
Transmisor / receptor LHE

Descripción *Payload Type* RTP y representación de datos

Versión: 0 - 2018/12/17

Resumen ejecutivo

Este documento incluye una descripción concisa de los principales aspectos relativos a la representación de vídeo y carga útil (*payload type*) para transmisión sobre RTP de LHE - versión adaptada a FPGA.

Colaboradores (por orden alfabético):

Tobías Alonso
Ángel López
Jorge E. López de Vergara
Eduardo Miravalls
Jesús Molina
David Muelas
Mario Ruiz
Gustavo Sutter

Índice

Estructura del contenedor LHE	3
División de datos codificados	3
Interfaz entre paquetizador y códec	4
Formato de carga útil	6
Cabecera RTP	6
Cabecera de la carga útil	7
Datos de la carga útil	8

1. Estructura del contenedor LHE¹

A continuación se presenta el diseño de cómo se deben almacenar los datos codificados con LHE en archivos, para permitir la transmisión de los mismos a través de redes TCP/IP. Asimismo, se describe la interfaz entre el bloque funcional de codificador y el elemento funcional de paquetizado y envío de datos, con el fin de establecer el método de comunicación y transferencia de datos entre ambos.

1.1. División de datos codificados

Para facilitar la paquetización de datos codificados con LHE, se define una estructura basada en bloques o “tiras” (*strips*) para los cuadros o *frames*. Todos los bloques de un mismo cuadro DEBEN tener el mismo tamaño, y definen el modo de segmentación de líneas. Cada bloque incluye píxeles correspondientes a un determinado número de líneas (*scanlines*), que en un caso extremo puede ser sólo una línea o una fracción de la misma. Un ejemplo de esquema de bloques de píxeles para un cuadro se muestra en la Fig. 2.1-1, en la que el cuadro se ha dividido en $M \cdot N$ bloques, cada uno con una longitud horizontal de 8 píxeles y una longitud vertical de 4 píxeles. Cada bloque debe ser decodificable de forma independiente al resto. Por ello, si se transmite una imagen completa (esto es, no es diferencial respecto del cuadro anterior), el primer píxel superior izquierdo de cada bloque se DEBE transmitir sin comprimir (esta cuestión se marca en el ejemplo de la Figura 1-1 marcando el píxel correspondiente en gris oscuro). No obstante, en el receptor lo habitual será que lleguen todos los bloques, por lo que la información interbloque (fronteras verticales y horizontales) PUEDE aprovecharse para mejorar la calidad de la imagen decodificada.

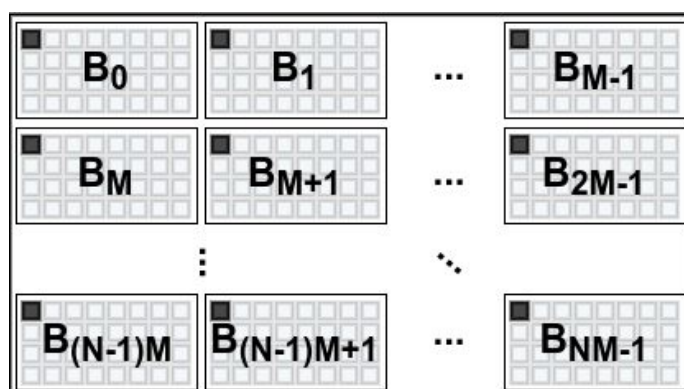


Fig. 2.1-1. Descomposición en bloques de píxeles

Una vez codificados, los bloques DEBEN acompañarse de una cabecera con metainformación para permitir que los datos sean paquetizados sin necesidad de decodificarlos. La estructura de cada bloque se especifica en la Fig. 2.1-2.

¹ A lo largo de las secciones 2 y 3 de este documento, los términos DEBE y PUEDE deben interpretarse según lo especificado en: Bradner, S. (1997). RFC 2119: Key words for use in RFCs to Indicate Requirement Levels.

