



# HOJAS DE CONTROL

---

SECRETARIA GENERAL DE LA UNION INTERNACIONAL DE TELECOMUNICACIONES

---

Ginebra, 20 de Mayo de 2006

UIT – SECTOR DE NORMALIZACIÓN DE LAS TELECOMUNICACIONES

**Asunto: Erratum 1 (05/2006) a la**

Recomendación UIT-T J.181 (06/2004), *Mensaje de aviso de inserción de programa digital para sistemas de televisión por cable*

*Añádase el apéndice I que fue omitido cuando la Rec. UIT-T J.181 (06/2004) fue publicada:*

## Apéndice I

### Prácticas recomendadas y guía de interpretación

#### I.1 Introducción

El objetivo de este apéndice es aportar más información a esta Recomendación; en aras de la concisión y la precisión, muchos temas se abordan brevemente.

#### I.2 Referencias informativas

- [1] Propuesta de Recomendación J.sapi, *API de empalme para inserción de programa digital*.
- [2] Recomendación UIT-T H.222.0 (2000) | ISO/CEI 13818-1:2000, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Sistemas*.
- [3] Recomendación UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Vídeo*.
- [4] ISO/CEI 13818-4:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 4: Conformance testing*.
- [5] SMPTE 312M (2000), *Television – Splice Points for MPEG-2 Transport Streams*.
- [6] SCTE 40 2003 (antes SCTE DVS/313), *Digital Cable Network Interface Standard*.
- [7] SCTE DVS/209 (edición original – 1 de febrero de 1999), *DPI System Physical Diagram*.

## **I.3 Definiciones y acrónimos**

### **I.3.1 Definiciones**

En este apéndice los términos tienen significados específicos. Dado que algunos de los términos definidos en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 [2] tienen un significado técnico muy específico, conviene que el lector se remita a la fuente para su definición. A continuación, se dan definiciones breves de los términos que se utilizan en este apéndice.

**I.3.1.1 unidad de acceso:** Representación codificada de una imagen de vídeo o una trama de audio [2].

**I.3.1.2 tono de aviso analógico:** En un sistema analógico, señal, que normalmente es una secuencia de tonos DTMF o un cierre de contacto, que indica al equipo de inserción publicitaria que va a empezar o terminar una pausa.

**I.3.1.3 espacio publicitario:** Espacio de tiempo que los servicios de programación proporcionan al operador CATV durante un programa; normalmente, el tiempo se vende a los anunciantes locales o se utiliza para promocionar la cadena.

**I.3.1.4 pausa publicitaria:** Espacio publicitario o inserción en curso.

**I.3.1.5 concatenación de bloques cifrados (CBC, *cipher block chaining*):** Método específico de criptación; uno de los utilizados en el DES.

**I.3.1.6 modo empalme de componente:** Modo del mensaje de aviso de inserción en el que la bandera de empalme de programa se pone a '0', lo que indica que cada PID/componente destinado al empalme se enumera por separado en la sintaxis que sigue. Los componentes que no figuran en el mensaje no se empalman.

**I.3.1.7 verificación por redundancia cíclica (CRC, *cyclic redundancy check*):** Método para verificar la integridad del mensaje transmitido.

**I.3.1.8 mensaje de inserción:** Véase mensaje.

**I.3.1.9 norma de criptación de datos (DES, *data encryption standard*):** Sistema de criptación de datos mediante claves simétricas.

**I.3.1.10 radiodifusión de vídeo digital (DVB, *digital video broadcasting*):** Consorcio internacional para el desarrollo de sistemas de televisión digital.

**I.3.1.11 mensaje de aviso de programa digital (DPI, *cue message*):** Véase mensaje.

**I.3.1.12 libro de código electrónico (ECB, *electronic code book*):** Método específico de criptación; uno de los utilizados en el DES.

**I.3.1.13 mensaje control de autorización (ECM, *entitlement control message*):** Mensajes privados de información sobre acceso condicional, que especifican las palabras de control y quizás otros parámetros específicos del flujo para aleatorización y/o control.

**I.3.1.14 mensaje gestión de autorización (EMM, *entitlement management message*):** Mensajes privados de información sobre acceso condicional, que especifican los niveles de autorización o los servicios de los decodificadores específicos. Pueden estar destinados a un único decodificador o a un grupo de decodificadores.

**I.3.1.15 evento:** Evento de empalme o evento de visionado, según se define a continuación.

**I.3.1.16 punto de entrada:** Punto del tren, adecuado para la entrada, situado en la frontera de una unidad de acceso.

**I.3.1.17 mensaje:** En el contexto de este documento, es el contenido de cualquier parámetro sección de información de empalme.

- I.3.1.18 punto de salida:** Punto del tren, adecuado para la salida, situado en la frontera de una unidad de acceso.
- I.3.1.19 indicador inicio de unidad de cabida útil (*payload\_unit\_start\_indicator*):** Bit de encabezamiento del paquete de transporte que señala, entre otras cosas, que empieza una señal en la cabida útil que sigue [2].
- I.3.1.20 identificador de paquete (PID, *packet identifier*):** Valor único de 13 bits que identifica los trenes elementales de un programa en un tren de transporte monoprograma o multiprograma [2].
- I.3.1.21 tren PID:** Tren de paquetes con el mismo PID dentro de un tren de transporte.
- I.3.1.22 campo de puntero:** Primer byte de la cabida útil de un paquete de transporte, necesario cuando en ese paquete empieza una sección [2].
- I.3.1.23 tiempo de presentación:** Tiempo o instante en el que una unidad de presentación se entrega al decodificador del sistema [2].
- I.3.1.24 programa:** Conjunto de trenes PID de vídeo, audio y datos que comparten un número de programa común en un MPTS [2].
- I.3.1.25 punto de entrada de programa:** Grupo de puntos de entrada del tren PID que tienen el mismo tiempo de presentación.
- I.3.1.26 punto de salida de programa:** Grupo de puntos de salida del tren PID que tienen el mismo tiempo de presentación.
- I.3.1.27 modo empalme de programa:** Modo del mensaje de aviso de inserción en el que la bandera de empalme de programa se pone a '1', lo que indica que el mensaje se refiere a un punto de empalme de programa y que hay que empalmar todos los PID/componentes del programa.
- I.3.1.28 punto de empalme de programa:** Punto de entrada o de salida de programa.
- I.3.1.29 descriptor de registro:** El descriptor de registro se lleva en la tabla de correspondencia de programas de un programa para indicar que, si se señalan eventos de empalme, las secciones de información de empalme se transportan en el tren PID dentro de ese programa. La presencia del descriptor de registro significa que el programa cumple esta Recomendación.
- I.3.1.30 reservado:** El término "reservado", cuando se utiliza en las cláusulas que definen el tren de bits codificado, indica que el valor se puede utilizar en el futuro para extender la norma. A menos que se indique otra cosa, en la Recomendación todos los bits reservados están puestos a '1'.
- I.3.1.31 evento de empalme:** Oportunidad de empalmar uno o más trenes PID.
- I.3.1.32 modo empalme inmediato:** Modo del mensaje de aviso de inserción en el que el dispositivo empalmador debe aprovechar la primera oportunidad para empalmar, según la tabla de información de empalme. Cuando no está en este modo, el mensaje lleva el parámetro "pts\_time", que indica el instante de empalme.
- I.3.1.33 punto de empalme:** Punto del tren PID, de salida o de entrada.
- I.3.1.34 evento de visionado:** Programa televisivo o trozo de material comprimido dentro de un servicio; se diferencia del evento de empalme en que éste es un punto en el tiempo.

## **I.3.2 Abreviaturas**

En esta Recomendación se utilizan las siguientes siglas.

