situación, el descompresor detectará la descompresión incorrecta del siguiente paquete con una gran probabilidad, aunque no sabrá la razón del fallo. El siguiente mecanismo permite que el descompresor decida si el contexto fue actualizado incorrectamente por un paquete anterior.

65 1) El descompresor siempre mantiene dos SN descomprimidos: el último (ref 0) y el anterior a este último (ref -1).

## ES 2 266 273 T3

2) Cuando se recibe un SN comprimido:

SN act = SN descomprimido usando ref 0 como valor de referencia descomprimir el resto del encabezamiento usando SN act

5

SI (el encabezamiento supera la prueba CRC)

ref -1 = ref 0

10

ref 0 = SN act

3) SI NO

15 SN act = SN descomprimido usando ref -1 como valor de referencia descomprimir resto del encabezamiento usando SN act

SI (el encabezamiento supera la prueba CRC)

20

ref0 = SN act; // obsérvese que ref -1 no ha cambiado

4) SI NO

// una de entre dos situaciones de error:

25

// 1) error de bit en el encabezamiento actual,

// 2) contexto de descompresión alterado. Usar 5.3.2.2.4 para

// diferenciar, (por ejemplo, considérese el contexto alterado

30

// solamente después de k errores de entre n)

La finalidad de este algoritmo es reparar el contexto. Si el encabezamiento generado en 3) supera la prueba CRC, se deben descomprimir además satisfactoriamente dos encabezamientos más antes de considerar que la reparación ha sido satisfactoria. De los tres encabezamientos satisfactorios, los primeros dos DEBERÍAN ser descartados y solamente entregar el tercero a las capas superiores. Si la descompresión de cualquiera de los tres encabezamientos no es satisfactoria, el descompresor DEBE descartar ese encabezamiento y los encabezamientos generados previamente, y trasladarse al estado de Contexto Estático.

40

5.3.2.3. *Realimentación en modo unidireccional*

Para mejorar el rendimiento correspondiente al modo unidireccional a través de un enlace que sí dispone de un canal de realimentación, el descompresor PUEDE enviar un acuse de recibo cuando la descompresión sea satisfactoria. La fijación a U del parámetro de modo en el paquete ACK indica que el compresor va a permanecer en el modo unidireccional. Cuando recibe un ACK(U), el compresor DEBERÍA reducir la frecuencia de los paquetes IR ya que la información estática se ha recibido correctamente, aunque no se necesario dejar de enviar paquetes IR. Si continúan llegando paquetes IR, el descompresor PUEDE repetir el ACK(U), aunque no DEBERÍA repetir el ACK continuamente.

50

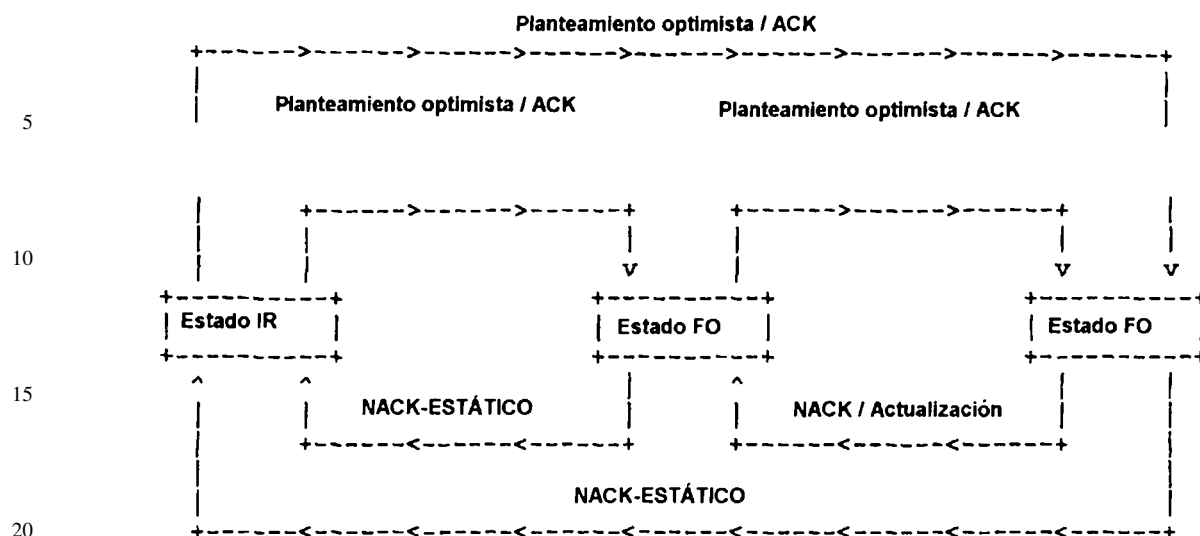
5.4. *Funcionamiento en el modo optimista bidireccional*

5.4.1. *Estados y lógica del compresor (modo O)*

A continuación se presenta la máquina de estados para el compresor en el modo optimista bidireccional. Después de la figura se proporcionan detalles de cada estado, las transiciones entre estados y la lógica de compresión.

60

65



5.4.1.1. Lógica de transición de estados

La lógica de transición para estados de compresión en el modo optimista bidireccional tiene mucho en común con la lógica del modo unidireccional. El principio del planteamiento optimista y las transiciones, debido a la necesidad de actualizaciones, funciona de la misma manera que la descrita en el capítulo 5.3.1. No obstante, en el modo optimista no existen temporizaciones. En su lugar, el modo optimista hace uso de la realimentación desde el descompresor al compresor tanto para transiciones a la dirección hacia atrás como para una transición hacia adelante mejorada.

5.4.1.1.1. Solicitudes de contexto, acuses de recibo negativos (acuses NACK)

5.4.1.1.2. Acuses de recibo opcionales

5.4.1.2. Lógica de compresión y paquetes usados

// ¿¿Existe alguna diferencia con respecto al modo unidireccional??

5.4.2. Estados y lógica del descompresor

Los estados de descompresión y la lógica de transición de estados son los mismos que para el caso unidireccional (ver sección 5.3.2). Lo que es diferente es la lógica de descompresión y realimentación.

5.4.2.1. Lógica de descompresión

5.4.2.1.1. Descompresión de indicaciones de tiempo basada en temporizadores

NOTA: Este mecanismo debe describirse aquí ya que no se puede usar en el modo unidireccional debido a que no se puede proporcionar ninguna indicación al compresor si el método basado en temporizadores no es satisfactorio.

5.4.2.2. Lógica de realimentación

La lógica de realimentación define qué mensajes de realimentación enviar debido a diferentes acontecimientos cuando funciona en los diversos estados.

// NOTA: Se debe reescribir y no referirse a paquetes FH, FO e IR

- En el estado NC:
  - Cuando un paquete FH se descomprime correctamente, enviar un ACK con el parámetro de modo fijado a O
  - Cuando se recibe un paquete FO o SO o no ha resultado satisfactoria la descompresión de un paquete FH, enviar un NACK ESTÁTICO con el parámetro de modo fijado a O
- En el estado SC:
  - Cuando se descomprime correctamente un paquete FH, enviar un ACK con el parámetro de modo fijado a O
  - Cuando se descomprime correctamente un paquete FO enviar opcionalmente un ACK con el parámetro de modo fijado a O

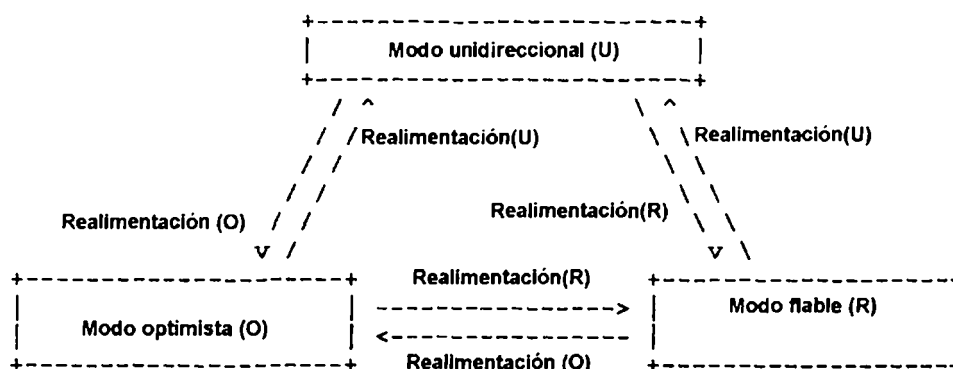


## ES 2 266 273 T3

- En el estado SC:
- Cuando se descomprime correctamente un paquete FO o FH, enviar un ACK con el parámetro de modo fijado a R
  - Cuando se recibe un paquete SO enviar un NACK con el parámetro de modo fijado a R
- En el estado FC:
- Cuando no ha resultado satisfactoria la descompresión de un paquete FO o FH, enviar un NACK ESTÁTICO con el parámetro de modo fijado a R
  - Cuando se descomprime correctamente un paquete SO de actualización, enviar periódicamente un ACK con el parámetro de modo fijado a R
  - Cuando no ha resultado satisfactoria la descompresión de un paquete SO, FO o FH, enviar un NACK con el parámetro de modo fijado a R.

### 5.6. Transiciones de modo

La decisión de cambiar de un modo de compresión a otro la toma el descompresor y en la siguiente figura se muestran las posibles transiciones de modo. Los capítulos subsiguientes describen cómo se realizan las transiciones junto con las excepciones para la funcionalidad de compresión y descompresión durante las transiciones.



#### 5.6.1. Compresión y descompresión durante transiciones de modo

Para poder funcionar correctamente, tanto el compresor como el descompresor deben disponer de una variable para cada modo según el cual debían funcionar. Los capítulos subsiguientes definen exactamente cuándo cambiar el valor de esta variable de modo y por lo tanto cambiar el modo de funcionamiento. Además, cuando la ROHC realiza una transición de un modo a otro, existen varios casos con reglas especiales para lo que está permitido realizar en los lados del compresor y el descompresor durante la fase de transición. Estas reglas especiales se definen por medio de parámetros de excepción que fijan varias reglas de excepción a seguir. Las descripciones de las transiciones en los capítulos subsiguientes se refieren a estos parámetros de excepción y definen cuándo y a qué valores se fijan. Todos los parámetros relacionados con el modo se enumeran a continuación con los posibles valores, explicaciones y reglas:

Parámetros para el lado del compresor:

- C\_MODO Los valores posibles para el parámetro C\_MODO son (U)UNIDIRECCIONAL, (O)OPTIMISTA y (R)FIABLE. Inicialmente el valor se DEBE fijar a U.
- C\_TRANS Los valores posibles para el parámetro C\_TRANS son (P)PENDIENTE y (D)REALIZADO. Inicialmente el valor se DEBE fijar a D. Cuando el parámetro se fija a P, se REQUIERE que el compresor únicamente use formatos de paquete comunes para todos los modos y NO PERMITIRÁ la transición al estado SO. También DEBE ignorarse una nueva solicitud de transición de modo cuando esté pendiente una transición ya iniciada.

Parámetros para el lado del descompresor:

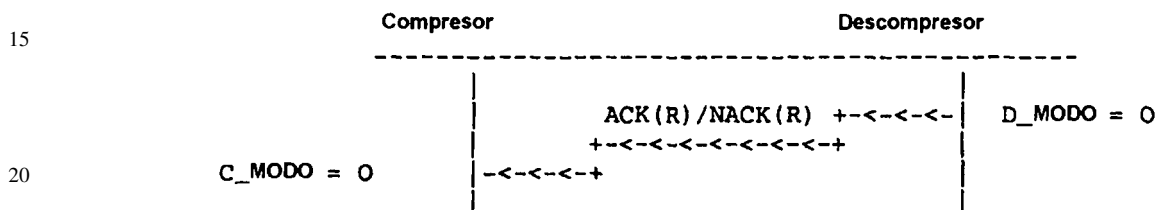
- D\_MODO Valores posibles para el parámetro D\_MODO son (U)UNIDIRECCIONAL, (O)OPTIMISTA y (R)FIABLE. Inicialmente el valor se DEBE fijar a U.
- D\_TRANS Valores posibles para el parámetro D\_TRANS son (I)INICIADO, (P)PENDIENTE y (D)REALIZADO. Inicialmente el valor se DEBE fijar a D. Las transiciones a modos nuevos están únicamente

## ES 2 266 273 T3

permitidas cuando el parámetro se fije a D. Cuando se fije a I, el descompresor debería enviar un NACK o ACK para todos los paquetes recibidos, lo cual puede dejar de realizarlo cuando el parámetro se fije a P o D.

### 5.6.2. Transición del modo Unidireccional a Optimista

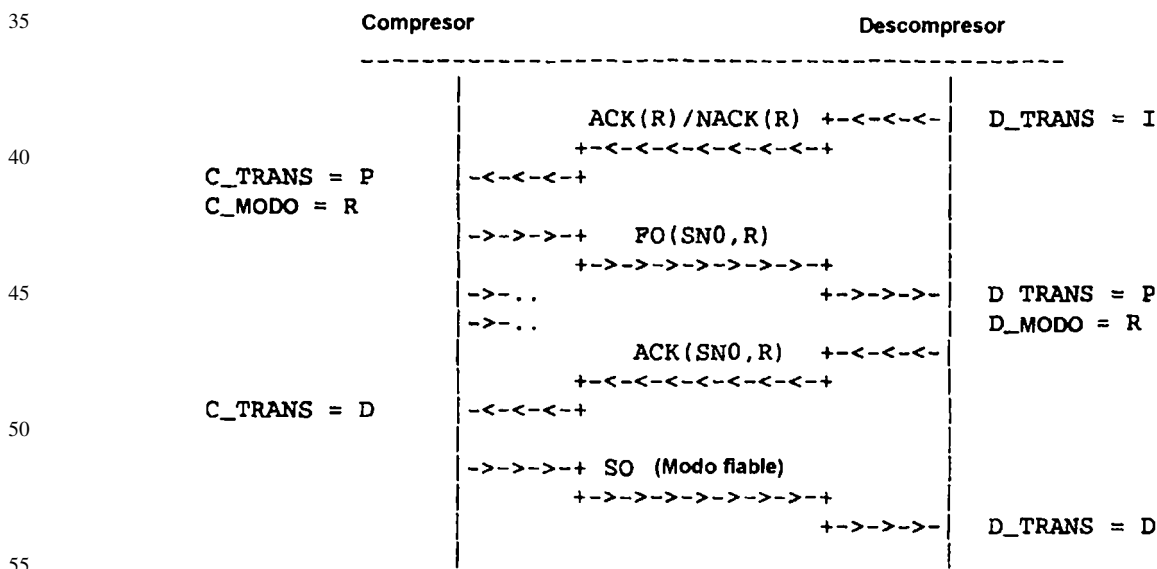
Siempre que haya un canal de realimentación disponible, el descompresor puede, en cualquier momento, decidir iniciar una transición del modo unidireccional al Optimista bidireccional. Se pueden usar todos los paquetes de realimentación con el parámetro de modo fijado a O y a continuación el descompresor puede comenzar directamente a funcionar en el modo Optimista. El compresor realiza una transición desde el modo unidireccional al optimista tan pronto como recibe cualquier paquete de realimentación con el parámetro de modo fijado a O. A continuación se describe el procedimiento de transición:



Si se pierde el paquete de realimentación, el compresor continuará funcionando en el modo unidireccional con restauraciones periódicas aunque en cuanto el descompresor envíe otro paquete de realimentación, el compresor también realizará una transición al modo optimista.

### 5.6.3. Del modo Optimista al Fiable

La transición del modo Optimista al Fiable está permitida únicamente después de que se haya descomprimido correctamente por lo menos un paquete, lo cual significa que se establece la parte estática del contexto. Para iniciar la transición se usa el paquete de realimentación bien ACK(R) o bien NACK(R) y el compresor DEBE siempre funcionar en el estado FO durante la transición. A continuación se describe el procedimiento de transición:



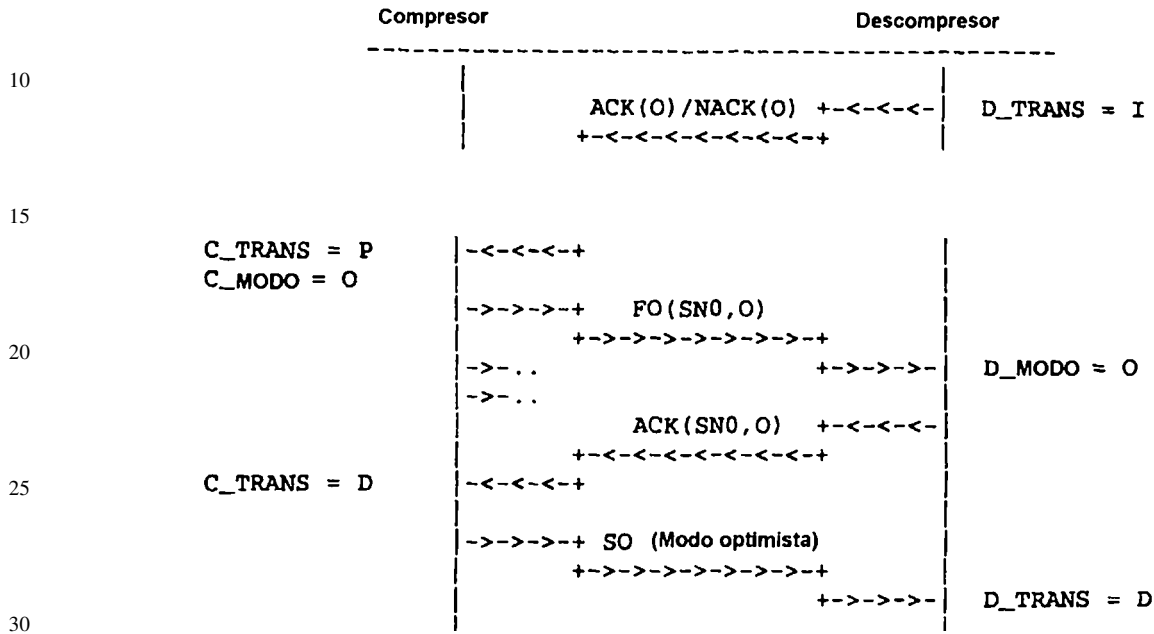
Mientras el descompresor no haya recibido un paquete FO con el parámetro de transición de modo fijado a R, debe permanecer en el modo Optimista. El compresor debe permanecer en el estado FO hasta que haya recibido un ACK correspondiente a un paquete FO enviado con el parámetro de transición de modo fijado a R (indicado por el número de secuencia).

### 5.6.4. Del modo Unidireccional al Fiable

Como la transición del modo Unidireccional al Optimista no requiere ningún intercambio de señalización de entrada en contacto, es posible realizar directamente una transición del modo Unidireccional al Fiable, siguiendo el mismo procedimiento de transición que en el punto 5.6.3 anterior.

5.6.5. Del modo Fiable al Optimista

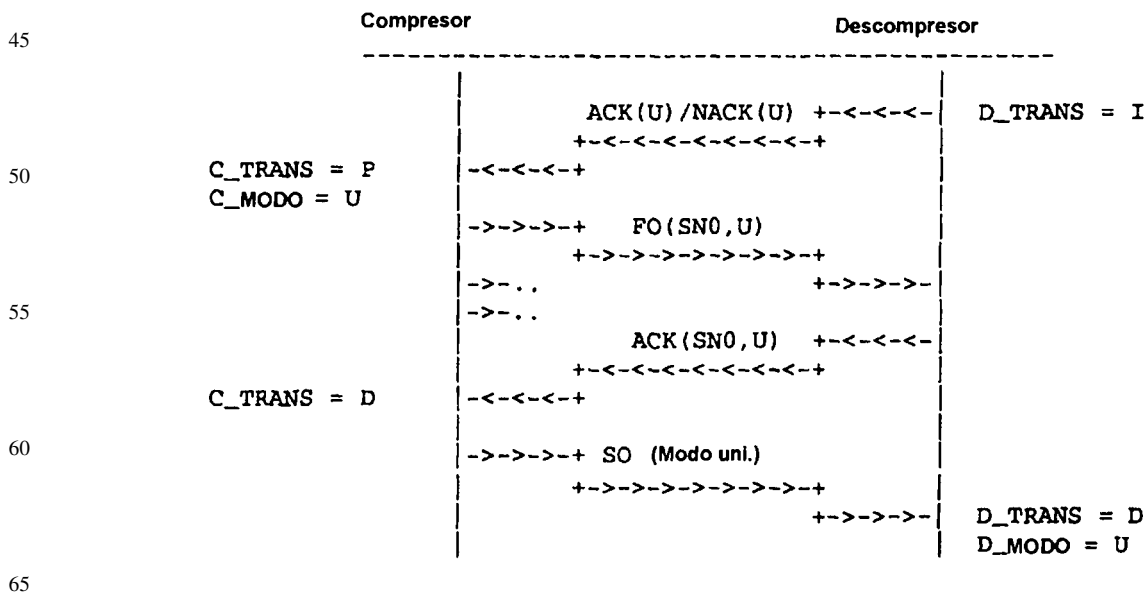
Para iniciar la transición del modo Fiable al Optimista se usa el paquete de realimentación bien ACK(O) o bien NACK(O) y el compresor DEBE siempre funcionar en el estado FO durante la transición. A continuación se describe el procedimiento de transición:



Mientras que el descompresor no haya recibido un paquete FO con el parámetro de transición de modo fijado a O, debe permanecer en el modo fiable. El compresor debe permanecer en el estado FO hasta que haya recibido una ACK correspondiente a un paquete FO enviado con el parámetro de transición de modo fijado a O (indicado por el número de secuencia).