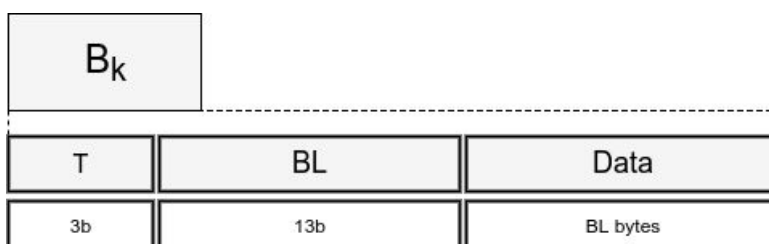


Fig. 2.1-2. Estructura de cada bloque de píxeles

Los campos que incluye esta metainformación son los siguientes:

- **Tipo de bloque (T):** 3 bits. indica el tipo de bloque (p.ej., si transmite información en términos absolutos o relativos).
- **Longitud del bloque (BL):** 13 bits. Indica la longitud en bytes de los datos codificados que incluye el bloque. A lo sumo se permiten bloques de 8191 bytes, si bien, en un paquete los bloques serán mucho menores, dada la MTU del nivel de enlace (típicamente 1500 bytes en Ethernet).
- **Data:** datos codificados del bloque, comenzando por el primer píxel superior izquierdo (que se transmitirá completo), y recorriendo el bloque de izquierda a derecha, y de arriba a abajo.

Esta estructura de bloques PUEDE estar presente en los ficheros para facilitar su paquetización sin que sea necesario realizar una decodificación de la carga multimedia. Todos estos datos DEBEN guardarse en formato de red (*network byte order, big endian*), para permitir la interoperabilidad entre arquitecturas heterogéneas.

1.2. Interfaz entre paquetizador y códec

Para minimizar el acoplamiento entre los módulos de codificación / decodificación (códec) y los módulos de paquetización y transmisión, se establecen dos modos de intercambio de datos:

1. **MODO 1, comunicación mediante transmisión de cuadros completos:** en este modo, las transferencias DEBEN indicar:
 - 1.1. Marca temporal de cada cuadro.
 - 1.2. Identificación de codificación completa o diferencial.
 - 1.3. Espacio de color de la codificación.
 - 1.4. El tamaño vertical en píxeles de cada bloque.
 - 1.5. El tamaño horizontal en píxeles de cada bloque.
 - 1.6. El tamaño vertical de la imagen en número de bloques.
 - 1.7. El tamaño horizontal de la imagen en número de bloques.

Estas normas DEBEN respetarse tanto al utilizar datos almacenados en ficheros como para cualquier otro tipo de transferencias. Esta información PUEDE transferirse mediante una especificación de cabecera de fichero LHE, que PUEDE seguir el formato de la cabecera LHE definida en la Sección 3.2.

2. **MODO 2, comunicación mediante transmisión de conjuntos de bloques:** en este modo, para cada conjunto de bloque los datos del modo 1 DEBEN ser proporcionados también. Adicionalmente, el códec DEBE proporcionar el desplazamiento del conjunto en bloques respecto del inicial.

2. Formato de carga útil²

Una vez se ha definido cómo se deben almacenar los datos codificados con LHE y el mecanismo de transferencia entre los elementos de codificación y envío por red, se definen a continuación cómo se deben estructurar y rellenar los campos de las cabeceras de los protocolos de red utilizados para la transmisión de datos.

2.1. Cabecera RTP

Como ya se comentó en el entregable E2.2³, el formato de la cabecera RTP estándar⁴ es el siguiente (ver Fig. 3.1-1):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=2		P	X	CC			M	PT							sequence number																
timestamp																															
synchronization source (SSRC) identifier																															

Fig 3.1-1. Cabecera RTP

Los valores a rellenar en dicha cabecera son los siguientes:

- Con el *timestamp* se puede conocer el *frame rate*. Este campo DEBE llevar unidades de 90 KHz para la transmisión de vídeo, siguiendo la convención para el formato de carga útil para vídeo, definido en la RFC 3551. Nótese que las frecuencias de refresco habituales son divisores de este número (24 Hz (HDTV), 25 Hz (PAL), 30 Hz (HDTV) y también 50 y 60 Hz).. Todos los paquetes que contengan bloques de un mismo cuadro DEBEN tener el mismo *timestamp*.
- Todos los paquetes DEBEN tener el mismo PT, que PUEDE ser 124 o DEBE ser otro valor asignado dinámicamente en la negociación previa a la transmisión del vídeo (por ejemplo, con RTSP).
- Todos los paquetes de una misma fuente de vídeo DEBEN tener el mismo SSRC durante una sesión multimedia. El valor del SSRC se genera aleatoriamente y DEBE coincidir con el que se envíe en la negociación previa a la transmisión del vídeo (por ejemplo, en RTSP).

² A lo largo de las secciones 2 y 3 de este documento, los términos DEBE y PUEDE deben interpretarse según lo especificado en: Bradner, S. (1997). RFC 2119: Key words for use in RFCs to Indicate Requirement Levels.

³ RacingDrones. (2017). “Estado del arte de codificación, protocolos de tiempo real y FPGAs.” Entregable 2.2.

⁴ Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (2003). *RFC 3550. RTP: A transport protocol for real-time applications*.

- El bit M se DEBE utilizar para indicar al receptor qué paquete contiene el último bloque de un cuadro. Esta indicación es habitual en RTP para paquetes de vídeo, y facilita al receptor saber cuándo debe tratar de recomponer la imagen.
- Los bits de P, X y CC se DEBEN transmitir a 0. Esto es, no hay bytes de *padding*, no hay extensión de cabecera RTP y sólo se transmite una única fuente de vídeo en un paquete de vídeo LHE.
- El número de secuencia será incremental para cada paquete transmitido, y DEBE ser distinto para paquetes de un mismo cuadro que transmitan distintos bloques del mismo, siguiendo lo indicado en la RFC 3550.

2.2. Cabecera de la carga útil

A continuación de la cabecera RTP, cada paquete DEBE incluir una cabecera de carga útil LHE. A continuación se detalla la descripción de la versión 1 de cabecera de carga útil, según muestra la Fig. 3.2-1:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=1	A	R	Profile					C	# Blocks							BH							BW								
IH																IW															
1 st Block id.																Payload															

Fig. 3.2-1. Versión 1 de Cabecera de la carga útil

Los campos de la versión 1 de la carga útil, mostrados en la Fig. 3.2-1, son los siguientes:

- V (*version*): indica la versión concreta de esta cabecera. Los paquetes conformados en base a esta especificación DEBEN tener valor 1 (01_b) en este campo.
- A: indica si se transmite Audio (1) o Vídeo (0) codificado con LHE. Los siguientes campos están definidos específicamente para vídeo, por lo que NO DEBEN tenerse en cuenta si el bit A está a 1.
- R (*Reserved*): bit reservado que DEBE ir a 0 para esta especificación. Con respecto a la versión inicialmente planteada, en la que este bit indicaba si se trataba de una imagen diferencial, en esta versión se puede identificar si cada bloque es diferencial respecto del anterior, evitando tener que enviar toda la imagen diferencial.
- Profile: indica el perfil de color utilizado. Los valores asignados son los siguientes:
 - 0 (0000b): YUV 4:0:0
 - 1 (0001b): YUV 4:2:0
 - 2 (0010b): YUV 4:2:2
 - 3 (0011b): YUV 4:4:4
 - Resto : RESERVADOS para futuros perfiles.