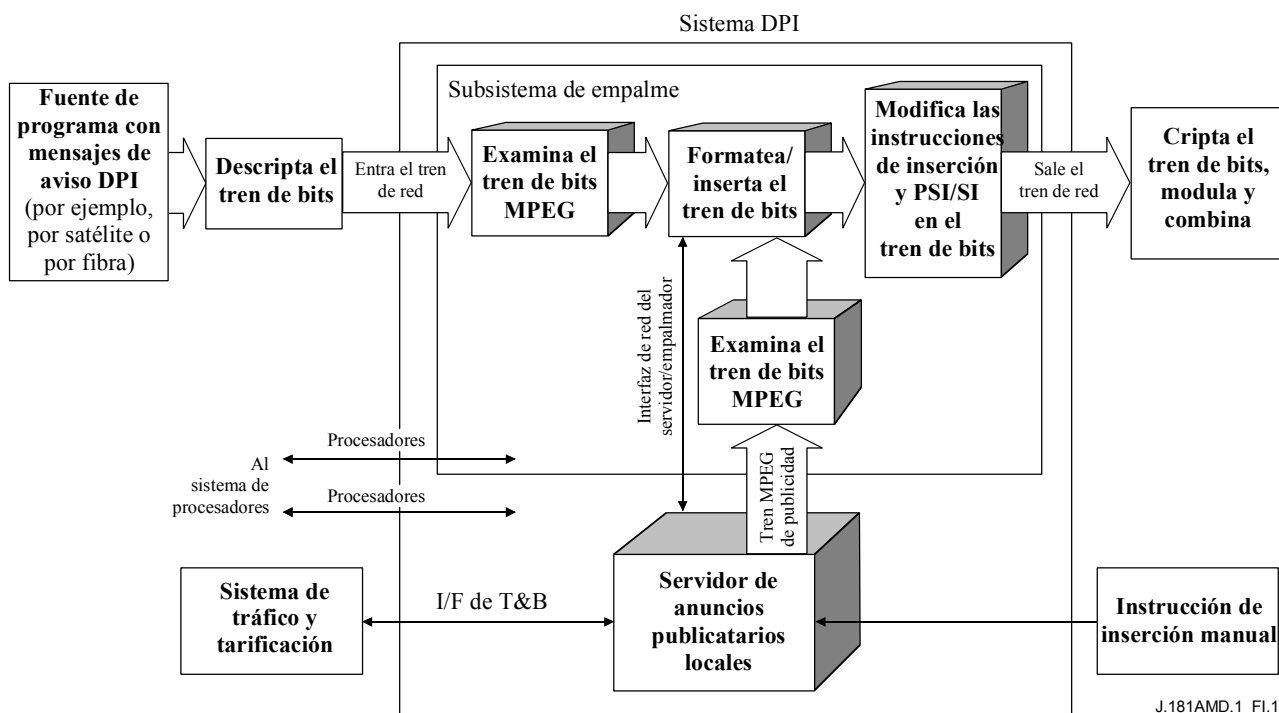
ATSC	Advanced Television Systems Committee
CBC	Concatenación de bloques cifrados ( <i>cipher block chaining</i> )
CBR	Velocidad binaria constante ( <i>constant bit rate</i> )

CRC	Verificación por redundancia cíclica ( <i>cyclic redundancy check</i> )
DES	Norma de criptación de datos ( <i>data encryption standard</i> )
DVB	Difusión de vídeo digital ( <i>digital video broadcasting</i> )
ECB	Libro de código electrónico ( <i>electronic code book</i> )
ECM	Mensaje control de autorización ( <i>entitlement control message</i> )
EMM	Mensaje gestión de autorización ( <i>entitlement management message</i> )
MPTS	Tren (flujo) de transporte multiprograma ( <i>multi program transport stream</i> )
PID	Identificador de paquetes ( <i>packet identifier</i> )
PMT	Tabla de correspondencia de programa ( <i>program map table</i> )
PTS	Indicación de tiempo de presentación ( <i>presentation time stamp</i> )
SPTS	Tren (flujo) de transporte de un solo programa ( <i>single program transport stream</i> )
T-STD	Decodificador-objetivo en sistemas de transporte ( <i>transport stream system target decoder</i> )
uimsbf	Entero sin signo, bit más significativo primero ( <i>unsigned integer, most significant bit first</i> )
VBR	Velocidad binaria variable ( <i>variable bit rate</i> )

#### **I.4 Visión general**

Esta Recomendación soporta el empalme de trenes de transporte MPEG-2 para inserción de programas digitales (DPI), por ejemplo, la inserción de anuncios publicitarios u otros tipos de contenidos diferentes. Se define un mecanismo de mensajería dentro del tren para señalar las oportunidades de empalme e inserción. El dispositivo empalmador no tiene por qué hacer caso de los eventos de empalme del mensaje de aviso DPI, puesto que este mensaje no es una instrucción de empalme sino un indicador de la presencia de un espacio publicitario. El aprovechamiento de dicho espacio es opcional.

Como se muestra en la figura I.1, los mensajes de aviso DPI los reciben los dispositivos empalmadores y servidores en la cabecera del sistema de cable, a fin de efectuar la inserción de locales mediante el empalme del tren de bits publicitario (cuyo contenido es normalmente comercial) en el tren de bits del contenido de programación. Esta Recomendación no diferencia entre un empalmador y un servidor, como sucede en la Recomendación J.sapi [1]. Cuando en esta Recomendación se utilizan los términos "empalmador" o "dispositivo empalmador" el significado de la oración se puede aplicar también a una combinación empalmador/servidor. En la práctica, son normalmente los servidores de publicidad (y no los empalmadores) los que analizan, interpretan e inician la acción especificada en los mensajes de aviso DPI. Puesto que los dispositivos empalmador y servidor se pueden combinar en un solo dispositivo, en este apéndice se usa a menudo servidor/empalmador para indicar uno o más dispositivos que realizan ambas funciones. La figura I.1 que es una versión modificada del que aparece en DVS/209 [7], describe la funcionalidad general y la interoperabilidad de los sistemas de cabecera que realizan esta función.



**Figura I.1/J.181 – Esquema del sistema de cabecera**

En la figura I.1 se supone que el tren de red es un tren de transporte compatible MPEG-2 (multiprograma o monoprograma), al que no se impone más restricción que la inclusión de los mensajes de aviso definidos. Se prevé que el empalme en la frontera de los paquetes de transporte, estipulado en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 [2] y SMPTE 312M [5], no será adecuada en redes de cable debido a la utilización del multiplexado estadístico con velocidad binaria variable (VBR) y del refresco progresivo (sin cuadros I), métodos que quizás exijan la eliminación de la capa de transporte.

Esta Recomendación especifica una técnica para transmitir notificaciones de llegada de puntos de empalme en el tren de transporte. Se define una tabla de información de empalme para notificar a los dispositivos situados más adelante eventos de emplame, tales como el inicio o fin de una pausa de la red. La tabla de información de empalme, que pertenece a un programa dado, se transporta en un identificador paquete (PID) específico incluido en la tabla de correspondencia de programas (PMT). Por ello, la notificación del evento de empalme puede atravesar casi todos los remultiplexores del tren de transporte sin necesidad de un procesamiento especial. Pero puede que los remultiplexores tengan que cumplir algunas condiciones cuando pasen mensajes de aviso DPI. Estas condiciones se explican en el cuerpo principal de esta Recomendación, y en este apéndice se dan más detalles.