5.6.6. Transición al modo Unidireccional

Es posible forzar una transición de vuelta al modo unidireccional si el descompresor desea realizarlo. Con independencia de a partir de qué modo comience, DEBE llevarse a cabo un intercambio de señalización de entrada en contacto a tres bandas para garantizar una transición correcta en el lado del compresor. A continuación se describe el procedimiento de transición:



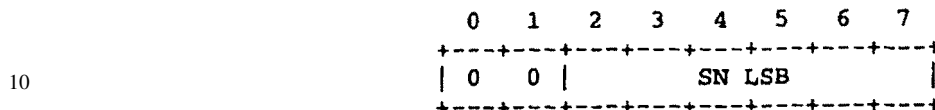
El descompresor debe continuar enviando la realimentación hasta que sepa que el compresor está preparado con la transición.

## ES 2 266 273 T3

### 5.7. Formatos de los paquetes

#### 5.7.1. Tipo de paquete 0: UO-0, R-0, R-0-CRC

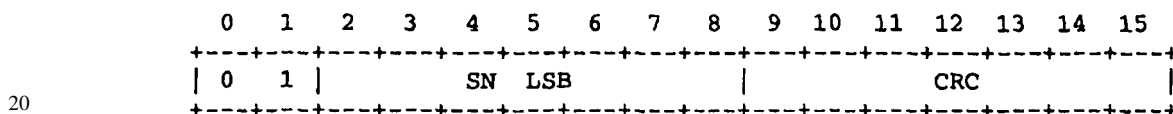
5 R-0



10

R-0-CRC

15



20

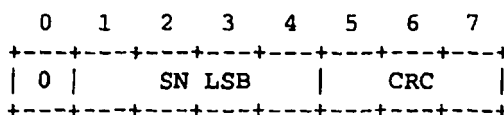
25 LSB SN: Bits menos significativos de SN RTP. (Obsérvese que, en R-0-CRC, LSB SN se encuentra a ambos lados del límite de un byte).

CRC: CRC de 7 bits, calculada según la sección 5.9.2.

30 Nota: Los encabezamientos R-0 NO DEBEN actualizar ninguna parte del contexto. Los encabezamientos R-0-CRC NO DEBEN actualizar ninguna parte del contexto no relacionada directamente con el SN RTP.

UO-0

35



40

LSB SN: Bits menos significativos de SN RTP (ver sección 4.5.1)

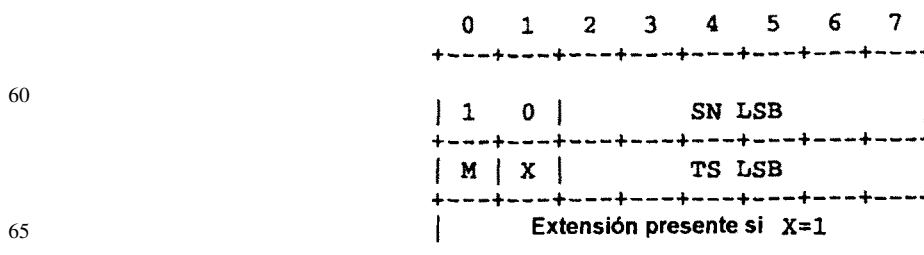
CRC: CRC de 3 bits (ver sección 5.9.2)

45 Nota: El encabezamiento UO-0 DEBE actualizar el valor actual del SN RTP. El encabezamiento UO-0 NO DEBE actualizar ninguna parte del contexto no relacionada directamente con el SN RTP.

#### 5.7.2. Tipo de paquete 1 (modo R): R-1, R-1-TS, R-1-ID

50 Este tipo de paquete se usa en el modo R cuando el SN RTP del encabezamiento que se debe comprimir difiere tanto con respecto al contexto que los LSB del paquete tipo 0 son demasiado pequeños, o cuando las funciones del SN RTP a la indicación de tiempo RTP o la ID IP cambian y es necesario comunicarlas al descompresor. También es posible proporcionar valores para otros campos usando una Extensión.

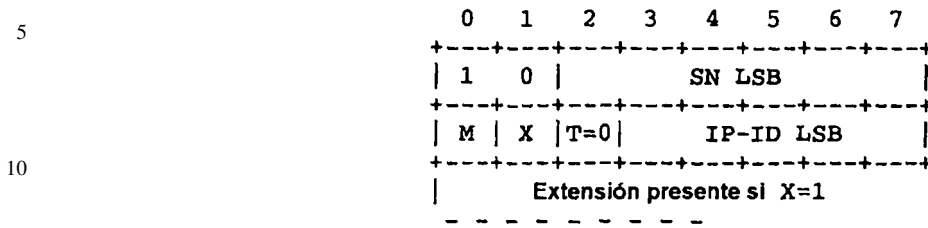
55 R-1



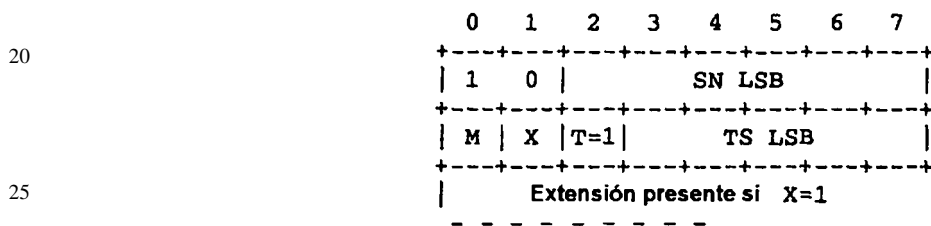
65

## ES 2 266 273 T3

### R-1-ID



### R-1-TS



LSB SN: Bits menos significativos de SN RTP, ver sección 4.5.1.

M: Bit marcador RTP, valor absoluto.

X: X=0 indica que no hay presente ninguna Extensión.

X=1 indica que hay presente un Extensión.

T: T=0 indica formato R-1-ID, T=1 indica formato R-1-TS.

LSB TS : Bits menos significativos de TS RTP escalada, ver sección 5.x.x.

LSB ID-IP: Bits menos significativos de la desviación de ID-IP desde SN, ver Sección 5.x.x.

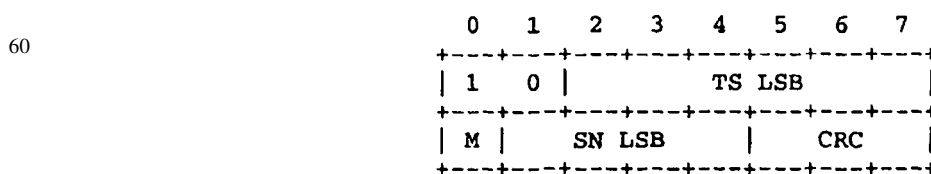
Extensión: Ver sección 5.7.5.

Nota: Los encabezamientos R-1-\* NO DEBEN actualizar ninguna parte del contexto.

#### 5.7.3. Tipo de paquete 1 (modos UO): UO-1, UO-1-ID, UO-1-TS

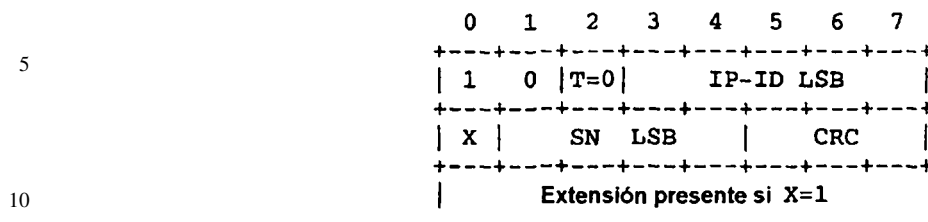
Este tipo de paquetes se usa en el modo U y en el modo O cuando el SN RTP del encabezamiento que se debe comprimir difiere tanto con respecto al contexto que el LSB RTP del tipo de paquete 0 es demasiado pequeño, o cuando las funciones desde el SN RTP a la indicación de tiempo RTP o la ID IP cambian y es necesario comunicarlas al descompresor. También es posible proporcionar valores que no sean de actualización para otros campos usando una Extensión.

### UO-1

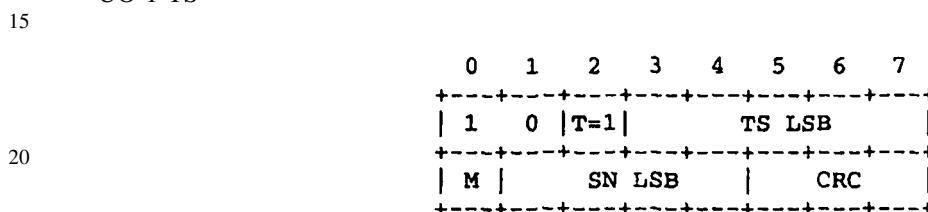


## ES 2 266 273 T3

### UO-1-ID



### UO-1-TS



LSB SN: Bits menos significativos de SN RTP, ver sección 4.5.1.

M: Bit marcador RTP, valor absoluto.

LSB TS: Bits menos significativos de TS RTP escalada, ver sección 5.x.x.

CRC: CRC de 3 bits (ver sección 5.9.2)

LSB ID-IP: Bits menos significativos de la desviación de ID-IP desde SN, ver Sección 5.x.x.

X: X=0 indica que no hay presente ninguna Extensión,

X=1 indica que hay presente un Extensión.

T: T=0 indica formato UO-1-ID, T=1 indica formato UO-1-TS.

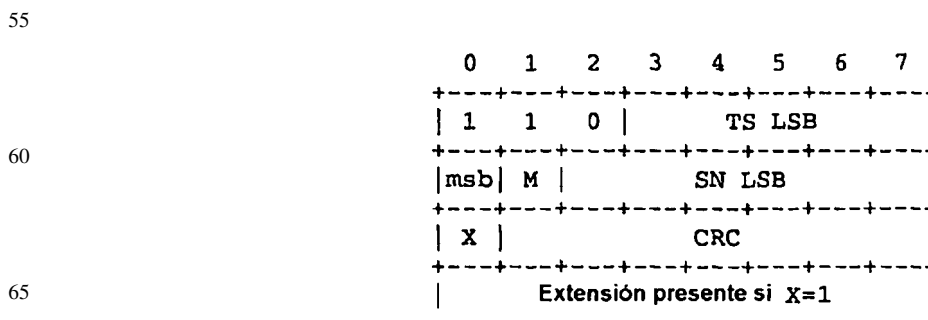
Extensión: Ver sección 5.7.5.

Nota: Los encabezamientos UO-1-\* NO DEBEN actualizar ninguna parte del contexto excepto partes relacionadas directamente SN RTP, TS RTP, e ID-IP. Los valores proporcionados en las extensiones, que no sean los correspondientes relacionados directamente con SN RTP, TS RTP, e ID-IP, NO DEBEN actualizar el contexto.

#### 5.7.4. Tipo de paquete 2: UOR-2

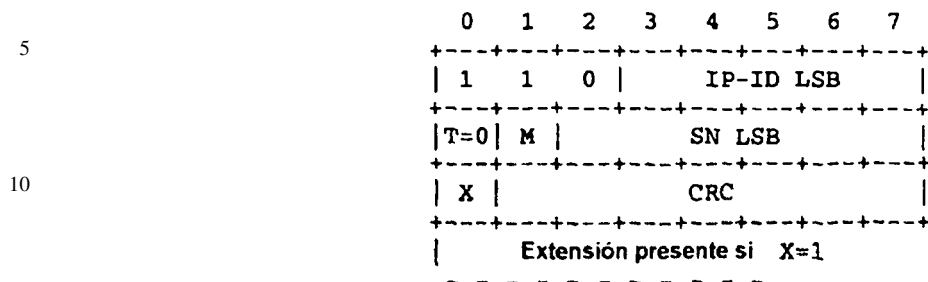
Este tipo de paquetes se usa en todos los modos cuando los cambios que requieren ser comunicados, o el SN RTP, no pueden encajar en los formatos anteriores. El formato UOR-2 siempre actualiza (o restaura) el contexto.

### UOR-2

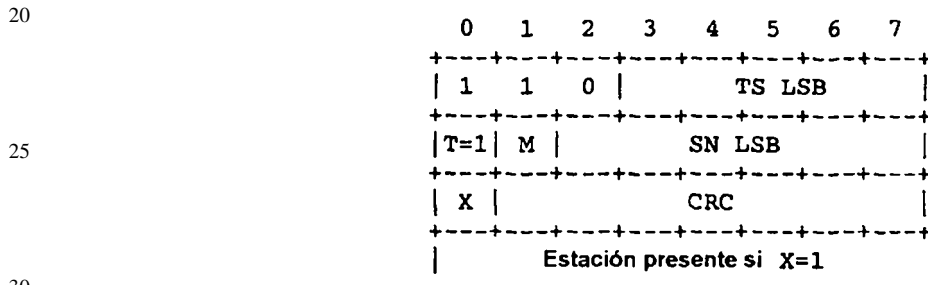


## ES 2 266 273 T3

### UOR-2-ID



### UOR-2-TS



LSB TS: Bits menos significativos de TS RTP escalada, ver sección 5.x.x.

msb: Extensión de un bit del campo LSB TS, de manera que el tamaño total de los LSB de la TS escalada es 6 bits. Este es el bit más significativo.

M: Bit marcador RTP, valor absoluto.

LSB SN: Bits menos significativos de SN RTP, ver sección 4.5.1.

LSB ID-IP: Bits menos significativos de la desviación de ID-IP desde SN, ver Sección 5.x.x.

X: X=0 indica que no hay presente ninguna Extensión,

X=1 indica que hay presente un Extensión.

CRC: CRC de 7 bits (ver sección 5.9.2)

T: T=0 indica formato UOR-2-ID, T=1 indica formato UOR-2-TS.

Extensión: Ver sección 5.7.5.

#### 5.7.5. Formatos de extensión

Los campos en las extensiones están concatenados con el campo correspondiente en el encabezamiento básico comprimido, si es que hubiera alguno. Los bits en una extensión son más significativos que los bits en el encabezamiento básico comprimido.

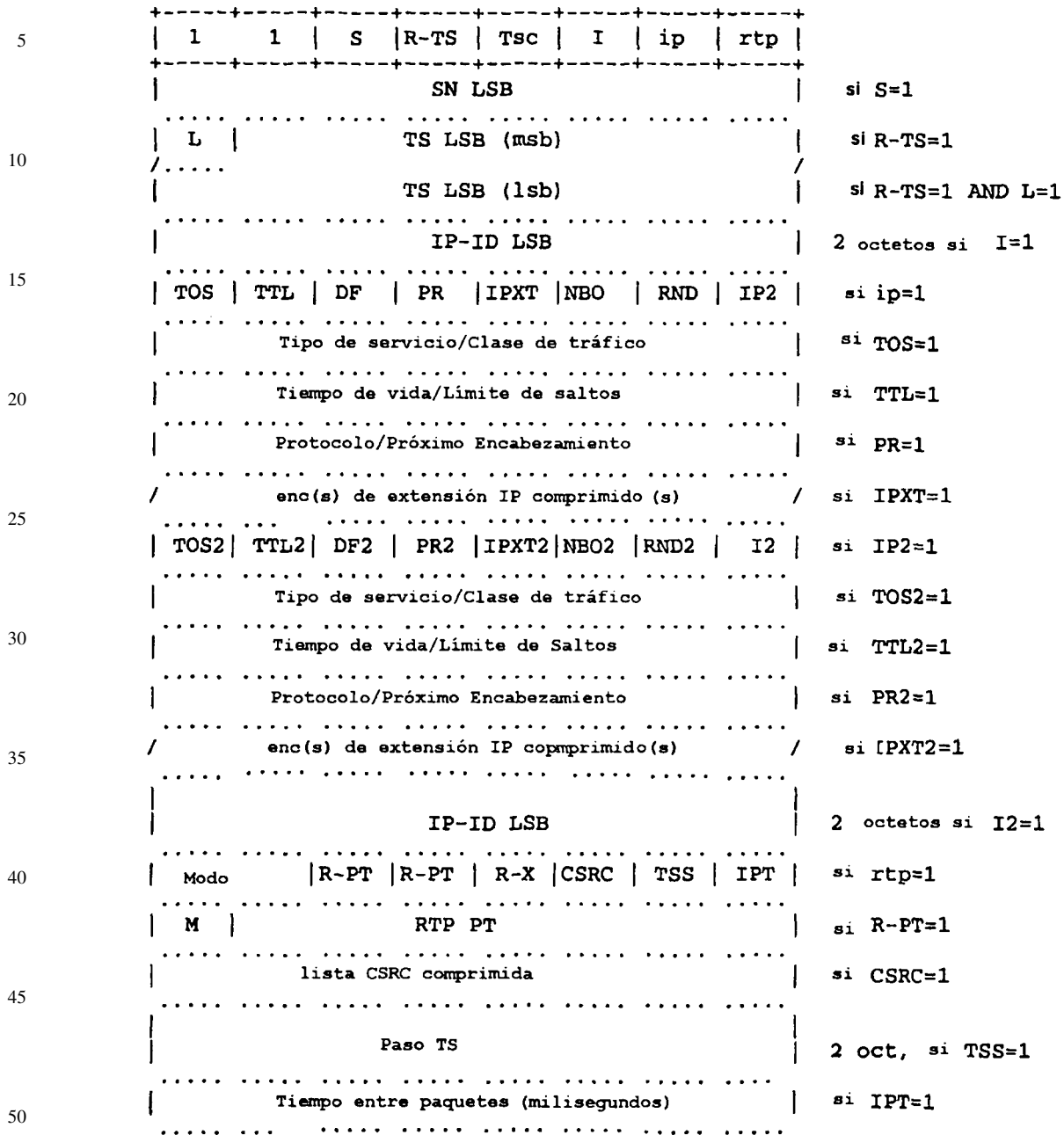
El campo LSB TS se escala en todas las extensiones, tal como lo está en el encabezamiento básico, excepto opcionalmente cuando se use la extensión 3 en la que la bandera Tsc puede indicar que el campo LSB TS RTP no está escalado. Cuando una extensión lleva un campo Paso-TS el valor de dicho campo se usa cuando se realice el escalado del campo LSB TS.

En las siguientes tres extensiones, la interpretación de los campos depende de si existe un bit T en el encabezamiento básico comprimido, y si es así, del valor de dicho campo. Si no hay ningún bit T, tanto LSB T como LSB -T significan LSB TS. Si existe un bit T,



## ES 2 266 273 T3

Extensión 3:



55 [Nota de Micke D: Este formato puede soportar dos encabezamientos IP, que es lo que se puede esperar del IP Móvil. En mi opinión, más es una exageración].

### 5.7.6. Paquetes de realimentación: REALIMENTACIÓN-3 y REALIMENTACIÓN-4

60 Cuando el tiempo de ida y vuelta entre el compresor y el descompresor es grande, puede haber varios paquetes en el trayecto. Por esta razón, el descompresor puede recibir varios paquetes antes de que el compresor pueda reaccionar a un paquete de REALIMENTACIÓN.

65 El descompresor NO DEBERÍA enviar paquetes de REALIMENTACIÓN para cada descompresión satisfactoria, descompresión no satisfactoria, o paquete recibido en un flujo de paquetes rechazados. En cambio, un compresor DEBERÍA limitar la velocidad a la cual se envían los paquetes REALIMENTACIÓN.

## ES 2 266 273 T3

Los siguientes son los formatos de los paquetes de REALIMENTACIÓN. Los mismos deben de ir precedidos de un CID que identifique a qué contexto se refieren.

Se prevé que los paquetes de REALIMENTACIÓN se pueden enviar intercalados con paquetes que no sean de REALIMENTACIÓN en un canal que vaya desde el descompresor al compresor. En tal caso, los formatos que se muestran a continuación irán precedidos de un CID con el tamaño, sea cual sea, que se haya negociado para ese canal. Si los CID de ese canal son más pequeños que los CID necesarios en los paquetes de REALIMENTACIÓN, en los formatos de REALIMENTACIÓN se DEBEN añadir bytes adicionales del CID. El número de bytes a añadir se sabe en el momento de establecimiento del canal, y no está indicado explícitamente. Si los CID del canal de REALIMENTACIÓN son más grandes que los correspondientes al canal directo, el valor numérico del CID se usa para identificar el contexto.

### REALIMENTACIÓN-3

15

20

25

30

35

40

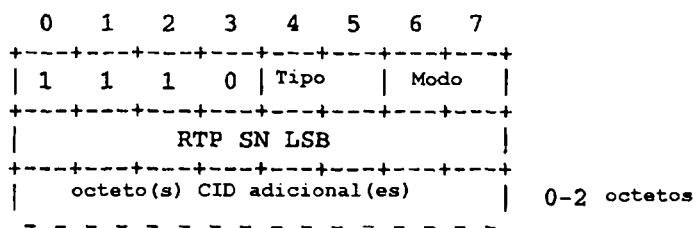
45

50

55

60

65



0-2 octetos

Tipo: 0= Reservado

1= ACK

2= NACK

3= NACK ESTÁTICO

Modo: 0= Reservado

1= Unidireccional

2= Optimista Bidireccional

3= Fiable Bidireccional

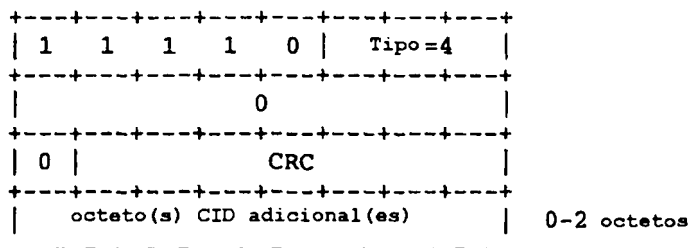
LSB SN RTP: LSB del número de Secuencia RTP de

ACK: el encabezamiento descomprimido satisfactoriamente que provocó la emisión del paquete de REALIMENTACIÓN,

NACK, NACK ESTÁTICO: el último encabezamiento descomprimido satisfactoriamente (del contexto), si es que hay alguno disponible. En cualquier otro caso cero.