- **C (Codec):** bits dedicados a la descripción de la versión de LHE utilizada para codificar la carga útil. Todos los paquetes de un mismo cuadro DEBEN estar codificados con la misma versión. Los valores definidos son los siguientes:
 - 0 (00_b): LHE básico
 - 1 (01_b): LHE avanzado
 - 2 (10_b): LHE2
 - Resto: RESERVADOS para futuras versiones.
- **# Blocks :** número de bloques consecutivos en el paquete. Como se conocen cuántos bloques van en el paquete y el identificador del primero, no es necesario indicar el resto de identificadores, al transmitirse de forma consecutiva. El número de bloques por paquete DEBE ser al menos 1. Para la versión actual se ha definido un máximo de 63 bloques consecutivos por paquete, si bien en un caso promedio el número de bloques será mucho menor. En el caso de utilizar esta cabecera para introducir metainformación en ficheros de vídeo LHE, este campo PUEDE dejarse a 0.
- **BH (Block Height):** longitud vertical menos uno medida en píxeles de los bloques de este paquete. Por ejemplo, si se envían bloques de una línea, el valor de este campo será 0. A lo sumo, los bloques podrán ser de 16 píxeles de alto, en cuyo caso se codificará el valor 15.
- **BW (Block Width):** longitud horizontal de los bloques de este paquete medida en píxeles, dividida por 16 y restando 1 al resultado. Por tanto, se supone que cualquier ancho de imagen será múltiplo de 16, lo cual es cierto para un amplio rango de resoluciones. Por ejemplo, si los bloques son de 128 píxeles de largo, se transmitirá el valor 7 (128/16-1). Para el caso de 4096, se transmitirá el valor 255 (4096/16-1).
- **IH (Image Height):** número total de bloques en vertical que componen el cuadro (N).
- **IW (Image Width):** número total de bloques en horizontal que componen el cuadro (M).
- **1st block id.:** identificador del primer bloque que aparece en el paquete. Si un paquete tiene varios bloques, el resto de bloques tendrá un identificador incremental respecto al primero, por lo que SOLO se transmite el identificador del primer bloque. El primer bloque de un cuadro DEBE ser 0, y el último N·M-1. Si se recibe un identificador de bloque que no esté entre dichos valores, DEBE descartarse el paquete por estar mal formado.

2.3. Datos de la carga útil

Finalmente, cada bloque debe seguir el siguiente formato, según se muestra en la Fig. 3.3-1:

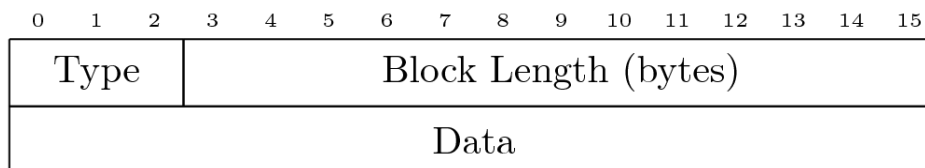


Fig. 3.3-1. Cabecera de un bloque de datos

- **Type:** indica el tipo de bloque que se transmite como carga útil. Los valores definidos son los siguientes:
 - 0 (000_b): bloque completo en valores absolutos.
 - 1 (001_b): codificación diferencial con respecto al mismo bloque del cuadro anterior.
 - Resto: RESERVADOS para futuros usos.

Nótese que, a diferencia de la revisión anterior de este documento, en esta versión no se indica *padding*, al haberse visto que no es necesario incluir explícitamente este valor. La codificación empleada conoce los símbolos posibles, por lo que es un valor que se puede obtener de forma implícita.
- **Block length:** longitud en bytes del bloque. Este campo indica cuántos bytes van a continuación (carga útil), sin contar los campos *Type* y *Block length*. El tamaño máximo en bytes de un bloque es de 8192 bytes, lo que permitiría llegar a transmitir *jumboframes*. No obstante, lo habitual será que los bloques sean inferiores a 1500 bytes para facilitar su transmisión sin fragmentación en redes Ethernet. En general se puede asumir que la MTU habitual en Internet será dicho valor de 1500 bytes, si bien es posible que se produzca fragmentación si se usa algún tipo de VPN.
- **Data:** Los datos de la imagen LHE codificada según las dimensiones de bloque definidas en la cabecera de la carga útil. Los datos ocupan un número entero de bytes (según el *block length*), no teniendo por qué estar alineados a 16 bits.