Esta Recomendación no trata las restricciones que puedan imponerse a los dispositivos empalmadores y en la sintaxis de la tabla de información de empalme (splice\_info\_table) no se menciona la calidad de la imagen ni del empalme. Esta Recomendación no garantiza que el empalme se realice sin problemas o consecuencias de algún tipo (empalme liso).

#### **I.4.1 Alcance**

La finalidad de este apéndice es servir de guía. Si contradice lo expuesto en el cuerpo principal de la Recomendación, prevalecerá esta última.

## **I.4.2 Objetivo**

El objetivo de este apéndice es ayudar a los diseñadores de equipos de empalme, diseñadores de equipos de inserción de espacios publicitarios, y a los que compran y usan dichos equipos. Se prevé que también interesará a las redes que originan mensajes de aviso DPI en sus emplazamientos de enlace ascendente y a los fabricantes de los equipos que realizan esta función. También se espera que ayude a la integración de sistemas de equipos relacionados con la publicidad, tanto en el extremo de origen como de recepción de los mensajes

Los fabricantes de equipos que pasan mensajes de aviso DPI en el tren MPEG encontrarán aquí información importante. Un ejemplo de tales equipos es el remultiplexor de cambio de velocidad, que realiza un procesamiento complejo del flujo. Cuando el flujo es demultiplexado, procesado y remultiplexado, es importante situar el mensaje de aviso DPI en la posición correcta con respecto al servicio de vídeo y las discontinuidades de la base de tiempo cercanas. Puede que este tipo de equipo se necesite también para modificar el mensaje antes de retransmitirlo.

## **I.5 Directrices de aplicación**

### **I.5.1 Fronteras prácticas de `splice_time()` en `splice_insert()`**

¿Con qué antelación con respecto al empalme hay que enviar el mensaje `splice_insert` (inserción de empalme) relativo a la imagen a la que se refiere, para que el sistema de inserción de anuncios publicitarios actúe sobre el mensaje con seguridad?

El "tiempo de armado" (*arm time*) es el tiempo en que un mensaje de aviso DPI debe preceder a la inserción real, y está comprendido entre 5 y 8 segundos. Esto es compatible con el tiempo rebobinado previo (*pre-roll*) de los tonos de aviso analógicos. El tiempo de armado no debe ser tan corto para que el espacio publicitario pase antes de que el sistema de inserción tenga tiempo de actuar. Se estima que el funcionamiento seguro necesita un mínimo de cuatro segundos.

En el caso de que se quiera ampliar el tiempo de armado por encima de los ocho segundos recomendados habrá que estudiar cuidadosamente las consecuencias, por lo que es prematuro especificar un valor máximo para el tiempo de armado.

La `splice_info_section` no tiene ninguna indicación de tiempo de presentación (PTS) o DTS asociados. Por consiguiente, no se define cuándo hay que decodificar el mensaje ni cuándo hay que presentarlo, es decir, cuándo se realiza. Al elegir un tiempo de armado mínimo de cuatro segundos se evita el problema de definir cómo se mide este tiempo; de esta forma, cualquier método de medición razonable será suficiente.

### **I.5.2 Exactitud del instante de tiempo del empalme**

Si bien tanto el modo empalme de programa como el modo empalme de componente permiten especificar los instantes precisos en que realizar los empalmes, o tiempos de empalme, no es necesario que el servidor/empalmador realice los empalmes en esos instantes, sobre todo si al hacerlo se obtiene un empalme de calidad inferior a la necesaria. En lugar de ello, conviene que el servidor/empalmador utilice los instantes de realización de empalmes indicados como una guía.

Por ejemplo, en el caso del vídeo, el empalmador puede insertar algunos cuadros negros para alinear un grupo de imágenes (GOP, *group of picture*), o, en vez de utilizar un cuadro B como punto de salida, puede utilizar un cuadro I o un cuadro P cercanos. En el caso del audio, normalmente el tiempo de presentación de vídeo no está alineado con el o los tiempos de presentación de audio y, por consiguiente, los empalmes de audio correspondientes se realizarán en instantes de tiempo que están "cerca" del instante de presentación de vídeo más próximo. Por tanto, la elección inteligente de los puntos de empalme concretos es uno de los factores que diferencian a los empalmadores.

El creador de mensajes de aviso puede ser lo suficientemente inteligente como para no especificar nunca el instante de presentación de un cuadro B como un punto de salida del sistema de provisión de contenidos de la red. Por ello, son escasos los motivos fundamentados para que el empalmador/servidor modifique el tiempo de empalme para iniciar la pausa. Igualmente, el creador del programa hará bien en elegir un cuadro I como punto de entrada en que se reanudará la transmisión del programa. No obstante, cada sitio remoto puede almacenar ficheros MPEG-2 locales de espacios publicitarios con longitud, estructura y naturaleza precisa; por ejemplo, cuando se utilizan distintas configuraciones de codificador. Por consiguiente, el creador de mensajes no puede garantizar la alineación correcta entre un punto de entrada determinado y cada punto de salida almacenado en los sitios remotos sin establecer limitaciones adicionales.

### 1.5.3 Uso y carácter unívoco de splice\_event\_id

Los mensajes de aviso se pueden crear de distintas maneras: con la información que lleva el material de origen audio/vídeo, mediante un sistema de disparo de eventos de enlace ascendente o un sistema de disparo de eventos de cabecera de cable. La combinación de estas fuentes en un servicio puede ocasionar la existencia de valores contradictorios del splice\_event\_id (identificador de evento empalme) para el mensaje de aviso. Por consiguiente, se recomienda realizar una partición del splice\_event\_id de 32 bits como se indica a continuación para que cada fuente tenga una gama única de valores splice\_event\_id:

Sintaxis	Bits	Tipo
Event_source	4	uimsbf
Event_number	28	uimsbf

Event\_source – Número asignado por el usuario para la fuente del mensaje de aviso.

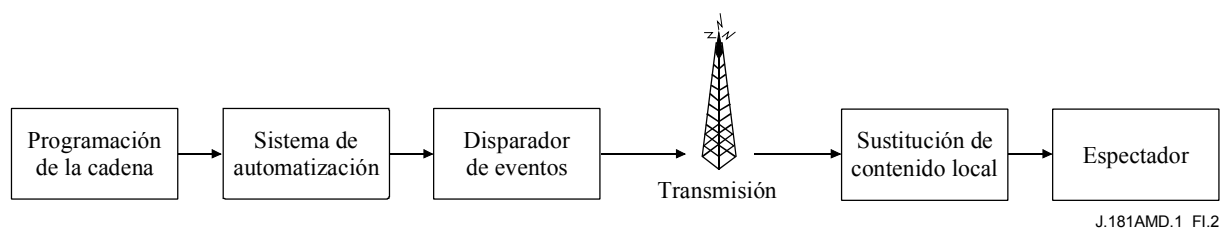
Event\_number – Número elegido por la fuente del evento para identificar un ejemplar del mensaje de aviso.

El splice\_event\_id es un identificador de un ejemplar de la oportunidad de modificar el multiplexado. Se necesitan al menos dos splice\_event distintos para realizar una inserción, uno de inicio y uno de fin (a menos que la duración se incluya en el mensaje que indica el fin del empalme, splice\_out).

Cada evento tiene que tener un splice\_event\_id único. El splice\_event\_id no se puede reutilizar antes de que la oportunidad de empalme que describe (total o parcialmente) haya pasado por completo. Se considera que el splice\_event\_id está en uso desde que aparece en el primer mensaje de aviso hasta más o menos un segundo después del instante del empalme asociado, y se le puede reutilizar inmediatamente después para un nuevo evento de empalme. No es posible utilizar el mismo splice\_event\_id para señalar el inicio y fin de un espacio publicitario si se señala el fin antes de que se ejecute el inicio. No obstante, se puede reutilizar el mismo splice\_event\_id si el primer mensaje de fin se envía después de ejecutado el inicio.