Octeto(s) CID adicional(es): Octetos adicionales del CID si existe una discordancia entre los espacios CID del canal directo y el canal de realimentación.

### REALIMENTACIÓN-4-RECHAZO



0-2 octetos

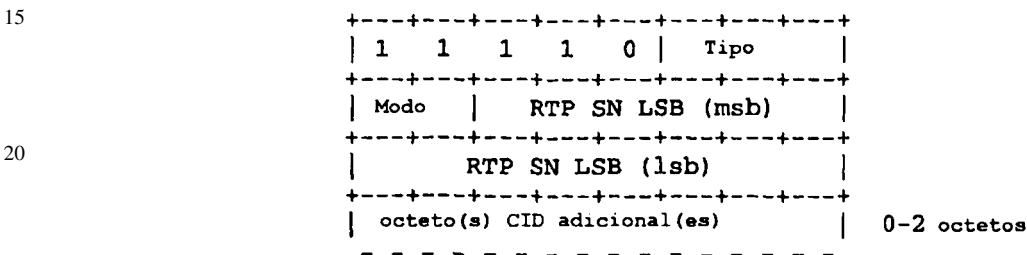
## ES 2 266 273 T3

Tipo: 4

5 CRC: CRC calculada sobre el paquete completo REALIMENTACIÓN-4-RECHAZO, incluyendo el CID, usando el polinomio para las CRC de 7 bits en la sección 5.9.2. A efectos del cálculo de CRC, el campo CRC se DEBE fijar a cero. Cuando se recibe un paquete REALIMENTACIÓN-4-RECHAZO, un compresor DEBE comprobar la CRC, y comprobar que los campos a cero del paquete REALIMENTACION-4-RECHAZO son realmente cero. Si estas comprobaciones no resultan satisfactorias, el paquete DEBE ser descartado y no se toma ninguna acción más.

10 Octeto(s) CID adicional(es): Octetos adicionales del CID si existe una discordancia entre los espacios CID del canal directo y el canal de realimentación.

### REALIMENTACIÓN-4



25 Tipo: 0, 5-7 = Reservado

1 = ACK

30 2 = NACK

3 = NACK ESTÁTICO

35 Modo: 0 = Reservado

1 = Unidireccional

2 = Optimista Bidireccional

40 3 = Fiable Bidireccional

LSB SN RTP (msb): Los 6 bits más significativos de LSB SN RTP de 14 bits.

45 LSB SN RTP (lsb): Los 8 bits menos significativos de LSB SN RTP de 14 bits.

Octeto(s) CID adicional(es): Octetos adicionales del CID si existe una discordancia entre los tamaños de los espacios CID del canal directo y el canal de realimentación.

50 Nota: El número de secuencia RTP que se debe utilizar es el mismo que para REALIMENTACIÓN-3.

#### 5.7.7 Paquetes IR e IR-DYN

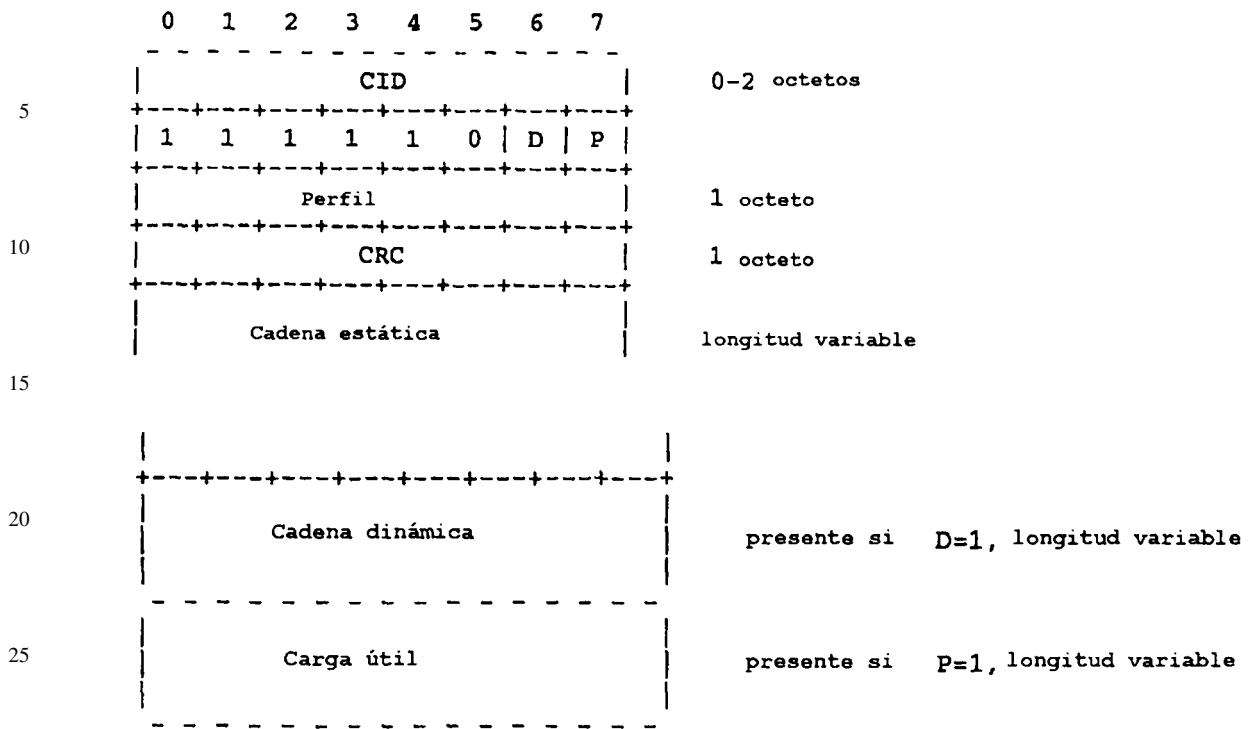
55 Los subencabezamientos que son compresibles se dividen en una parte ESTÁTICA y una parte DINÁMICA. Estas partes se pueden encontrar en las secciones 5.7.6.3-\*

La estructura de una cadena de subencabezamientos queda determinada por cada encabezamiento que tiene un campo de Siguiente Encabezamiento, o Protocolo. Este campo identifica el tipo del siguiente encabezamiento. Cada una de las siguientes partes Estáticas contiene ese campo y permite un análisis sintáctico de la cadena Estática.

##### 60 5.7.7.1 Estructura básica del paquete IR

Este tipo de paquete comunica la parte estática del contexto, es decir, el valor de las funciones SN constantes. Opcionalmente, también puede comunicar la parte dinámica del contexto, es decir, los parámetros de funciones SN no constantes. Opcionalmente también puede comunicar la carga útil de un paquete original, si es que hubiera alguna.

## ES 2 266 273 T3



D: D=1 indica que la cadena dinámica está presente.

P: P=1 indica que hay presente una carga útil (se podría deducir a partir de la cadena estática más la longitud de la trama).

Perfil: Indica transporte/aplicación de este flujo continuo.

CRC: Suma de comprobación de 8 bits que abarca el encabezamiento IR completo, incluyendo el CID, excluyendo la Carga útil, calculada según el capítulo 5.9.x.

Cadena estática: Una cadena de información estática de subencabezamiento.

Cadena dinámica: Una cadena de información dinámica de subencabezamientos. La información dinámica que está presente se deduce a partir de la cadena Estática.

Carga útil: carga útil del paquete original correspondiente, si es que hay alguna.

### 5.7.7.2. Estructura básica del paquete IR-DYN

Este tipo de paquete comunica la parte dinámica del contexto, es decir, los parámetros de funciones SN no constantes.



## ES 2 266 273 T3

### 5.7.7.4. Inicialización del Encabezamiento IPv4 [IPv4, sección 3.1]

Parte estática:

5 Versión, Banderas, Tiempo de Vida, Protocolo, Dirección de Origen, Dirección de Destino.

```

  +---+---+---+---+---+---+---+---+
  | Versión = 4 | 0 | DF | 0 |
  +---+---+---+---+---+---+---+---+
10 |           Tiempo de Vida           |
  +---+---+---+---+---+---+---+---+
  |           Protocolo                 |
  +---+---+---+---+---+---+---+---+
15 /           Dirección de Origen       /   4 octetos
  +---+---+---+---+---+---+---+---+
  /           Dirección de Destino       /   4 octetos
  +---+---+---+---+---+---+---+---+

```

20 Parte dinámica:

Tipo de Servicio, Identificación

```

25 +---+---+---+---+---+---+---+---+
  |           Tipo de servicio           |
  +---+---+---+---+---+---+---+---+
  /           Identificación             /   2 octetos
30 +---+---+---+---+---+---+---+---+

```

Eliminado:

35 IHL (debe ser 5)

Longitud Total (deducida en paquetes descomprimidos))

Bandera MF (bandera de Más Fragmentos, debe ser 0)

40 Desviación de los Fragmentos (debe ser 0)

Suma de Comprobación de Encabezamientos (deducida en paquetes descomprimidos)

45 Opciones, Relleno (no deben estar presentes)

### 5.7.7.5. Inicialización de Encabezamiento UDP [RFC-768]

Parte estática:

```

50 +---+---+---+---+---+---+---+---+
  /           Puerto de origen           /   2 octetos
  +---+---+---+---+---+---+---+---+
  /           Puerto de destino          /   2 octetos
55 +---+---+---+---+---+---+---+---+

```

Parte dinámica:

```

60 +---+---+---+---+---+---+---+---+
  /           Suma de comprobación       /   2 octetos
  +---+---+---+---+---+---+---+---+

```

65 Eliminado:

Longitud

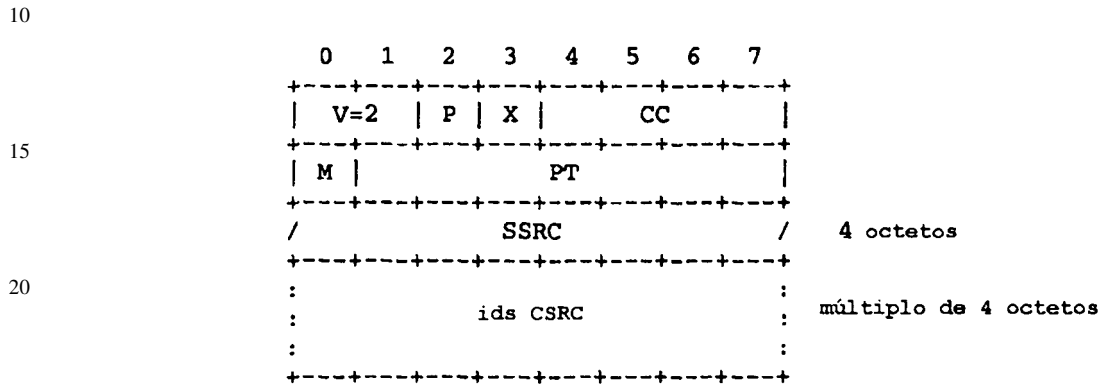
## ES 2 266 273 T3

El campo Longitud el encabezamiento UDP DEBE concordar con el(los) campo(s) de Longitud de los subencabezamientos precedentes, es decir, no debe haber ningún relleno después de la carga útil UDP que esté cubierta por la Longitud IP.

### 5 5.7.7.6. Inicialización del Encabezamiento RTP [RTP]

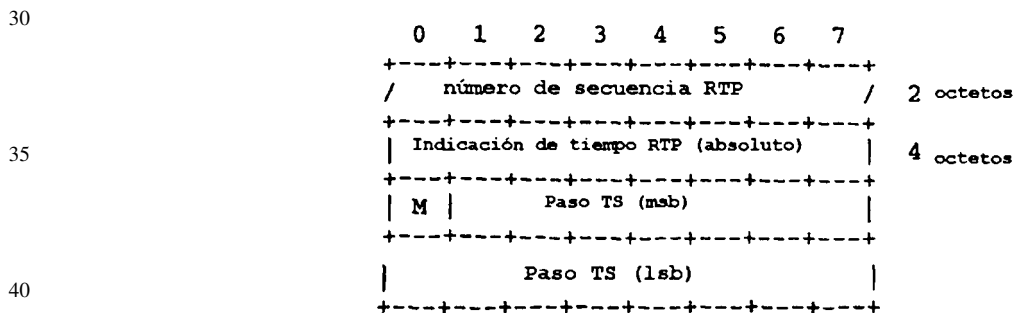
Parte estática:

Identificadores P, X, CC, PT, SSRC, CSRC



Parte dinámica:

M, número de secuencia, indicación de tiempo, diferencia delta de la indicación de tiempo.



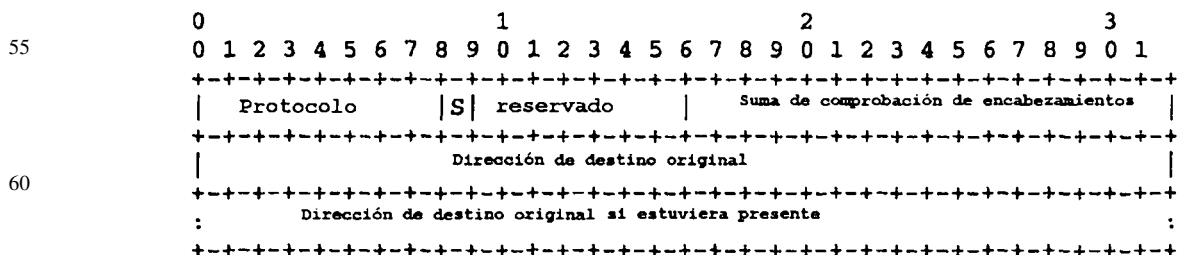
Eliminado:

Nada.

### 5.7.7.7. Encabezamiento de Encapsulación Mínima [RFC-2004, sección 3.1]

Parte ESTÁTICA:

Encabezamiento completo.



Parte DINÁMICA:

Vacía.

Eliminado:

Nada.

5 5.8. *Compresión Basada en Listas*

En dos casos, la información del paquete que debe comprimirse se puede describir como una lista ordenada, es decir, gran parte constante entre los paquetes, pero en la cual se realizan ocasionalmente adiciones y eliminaciones en el transcurso del flujo continuo de paquetes. Esta sección describe el esquema de compresión para esta información, denominado “compresión basada en listas”. Los dos casos son: listas CSRC en paquetes RTP, y cadenas de encabezamientos de extensión en paquetes IP.

[[Nota del editor: Esta sección está siendo todavía discutida de forma activa en la lista de correo. Se cree que pronto aparecerá una versión algo más simplificada]].

15 5.8.1. *Compresión CSRC*

La Listas de Fuentes Contribuyentes (CSRC) en un encabezamiento RTP contiene los identificadores de las Fuentes de Sincronización (SSRC) de las fuentes contribuyentes para la carga útil en el paquete actual.

Una lista CSRC contiene como mucho 15 identificadores, debido al tamaño de 4 bits del campo Cómputo CSRC (CC) en el encabezamiento RTP. Cada identificador de 32 bits es seleccionado aleatoriamente por la fuente de sincronización original de manera que es globalmente exclusivo dentro de una sesión RTP.

25 El esquema de compresión introducido en el presente caso hará uso de los hechos mencionados anteriormente. Para mantener la transparencia, se conserva el orden de los identificadores durante la compresión. En otras palabras, la lista CSRC se comprime realmente como una lista, no como un conjunto.

30 5.8.1.1. *Clasificación de Transformaciones para la Lista CSRC*

Una lista CSRC determinada (lista\_act) se puede clasificar como perteneciente a uno de los siguientes casos de transformación cuando se compara con una lista CSRC de referencia (lista\_ref).

- 35 - Caso de Transformación A: la lista\_act puede obtener a partir de la lista\_ref simplemente añadiendo algunas CSRC; las posiciones relativas de las CSRC comunes a la lista\_act y la lista\_ref son las mismas.
- Caso de Transformación B: la lista\_act se puede obtener a partir de la lista\_ref simplemente eliminando algunas CSRC; las posiciones relativas de las CSRC comunes a la lista\_act y la lista\_ref son las mismas.
- 40 - Caso de Transformación C: la totalidad del resto de casos de transformación que no quedan cubiertos por el caso transformación A y B.

5.8.1.2. *Esquemas de Codificación*

45 Para afrontar los 3 casos de transformación mencionados anteriormente, se usan cuatro esquemas de codificación. Cada esquema afronta uno o más de los casos de transformación mencionados anteriormente. Los cuatro esquemas de codificación y los casos de transformación que afrontan se enumeran en la siguiente tabla.

50

Esquema de Codificación	Caso de Transformación
Solamente Inserción	A
Solamente Eliminación	B
Genérico	mezcla de A, B, C
Sin Comprimir	mezcla de A, B, C

60

- 65 - En el esquema de solamente inserción, en comparación con la lista CSRC en la lista\_ref, las CSRC recién añadidas en la lista\_act se envían junto con las posiciones de las CSRC en la lista\_ref, antes de lo cual se insertarán las CSRC nuevas.

## ES 2 266 273 T3

- En el esquema de solamente eliminación, se envían las posiciones de las CSRC, las cuales están en la lista\_ref, aunque no en la lista\_act.
- En el esquema genérico, para una CSRC determinada en la lista\_act, la misma se envía comprimida solamente con un campo de posición si la CSRC está también en la lista\_ref, o se envía sin comprimir.

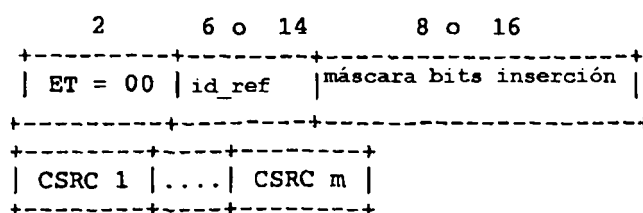
La totalidad de los 3 esquemas mencionados anteriormente generan un formato comprimido de la lista CSRC. La lista CSRC también se puede enviar en un formato no comprimido.

### 5.8.1.3. Formato de la lista CSRC comprimida

El formato de la lista CSRC comprimida usando los cuatro esquemas es el siguiente.

#### 5.8.1.3.1 Esquema de Solamente Inserción

##### 5.8.1.3.1.1 Modo R



\* id\_ref - los LSB del número de secuencia RTP de la lista\_ref

6 bits: "0" + LSB de 5 bits del número de secuencia RTP

14 bits: "1" + LSB de 13 bits del número de secuencia RTP

\* máscara de bits de inserción: la máscara de bits que indica la posición de las siguientes CSRC que se deben insertar en la lista\_ref para reconstruir la lista\_act.

La longitud de la máscara de bits de inserción puede ser 8 bits ó 16 bits.

8 bits: "0" + máscara de bits de inserción de 7 bits

16 bits: "1" + máscara de bits de inserción de 15 bits

Para construir la máscara de bits de inserción y la siguiente lista de CSRC insertadas, se realizan las siguientes etapas.

\*\* Como punto de partida se generan una lista de "0" y una lista de CSRC insertadas vacía. El número de elementos "0" en la lista de "0" es igual al número de CSRC en la lista\_ref. El "0" iésimo en la lista de "0" se corresponde con la CSRC iésima en la lista\_ref.

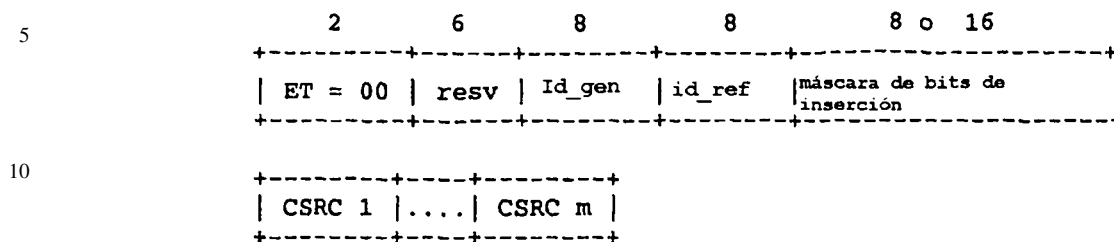
\*\* Comparando la lista\_act con la lista\_ref, si entre el elemento iésimo y el elemento (i+1)ésimo en la lista\_ref se inserta una CSRC nueva, entre el "0" iésimo y el "0" (i+1)ésimo en la lista original de "0" se inserta un "1". La CSRC nueva se debería añadir al final de la lista de CSRC insertadas. Este procedimiento se repite hasta que se han procesado la totalidad de las m CSRC nuevas. Si la longitud de la máscara de bits nueva es inferior a 7 bits ó 15 bits, se debería añadir un "0" adicional al final hasta que alcance los 7 bits ó 15 bits.

Cuando el descompresor recibe la máscara de bits de inserción, realiza una exploración de izquierda a derecha. Cuando se observa un "0", el descompresor copia la CSRC correspondiente de la lista\_ref a la lista\_act; cuando se observa un "1", el descompresor añade la CSRC correspondiente a la lista\_act.

\* CSRC i (i=1..m): las CSRC que se deben insertar; suponiendo que el número de CSRC a insertar es m.