La industria de la publicidad utiliza otra información además del splice\_event\_id para identificar el material real que se está reproduciendo y el momento de su inserción: el nombre del servicio, el tiempo, la agencia de publicidad y el número del anuncio publicitario.

Siempre y cuando Event\_source sea único para cada punto en el que se puedan insertar mensajes de aviso en la cadena siguiente (véase la figura I.2), los identificadores de evento no entrarán en colisión.



**Figura I.2/J.181 – Puntos de inserción del mensaje de aviso**

Se sugieren los siguientes valores para Event\_source:

**Cuadro I.1/J.181 – Valores para Event\_source**

Fuente del mensaje de aviso	Valor de Event_source	Ejemplo de gamas de splice_event_id
Aviso incluido en el material fuente original	0	0x00000000, 0x00000001 .... 0x0fffffff
Aviso creado por la conmutación del sistema de automatización	4	0x40000000, 0x40000001 .... 0x4fffffff
Aviso creado por el sistema de disparo de eventos en directo	6	0x60000000, 0x60000001 .... 0x6fffffff
Aviso creado por el sistema de sustitución de contenido local	12	0xC0000000, 0xC0000001 .... 0xCfffffff

#### **I.5.4 Uso de la instrucción splice\_schedule( )**

La instrucción splice\_schedule() (planificación de empalmes), se usa cuando las pausas comerciales son largas y nunca en el caso de un solo evento. Las implementaciones DPI actuales se centran en el soporte de un solo evento, para el que no se usa esta instrucción.

#### **I.5.5 Modo empalme de componente**

La principal razón para aplicar el modo empalme de componente es permitir la sustitución o el paso de distintos flujos elementales de cualquier tipo, de forma que se ajuste al contenido de programa y la finalidad del anunciante.

Por ejemplo, durante una pausa comercial local, puede convenir que continúen los datos informativos (por ejemplo, información financiera) que transmite el proveedor de red, aunque el espectador esté viendo un comercial insertado localmente. A la inversa, si el espectador tiene un adaptador multimedia o decodificador sofisticado, se le puede ofrecer dentro del espacio publicitario información que podrá bajar para verla en otro momento. En este caso, si el contenido de datos excede la duración del comercial, se puede seguir pasando datos de publicidad aunque se haya reanudado el programa.

La metodología que se define en esta Recomendación permite que durante una pausa comercial se transmitan algunos tipos de datos, mientras que otros se bloquean o sustituyen. Corresponde al insertador de mensaje decidir qué datos pasar y cuáles bloquear. El dispositivo empalmador puede tener un comportamiento diferente si tiene un mayor conocimiento o si recibe una orden de una entidad de la cabecera de la red.

El objetivo de esta Recomendación es que, tanto en el modo empalme de programa como en el modo empalme de componente, el empalmador tenga gran libertad para elegir las unidades de acceso (vídeo y audio) en las que realizará el empalme. Algunos equipos podrán hacer empalmes con precisión a nivel de cuadro y otros no.



El splice\_time señalado puede o no estar en un cuadro ancla o I, y puede que el empalmador tenga que hacer un redondeo de cuadro según sus capacidades.

Parece haber consenso en la industria de que en el modo empalme de programa es más adecuado establecer el instante o tiempo de presentación de un cuadro de vídeo y dejar al empalmador la elección de la trama de audio más cercana o que se ajuste a sus métodos. Pero esto no se especifica en la Recomendación y es previsible que la industria encuentre la mejor manera de utilizar las herramientas (especialmente si difieren de las estipuladas aquí).

En el modo empalme de componente, el proceso es el mismo. Conviene usar un único splice\_time (denominado splice\_time por defecto), siendo opcional que cada componente tenga un splice\_time propio. La industria ha expresado preocupación por el hecho de que el creador de un mensaje facilite el tiempo exacto de todos los componentes para "ayudar" al empalmador, lo que posiblemente complicará el trabajo del empalmador (o hará que el empalmador no tenga en cuenta el splice\_time de algunos componentes).

Se entiende que SMPTE 312M [5] necesita un tiempo para cada componente con objeto de definir la unidad de acceso exacta de cada componente que hay que empalmar. La industria del cable está buscando una práctica diferente a la de tiempo por componente. Esta Recomendación permite que los componentes empiecen o terminen en instantes muy distintos de los correspondientes al inicio o fin de la pausa. Por ejemplo, es probable que haya que descargar en el adaptador multimedios un componente "datos" tal como un aplicativo (applet) Java, unos segundos antes del pase del comercial para que éste sea posible. Esta es la utilización principal de un único splice\_time por componente.

#### **I.5.5.1 Instrucciones de empalme de componente erróneas**

Cuando un mensaje empalme (en modo componente) especifica banderas de componente no válidas, el empalme se efectúa para todos los componentes correctamente identificados. El objetivo es que la inserción tenga éxito si es posible, incluso si la cadena completa de dispositivos infringe la norma. Los distintos casos posibles se explican a continuación.

Si un dispositivo elimina un flujo de audio de entre varios, después de insertado el mensaje de empalme, pero no consigue actualizar correctamente los mensajes de empalme (porque no los reconoce o por otras razones), la combinación de flujo y mensaje en el servidor/empalmador será no válida. Sin embargo, el servidor/empalmador puede aún realizar una inserción válida.

Si un dispositivo añade un componente (por ejemplo, un canal de datos), pero no consigue actualizar el mensaje de empalme (por las mismas razones aducidas en el párrafo anterior), puede aún realizar una inserción válida.

Siempre que se produzca una discrepancia entre el flujo real y el mensaje de empalme, el servidor/empalmador hará la inserción, a fin de que la cadena presente mayor robustez contra errores.

#### **I.5.6 Funcionalidad del rebobinado – Ejecución del rebobinado previo**

En los sistemas analógicos se necesitaba la función rebobinado previo (*pre-roll*) para dar tiempo al magnetoscopio a estabilizar la velocidad de la cinta con el inicio del espacio publicitario. Los tonos de aviso se enviaban normalmente entre 5 y 8 segundos antes del espacio (cada red utilizaba un tiempo propio y la compatibilidad era variable). El hecho de que el tono de aviso llegara antes se mantuvo durante la época de la inserción híbrida de los espacios publicitarios, en la que los magnetoscopios se fueron reemplazando por servidores y decodificadores MPEG, pero la inserción seguía siendo analógica. El envío anticipado del tono de aviso es también útil para la inserción digital del espacio publicitario, pues da al equipo de inserción el tiempo necesario para acceder a su base de datos de espacios publicitarios, determinar cuál pasar a continuación, acceder a sus lectores de discos y alimentar al decodificador MPEG.

La instrucción `splice_insert()` (inserción de empalme) se ha construido de manera que permite utilizar el rebobinado previo. También se puede repetir la instrucción `splice_insert()` para dar al sistema más robustez contra los errores.

La variante más sencilla es enviar la instrucción `splice_insert()` una vez aproximadamente 8 segundos antes del espacio publicitario, de forma semejante a la inserción analógica.

Se consigue una mejora si se envía la instrucción tres veces: 8, 6 y 4 segundos antes del espacio publicitario, con lo que se aumenta la redundancia y se evitan las oportunidades de pérdida de inserción. Perder un espacio publicitario a nivel nacional es un error muy caro. Cuando hay muchos mensajes para un solo espacio publicitario, el campo `splice_event_id` de todos esos mensajes debe ser idéntico. Cuando el primer tiempo de empalme sólo es aproximado y en los siguientes mensajes se especifica un tiempo de empalme con más precisión, se admite que los mensajes tengan tiempos de empalme diferentes. El empalmador sólo actúa sobre el último tiempo.

### **I.5.7 Acceso condicional y criptación del tono de aviso**

#### **I.5.7.1 ¿Qué criptar?**

El formato de la sección DPI permite que la cabida útil sea opcionalmente criptada. Esto incluye todos los datos desde "`splice_command_type`" hasta "`E_CRC_32`". Este mecanismo permite dar protección a las instrucciones presentes o futuras, protección que se necesita para luchar contra el pirateo (por ejemplo, los "commercial killers" o filtros de propaganda) y la manipulación malintencionada de los datos, y en aras de la confidencialidad cuando se utilizan dispositivos de inserción en cascada de inserciones publicitarias.

Esta especificación exige que haya un CRC criptado, de manera que cualquier receptor pueda verificar que, desde el origen, los datos criptados no han sufrido modificación, en especial debida al ruido, aunque un CRC normal suele detectar este tipo de corrupción. El CRC criptado tiene el fin fundamental para actuar como firma para saber si el receptor está autorizado a recibir el mensaje (es decir, tiene la palabra de control correcta), y también se utiliza para detectar las modificaciones intencionales de los datos criptados, pero no hace falta cuando no se cripta la sección.

#### **I.5.7.2 Funcionamiento del dispositivo de inserción de aviso**

El dispositivo de inserción de aviso se usa para crear las secciones de información de empalme, `splice_info_sections`, e insertarlas en el flujo de transporte. Cuando se necesita criptación, el insertador de pulso utiliza una clave fija, que introduce un operador, más el algoritmo seleccionado para criptar la sección antes de transmitirla.

Este dispositivo debe tener la capacidad de mantener múltiples claves simultáneas para el mismo programa, puesto que cada insertador de espacio publicitario en una cascada necesitará una o más claves diferentes para describir `splice_info_section`.

El dispositivo de inserción de aviso sólo necesita implementar uno de los métodos de criptación normalizados. En la práctica recomendada, el dispositivo de inserción de aviso tiene la capacidad de implementar todos los algoritmos normalizados.

#### **I.5.7.3 Funcionamiento del dispositivo de inserción de espacios publicitarios**

El dispositivo de inserción de espacios publicitarios procesa las `splice_info_sections` (secciones de información de empalme). Cuando la `splice_info_section` está criptada, el insertador de espacio publicitario sólo actuará sobre la sección si el proceso de descryptado es exitoso, lo que consiste en comprobar el CRC criptado antes de interpretar los datos. Si no lo logra, el dispositivo no está autorizado para interpretar la información que contiene la sección.

El insertador deja pasar la `splice_info_section` criptada hasta el próximo dispositivo con independencia de si el proceso de descryptado fue exitoso o no lo fue. Por motivos de seguridad, nunca revela el contenido de una sección descryptada.

Una vez descrita la sección, el dispositivo puede actuar sobre la información que contiene o descartarla. En el caso de una sección enviada sin criptación, la operación es la misma.

El dispositivo de inserción de espacio publicitario debe tener la capacidad de mantener 256 claves de descripción diferentes, y la que tiene que utilizar va indicada en el campo `cw_index` de la sección. El método de intercambio de claves queda fuera del alcance de este apéndice.

En el funcionamiento normal, se supone que se utilizan una o dos claves para cada par de dispositivos de envío/recepción. Hay un gran número de claves para que el insertador de anuncios se conecte con múltiples programas en un flujo de transporte, o para que los insertadores en cascada utilicen distintas gamas de `cw_index` al conectarse al mismo programa.

En el dispositivo de inserción de anuncios publicitarios hay que implementar todos los algoritmos de descripción definidos en esta Recomendación, con lo que podrá recibir las secciones criptadas de cualquier dispositivo de inserción de avisos. El dispositivo de inserción de avisos puede elegir cualquier algoritmo normalizado para proteger la sección, y el insertador de espacio publicitario debe poder describir cualquier mensaje que esté autorizado a recibir.

#### **I.5.7.4 Teoría del funcionamiento**

##### **I.5.7.4.1 Criptación o aleatorización**

El Grupo de Trabajo sobre DPI está estudiando la aleatorización del tren de inserción de programa digital. En este contexto, "aleatorización" significa que se aplica el algoritmo DES o el algoritmo común de aleatorización común (CSA, *common scrambling algorithm*) DVB, que se están utilizando en los servicios de vídeo y audio.

Para encontrar una solución, hay que tener en cuenta que la funcionalidad inserción de anuncios publicitarios, no suele estar incluida en el adaptador multimedios. Se ha previsto este modelo, pero no es el principal. La mayoría de los sistemas emplean un dispositivo de inserción de anuncios publicitarios en una cabecera digital (o cualquier otro punto de distribución), que suele ser una computadora. Un desaleatorizador necesita un circuito de diseño específico para desaleatorizar el tren. Puesto que todo el paquete de transporte está aleatorizado, el dispositivo de inserción de anuncio publicitario no tiene acceso a los datos de encabezamiento, como sucede en el modelo en uso. Es probable que haya que modificar partes de la información, como `pts_adjustment`, incluso si la sección no está desaleatorizada. Hay que tener en cuenta factores similares al crear el tren.

El modelo que se usa con `splice_info_section` se asemeja más al modelo de mensaje control de autorización (ECM, *entitlement control message*) que al modelo de tren elemental. Una sección ECM se cripta (y describe) de manera independiente y está autorizada por algún mecanismo externo. En el caso del ECM, la autorización la controla el mensaje gestión de autorización (EMM, *entitlement management message*). En el caso de `splice_info_section`, se trata de una autorización manual.

El modelo de clave fija se eligió por motivos de simplicidad. No se definió la distribución de la clave y se puede utilizar cualquier mecanismo que lleve la clave al descriptor cuando corresponde y con seguridad. Es lícito pensar que algún comité normalizará algún tipo de ECM y/o EMM para distribuir las claves en el flujo de transporte.

##### **I.5.7.4.2 Métodos de criptación normalizados**

La norma prevé tres tipos de algoritmo con el descriptor DES como bloque de construcción básico. En el caso del DES triple también es necesario contar con un criptador DES. Ya existen implementaciones de soporte lógico para el algoritmo DES.

##### **I.5.7.4.2.1 Relleno de sección**

Todos los algoritmos tipo DES (criptación por bloque en vez de por tren) exigen que la longitud de los datos sea un múltiplo exacto de 8 bytes, y para conseguirlo se suele necesitar el relleno. El

campo relleno para alineación, `alignment_stuffing`, puede estar presente vaya o no criptada la sección. El número de bytes de relleno se determina durante la interpretación de los datos. Nunca se utilizan los bytes de relleno, por lo que pueden tomar cualquier valor.

Para determinar la longitud del relleno se parte del hecho de que se conoce la longitud total del campo `section_length` (longitud de sección). También se conoce el tamaño exacto del encabezamiento, lo que sirve para determinar el inicio de la instrucción empalme, `splice_command`. La sintaxis de las instrucciones indica con detalle la longitud, que es:

`<section_length> +3 – <end_of_command> – <length_trailing>`

donde `<length_trailing>` es cuatro para las secciones no criptadas y ocho para las criptadas.