## ES 2 266 273 T3

### 5.8.1.3.1.2 Modos UO



15 \*resv: reservado

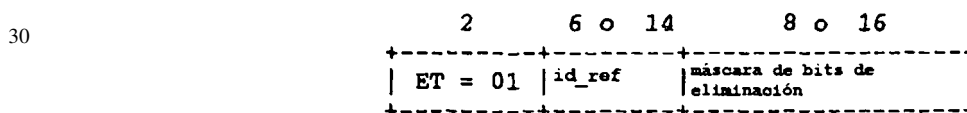
\* id\_Gen: se usa para identificar un conjunto de paquetes que pertenece a la misma generación

20 \* id\_ref: la id\_Gen transportada en la lista\_ref

\* máscara de bits de inserción y CSRC i: igual que lo definido para el modo R

### 5.8.1.3.2 Esquema de Solamente Eliminación

#### 5.8.1.3.2.1 Modo R



35 \* id\_ref: lo mismo que lo definido en la sección 3.1.1.

\* máscara de bits de eliminación: la máscara de bits que indica la CSRC en la lista\_ref que se debe eliminar para reconstruir la lista\_act. Un "1" en el bit iésimo en la máscara de bits de eliminación significa que la CSRC iésima en la lista\_ref no está en la lista\_act, mientras que un "0" significa que todavía está presente en la lista\_act.

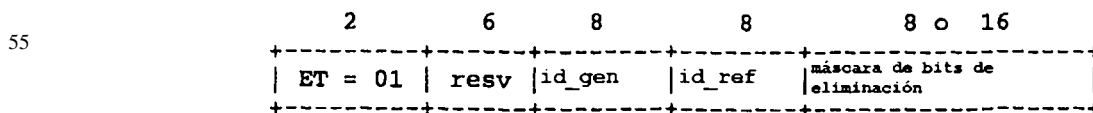
La longitud de la máscara de bits de inserción puede ser 8 bits ó 16 bits.

8 bits: "0" + máscara de bits de inserción de 7 bits

16 bits: "1" + máscara de bits de inserción de 15 bits

El formato que se debe utilizar depende del número de CSRC en la lista\_ref. Si es inferior a 8, en ese caso se puede usar la máscara de bits de eliminación de 8 bits; en cualquier otro caso es necesario el formato de 16 bits.

#### 5.8.1.3.2.2 Modo UO



60 \* Id\_gen, resv e id\_ref: lo mismo que lo definido en la sección 3.1.2.

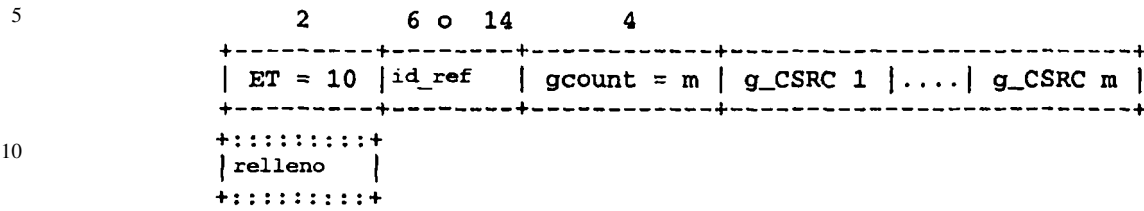
\* máscara de bits de eliminación: lo mismo que lo definido para el modo R

65

# ES 2 266 273 T3

## 5.8.1.3.3 Esquema Genérico

### 5.8.1.3.3.1 Modo R



\* id\_ref: lo mismo que lo definido en la sección 3.1.1.

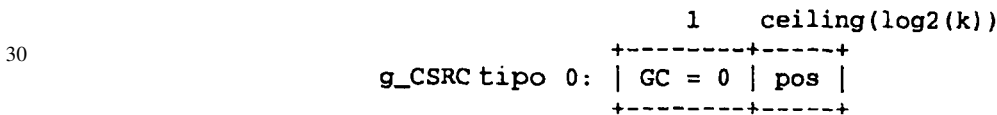
\* gcount: el número del siguiente campo g\_CSRC; la longitud es 4 bits.

\* g\_CSRC i (i=1..m): se corresponde con la CSRC iésima en la lista\_act. El orden de g\_CSRC representa el orden de las CSRC en la lista\_act. Se definen dos tipos g\_CSRC.

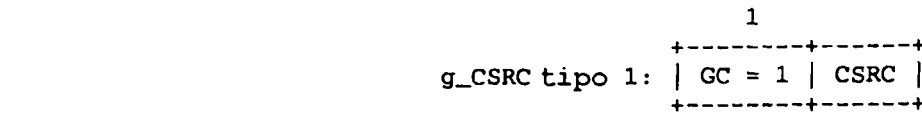
\*\* el tipo de g\_CSRC 0 se usa para comprimir una CSRC que se encuentra tanto en la lista\_ref como en la lista\_act.

\*\* el tipo de g\_CSRC 1 se usa para transportar una CSRC que se encuentra en la lista\_act aunque no en la lista\_ref.

El formato de g\_CSRC se define de la manera siguiente.

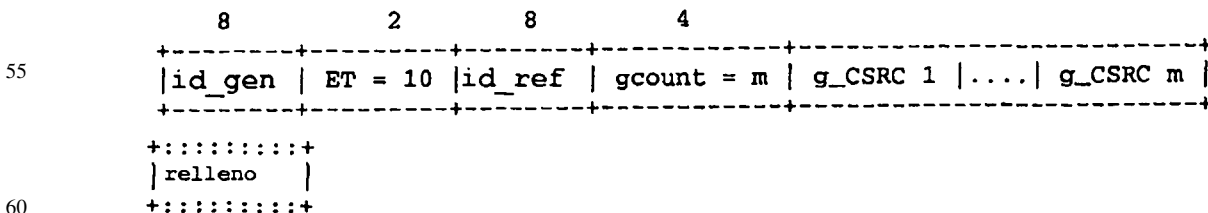


- pos: la posición de una CSRC en la lista\_ref; la longitud es ceiling(log2(k)) en la que k es el número de CSRC en la lista\_ref. Como el valor de k es conocido tanto para el compresor como para el descompresor, no es necesario que la longitud del campo pos sea transportada en la lista comprimida



\* relleno: puede que sean necesarios bits de relleno para mantener la alineación de los bits

### 5.8.1.3.3.2 Modo UO

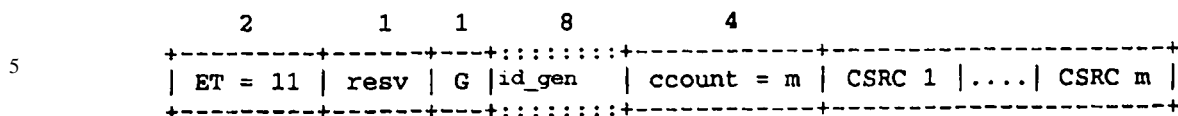


\* Id\_Gen e id\_ref: lo mismo que lo definido en la sección 3.2.1.2.

\* gcount, g\_CRSC y relleno: lo mismo que lo definido para el modo R

## ES 2 266 273 T3

### 3.4 Sin compresión (común para el modo R y los modos UO)



- 10 \* resv: reservado
- \* G: indicando la presencia del campo id\_gen
- \* Id\_gen: lo mismo que lo definido en la sección 3.1.2.
- 15 - En el modo R, id\_gen no está presente.
- Durante la transición desde cualquier modo al modo U o al modo O, no se debería enviar la id\_gen. La lista sin id\_gen no se debería usar como referencia para comprimir y descomprimir una lista CSRC.
- 20 - En los modos UO, id\_gen está presente si la misma se puede usar como referencia para comprimir o descomprimir la lista CSRC subsiguiente.
- \* ccount: el número de CSRC en la lista CSRC
- 25 \* CSRC i (i=1..m): la lista CSRC original en la lista\_act

#### 5.8.2. Compresión de Encabezamientos para Encabezamientos de Extensión IPv6

30 Los encabezamientos de extensión IPv6 se codifican en forma de una lista de elementos. Cada elemento es uno de los encabezamientos de extensión. La longitud de cada encabezamiento de extensión puede variar de uno a otro. Cuando se usa más de un encabezamiento de extensión en el mismo paquete, el orden de estos encabezamientos de extensión se recomienda en el RFC 2460, aunque no de forma obligatoria. Por lo tanto, aunque no es probable que ocurra, el orden de los encabezamientos de extensión puede variar durante la misma sesión. Adicionalmente, se pueden añadir o eliminar uno o más encabezamientos de extensión durante la sesión y el contenido de cada encabezamiento de extensión puede variar. Por esta razón, los encabezamientos de extensión IPv6 se clasifican como una lista de elementos y se puede aplicar el mecanismo de compresión de lista de elementos.

35

40 La compresión de encabezamientos de extensión IPv6 al nivel de las listas es similar a la correspondiente a las entradas CSRC (sección 5.8.1). Al valor comprimido de la lista de encabezamientos de extensión se le hace referencia como lista comprimida de encabezamientos de extensión. La compresión de los encabezamientos de extensión IPv6 al nivel de los elementos, es decir, el esquema de compresión usado para cada tipo de encabezamiento de extensión, se define en la presente subsección. El encabezamiento de extensión de referencia usado para comprimir un encabezamiento de extensión determinado es el encabezamiento de extensión de la lista de referencia que tiene el mismo tipo.

45 Al valor comprimido de un encabezamiento de extensión se le hace referencia como encabezamiento de extensión comprimido.

##### 5.8.2.1. Terminología

- 50 \* lista de encabezamientos de extensión: la lista de encabezamientos de extensión IPv6
- \* encabezamiento de extensión comprimido: el valor comprimido de un encabezamiento de extensión IPv6
- \* lista comprimida de encabezamientos de extensión: la lista de encabezamientos de extensión comprimidos
- 55 \* lista de encabezamientos de extensión de referencia: la lista de encabezamientos de extensión que se usa como referencia para comprimir una lista de encabezamientos de extensión
- \* encabezamientos de extensión de referencia: un encabezamiento de extensión en la lista de referencia que presenta el mismo tipo y se usa como la referencia para comprimir un encabezamiento de extensión determinado
- 60

##### 5.8.2.2. Clasificación de Transformaciones y Esquemas de Codificación

###### 5.8.2.2.1 Clasificación de Transformaciones

65 Una lista determinada de encabezamientos de extensión (lista\_act) se puede clasificar como perteneciente a uno de los siguientes casos de transformación cuando se compara con una lista de encabezamientos de extensión de referencia (lista\_ref).

## ES 2 266 273 T3

- Caso de Transformación A: la lista\_act se puede obtener a partir de la lista\_ref simplemente añadiendo (insertando) algunos encabezamientos de extensión; las posiciones relativas de los encabezamientos de extensión comunes a la lista\_act y la lista\_ref son las mismas.
- 5 - Caso de Transformación B: la lista\_act se puede obtener a partir de la lista\_ref simplemente eliminando algunos encabezamientos de extensión; las posiciones relativas de los encabezamientos de extensión comunes a la lista\_act y la lista\_ref son las mismas.
- 10 - Caso de Transformación C: la lista\_act se puede obtener a partir de la lista\_ref simplemente modificando el contenido de algunos encabezamientos de extensión; las posiciones relativas de los encabezamientos de extensión siguen siendo las mismas.
- Caso de Transformación D: la totalidad del resto de casos de transformación que no quedan cubiertos por el caso de transformación A, B y C.

### 5.8.2.2.2 Esquemas de Codificación

Para afrontar los 4 casos de transformación mencionados anteriormente, se usan cuatro esquemas de codificación. Cada esquema afronta uno más de los casos de transformación mencionados anteriormente. Los cuatro esquemas de codificación y los casos de transformación que afrontan se enumeran en la siguiente tabla.

Esquema de Codificación	Valor ET	Caso de Transformación
Solamente Inserción	00	A
Solamente Eliminación	01	B
Sol. Cambio Contenido	10	C
Sin comprimir	11	mezcla de A,B,C,D

\* En el esquema de solamente inserción, en comparación con la lista\_ref, el valor no comprimido de los encabezamientos de extensión recién añadidos en la lista\_act se envía junto con las posiciones de los encabezamientos de extensión en la lista\_ref, antes de lo cual se insertarán los encabezamientos de extensión nuevos.

\* En el esquema de solamente eliminación, se envían las posiciones de los encabezamientos de extensión, que están en la lista\_ref aunque no en la lista\_act.

\* En el esquema de solamente cambio, se envían las posiciones del encabezamiento de extensión cuyo contenido cambia así como su valor no comprimido o comprimido. (Obsérvese que en la fase actual, se usa solamente un encabezamiento de extensión no comprimido).

Los 3 esquemas mencionados anteriormente generan un formato comprimido de la lista de encabezamientos de extensión. La lista\_act también se puede enviar en un formato no comprimido.

### 5.8.2.2.3 Tratamiento Especial

#### 5.8.2.2.3.1 Tratamiento Especial del AH

El campo de número de secuencia en el AH contiene un valor contador creciente de forma monótona correspondiente a una asociación de seguridad. Por esta razón, cuando se compara la lista\_act con la lista\_ref, si el número de secuencia en el AH cambia y el campo SPI no cambia, no es necesario clasificar el AH como cambiado.

Si el número de secuencia en el AH aumenta linealmente a medida que aumenta el número de secuencia RTP, no es necesario enviarlo. El descompresor aplica una extrapolación lineal para reconstruir el número de secuencia en el AH. De otro modo, se debería incluir un número de secuencia comprimido en el campo de elemento de compresión de encabezamientos de extensión IPv6 en el Encabezamiento "11" de Extensión PT2.

El campo de datos de autenticación en el AH cambia de un paquete a otro y se debería enviar en todos los paquetes. Si se envía el AH no comprimido, el campo de datos de autenticación se envía dentro del AH no comprimido; en cualquier otro caso, se envía después de los encabezamientos de extensión comprimidos IP/UDP/RTP e IPv6 y antes de la carga útil.

5.8.2.2.3.2 Encabezamiento de Carga Útil de Seguridad de Encapsulación

Si se usa el Encabezamiento de Carga Útil de Seguridad de Encapsulación (ESP), los encabezamientos UDP y RTP están ambos cifrados y no se pueden comprimir. En este caso, es necesario definir un formato de paquete comprimido especial en la ROHC.

En el ESP, los únicos campos que se pueden comprimir son el SPI y el número de secuencia.

\* En el caso de que el campo SPI cambie, se envía el ESP no comprimido.

\* En el caso de que no se produzca ningún en el campo SPI, el ESP no se considera como cambiado.

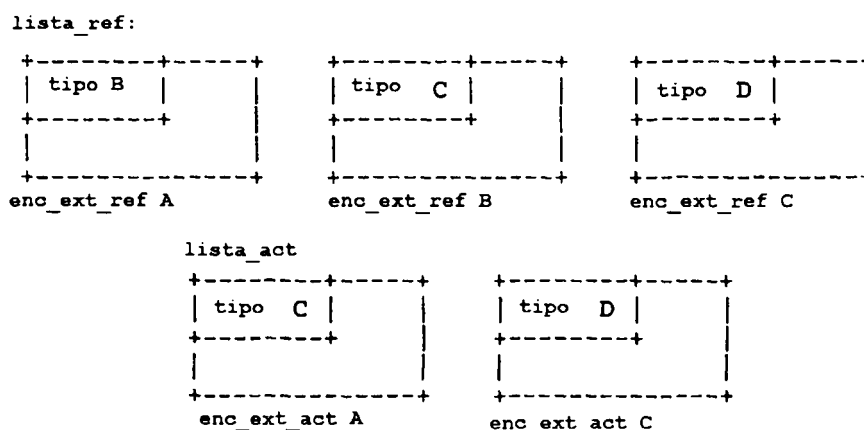
El número de secuencia en ESP presenta el mismo comportamiento que el mismo campo que en el AH. Si el mismo se incrementa linealmente, no es necesario enviarlo. En cualquier otro caso, se debería enviar un número de secuencia comprimido en el campo de elemento de compresión de Encabezamientos de Extensión IPv6 en el encabezamiento "11" de extensión PT2.

5.8.2.2.3.3 Tratamiento Especial del Campo de Siguiete Encabezamiento

El campo de siguiete encabezamiento en un encabezamiento de extensión cambia siempre que cambia el tipo del encabezamiento que viene inmediatamente a continuación, por ejemplo, se inserta después del mismo un encabezamiento de extensión nuevo, se elimina de la lista el encabezamiento de extensión subsiguiente inmediato, o se cambia el orden de varios encabezamientos de extensión. De este modo, en particular, puede que no sea extraño que para un encabezamiento de extensión determinado, únicamente cambie el campo de siguiete encabezamiento aunque no cambien los campos restantes. Por esta razón, es necesario tratar de una manera especial el campo de siguiete encabezamiento en cada encabezamiento de extensión.

En el caso de que únicamente cambie el campo de siguiete encabezamiento, el encabezamiento de extensión se debería considerar como no cambiado. El tratamiento especial del cambio del campo de siguiete encabezamiento se define de la manera siguiente.

\* En el caso de que se elimine de la lista un encabezamiento de extensión subsiguiente, el valor nuevo del campo de siguiete encabezamiento se puede obtener a partir de la lista de encabezamientos de extensión de referencia. Por ejemplo, supóngase que la lista de encabezamientos de extensión de referencia (lista\_ref) consta del encabezamiento de extensión A, B y C (enc\_ext\_ref A, B, C), y la lista de encabezamientos de extensión actual (lista\_act) solamente consta de los encabezamientos de extensión A y C (enc\_ext\_act A, C). El orden y el valor del campo de siguiete encabezamiento de dichos encabezamientos de extensión son los siguientes.



Comparando el enc\_ext\_act A de la lista\_act y el enc\_ext\_ref A de la lista\_ref, el valor del campo de siguiete encabezamiento se cambia del "tipo B" al "tipo C" debido a la eliminación del campo de extensión B.

No es necesario enviar al descompresor el valor nuevo del campo de siguiete encabezamiento en enc\_ext\_act A, es decir, "tipo C", ya que cuando el descompresor detecta (observando la codificación a nivel de listas) que se elimina de la lista el encabezamiento de extensión siguiete inmediato B, recupera el campo de siguiete encabezamiento en enc\_ext\_ref B y lo usa para sustituir el campo de siguiete encabezamiento en el enc\_ext\_act A.

\* En el caso de que se inserte un encabezamiento de extensión nuevo después de un encabezamiento de extensión existente, el campo de siguiete encabezamiento en el encabezamiento de extensión nuevo transporta su propio tipo, en lugar del tipo del encabezamiento de extensión que viene a continuación. Por ejemplo, supóngase que la lista de encabezamientos de extensión de referencia (lista\_ref) consta del encabezamiento de extensión A y C (enc\_ext\_ref A,

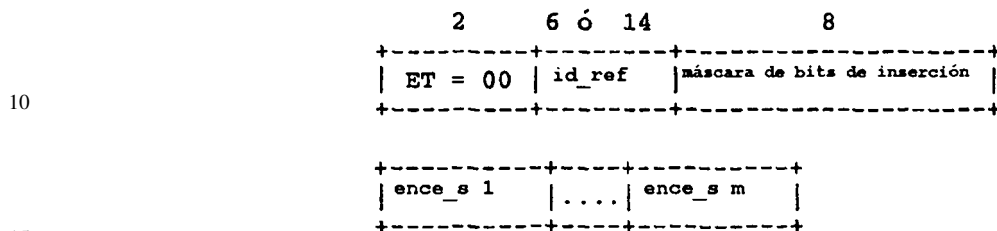


## ES 2 266 273 T3

### 5.8.2.3.2 Formato del encabezamiento de extensión IPv6 comprimido

#### 5.8.2.3.2.1 Esquema de Solamente Inserción

##### 5.8.2.3.2.1.1 Modo R



\*id\_ref - los LSB de número de secuencia RTP de la lista\_ref

6 bits: "0" + LSB de 5 bits

14 bits: "1" + LSB de 13 bits

\* ence\_s i (i=1..m): el valor sin comprimir del encabezamiento de extensión nuevo en la lista\_act; m es el número de encabezamientos de extensión nuevos que se debe añadir.

\* máscara de bits de inserción: la máscara de bits que indica la posición del siguiente ence\_s a insertar en la lista\_ref para reconstruir la lista\_act.

La longitud de la máscara de bits de inserción es 8 bits.

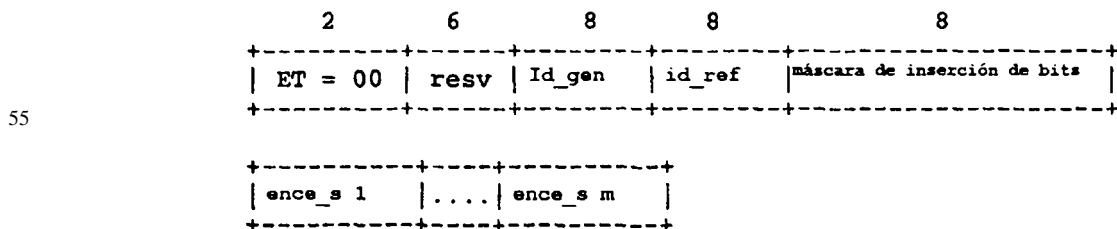
Para construir la máscara de bits de inserción y la sucesiva lista de ence\_s insertados, se realizan las siguientes etapas.

\*\* Como punto de partida se generan una lista de "0" y una lista vacía de ence\_s insertados. El número de elementos "0" en la lista de "0" es igual al número de encabezamientos de extensión en la lista\_ref. El "0" iésimo en la lista de "0" se corresponde con el encabezamiento de extensión iésimo en la lista\_ref.