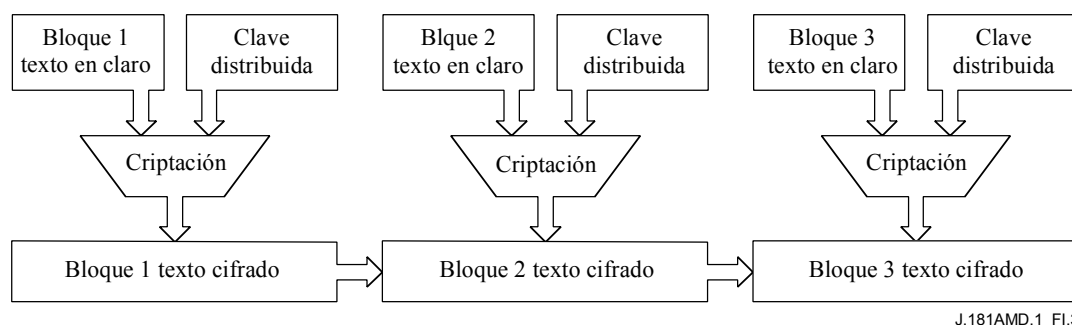
Con respecto a la aplicación del algoritmo, resulta más fácil no tener en cuenta el relleno, trabajar desde el principio y describir e interpretar la instrucción, y luego empezar desde el final para buscar el CRC.

#### I.5.7.4.2.2 Algoritmo DES en modo libro de código electrónico (ECB)

El DES ECB necesita una clave de 56 bits más 8 bits de paridad para conformar una clave de 64 bits, que es la que usa el algoritmo para criptar o descripiar un flujo. El algoritmo DES es simétrico, es decir que tanto el criptador como el descripiador utilizan la misma clave, pero en la práctica los algoritmos son ligeramente diferentes para conseguir la simetría.

Se propone que la clave completa de 64 bits se distribuya entre dos dispositivos. La clave se inserta como número hexadecimal de 16 dígitos, por ejemplo, `0x123456789ABCDEF0`. En notación técnica, el primer dígito es el más significativo, y el bit más significativo (MSB, *most significant bit*) de este cuarteto lo es también de la clave (bit 63). Los algoritmos de cifrado suelen utilizar la notación FIPS para representar las claves. En este caso, el bit más a la izquierda de la clave se numera como bit 1 y se sigue este orden hasta el bit 64 de la derecha. Por consiguiente, el bit 63 del valor de clave distribuida se cargará en el bit 1 del registro de clave inicial. Del mismo modo, el bit 0 del valor de clave se carga en el bit 64 del registro de clave.

El método de criptación en modo libro de código electrónico utiliza la *misma* clave para cada bloque de 8 bytes del mensaje original. En la figura I.3 se muestra un ejemplo de criptación de un mensaje simple de tres bloques; las flechas representan las operaciones. En el primer nivel, la clave se carga en el registro de clave. De izquierda a derecha, los datos pasan por el registro de criptación, y en el último nivel se aplica el algoritmo DES.

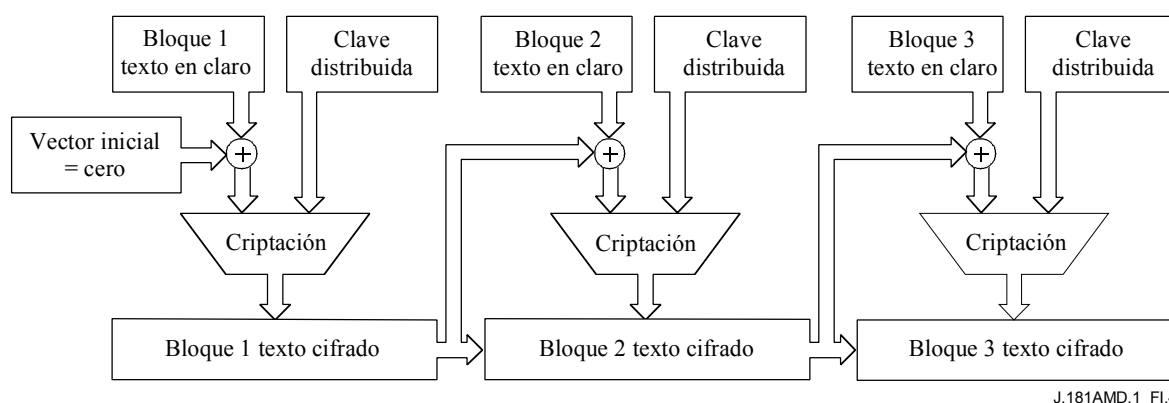


**Figura I.3/J.181 – Ejemplo de DES con ECB**

#### I.5.7.4.2.3 Algoritmo DES en modo concatenación de bloques cifrados (CBC)

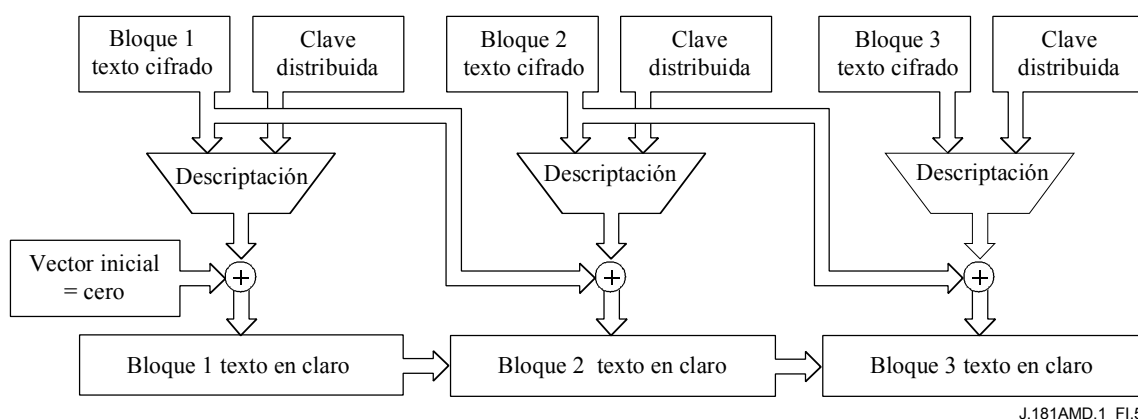
El DES CBC necesita una clave de 56 bits más 8 bits de paridad para conformar una clave de 64 bits, que es la que se usa para criptar o descripiar un tren de bits. Pueden utilizarse las mismas normas de ordenación de bits y los mismos métodos de distribución de claves que para el ECB descrito en I.5.7.4.2.2.

El método de criptación en modo concatenación de bloques cifrados utiliza una clave diferente para cada bloque de 8 bytes del mensaje original.



**Figura I.4/J.181 – Ejemplo de criptación DES con CBC**

En la figura I.4 se observa que los datos de texto en claro son modificados por el resultado del bloque de cifrado anterior. En cuanto al primer bloque, hace falta un "vector inicial" para aplicar la OR-exclusiva. El hecho de que el vector inicial, se conozca o no, no aporta más seguridad, por lo que para este sistema el vector inicial puede ser cero, lo que elimina la necesidad de distribuirlo con la clave.