\*\* Comparando la lista\_act con la lista\_ref, si se inserta un encabezamiento de extensión nuevo entre el elemento iésimo y el elemento (i+1)ésimo en la lista\_ref, se inserta un "1" entre el "0" iésimo y el "0" (i+1)ésimo en la lista de "0" original. El encabezamiento de extensión nuevo se debería añadir al final de la lista de CSRC insertadas. Este procedimiento se repite hasta que se han procesado la totalidad de los m encabezamientos de extensión nuevos. Si la longitud de la máscara de bits de inserción nueva es inferior a 7 bits, se debería añadir un "0" adicional al final hasta que se alcancen los 7 bits.

Cuando el descompresor recibe la máscara de bits de inserción, realiza una exploración de izquierda a derecha. Cuando se observa un "0", el descompresor copia el encabezamiento de extensión correspondiente de la lista\_ref a la lista\_act; cuando se observa un "1", el descompresor añade el ence\_s correspondiente a la lista\_act.

#### 3.2.1.2 Modos UO



\* resv: reservado

\* id\_Gen: se usa para identificar un conjunto de paquetes que pertenece a la misma generación

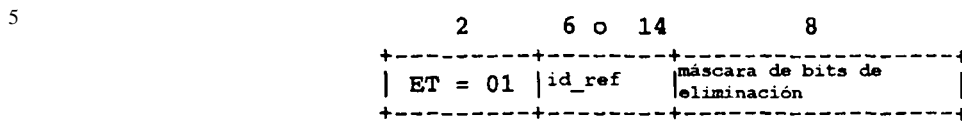
\* id\_ref: la id\_Gen transportada en la lista\_ref

\* máscara de bits de inserción y ence\_s i: lo mismo que lo definido para el modo R

## ES 2 266 273 T3

### 5.8.2.3.2.2 Esquema de Solamente Eliminación

#### 5.8.2.3.2.2.1 Modo R

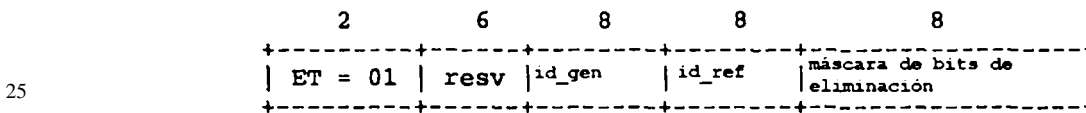


10 \* id\_ref: lo mismo que lo definido en la sección 3.2.1.1.

15 \* máscara de bits de eliminación: la máscara de bits que indica el encabezamiento de extensión en la lista\_ref que se debe eliminar para reconstruir la lista\_act. Un "1" en el bit iésimo en la máscara de bits de eliminación significa que el encabezamiento de extensión iésimo en la lista\_ref no se encuentra en la lista\_act, mientras que un "0" significa que todavía está presente en la lista\_act.

La longitud de la máscara de bits de inserción puede ser 8 bits.

#### 20 3.2.2.2 Modo UO

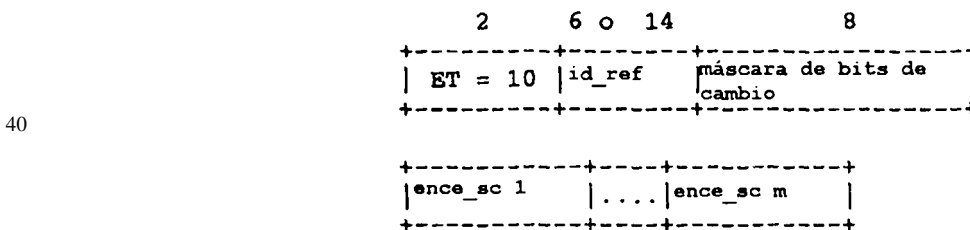


\* id\_Gen, resv e id\_ref: lo mismo que lo definido en la sección 3.2.1.2.

30 \* máscara de bits de eliminación: lo mismo que lo definido para el modo R

### 5.8.2.3.2.3 Esquema de Solamente Cambio del Contenido

#### 5.8.2.3.2.3.1 Modo R

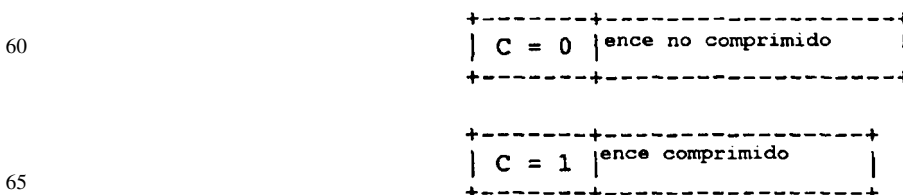


45 \* id\_ref: lo mismo que lo definido en la sección 2.3.1.1

50 \* máscara de bits de cambio: una máscara de bits que indica la posición del encabezamiento de extensión que se cambia. Un "1" en el bit iésimo en la máscara de bits de cambio significa que el encabezamiento de extensión iésimo en la lista\_ref no es el mismo que la extensión iésima en la lista\_act, mientras que un "0" significa que son los mismos.

55 \* ence\_sc: se corresponde con el encabezamiento de extensión cuyo contenido cambia cuando se compara con el encabezamiento de extensión en la misma posición en la lista\_ref. Sus posiciones en la lista\_act se indican en el campo de máscara de bits de cambio.

ence\_sc:

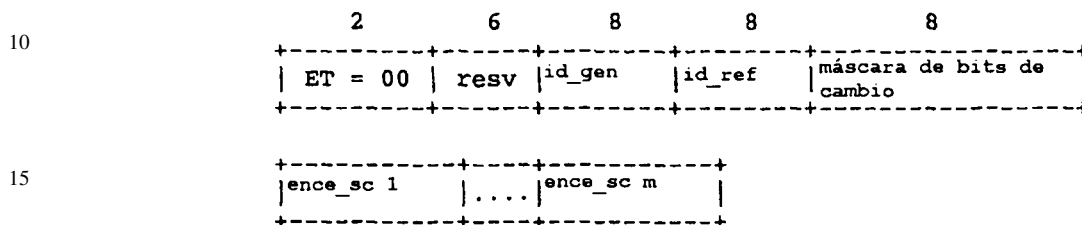


## ES 2 266 273 T3

- ence no comprimido: encabezamiento de extensión no comprimido

- ence comprimido: encabezamiento de extensión comprimido usando como referencia el encabezamiento de extensión en la misma posición en la lista\_ref. Los mecanismos de compresión para los tipos diferentes de encabezamientos de extensión son diferentes entre ellos y el FFS.

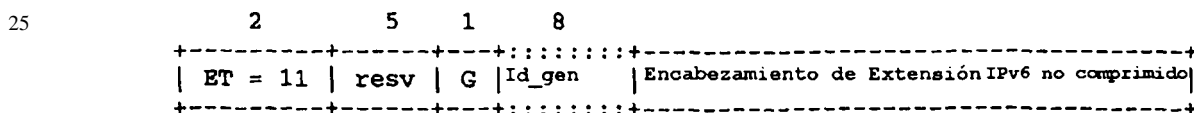
### 5.8.2.3.2.3.2 Modo UO



\* id\_Gen, resv, id\_ref: lo mismo que lo definido en la sección 3.2.1.2.

\* máscara de bits de cambio y ence\_sc i: lo mismo que lo definido para el modo R

### 5.8.2.3.2.4 Esquema No Comprimido (común para el modo R y los modos UO)



\* resv: reservado

\* G: indica la presencia de id\_Gen

\* id\_Gen: lo mismo que lo definido en la sección 3.2.1.2.

- En el modo R, id\_gen no está presente.

- Durante la transición desde cualquier modo al modo U o el modo O, no se debería enviar la id\_gen. La lista sin id\_gen no se debería usar como referencia para comprimir y descomprimir una lista de encabezamientos de extensión.

- En los modos UO, la id\_gen no está presente si la misma se puede usar como referencia para comprimir y descomprimir la subsiguiente lista de encabezamientos de extensión.

## 5.9. Comprobaciones CRC de compresión de encabezamientos, alcance y polinomios

Este capítulo describe cómo calcular las CRC usadas en encabezamientos de paquetes definidos en el presente documento.

### 5.9.1. Comprobaciones CRC de paquetes IR e IR-DYN

La CRC en el paquete IR e IR-DYN se calcula sobre el paquete completo IR o IR-DYN, excluyendo la Carga Útil e incluyendo el CID. Con el fin de calcular la CRC, el campo CRC en el encabezamiento se fija a cero.

El contenido inicial del registro CRC se debe fijar previamente todo a 1.

El polinomio CRC que se debe utilizar es:

$$C(x) = 1 + x + x^2 + x^8$$

### 5.9.2. Comprobaciones CRC en paquetes comprimidos

La CRC en paquetes comprimidos se calcula sobre el encabezamiento original completo, antes de la compresión:

[[Nota del editor: En la reunión del WG del Pittsburgh, dijimos que la tara del cálculo de la CRC se puede reducir calculando la CRC en primer lugar sobre las partes estáticas del paquete y a continuación sobre las partes dinámicas. Esto requiere ser descrito detalladamente en el presente documento]].

## ES 2 266 273 T3

El polinomio que se debe utilizar para la CRC de 3 bits es:

$$C(x) = 1 + x + x^3$$

El polinomio que se debe utilizar para la CRC de 7 bits es:

$$C(x) = ??? \text{ [debe definirse]}$$

### 6. Aspectos de la implementación

El presente documento especifica mecanismos para el protocolo, aunque se deja que los desarrolladores tomen decisiones sobre gran parte del uso de estos mecanismos. Este capítulo está destinado a proporcionar pautas, ideas y sugerencias para implementar el esquema.

#### 6.1. Descompresión inversa

Este capítulo describe un funcionamiento opcional del descompresor para reducir paquetes descartados debido a un contexto no válido.

Una vez que un contexto se convierte en no válido (por ejemplo, en el caso de que se hayan producido más pérdidas de paquetes consecutivos de lo esperado), los paquetes comprimidos subsiguientes no se pueden descomprimir inmediatamente de forma correcta. La descompresión inversa tiene como objetivo descomprimir dichos paquetes posteriormente en lugar de descartarlos, almacenándolos hasta que contexto haya sido actualizado y validado y a continuación intentando la descompresión.

Sea la secuencia de paquetes almacenados  $i, i+1, \dots, i+k$ , en la que  $i$  es el primer paquete e  $i+k$  es el paquete antes de que se actualizara el contexto. El descompresor intentará recuperar los paquetes almacenados en orden inverso, es decir, comenzando con  $i+k$ , y actuando hacia  $i$ . Cuando se ha reconstruido un paquete almacenado, se verifica si es correcto usando su CRC. Los paquetes que no transportan una CRC no se deben entregar a capas superiores. Los paquetes en las que la CRC es satisfactoria, se entregan a capas superiores en el orden original, es decir,  $i, \dots, i+k$ .

Obsérvese que esta descompresión inversa introduce un almacenamiento temporal mientras se espera la validación del contexto y por lo tanto introduce un retardo adicional. De este modo, la misma debería usarse solamente cuando sea aceptable cierta cantidad de retardo. Por ejemplo, para paquetes de vídeo pertenecientes al mismo cuadro de vídeo, el retardo del tiempo de llegada de los paquetes no provoca un retardo del tiempo de presentación. Las aplicaciones de flujo continuo insensibles a los retardos también pueden ser tolerantes a dicho retardo. Si el descompresor no puede determinar si la aplicación puede tolerar el retardo, el mismo no debería realizar la descompresión inversa.

La siguiente descripción ilustra el procedimiento de descompresión con cierto detalle:

1. El descompresor almacena paquetes comprimidos que no se pueden descomprimir correctamente debido a un contexto no válido.

2. Cuando el descompresor ha recibido un paquete de actualización de contexto y el contexto ha sido validado, comienza a recuperar los paquetes almacenados en orden inverso. La descompresión se lleva a cabo seguida del último paquete descomprimido hacia su paquete anterior como si los dos paquetes estuvieran reordenados. Después de esto, el descompresor comprueba si el encabezamiento reconstruido es correcto usando la CRC.

3. Si la CRC indica una descompresión satisfactoria, el descompresor almacena el paquete completo e intenta descomprimir el paquete precedente. De esta manera, se recuperan los paquetes almacenados hasta que no quedan paquetes comprimidos. Para cada paquete, el descompresor comprueba si los encabezamientos descomprimidos son correctos usando la CRC de la compresión de encabezamientos.

4. Si la CRC indica un paquete descomprimido incorrectamente, el intento de descompresión inversa debe finalizar y se deben descartar la totalidad del resto de paquetes no comprimidos.

5. Finalmente, el descompresor reenvía todos los paquetes descomprimidos correctamente hacia las capas superiores en el orden original.

#### 6.2. RTCP

El RTCP es el Protocolo de Control RTP, [RTP]. El RTCP se basa en la transmisión periódica de paquetes de control a todos los participantes en una sesión, usando el mismo mecanismo de distribución que para los paquetes de datos. Su función principal es proporcionar una realimentación desde los receptores de datos sobre la calidad de la distribución de datos. La información de realimentación se puede usar para aspectos relacionados con funciones de control de la congestión, y puede resultar directamente útil para el control de codificaciones adaptativas.

## ES 2 266 273 T3

En una sesión RTP, habrá dos tipos de flujos continuos de paquetes; uno con el encabezamiento RTP y datos de aplicación, y un segundo flujo continuo con la información de control RTCP. La diferencia entre los flujos continuos en el nivel del transporte es los números de puerto UDP, La cual es más uno para el RTCP. La implementación del compresor de encabezamientos ROHC dispone de varias formas de tratar el flujo continuo RTCP.

5

1. Una entidad de compresor/descompresor para ambos flujos continuos y transportada en el mismo canal usando identificadores CID para diferenciarlas. En el flujo continuo RTCP, se utilizará básicamente sólo la compresión IP/UDP.

10

2. Dos entidades de compresor/descompresor, una para RTP y otra para RTCP, y los flujos continuos transportados en su propio canal. Esto significa que no compartirán el mismo espacio de números CID.

### 7. Tareas adicionales

15

(Editor: Esta sección es “tareas adicionales\_” en particular ya que es necesario integrarla en el resto del documento).

### 7.3. Tunelización

20

#### 7.3.1. Compresión de Encabezamientos para un Encabezamiento de Tunelización IPv4

25

Para encaminar los paquetes hacia el nodo móvil que está en un enlace externo, el agente local del nodo móvil puede encapsular el paquete original en un encabezamiento IP y tunelizar el paquete hacia la dirección de auxilio del nodo móvil. En el caso de una dirección de auxilio de un agente externo en el IPv4 Móvil, el encabezamiento de tunelización en cada paquete tunelizado será eliminado por el agente externo antes de transferirlo al nodo móvil a través de la interfaz aérea; de esta razón no existe ninguna necesidad de comprimir el encabezamiento de tunelización. En el caso de que el nodo móvil use una dirección de auxilio concomitante, el paquete tunelizado será enviado a la estación móvil a través de la interfaz aérea, y es necesario aplicar la compresión al encabezamiento de tunelización.

30

#### 7.3.1.1. Tipo de los Campos del Encabezamiento de Tunelización IPv4 Móvil

35

La siguiente tabla resume la clasificación de los diversos campos definidos en diferentes encabezamientos de tunelización usados en el IPv4 Móvil. En la columna del Esquema de Encapsulación (Esquema Enc.), se incluyen tres métodos de encapsulación - IP en Encapsulación IP (IIE), Encapsulación Mínima (ME), Encapsulación Genérica de Encaminamiento (GRE).

(Nota del editor: Armonizar con la forma en la que esto se describe en el documento ROHC)

40

(Tabla pasa a página siguiente)

45

50

55

60

65

## ES 2 266 273 T3

Esquema Enc	Tipo Encab.	Estático			No estático	
		Esencial		No esencial		
IIE	encab. inter.	igual que en la tabla del Apéndice A en el proyecto de Internet ACE				
	encab. ext.	igual que en la tabla del Apéndice A en el proyecto de Internet ACE				
ME	Encab. IP	igual que en la tabla del Apéndice A en el proyecto de Internet ACE				
	Encab. reenv. mín.	Protocolo			Suma comprob. encab.	Direcc. Dest. Origi. Direcc. Origen Origi.
GRE	Encab. inter.	igual que en la tabla del Apéndice A en el proyecto de Internet ACE				
	Encab. exter.	igual que en la tabla del Apéndice A en el proyecto de Internet ACE				
	Encab. GRE	Protocolo Ver	Número secuencia	C, R, K, S, s bits Banderas Suma de comprobación Desviación Clave Encaminamiento		

### 7.3.1.2. Compresión de Encabezamientos de Tunelización en MIPv4

En el MIPv4 se han especificado tres esquemas de encapsulación. Para esquemas de encapsulación diferentes, los métodos de compresión son diferentes entre ellos.

#### 7.3.1.2.1. Encapsulación IP en IP en el IPv4

Usando la IP en Encapsulación IP, el encabezamiento IP interior original no se modifica en absoluto y por lo tanto se puede comprimir como si no estuviera encapsulado. El encabezamiento exterior se comprime en el nivel de las IP, mientras que el encabezamiento interior se comprime en la ROHC.

#### 7.3.1.2.2. Encapsulación Mínima en IPv4

Con la Encapsulación Mínima, se modifica el encabezamiento IP original y se inserta el Encabezamiento de Reenvío Mínimo entre el encabezamiento IP modificado y la carga útil IP original. El encabezamiento IP modificado más los encabezamientos UDP/RTP se comprimen tal como se define en la ROHC.

El esquema de compresión para el Encabezamiento de Reenvío Mínimo es similar al esquema aplicado al encabezamiento IP. Los campos no esenciales estáticos y variables en el Encabezamiento de Reenvío Mínimo se envían en el estado de Encabezamiento Completo y Restauración. Cuando se produce cualquier cambio en cualquier campo no esencial en el Encabezamiento de Reenvío Mínimo, se debería enviar un encabezamiento comprimido con una máscara de bits que indique el cambio.

#### 7.3.1.2.3. Encapsulación Genérica de Encaminamiento en el IPv4

Con la Encapsulación Genérica de Encaminamiento, el paquete IP original se encapsula en un encabezamiento IP exterior. Se inserta un encabezamiento GRE entre el encabezamiento interior y el encabezamiento exterior. El encabezamiento IP/UDP/RTP original se comprime como si no hubiera encapsulación. El encabezamiento IP exterior se comprime en el nivel de las IP.

## ES 2 266 273 T3

El esquema de compresión para el encabezamiento GRE es similar al esquema aplicado en el encabezamiento IP. Todos los campos no esenciales estáticos y variables en el encabezamiento GRE se envían en el estado de Encabezamiento Completo y restauración. Cuando se produce cualquier cambio en cualquier campo no esencial en el encabezamiento GRE, se debería enviar un encabezamiento comprimido con una máscara de bits que indique el cambio. Si está presente el número de secuencia en el encabezamiento GRE, el esquema para comprimir el número de secuencia podría ser VLE, tal como se define en el proyecto ACE.

### 7.4. Tráfico UDP no RTP

(Nota del editor: Este texto es de draft-koren-avt-crtp-enhance-01.txt que se debe añadir a la rohc - todavía no en concordancia con el resto del documento)

[Nota de Micke: Este debería ser un perfil independiente].

[Nota de cabo: Ni siquiera estoy seguro de que deseemos mantenerlo en la ROHC. La bandera de memoria caché negativa simplemente se podría añadir al I/R, evidentemente, por ejemplo, como un perfil.

Obsérvese que los flujos continuos con números de puerto impares no deberían ser nunca flujos continuos RTP, por lo tanto se podría considerar que esta situación simplifica la compresión RTCP.

No estoy seguro de que el caso sobre CRTP e2e en túneles se pueda aplicar mucho a la ROHC. Debemos explicar mejor los paquetes de RECHAZO que aparecen anteriormente].

#### 2.1 La bandera de flujo continuo en memoria caché negativa

Ciertos flujos continuos, de los que se sabe o sospecha que no son RTP, se pueden situar en una “memoria caché negativa” en el compresor, con lo que solamente se comprimen los encabezamientos IP y UDP. Es ventajoso notificar al descompresor que el flujo continuo comprimido está en la memoria caché negativa: para dichos flujos continuos el contexto es más breve - no existe la necesidad de incluir el encabezamiento RTP, y se pueden evitar todos los cálculos relacionados con el RTP.

En esta mejora, se añade un nuevo bit de bandera “N” al paquete de ENCABEZAMIENTO\_COMPLETO que inicializa un contexto en el descompresor. El bit ocupado por la bandera nueva se fijó siempre previamente a cero. Si la bandera N se fija a 1, esto indica que en este contexto no se transmitirán paquetes RTP\_COMPRIMIDO. Esta bandera es solamente una optimización y el descompresor puede elegir ignorarla.

[[Editor: Las bandera de memoria caché negativa podría ser parte de la información de perfil]]

#### 2.2 Rechazo de un flujo continuo comprimido nuevo

En un enlace de punto a punto, los dos nodos pueden llegar a un acuerdo sobre el número de sesiones comprimidas que están preparados para soportar para este enlace. En un esquema de extremo-a-extremo, un anfitrión puede tener sesiones comprimidas con muchos anfitriones y finalmente puede quedarse sin recursos. Cuando se negocia el túnel de extremo-a-extremo, puede que el número de contextos necesarios no sea predecible. Esta mejora permite que el número negociado de contextos sea mayor de lo que se podría aceptar si se establecen muchos túneles.

En ese caso, a medida que se consumen los recursos de los contextos, puede que se rechace un intento de restablecer un contexto nuevo.

El compresor inicia una compresión de un flujo continuo enviando un paquete ENCABEZAMIENTO\_COMPLETO. Actualmente, si el descompresor dispone de recursos insuficientes para descomprimir el flujo continuo nuevo, puede enviar un paquete ESTADO\_CONTEXTO para invalidar el flujo continuo recién comprimido. El compresor no conoce la razón de la invalidación: habitualmente esto ocurre cuando el descompresor se desincroniza debido a la pérdida de paquetes. Lo más probable es que el compresor vuelva a intentar comprimir este flujo continuo enviando otro ENCABEZAMIENTO\_COMPLETO.

Esta mejora específica que el descompresor puede rechazar la compresión de un flujo continuo enviando un mensaje RECHAZO al compresor. Un mensaje RECHAZO le comunica al compresor que deje de comprimir este flujo continuo.

El mensaje RECHAZO es una [[realimentación-4 de tipo RECHAZO]] con una bandera adicional:

Código de tipo = 1 :ESTADO\_CONTEXTO para flujos continuos CID de 8 bits

Código de tipo = 2 : ESTADO\_CONTEXTO para flujos continuos CID de 16 bits

[[Editor: Esto será tratado por el paquete de realimentación RECHAZO]]

## ES 2 266 273 T3

El compresor puede decidir esperar un rato antes de intentar comprimir flujos continuos adicionales destinados al anfitrión rechazador.

8. *La sección 8 se ha eliminado*

5

9. *Consideraciones sobre seguridad*

Como el cifrado elimina la redundancia de la que intentan aprovecharse los esquemas de compresión de encabezamientos, se encuentra cierto incentivo en dejar de lado el cifrado de encabezamientos para posibilitar un funcionamiento sobre enlaces con un ancho de banda reducido. No obstante, para los casos en los que el cifrado de datos (y no encabezamientos) es suficiente, el RTP especifica un método de cifrado alternativo en el cual únicamente se cifra la carga útil RTP y los encabezamientos se dejan sin cifrar. Dicha situación seguiría permitiendo la aplicación de la compresión de encabezamientos.

10

La compresión ROHC es transparente con respecto a los campos de número de secuencia RTP e indicación de tiempo RTP, por lo que los esquemas de cifrado de carga útil pueden fiarse de los valores de dichos campos.

15

Un compresor de encabezamientos que funcione de forma defectuosa o malintencionado podría provocar que el descompresor de encabezamientos reconstituiera paquetes que no se corresponden con los paquetes originales aunque siguen disponiendo de encabezamientos IP, UDP y RTP válidos y también posiblemente de sumas de comprobación UDP válidas. Dicha alteración se puede detectar con mecanismos de integridad y autenticación de extremo-a-extremo los cuales no se verán afectados por la compresión. Por otra parte, este esquema de compresión de encabezamientos usa una suma de comprobación interna para la verificación de encabezamientos reconstruidos. Esto reduce la probabilidad de producir encabezamientos descomprimidos que no se correspondan con los originales sin que la situación sea percibida.

20

25

Los ataques en forma de negación de servicios son posibles si un intruso puede introducir (por ejemplo) paquetes falsos de tipo ESTÁTICO, DINÁMICO o REALIMENTACIÓN en el enlace y provocar por lo tanto la reducción de la eficacia de la compresión. No obstante, un intruso que tenga la capacidad de inyectar paquetes arbitrarios en la capa de enlace de esta manera plantea cuestiones adicionales de seguridad que van más allá de las correspondientes relacionadas con el uso de la compresión de encabezamientos.

30

10. *Acuses de recibo*

Cuando se diseñó este protocolo, las ideas anteriores de compresión de encabezamientos descritas en [CJHC], [IPHC] y [CRTP] han sido fuentes importantes de información.

35

Gracias a Takeshi Yoshimura de NTT DoCoMo por proporcionar la sección de descompresión inversa (6.1). Gracias también a Anton Martensson por muchas contribuciones valiosas al proyecto y a Andreas Jonsson (Universidad de Lulea), el cual realizó una gran tarea apoyando este trabajo en su estudio de los patrones de cambio de los campos de encabezamientos. Gracias también a la totalidad del resto que han proporcionado comentarios.

40

11. *Consideraciones sobre la propiedad intelectual*

(Nota del editor: esta sección irá a [www.ietf.org/ipr](http://www.ietf.org/ipr) y será sustituida por la referencia estándar correspondiente a la misma, aunque por ahora se deje en el proyecto para simplificar el trabajo que se haga sobre el mismo).

45

Esta propuesta está en conformidad con el RFC 2026.

Telefonaktiebolaget LM Ericsson y sus filiales, según la política corporativa, ofrecerán, para presentaciones realizadas justamente por sus empleados que sean adoptadas o recomendadas como normativa por el IETF, concesión de licencias de patentes de la forma siguiente:

50

Si parte(s) de una presentación de empleados de Ericsson está (están) incluida(s) en una normativa y Ericsson tiene patentes y/o una(s) solicitud(es) de patente que son esenciales para la implementación de dicha(s) parte(s) incluida(s) en dicha normativa, Ericsson está preparada para conceder, basándose en el principio de reciprocidad (retorno de conocimientos), una licencia sobre dicha(s) parte(s) incluida(s) con unos términos y condiciones razonables, no discriminatorios.

55

Para evitar dudas, esta garantía general de concesión de licencias de patentes se aplica a la presente propuesta.

60

Nokia ha presentado solicitudes de patente que posiblemente podrían tener alguna relación técnica con esta contribución.

Matsushita ha presentado solicitudes de patente que posiblemente podrían tener alguna relación técnica con esta contribución. Si parte(s) de la contribución del empleado Matsushita está (están) incluida(s) en una normativa y Matsushita dispone de patentes y/o una(s) solicitud(es) de patente que son esenciales para la implementación de dicha(s)

65

## ES 2 266 273 T3

parte(s) incluida(s) en dicha normativa, Matsushita está preparado para conceder - basándose en el principio de reciprocidad (retorno de conocimientos) - una licencia sobre dicha(s) parte(s) incluida(s) con unos términos y condiciones razonables, no discriminatorios (según el párrafo 10.3.3 del RFC 2026).

5 NTT DoCoMo, Inc. también declara que este texto puede estar relacionado con su patente, y ofrece concesión de licencias de patente de la manera siguiente:

Si parte(s) de este texto proporcionado por empleados de NTT DoCoMo está (están) incluida(s) en una normativa y NTT DoCoMo dispone de patentes y/o una(s) solicitud(es) de patente que son esenciales para la implementación de dicha(s) parte(s) incluida(s) en dicha normativa, NTT DoCoMo está preparada para conceder - basándose en el principio de reciprocidad (retorno de conocimientos) - una licencia sobre dicha(s) parte(s) incluida(s) con unos términos y condiciones razonables, no discriminatorios.

### 12. Referencias

15

[UDP] Jon **Postel**, “User Datagram Protocol”, RFC 768, agosto 1980.

[IPv4] Jon **Postel**, “Internet Protocol”, RFC 791, septiembre 1981.

20

[IPv6] Steven **Deering**, Robert **Hinden**, “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, diciembre 1998.

[RTP] Henning **Schulzrinne**, Stephen **Casner**, Ron **Frederick**, Van **Jacobson**, “RTP: A Transport Protocol for Real-Time Applications”, RFC 1889, enero 1996.

25

[HDLC] William **Simpson**, “PPP in HDLC-like framing”, RFC 1662, 1994.

[VJHC] Van **Jacobson**, “Compressing TCP/IP Headers for Low-Speed Serial Links”, RFC 1144, febrero 1990.

30

[IPHC] Mikael **Degermark**, Bjorn **Nordgren**, Stephen **Pink**, “IP Header Compression”, RFC 2507, febrero 1999.

[CRTP] Steven **Casner**, Van **Jacobson**, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links”, RFC 2508, febrero 1999.

35

[PPPHC] Mathias **Engan**, Steven **Casner**, Carsten **Bormann**, “IP Header Compression over PPP”, RFC 2509, febrero 1999.

[CRTPC] Mikael **Degermark**, Hans **Hannu**, Lars-Erik **Jonsson**, Krister **Svanbro**, “CRTP over cellular radio links”, Internet Draft (work in progress), diciembre 1999. <draft-degermark-crtip-cellular-01.txt>

40

[REQ] Mikael **Degermark**, “Requirements for robust IP/UDP/RTP header compression”, Internet Draft (work in progress), junio 2000. <draft-ietf-rohc-rtp-requirements-01.txt>

[LLG] Krister **Svanbro**, “Lower Layer Guidelines for Robust Header Compression”, Internet Draft (work in progress), mayo 2000. <draft-ietf-rohc-lower-layer-guidelines-00.txt>

45

[CELL] Lars **Westberg**, Morgan **Lindqvist**, “Realtime traffic over cellular access networks”, Internet Draft (work in progress), May 2000. <draft-westberg-realtime-cellular-02.txt>

[WCDMA] “Universal Mobile Telecommunications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.1.0)”. ETSI TR 101 112 V3.0.1, noviembre 1997.

50

### 13. Direcciones de los autores

55

Carsten Bormann  
Universitaet Bremen TZI  
Postfach 330440  
D-28334 Bremen, GERMANY

Tel: +49 421 218 7024  
Fax: +49 421 218 7000  
EMail: cabo@tzi.org

60

Mikael Degermark  
The University of Arizona  
Dept of Computer Science  
P.O. Box 210077

Tel: +1 520 621-3498  
Fax: +1 520 621-4642  
Email: micke@cs.arizona.edu

65

Tucson, AZ 85721-0077, USA

## ES 2 266 273 T3

Lars-Erik Jonsson  
Ericsson Erisoft AB  
SE-971 28 Lulea, Sweden

Tel: +46 920 20 21 07  
Fax: +46 920 20 20 99  
EMail: lars-erik.jonsson@ericsson.com

Zhigang Liu  
Nokia Research Center  
6000 Connection Drive  
Irving, TX 75039, USA

Tel: +1 972 894-5935  
Fax: +1 972 894-4589  
EMail: zhigang.liu@nokia.com

### Apéndice A. Clasificación detallada de campos de encabezamientos

La compresión de encabezamientos es posible debido al hecho de que la mayoría de campos de encabezamientos no varían aleatoriamente de un paquete a otro. Muchos de los campos presentan un comportamiento estático o cambios de una manera más o menos predecible. Cuando se diseña un esquema de compresión de encabezamientos, tiene una importancia fundamental entender el comportamiento de los campos detalladamente.

En el presente apéndice, todos los campos de encabezamientos IP, UDP y RTP, se clasifican y analizan en dos etapas. En primer lugar, se dispone de una clasificación general en A.1 en la que los campos se clasifican basándose en información y suposiciones estables. La clasificación general no tiene en cuenta las características de cambio de los campos variables ya que los mismos variarán más o menos dependiendo de la implementación y de la aplicación usadas. A continuación, en A.2 se realiza un análisis menos estable aunque más detallado que tiene en cuenta las características de cambio. Finalmente, A.3 resume el presente apéndice con conclusiones sobre cómo debería tratar los diversos campos de los encabezamientos el esquema de compresión de encabezamientos para optimizar la compresión y la funcionalidad.

#### A.1. Clasificación general

A nivel general, los campos de los encabezamientos se dividen en 5 clases:

DEDUCIDO	Estos campos contienen valores que se pueden deducir a partir de otros valores, por ejemplo el tamaño de la trama que transporta el paquete, y por lo tanto no deben ser tratados en absoluto por el esquema de compresión.
ESTÁTICO	Se espera que estos campos sean constantes durante toda la vida útil del flujo continuo de paquetes. De alguna manera, la información estática se debe comunicar una vez.
ESTÁTICO-DEF	Campos ESTÁTICOS cuyos valores definen un flujo continuo de paquetes. En general se tratan como el ESTÁTICO.
ESTÁTICO-CONOCIDO	Se espera que estos campos ESTÁTICOS tengan unos valores bien conocidos y por lo tanto no sea necesario comunicarlos en absoluto.
VARIABLE	Se espera que estos campos varíen de alguna manera, bien de forma aleatoria, bien dentro de un conjunto o intervalo de valores limitado, o de alguna otra manera.

En la presente sección, cada uno de los campos de los encabezamientos IP, UDP y RTP se asigna a una de estas clases. Para todos los campos excepto para los casos clasificados como VARIABLE, también se mencionan los motivos de la clasificación. Los campos VARIABLES se examinan y clasifican adicionalmente en A.2 basándose en su comportamiento de cambio esperado.

##### A.1.1. Campos de encabezamientos IPv6

Campo	Tamaño (bits)	Clase
Versión	4	ESTÁTICO-CONOCIDO
Clase de Tráfico	8	VARIABLE
Etiqueta de Flujo	20	ESTÁTICO-DEF
Longitud de Carga Útil	16	DEDUCIDO

## ES 2 266 273 T3

(Continuación)

Campo	Tamaño (bits)	Clase
Siguiente Encabezamiento	8	ESTÁTICO-CONOCIDO
Límite de Saltos	8	VARIABLE
Dirección de Origen	128	ESTÁTICO-DEF
Dirección de Destino	128	ESTÁTICO-DEF

### *Versión*

El campo versión estipula en qué versión IP se basa el paquete. Los paquetes con valores diferentes en este campo deben ser tratados por pilas IP diferentes. Para la compresión de encabezamientos, también se deben usar perfiles de compresión diferentes. Cuando el compresor y el descompresor han negociado qué perfil usar, la versión IP también es conocida para ambas partes. Por esta razón, el campo se clasifica como ESTÁTICO-CONOCIDO.

### *Etiqueta de Flujo*

Este campo se puede usar para identificar paquetes que pertenecen a un flujo de paquetes específico. Si no se usa, el valor se debería fijar a cero. En cualquier otro caso, todos los paquetes pertenecientes al mismo flujo continuo deben tener el mismo valor en este campo, siendo uno de los campos que define el flujo continuo. Por esta razón, el campo se clasifica como ESTÁTICO-DEF.

### *Longitud de la Carga Útil*

Se espera que la información sobre la longitud de los paquetes (y a continuación también la longitud de la carga útil) sea proporcionada por la capa de enlace. Por esta razón, el campo se clasifica como DEDUCIDO.

### *Siguiente Encabezamiento*

Se espera que este campo presente el mismo valor en todos los paquetes de un flujo continuo de paquetes. En cuanto al número de versión, un cierto perfil de compresión únicamente puede tratar un siguiente encabezamiento sucesivo lo cual significa que este valor es conocido cuando se ha negociado el perfil. Por esta razón, el campo se clasifica como ESTÁTICO-CONOCIDO.

### *Direcciones de Origen y de Destino*

Estos campos son parte de la definición de un flujo continuo y por lo tanto deben ser constantes para todos los paquetes del flujo continuo. Por esta razón, los campos se clasifican como ESTÁTICO-DEF.

El resumen de los bits correspondientes a las clases proporciona:

Clase	Tamaño (octetos)
DEDUCIDO	2
ESTÁTICO-DEF	34.5
ESTÁTICO-CONOCIDO	1.5
VARIABLE	2

## ES 2 266 273 T3

### A.1.2. Campos de encabezamientos IPv4

Campo	Tamaño (bits)	Clase
5 Versión	4	ESTÁTICO-CONOCIDO
Longitud de Encabezamiento	4	ESTÁTICO-CONOCIDO
10 Tipo de Servicio	8	VARIABLE
Longitud del Paquete	16	DEDUCIDO
Identificación	16	VARIABLE
15 Bandera Reservada	1	ESTÁTICO-CONOCIDO
Bandera de Posibilidad de Fragmentación	1	ESTÁTICO
20 Bandera de Último Fragmento	1	ESTÁTICO-CONOCIDO
Desviación de los Fragmentos	13	ESTÁTICO-CONOCIDO
Tiempo de Vida	8	VARIABLE
25 Protocolo	8	ESTÁTICO-CONOCIDO
Suma de Comprobación del Encabezamiento	16	DEDUCIDO
30 Dirección de Origen	32	ESTÁTICO-DEF
Dirección de Destino	32	ESTÁTICO-DEF

#### 35 *Versión*

El campo versión estipula en qué versión IP se basa el paquete y los paquetes con valores diferentes en este campo deben ser tratados por pilar IP diferentes. Para la compresión de encabezamientos, también deben usarse perfiles de compresión diferentes. Cuando el compresor y el descompresor han negociado qué perfil usar, la versión IP también es bien conocida para ambas partes. Por esta razón, el campo se clasifica como ESTÁTICO-CONOCIDO.

#### *Longitud del Encabezamiento*

45 Siempre que no haya opciones presentes en el encabezamiento IP, la longitud del encabezamiento es constante y bien conocida. Si hay opciones, los campos serían ESTÁTICO, aunque suponemos que no hay opciones. Por esta razón, el campo se clasifica como ESTÁTICO-CONOCIDO.

#### *Longitud de los Paquetes*

50 Se espera que la información sobre la longitud de los paquetes sea proporcionada por la capa de enlace. Por esta razón, el campo se clasifica como DEDUCIDO.

#### *Banderas*

55 La bandera Reservada se debe fijar a cero y por lo tanto se clasifica como ESTÁTICO-CONOCIDO. La bandera Posibilidad de Fragmentación será constante para todos los paquetes en un flujo continuo y por lo tanto se clasifica como ESTÁTICA. Finalmente, se espera que el bit de Último Fragmento sea cero ya que NO se espera que haya fragmentación, debido al pequeño tamaño esperado de los paquetes. Por esta razón, el bit de Último Fragmento se clasifica como ESTÁTICO-CONOCIDO.

#### *Desviación de los Fragmentos*

65 Suponiendo que no se produce ninguna fragmentación, la desviación de los fragmentos es siempre cero. Por esta razón, el campo se clasifica como ESTÁTICO-CONOCIDO.

## ES 2 266 273 T3

### *Protocolo*

Se espera que este campo presente el mismo valor en todos los paquetes de un flujo continuo de paquetes. En cuanto al número de versión, un cierto perfil de compresión solamente puede tratar un siguiente encabezamiento específico lo cual significa que este valor es bien conocido cuando se ha negociado el perfil. Por esta razón el campo se clasifica como ESTÁTICO-CONOCIDO.

### *Suma de Comprobación de los Encabezamientos*

La suma de comprobación de los encabezamientos protege los saltos individuales contra el procesado de un encabezamiento alterado. Cuando se ha comprimido casi toda la información de encabezamiento IP, no hay necesidad de disponer de esta suma de comprobación adicional; en su lugar la misma se puede regenerar en el lado del descompresor. Por esta razón, el campo se clasifica como DEDUCIDO.

### *Direcciones de Origen y de Destino*

Estos campos son parte de la definición de un flujo continuo y por lo tanto deben ser constantes para todos los paquetes en el flujo continuo. Por esta razón, los campos se clasifican como ESTÁTICO-DEF.

El resumen de los bits correspondientes a las clases proporciona:

Clase	Tamaño (octetos)
DEDUCIDO	4
ESTÁTICO	1 bit
ESTÁTICO-DEF	8
ESTÁTICO-CONOCIDO	3 + 7 bits
VARIABLE	4

### *A.1.3. Campos de encabezamientos UDP*

Campo	Tamaño (bits)	Clase
Puerto de Origen	16	ESTÁTICO-DEF
Puerto de Destino	16	ESTÁTICO-DEF
Longitud	16	DEDUCIDO
Suma de Comprobación	16	VARIABLE

### *Puertos de Origen y de Destino*

Estos campos son parte de la definición de un flujo continuo y por lo tanto deben ser constantes para todos los paquetes en el flujo continuo. Por esta razón, los campos se clasifican como ESTÁTICO-DEF.

### *Longitud*

Este campo es redundante y por lo tanto se clasifica como DEDUCIDO.

El resumen de los bits correspondientes a las clases proporciona:

## ES 2 266 273 T3

Clase	Tamaño (octetos)
DEDUCIDO	2
ESTÁTICO-DEF	4
VARIABLE	2

### A.1.4. Campos de encabezamientos RTP

Campo	Tamaño (bits)	Clase
Versión	2	ESTÁTICO-CONOCIDO
Relleno	1	ESTÁTICO
Extensión	1	ESTÁTICO
Contador CSRC	4	VARIABLE
Marcador	1	VARIABLE
Tipo de Carga Útil	7	VARIABLE
Número de Secuencia	16	VARIABLE
Indicación de Tiempo	32	VARIABLE
SSRC	32	ESTÁTICO-DEF
CSRC	0(-480)	VARIABLE

#### *Versión*

Existe solamente una versión RTP de trabajo y la misma es la versión 2. Por esta razón el campo se clasifica como ESTÁTICO-CONOCIDO.

#### *Relleno*

El uso de este campo depende de la aplicación, aunque cuando se usa el relleno de la carga útil es probable que esté presente en todos los paquetes. Por esta razón, el campo se clasifica como ESTÁTICO.

#### *Extensión*

Si la aplicación usa extensiones RTP, es probable que haya una extensión presente en todos los paquetes (aunque el uso de las extensiones es muy poco frecuente). No obstante, en aras de una mayor seguridad este campo se clasifica como ESTÁTICO y no ESTÁTICO-CONOCIDO.

#### *SSRC*

Este campo es parte de la definición de un flujo continuo y por lo tanto debe ser constante para todos los paquetes en el flujo continuo. Por esta razón, el campo se clasifica como ESTÁTICO-DEF.