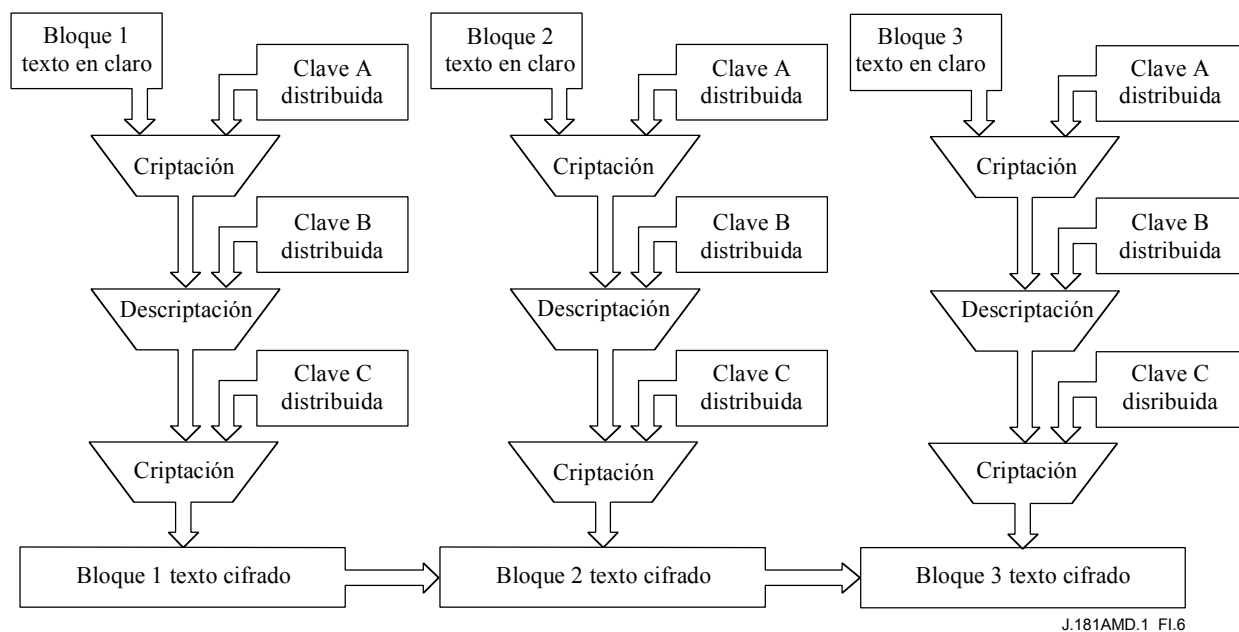
**Figura I.5/J.181 – Ejemplo de descryptación DES con CBC**

En la figura I.5 se observa que la descryptación DES CBC es ligeramente diferente de la criptación. El bloque OR-exclusiva necesita que el bloque anterior de datos criptados llegue a la clave de cifrado correcta para aplicar el algoritmo DES. En el descryptador, este bloque criptado se deriva de los datos transmitidos.

#### **I.5.7.4.2.4 Algoritmo de DES triple en modo ECB (método EDE)**

El DES triple utiliza el algoritmo DES normalizado, pero lo aplica 3 veces a cada bloque de datos de entrada, con lo que se consigue mucha más seguridad. El inconveniente es que hay que distribuir una clave de 192 bits (que son tres claves de 64 bits).

Mediante dos algoritmos y tres vueltas (la criptación y la descriptación son diferentes), se pueden generar ocho variantes diferentes de DES<sup>1</sup> triple, que se designan con un código de tres letras. De entre ellas se elige la más común para el sistema de empalme, la llamada EDE (*encryption-decryption-encryption*), pues en la primera vuelta se cripta, en la segunda se descripta, y en la tercera se cripta. Para facilitar la operación, el DES triple se usa en modo ECB<sup>2</sup>.

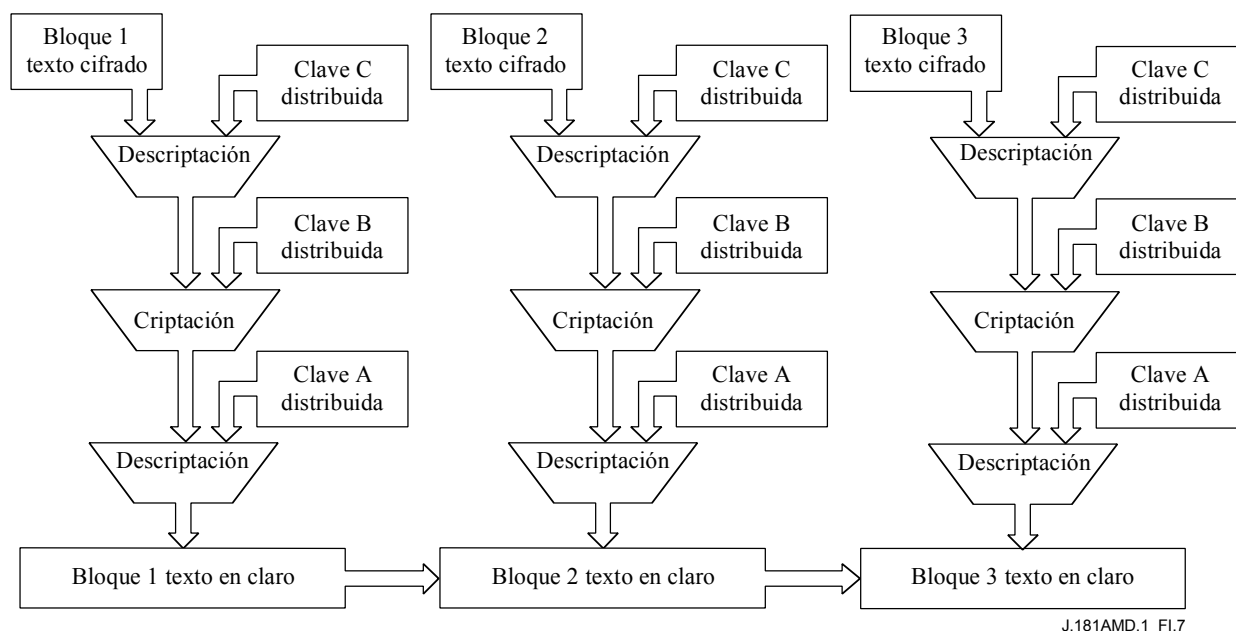


**Figura I.6/J.181 – Ejemplo de criptación DES triple con ECB**

La descriptación con DES triple se hace con la aplicación inversa del algoritmo de criptación y por ello, el diagrama de bloques de la descriptación es ligeramente diferente, como se muestra en la figura I.7.

<sup>1</sup> El DES triple también tiene dos claves, lo que se consigue mediante la simulación CLAVE A == CLAVE C.

<sup>2</sup> También se puede usar el DES triple en modo CBC, pero este modo no es un método normalizado.



**Figura I.7/J.181 – Ejemplo de descripción DES triple con ECB**

En DES triple se distribuyen tres claves de 64 bits, denominadas A, B y C, cuya ordenación es igual a la de la clave de las variantes DES de una sola vuelta. En los diagramas de bloques vemos que en la criptación se usa primero la clave A, luego la B y después la C. En la descripción, las claves se aplican en orden inverso.

#### **I.5.7.4.3 Métodos de criptación privados**

Casi la mitad de los valores de algoritmo de criptación, (`encryption_algorithm`), se han reservado para uso privado, por lo que esta Recomendación no los asigna. Por su índole, se prestan a interpretaciones erróneas, lo que imposibilita la normalización de un método para seleccionar los algoritmos privados que crean los usuarios, a menos que exista una autoridad de registro. Puesto que no existe tal autoridad, no se ha definido un método de coordinación de algoritmos.

El problema se plantea cuando dos entidades independientes utilizan el mismo valor de `encryption_algorithm` para algoritmos diferentes. Cuando un dispositivo de inserción de aviso de la entidad A cripta la sección, que luego recibe la entidad B, la descripción no funcionará porque el equipo cree que no está autorizado, puesto que falla el CRC, aunque en ambos extremos se haya usado la misma clave.