## ES 2 266 273 T3

El resumen de los bits correspondientes a las clases proporciona:

	Clase	Tamaño (octetos)
5	ESTÁTICO	2 bits
	ESTÁTICO-DEF	4
10	ESTÁTICO-CONOCIDO	2 bits
	VARIABLE	7,5(-67,5)

### 15 A.1.5. Resumen para IP/UDP/RTP

Si se resume esta situación para IP/UDP/RTP se obtiene:

	Clase \ Ver IP	IPv6 (octetos)	IPv4 (octetos)
20	DEDUCIDO	4	6
	ESTÁTICO	2 bits	3 bits
25	ESTÁTICO-DEF	42,5	16
	ESTÁTICO-CONOCIDO	1 + 6 bits	4 + 1 bit
30	VARIABLE	11,5(-71,5)	13,5(-73,5)
	Total	60(-120)	40(-100)

### 35 A.2. Análisis de patrones de cambio de campos de encabezamientos

Para diseñar mecanismos adecuados con vistas a una compresión eficaz de todos los campos de encabezamientos, se deben analizar sus patrones de cambio. Por esta razón, se realiza una clasificación ampliada basándose en la clasificación general de A.1, considerando los campos que se etiquetaron como VARIABLE en dicha clasificación. Las diferentes aplicaciones usarán los campos de diferentes maneras, lo cual puede afectar a su comportamiento. Cuando sea éste el caso, se describirá un comportamiento típico para el audio y el vídeo conversacional.

Los campos VARIABLE se dividen en cinco subclases diferentes:

45	ESTÁTICO	Estos son campos que se clasificaron como VARIABLE de una forma general, aunque en este caso se clasifican como ESTÁTICOS debido a ciertas suposiciones adicionales.
50	SEMIESTÁTICO	Estos campos son ESTÁTICOS la mayor parte del tiempo. No obstante, el valor cambia ocasionalmente aunque vuelve a su valor original después de un número conocido de paquetes.
55	RARAMENTE-VARIABLE (RC)	Estos son campos que cambian sus valores ocasionalmente y a continuación mantienen sus valores nuevos.
	ALTERNO	Estos campos alternan entre un número pequeño de valores diferentes.
60	IRREGULAR	Finalmente, éstos son los campos para los cuales no se puede identificar ningún patrón de cambio útil.

Para aumentar adicionalmente las posibilidades de clasificación sin incrementar la complejidad, la clasificación se puede realizar bien según los valores del campo y/o bien según los valores de las diferencias delta para el campo.

65 Cuando se realiza la clasificación, se estipulan también otros detalles referentes a una posible información adicional sobre los valores de los campos y/o las diferencias delta de los campos, según la clasificación. Para los campos clasificados ESTÁTICO o SEMIESTÁTICO, uno de los casos podría ser que el valor del campo no sea solamente ESTÁTICO sino también bien CONOCIDO *a priori* (dos estados para los campos SEMIESTÁTICO). Para campos

## ES 2 266 273 T3

con un comportamiento de cambio no irregular, podría saberse que los cambios habitualmente están dentro de un intervalo LIMITADO en comparación con el cambio máximo para el campo. Para otros campos, los valores son completamente DESCONOCIDOS.

5 La tabla A.1 clasifica todos los campos VARIABLE basándose en sus patrones de cambio esperados, especialmente para el audio y el vídeo conversacional.

TABLA A.1

*Clasificación de los campos de encabezamientos VARIABLE*

10

Campo	Valor/Delta	Clase	Información	
15 Id IPv4:	Secuencial	Delta	ESTÁTICO	CONOCIDO
	Salto sec.	Delta	RC	LIMITADO
	Aleatorio	Valor	IRREGULAR	DESCONOCIDO
20 TOS IP / Clase de Tr.	Valor	RC	DESCONOCIDO	
TTL IP / Límite saltos	Valor	ALTERNO	LIMITADO	
25 Sum. Comp. UDP	Deshabil.	Valor	ESTÁTICO	CONOCIDO
	Habilita.	Valor	IRREGULAR	DESCONOCIDO
30 Cont. CSRC RTP:	Sin mez.	Valor	ESTÁTICO	CONOCIDO
	Mezclado	Valor	RC	LIMITADO
Marcador RTP	Valor	SEMIESTÁTICO	CONOC./CONOC.	
35 Tipo de carga útil RTP	Valor	RC	DESCONOCIDO	
Número de secuencia RTP	Delta	ESTÁTICO	CONOCIDO	
Indicación de tiempo RTP	Delta	RC	LIMITADO	
40 Lista CSRC RTP:	Sin mez.	-	-	-
	Mezclado	Valor	RC	DESCONOCIDO

45

Las siguientes subsecciones describen detalladamente los diversos campos de los encabezamientos. Obsérvese que la tabla A.1 y las siguientes descripciones no consideran los cambios provocados por pérdida o reordenación antes del punto de compresión.

50

### A.2.1. Identificación IPv4

55

El campo de identificación (ID IP) del encabezamiento IPv4 se encuentra en ese lugar para identificar qué fragmentos constituyen un datagrama cuando se vuelven a ensamblar datagramas fragmentados. La especificación IPv4 no especifica exactamente cómo se van a asignar valores a este campo, solamente que cada paquete debería obtener una ID IP que sea exclusiva para el par origen-destino y el protocolo durante el tiempo que el datagrama (o cualquiera de sus fragmentos) pudiera estar vivo en la red. Esto significa que la asignación de valores de ID IP se puede realizar de varias maneras, las cuales se han dividido en tres clases.

60

#### *Secuencial*

65

Esta política de asignación mantiene un contador dependiente para cada flujo de paquetes salientes y por lo tanto el valor de ID IP se incrementará en uno para cada paquete en el flujo continuo. Por esta razón, el valor de la diferencia delta del campo es constante y bien conocido *a priori*. Cuando se usa el RTP además del UDP y el IP, el valor de ID IP sigue al número de secuencia RTP. Esta política de asignación es la más deseable a efectos de la compresión de encabezamientos aunque su uso no es tan común como debería ser. La razón es que se puede realizar solamente si el UDP y el IP se implementan conjuntamente de manera que el UDP, el cual separa flujos continuos de paquetes por la identificación del puerto, puede hacer que el IP use contadores de ID independientes para cada flujo continuo de paquetes.

*Salto secuencial*

Esta es la política de asignación más común en las filas IP actuales. La diferencia con respecto al método secuencial es que se usa solamente un contador para todas las conexiones. Cuando el emisor está ejecutando más de un flujo continuo de paquetes simultáneamente, la ID IP se puede incrementar en más de uno. Los valores de ID IP serán mucho más predecibles y requerirán menos bits que se deben transferir que los valores aleatorios, y habitualmente el incremento de paquete-a-paquete (determinado por el número de flujos continuos activos de paquetes salientes y frecuencias de emisión) será limitado.

10 *Aleatorio*

Algunas pilas IP asignan valores de ID IP usando un generador de números pseudoaleatorios. De este modo, no existe ninguna correlación entre los valores ID de los datagramas subsiguientes. Por esta razón, no existe ninguna manera de predecir el valor ID IP para el siguiente datagrama. A efectos de la compresión de encabezamientos, esto significa que es necesario enviar el campo ID IP sin comprimir con cada datagrama, dando como resultado dos octetos adicionales del encabezamiento. Las pilas IP en los terminales celulares NO DEBERÍAN usar esta política de asignación de ID IP.

Debería observarse que la ID es un mecanismo IPv4 y por lo tanto no es necesaria en absoluto en los perfiles IPv6. Para el IPv4, la ID se podría tratar de tres maneras diferentes. En primer lugar, se dispone de la solución ineficaz aunque fiable en la que el campo ID se envía tal como está en todos los paquetes, incrementando los encabezamientos comprimidos con dos octetos. Esta es la mejor forma de tratar el campo ID si el emisor usa la asignación aleatoria del campo ID. En segundo lugar, pueden existir soluciones con mecanismos más flexibles que requieran menos bits para el tratamiento de las ID siempre que se use la asignación de saltos secuenciales. Probablemente, dichas soluciones requerirán incluso más bits si el emisor usa la asignación aleatoria. De este modo, la información sobre la política de asignación del emisor podría ser útil cuando se realice una elección entre las dos soluciones anteriores. Finalmente, incluso para el IPv4, la compresión de encabezamientos se podría diseñar sin ninguna información adicional para el campo ID incluido en encabezamientos comprimidos. Para usar dichos esquemas, debe saberse que el emisor hace uso de la política de asignación secuencial pura para el campo ID. Puede que no sea posible conocer dicha elección, lo cual implica que la aplicabilidad de dichas soluciones es muy dudosa. No obstante, los diseñadores de pilas IPv4 para terminales celulares DEBERÍAN usar la política secuencial.

A.2.2. *Clase de Tráfico / Tipo De Servicio IP*

Se espera que el campo de Clase de Tráfico (IPv6) o de Tipo De Servicio (IPv4) sea constante durante la vida útil de un flujo continuo de paquetes o que cambie relativamente poco.

A.2.3. *Límite de Saltos / Tiempo de Vida IP*

Se espera que el campo de Límite de Saltos (IPv6) o de Tiempo De Vida (IPv4) sea constante durante el tiempo de vida de un flujo continuo de paquetes o que alterne entre un número limitado de valores debido a cambios de la ruta.

A.2.4. *Suma de Comprobación UDP*

La suma de comprobación UDP es opcional. Si se deshabilita, su valor es constantemente cero y se podría comprimir. Si se habilita, su valor depende de la carga útil, lo cual a efectos de la compresión es equivalente a que cambie aleatoriamente con cada paquete.

A.2.5. *Contador CSRC RTP*

Este es un contador que indica el número de elementos CSRC presentes en la lista CSRC. Se espera que este número sea prácticamente constante basándose en cada paquete individual y que cambie en una magnitud reducida. Mientras no se use un mezclador RTP, el valor de este campo es cero.

55 *A.2.6. Marcador RTP*

Para el audio, el bit marcador se debería activar solamente en el primer paquete de una secuencia hablada mientras que para el vídeo se debería activar en el último paquete de cada imagen. Esto significa que en ambos casos el marcador RTP se clasifica como SEMIESTÁTICO con valores bien conocidos para ambos estados.

60

A.2.7. *Tipo de Carga Útil RTP*

Se espera que los cambios del tipo de carga útil RTP dentro de un flujo continuo de paquetes sean poco frecuentes. Las aplicaciones se podrían adaptar a la congestión cambiando el tipo de carga útil y/o los tamaños de las tramas, aunque no se espera que esto ocurra frecuentemente.

65

## ES 2 266 273 T3

### A.2.8. Número de Secuencia RTP

El número de secuencia RTP se incrementará en uno por cada paquete enviado.

### 5 A.2.9. Indicación de Tiempo RTP

En el caso del audio:

10 Siempre que no haya pausas en el flujo continuo de audio, la indicación de tiempo RTP se incrementará en una diferencia delta constante, correspondiente al número de muestras en la trama de voz. De este modo, seguirá en gran medida al número de secuencia RTP. Cuando haya un periodo de silencio y comience una nueva secuencia hablada, la indicación de tiempo realizará un salto de forma proporcional a la longitud del periodo de silencio. No obstante, probablemente el incremento estará dentro de un intervalo relativamente limitado.

15 En el caso del vídeo:

El cambio de indicación de tiempo entre dos paquetes consecutivos bien será cero o bien se incrementará en un múltiplo de un valor fijo correspondiente a la frecuencia de reloj de la imagen. La indicación de tiempo también se puede decrementar en un múltiplo del valor fijo si se usan imágenes B. El intervalo de diferencia delta, expresado como un múltiplo de la frecuencia de reloj de la imagen, en la mayoría de los casos es muy limitado.

### A.2.10. Fuentes Contribuyentes RTP (CSRC)

25 Se espera que los participantes en una sesión, los cuales están identificados por los campos CSRC, sean prácticamente los mismos basándose en cada paquete individual con relativamente pocas adiciones o eliminaciones. Mientras no se usen mezcladores RTP, no habrá presentes campos CSRC en absoluto.

### A.3. Estrategias de compresión de encabezamientos

30 Esta sección explica adicionalmente lo que se ha realizado en las secciones anteriores. Basándose en las clasificaciones, se proporcionan recomendaciones sobre cómo tratar los diversos campos en el proceso de compresión de encabezamientos. Son posibles siete acciones diferentes y las mismas se enumeran conjuntamente con los campos a los cuales se aplica cada acción.

#### 35 A.3.1. No enviar en absoluto

Los campos que presentan unos valores bien conocidos *a priori* no deben ser enviados en absoluto. Los mismos son:

- 40 - Versión IP
- Longitud de Carga Útil IPv6
- Siguiendo Encabezamiento IPv6
- 45 - Longitud de Encabezamiento IPv4
- Bandera Reservada IPv4
- 50 - Bandera de Último Fragmento IPv4
- Desviación de Fragmento IPv4
- Protocolo IPv4
- 55 - Suma de Comprobación UDP (si está deshabilitada)
- Versión RTP

#### 60 A.3.2. Transmitir sólo inicialmente

Los campos que son constantes durante toda la vida útil del flujo continuo de paquetes se deben transmitir y entregar correctamente al descompresor solamente una vez. Los mismos son:

- 65 - Dirección de Origen IP
- Dirección de Destino IP

## ES 2 266 273 T3

- Etiqueta de Flujo IPv6
- Bandera de Posibilidad de Fragmentación IPv4
- 5 - Puerto de Origen UDP
- Puerto de Destino UDP
- 10 - Bandera de Relleno RTP
- Bandera de Extensión RTP
- SSRC RTP

### 15 A.3.3. *Transmitir inicialmente, pero estar preparado para actualizar*

Los campos que cambian sólo ocasionalmente se deben transmitir inicialmente aunque también debe disponerse de una forma de actualizar estos campos con nuevos valores si los mismos cambian. Estos campos son:

- 20 - Clase de Tráfico IPv6
- Límite de Saltos IPv6
- Tipo De Servicio (TOS) IPv4
- 25 - Tiempo De Vida (TTL) IPv4
- Contador CSRC RTP
- 30 - Tipo de Carga Útil RTP
- Lista CSRC RTP

### 35 A.3.4. *Estar preparado para actualizar o enviar tal como estén frecuentemente*

Para campos que normalmente bien son constantes o bien cuyos valores se pueden deducir a partir de algún otro campo aunque frecuentemente divergen con respecto a dicho comportamiento, debe existir una forma eficaz de actualizar el campo o de enviarlo tal como está en algunos paquetes. Dichos campos son:

- 40 - Identificación IPv4 (si no se ha asignado secuencialmente)
- Marcador RTP
- Indicación de Tiempo RTP

### 45 A.3.5. *Garantizar robustez continua*

Campos que se comportan como un contador con una diferencia delta fija para TODOS los paquetes, el único requisito sobre la codificación de la transmisión es que las pérdidas de paquetes entre el compresor y el descompresor deben ser tolerables. Si existe más de uno de dichos campos, se pueden comunicar todos ellos conjuntamente. Dichos campos también se pueden usar para interpretar los valores correspondientes a campos enumerados en la sección anterior. Los campos que presentan este comportamiento de contador son:

- 55 - Identificación IPv4 (si se asigna secuencialmente)
- Número de Secuencia RTP

### 60 A.3.6. *Transmitir tal como están en todos los paquetes*

Los campos que presentan valores completamente aleatorios para cada paquete se deben incluir tal y como están en todos los encabezamientos comprimidos. Dichos campos son:

- 65 - Identificación IPv4 (si se asigna aleatoriamente)
- Suma de Comprobación UDP (si está habilitada)

## ES 2 266 273 T3

### A.3.7. Establecer y estar preparado para actualizar la diferencia delta

Finalmente, existe un campo que habitualmente se incrementa en una diferencia delta fija y está en correlación con otro campo. Para este campo, tendría sentido hacer que dicha diferencia delta fuera parte del estado del contexto. En ese caso, debe ser posible iniciar y actualizar la diferencia delta de la misma forma que los campos enumerados en A.3.3. El campo al cual se aplica esto es:

- Indicación de Tiempo RTP

#### 10 Apéndice E - Ejemplos de Codificación

[[Nota del editor: Es necesario actualizar esta descripción con la terminología actual]].

#### 15 E.1. VLE básica

Los siguientes ejemplos ilustran el funcionamiento de la VLE en varios escenarios. Los valores de los campos usados en los ejemplos se podrían corresponder con cualquier campo que se desee comprimir. Los ejemplos ilustran el escenario en el que el campo comprimido presenta una resolución de un bit.

#### 20 Ejemplo 1

*Funcionamiento normal (sin pérdida de paquetes antes del compresor, sin reordenación antes del compresor)*

Supóngase que los paquetes con campos de encabezamiento 279, 280, 281, 282 y 283 se han enviado, y el 279 y el 283 son campos de paquetes de referencia potenciales.

La ventana VLE actual es {279, 283}.

y un paquete con valor de campo = 284 se recibe a continuación, la VLE calcula los siguientes valores

Valor Nuevo	VMax	Vmin	r	Nº de LSB
284	283	279	$\max[ 284-279 ,  284-283 ]=5$	4

La ventana se queda sin modificar si se considera que el paquete nuevo {284} no es una referencia potencial. En este caso el campo se codifica usando 4 bits, y el valor codificado real es los 4 bits menos significativos de 284 (10011100) los cuales = 1100.

#### 40 Ejemplo 2

*Pérdida de paquetes antes del compresor*

Supóngase que se han enviado los paquetes con campos de encabezamiento 279, 280, 281, 282, y 283, y el 279 y 283 son campos de paquetes de referencia potenciales de tal manera que la VSW es nuevamente {279, 283}.

Si a continuación se recibe un paquete con valor de campo = 290, la VLE calcula los siguientes valores

Valor Nuevo	VMax	Vmin	r	Nº de LSB
290	283	279	$\max[ 290-283 ,  290-279 =11$	5

Por lo tanto el campo se codifica usando 5 bits. El valor codificado real es los 5 LSB de 290 (100100010) los cuales = 00010.

Si se supone que el valor nuevo es una referencia potencial, la VSW es {279, 283, 290}.

#### Ejemplo 3

65 *Desordenación de Paquetes antes del compresor*

Supóngase que se han enviado los paquetes con campos de encabezamiento 279, 280, 281, 282 y 283, y el 279 y 283 son campos de paquetes de referencia potenciales de tal manera que la VSW es nuevamente {279, 283}.

## ES 2 266 273 T3

Si a continuación se recibe un paquete con el valor de campo = 278, la VLE calcula los siguientes valores

	Valor Nuevo	VMax	Vmin	r	Nº de LSB
5	278	283	279	$\max[ 278-283 ,  278-279 ]=5$	4

10 Por lo tanto el campo se codifica usando 4 bits. El valor codificado real es los 4 LSB de 278 (10010110) los cuales = 0110.

Si se considera que el valor nuevo es una referencia potencial, la VSW nueva es {283, 290, 278}.

15 En cualquier caso, los campos codificados con VLE deben venir acompañados de algunos bits para identificar los posibles tamaños diferentes de los campos codificados. Los tamaños de este campo de bits pueden variar dependiendo del número de tamaños diferentes que se desee permitir. El planteamiento en el ACE es permitir solamente unos pocos tamaños diferentes para formatos de encabezamientos alineados según los bits. Se usa la codificación Huffman de la longitud para conseguir alguna eficacia adicional, basándose en la frecuencia esperada de necesidad de los diferentes tamaños. De hecho en el encabezamiento comprimido ACE se envían 1 ó 2 bits adicionales.

20 El comportamiento del descompresor en todos los casos de los ejemplos es el mismo - usa como referencia un valor específico de un campo de encabezamiento descomprimido. El encabezamiento que se debe utilizar se podría indicar mediante la presencia de una suma de comprobación en el paquete del encabezamiento comprimido, o por otros medios. Por definición, debe ser uno de los valores de la ventana del compresor.

30 Por ejemplo, supóngase que el último paquete descomprimido correctamente que se clasifica como referencia fue el paquete con el campo de encabezamiento = 291. A continuación supóngase que se recibe el valor del campo codificado de 303 (10001111) y = 01111. Los dos valores más próximos a 291 que tienen los bits LSB = 01111 son el 271 y el 303. El 303 es más próximo, por lo que se selecciona correctamente como el valor del campo no comprimido.

### E.2. VLE Basada en Temporizadores

35 Como ejemplo de funcionamiento, considérese el caso de un códec de voz (20 ms), tal que Paso\_TS = 160. Supóngase que T\_actual y p\_TS\_actual son, respectivamente, 357 y 351, y que se dispone de una ventana deslizante TSW la cual contiene los siguientes 4 valores para las entradas:

	j	T_j	p_TS_j
40	1	9	7
	2	8	6
	3	7	4
45	4	3	1

la j anterior es el número de paquete.

En este caso tiene

50  $\text{Fluctuación\_red}(1)=|(357-9)-(351-7)|=4$  (80 ms Fluctuación de Red)

$\text{Fluctuación\_red}(2)=|(357-8)-(351-6)|=4$  (80 ms Fluctuación de Red)

55  $\text{Fluctuación\_red}(3)=|(357-7)-(351-4)|=3$  (60 ms Fluctuación de Red)

$\text{Fluctuación\_red}(4)=|(357-3)-(351-1)|=4$  (80 ms Fluctuación de Red)

Por lo tanto  $\text{Fluctuación\_Red\_Max} = 4$ .

60 Se supone una fluctuación CD-CC máxima de 2 (40 ms); la fluctuación total que se debe tratar en este caso es por lo tanto

$$J = 4 + 2 + 2 = 8 \text{ paquetes (160 ms)}$$

65

y  $k = 5$  bits (ya que  $2 * 5 + 1 < 2^5$ ). El compresor envía los 5 LSB de p\_TS\_actual al descompresor (351 = 101011111, por lo tanto el valor de TS codificado = 11111).

## ES 2 266 273 T3

Cuando el descompresor recibe este valor, en primer lugar intenta realizar una estimación de la indicación de tiempo calculando la diferencia de tiempo entre la última referencia establecida y el paquete actual  $T_{\text{actual}} - T_{\text{ref}}$ , en la que  $T_{\text{ref}}$  es el valor del tiempo del reloj de pared en el cual el descompresor recibió los encabezamientos de referencia.

5 Ese valor se suma a  $p_{\text{TS\_ref}}$ , la TS RTP empaquetada del encabezamiento de referencia, para obtener la estimación.

Supóngase que en el descompresor como referencia se usa el paquete N° 3:

- 10
- $T_{\text{actual}} = 359$
  - $T_{\text{ref}} = 7$
  - 15 -  $p_{\text{TS\_ref}} = 4$

Nota:

20  $T_{\text{actual}}$  se selecciona en este caso como cualquier valor; la diferencia entre el mismo y  $T_{\text{ref}}$  representa la longitud del intervalo de silencio tal como se observa en el descompresor. En ese caso:

$$T_{\text{actual}} - T_{\text{ref}} = 359 - 7 = 352$$

25 
$$p_{\text{TS\_actual}}(\text{estimación}) = 352 + 4 = 356$$

El descompresor busca el valor más próximo a 356 que presente, en este caso, los bits LSB = 11111. En este caso el valor es 351, la  $p_{\text{TS}}$  original.

30 En cambio, si el compresor fuera a enviar el salto de la indicación de tiempo simplemente como la diferencia en las Indicaciones de Tiempo RTP empaquetadas consecutivas, dicho valor sería

$$p_{\text{TS\_actual}} - p_{\text{TS\_ref}} = 351 - 4 = 347 = 101011011$$

35 Por lo tanto se enviarían más del doble de los bits para un intervalo de silencio de

$$347 \text{ (20 ms)} = 6,94 \text{ segundos}$$

40 Debido a los requisitos básicos del tiempo real para conversación, se espera que la fluctuación acumulada en el funcionamiento normal sea como mucho solamente unas pocas veces el paso de T para la voz. Por esta razón, los formatos de carga útil FO en la sección 4.3 se optimizan (en términos de representar diferentes valores codificados de TS de longitud k) para el caso de  $k=4$  (gestiona hasta 16 diferencias en la indicación de tiempo). Los formatos restantes asimismo permiten tratar una amplia gama de condiciones de fluctuación (más allá de simplemente la voz).

45 Este Proyecto de Internet vence el 17 de marzo de 2001.

50

55

60

65

REIVINDICACIONES

- 5 1. Método para comprimir un flujo continuo de paquetes de Protocolo de Tiempo Real, RTP, que llega a un compresor, que comprende las etapas siguientes:
- adquirir un patrón en el compresor, comprendiendo el patrón una función de la indicación de tiempo, TS, una función del bit marcador, bit M, un cociente y un incremento de TS;
- 10 certificar que un descompresor está sincronizado con el compresor según el patrón, comprendiendo la etapa de certificación el envío del patrón al descompresor; y
- enviar un paquete comprimido según el patrón, estando **caracterizado** el método porque:
- 15 la adquisición del patrón comprende la expresión de la función de TS como una función escalonada del número de secuencia, SN, del paquete, presentando la función escalonada por lo menos un escalón.
2. Método de compresión según la reivindicación 1, en el que el patrón comprende una función del bit M en la que el bit M se activa para un último paquete del escalón.
- 20 3. Método de compresión según la reivindicación 2, en el que el bit M se activa únicamente para el último paquete del escalón.
4. Método según la reivindicación 3, en el que la etapa de certificación comprende además la recepción de una indicación que tiene un bit marcador activado.
- 25 5. Método según la reivindicación 3, en el que la etapa de certificación comprende además:
- recibir un primer ack; y
- 30 recibir un segundo ack.
6. Método según la reivindicación 3, en el que la etapa de certificación comprende además la detección, según el patrón de por lo menos dos paquetes.
- 35 7. Método según la reivindicación 6, en el que la etapa de detección según el patrón comprende el acuse de recibo de por lo menos dos paquetes.
8. Método según la reivindicación 6, en el que dichos por lo menos dos paquetes comprenden un primer paquete y un segundo paquete y las etapas anteriores a la detección según el patrón comprenden:
- 40 recibir un primer acuse de recibo que tiene por lo menos el primer paquete; y
- recibir un segundo acuse de recibo que tiene por lo menos el segundo paquete.
- 45 9. Método según la reivindicación 3, en el que el flujo continuo de paquetes RTP comprende un primer paquete que tiene un primer número de secuencia y un primer bit M, comprendiendo dicho flujo continuo un segundo paquete que tiene un segundo número de secuencia y un segundo bit M, comprendiendo el método además:
- 50 adquirir el primer paquete y el segundo paquete; y
- detectar que el segundo número de secuencia es uno más que el primer número de secuencia y que el primer bit M y el segundo bit M están activados.
- 55 10. Método de compresión según la reivindicación 3, en el que la etapa de envío del patrón comprende además el envío explícito del patrón desde el compresor al descompresor.
11. Compresor para comprimir un flujo continuo de paquetes de Protocolo de Tiempo Real, RTP, que comprende:
- 60 unos medios para adquirir un patrón en el compresor, comprendiendo el patrón una función de la indicación de tiempo, TS, una función del bit marcador, bit M, un cociente y un incremento de TS:
- unos medios para certificar que un descompresor está sincronizado con el compresor según el patrón, comprendiendo dichos medios de certificación unos medios para enviar el patrón al descompresor; y
- 65 unos medios para enviar un paquete comprimido según el patrón;

## ES 2 266 273 T3

estando **caracterizado** el compresor porque:

los medios para adquirir el patrón están adaptados para expresar la función de TS como una función escalonada del número de secuencia, SN, del paquete, presentando la función escalonada por lo menos un escalón.

5

12. Compresor para comprimir según la reivindicación 11, en el que el patrón comprende una función del bit M en la que el bit M está activado para un último paquete del escalón.

10

13. Compresor para comprimir según la reivindicación 12, en el que el bit M se activa solamente para el último paquete del escalón.

14. Compresor según la reivindicación 13, en el que los medios de certificación comprenden además unos medios para recibir una indicación que tiene un bit marcador activado.

15

15. Compresor según la reivindicación 13, en el que los medios de certificación comprenden además:

unos medios para recibir un primer ack; y

unos medios para recibir un segundo ack.

20

16. Compresor según la reivindicación 13, en el que los medios de certificación comprenden además unos medios para detectar, según el patrón, por lo menos dos paquetes.

25

17. Compresor según la reivindicación 16, en el que los medios para la detección según el patrón comprenden unos medios para acusar el recibo de dichos por lo menos dos paquetes.

18. Compresor según la reivindicación 16, en el que dichos por lo menos dos paquetes comprenden un primer paquete y un segundo paquete, que comprenden además:

30

unos medios para recibir un primer acuse de recibo que tiene por lo menos el primer paquete; y

unos medios para recibir un segundo acuse de recibo que tiene por lo menos el segundo paquete.

35

19. Compresor según la reivindicación 13, en el que el flujo continuo de paquetes RTP comprende un primer paquete que tiene un primer número de secuencia y un primer bit M, comprendiendo dicho flujo continuo un segundo paquete que tiene un segundo número de secuencia y un segundo bit M, comprendiendo además el compresor:

unos medios para adquirir el primer paquete y el segundo paquete; y

40

unos medios para detectar que el segundo número de secuencia es uno más que el primer número de secuencia y que el primer bit M y el segundo bit M están activados.

20. Compresor para comprimir según la reivindicación 13, en el que los medios para enviar el patrón comprenden además unos medios para enviar explícitamente el patrón desde el compresor al descompresor.

45

50

55

60

65

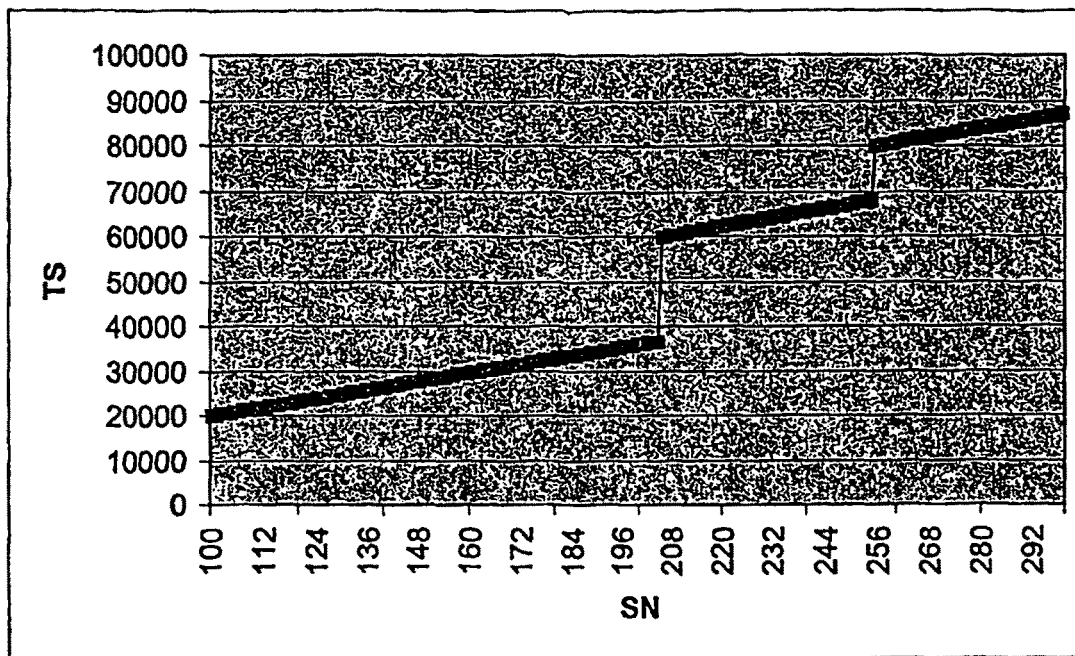


Fig. 1: patrón observado típico para audio

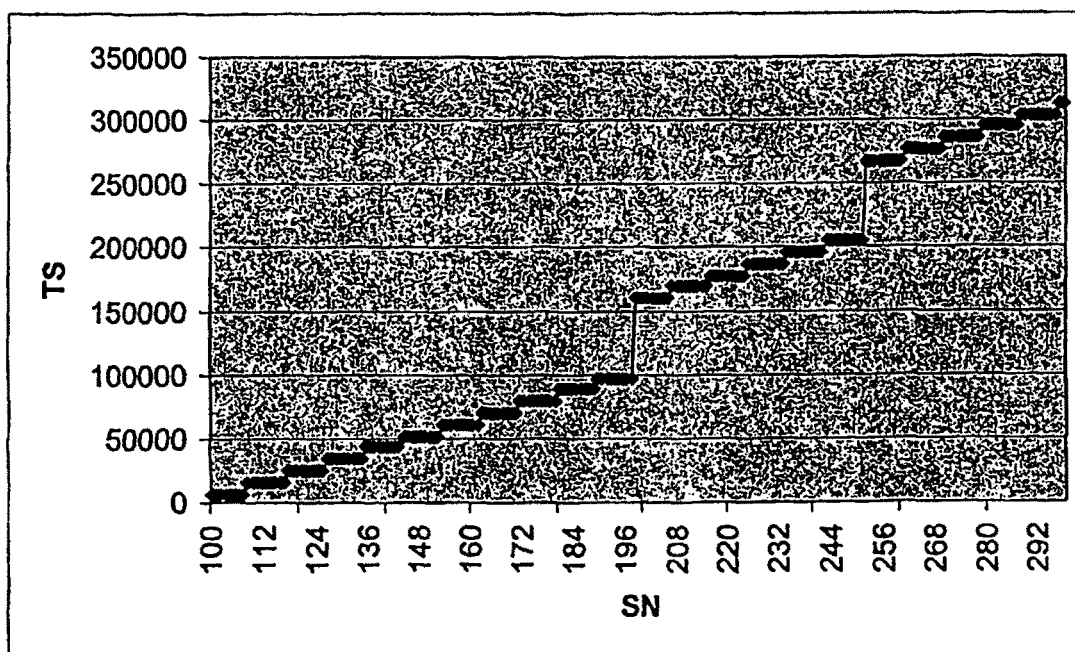


Fig. 2: patrón observado típico para vídeo

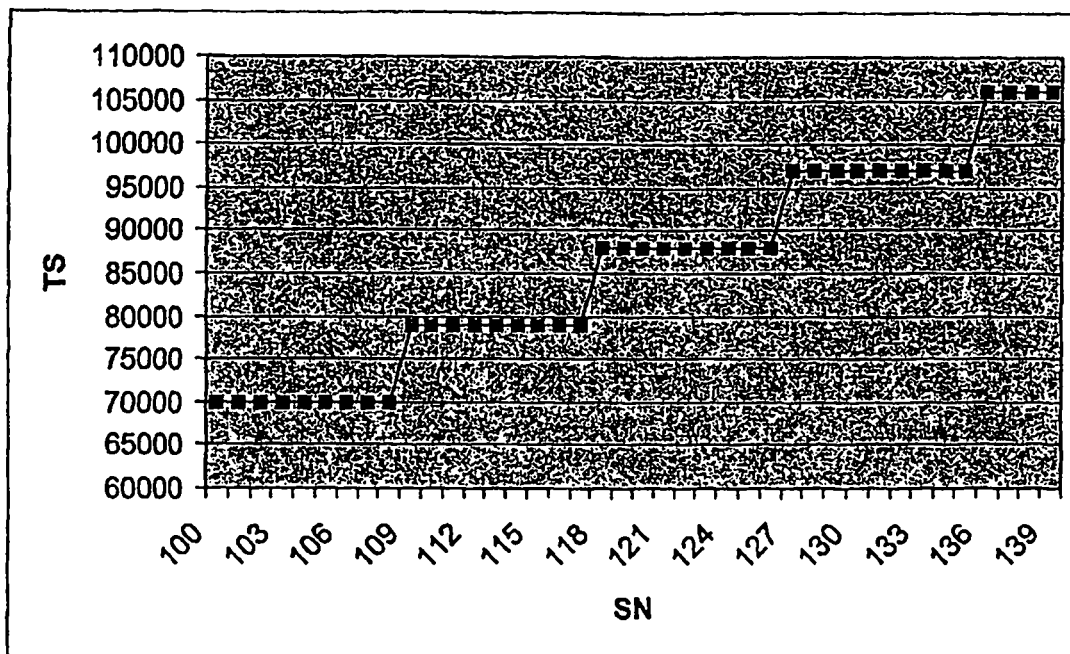


Fig. 3a: Patrón SO, TS en función de SN

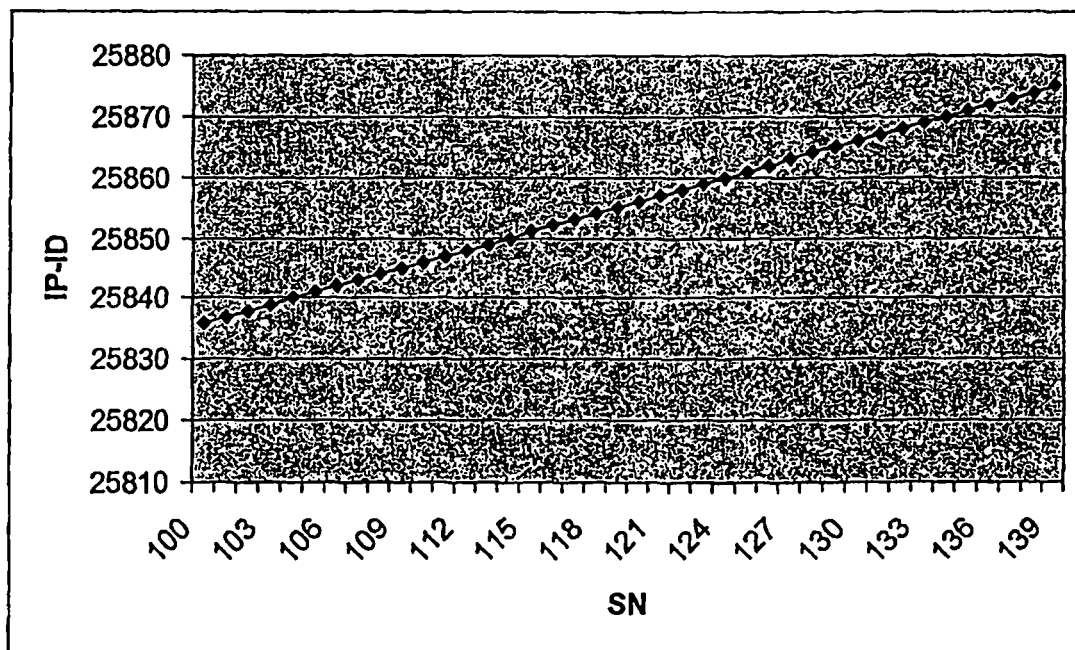


Fig. 3b: Patrón SO, ID-IP en función de SN

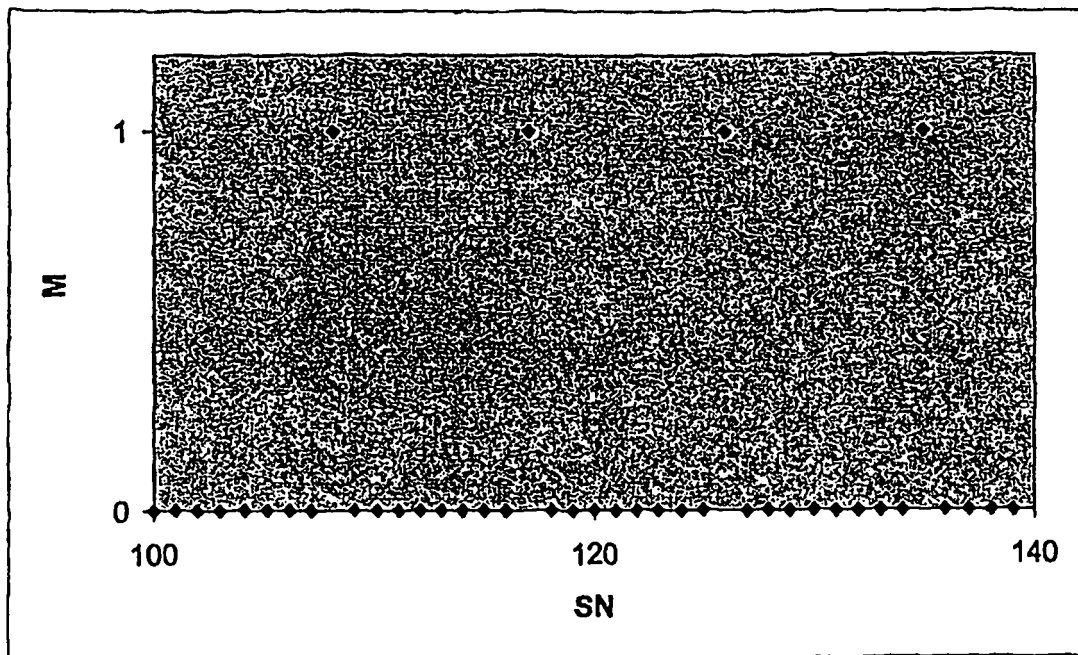


Fig. 3c: Patrón SO, bit M en función de SN

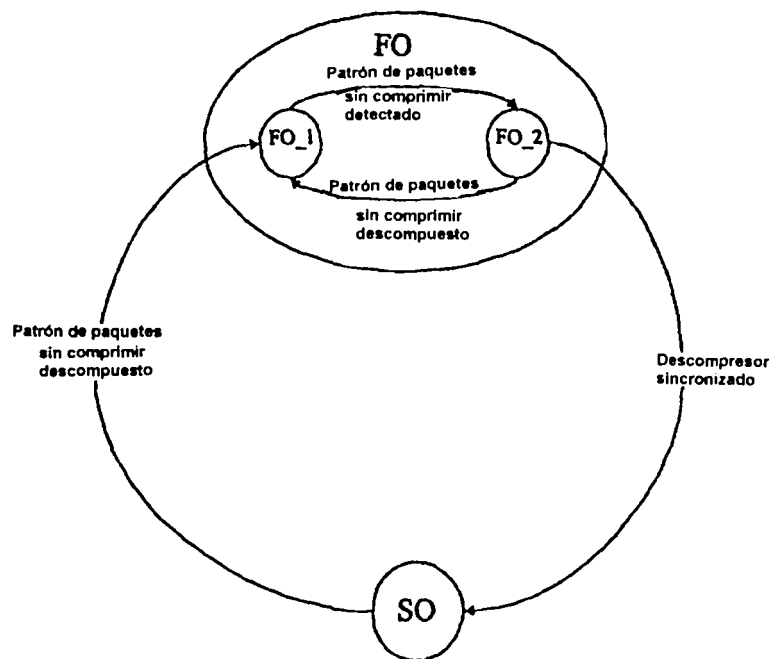


Fig. 4: Lógica del compresor