### **I.5.8 Uso del identificador exclusivo de program (`unique_program_id`)**

#### **I.5.8.1 ¿Qué es un "programa"?**

Dos contenidos que ofrece una cadena se dividen normalmente en varios programas. Un programa puede tener prácticamente cualquier duración, aunque la mayoría suelen ser de 30 ó 60 minutos. Ciertos programas, como las noticias deportivas o los avances deportivos, duran escasos minutos, mientras que otros tipos, como las películas, las retransmisiones deportivas o las ceremonias de entrega de premios en directo, pueden durar varias horas.

#### **I.5.8.2 ¿Qué es "program\_id"?**

El identificador de programa (`program_id`) es una manera "abreviada" de identificar un programa específico en una cadena, la cual proporciona su valor.

### **I.5.8.3    ¿Por qué hay que identificar y diferenciar los programas?**

Para el anunciante, algunos programas tienen más valor que otros. El valor relativo de un programa no depende de su duración ni de su hora de emisión, sino de su capacidad para captar audiencia.

Algunos anunciantes compran tiempo de publicidad en función de las franjas horarias (por ejemplo, de 10 a 16 horas, de 20 a 23 horas, etc.), mientras que otros quieren que sus anuncios publicitarios pasen con un programa específico, incluso como pausa específica dentro de ese programa. La parrilla tiene programas regulares (informativos o episodios de una serie), o especiales, como los partidos de béisbol, las ceremonias de entrega de premios o las entrevistas en directo.

Casi siempre los programas especiales son más caros que los regulares. Los anunciantes que aceptan pagar estos precios más altos esperan que sus anuncios publicitarios pasen dentro del especial que compraron, y no están dispuestos a pagar si éstos no se emiten en ese contexto.

### **I.5.8.4    ¿Por qué la hora de emisión no identifica el programa?**

Las redes pueden modificar la programación en el último momento. A veces, se informa de dichos cambios a las compañías asociadas antes de enviar la programación de los anuncios publicitarios a las cabeceras de la red, pero puede suceder que esta información no esté disponible hasta que se hayan ubicado los programas y las pausas publicitarias en la parrilla.

Con frecuencia, las modificaciones de programación son resultado de cambios no planificados que se hacen a último momento en los programas en vivo, como retransmisiones deportivas, ceremonias de entrega de premios y producción de informativos en directo. Estos programas pueden ubicarse en la parrilla en el último minuto, pueden durar más o menos tiempo que el previsto en principio o se pueden cancelar por motivos meteorológicos o de otro tipo.

Si la parrilla se hace basándose únicamente en el "tiempo" (o instante de emisión), un anuncio cuyo pase estaba previsto en principio dentro de un programa especial puede quedar incluido en un programa "regular", y cuando esto sucede, el anunciante no lo pagará. En el peor de los casos, la compañía asociada local puede cobrar inadvertidamente el anuncio como si hubiese pasado durante un programa especial, aunque de hecho no haya sido así, lo que puede resultar perjudicial para la relación con el cliente.

### **I.5.8.5    ¿Cómo puede solucionar el problema el `unique_program_id` ?**

Si un evento de empalme se identifica con un `unique_program_id`, los vendedores de sistemas de venta/tráfico (sistemas informatizados que ubican en la parrilla los comerciales que el departamento de ventas de espacios publicitarios especifica para sus clientes podrán especificar cuál comercial se transmite con qué programa. También podrán indicar cuáles son los comerciales que los pueden sustituir (sin pérdidas de ingresos) si dichos anuncios publicitarios no se difunden a la hora prevista.

Gracias a esta información, los vendedores de sistemas de inserción de publicidad sabrán cuál es el comercial que hay que transmitir cuando el `unique_program_id` del evento empalme no coincida con el `unique_program_id` del mensaje publicitario previsto para ese horario. La adición de este campo y la implementación de la funcionalidad necesitará un esfuerzo por parte de los vendedores de sistemas de inserción de publicidad. Los antiguos sistemas analógicos no contaban con esta funcionalidad, aunque es extremadamente importante.

## **I.5.9    Uso de los campos del espacio publicitario**

### **I.5.9.1    ¿Qué es un espacio publicitario?**

Un espacio publicitario es una oportunidad que ofrece la red a las empresas afiliadas locales para insertar un evento comercial en un programa. El comienzo de un espacio se indica mediante un evento empalme en el tren de programas. Su duración puede variar entre unos pocos segundos y varios minutos.



Normalmente, un espacio publicitario dura sesenta segundos, aunque los hay de noventa segundos y hasta de dos minutos. Puesto que la mayoría de los anuncios individuales duran treinta segundos, en cada espacio se insertan dos o más.

#### **1.5.9.2 ¿Cuántos espacios publicitarios hay en un programa?**

La duración del programa es el predictor más común de la cantidad de espacios publicitarios que contendrá; por regla general, hay un espacio publicitario de sesenta segundos por cada treinta minutos de programa. El número exacto de espacios publicitarios se indica en la parrilla de programas que proporciona la cadena. Si bien el número y la longitud de los espacios publicitarios pueden variar según el programa, la parrilla de programación le permite a la empresa afiliada local conocer con antelación cuántos espacios publicitarios es previsible que haya en un programa.

#### **1.5.9.3 ¿Por qué es importante identificar los espacios publicitarios de un programa?**

En el más simple de los casos, el anunciante puede haber comprado una "posición" específica dentro de un programa, ya que no es lo mismo que ese anuncio pase en la primera, la tercera o la sexta posición del programa. El anunciante puede especificar (y pagar un recargo) que quiere "el último espacio publicitario" de la retransmisión de un partido de baloncesto, porque supone que en caso de un partido reñido, es más probable que los espectadores se queden viendo el programa.

En otros casos, puede suceder que varios anunciantes (en desconexiones locales o nacionales/regionales) se hayan puesto de acuerdo para dividir los espacios publicitarios de un programa con arreglo a sus intereses. En este caso, es importante que cada anunciante sepa el espacio publicitario asociado con cada evento empalme para mantener sus tablas de inserción "sincronizadas" con el programa.

#### **1.5.9.4 ¿Cómo asegura esta función el campo "espacio publicitario"?**

Si cada espacio publicitario del programa está identificado unívocamente, los sistemas de venta y tráfico, así como los vendedores de equipos de inserciones publicitarias, pueden asociar la identificación del espacio publicitario con los anuncios que deben pasarse en un espacio publicitario específico.

Si por algún motivo (fallo de la cadena o del afiliado local) se "omite" un evento empalme, el siguiente evento empalme, con su propio campo identificador de espacio publicitario, permitirá que las inserciones publicitarias se resincronicen con el evento correcto.

#### **1.5.9.5 ¿Qué hace el campo cómputo de espacios publicitarios?**

Este campo (`avail_count`) cuenta el número total de espacios publicitarios previstos en un programa completo. Esta información permite al dispositivo de inserción de publicidad garantizar que el número previsto de espacios publicitarios se corresponde con el número que ejecuta la cadena.

El valor que figura en el campo espacio publicitario puede ser superior al valor del campo `avail_count`. Esto ocurre, por ejemplo, cuando una retransmisión deportiva dura más de lo previsto. Si el proveedor de contenido de la cadena envía mensajes de aviso durante este periodo "de prórroga", el valor incluido en el campo espacio publicitario será superior al del campo `avail_count`.

### **1.5.10 Utilización de los avisos de inserción**

La utilización de instrucciones de empalme para actualizar, modificar, anular y terminar eventos se aclara en las cláusulas siguientes.

#### **1.5.10.1 Iniciar una pausa**

El mensaje de aviso DPI indica al servidor/empalmador una oportunidad de empalme para pasar de la programación de la red a una interrupción publicitaria o a la inversa. En la instrucción "splice insert" (inserción de empalme) el indicador sale de la red, `out_of_network`, se fija a "1" cuando el empalme se hace para salir de la red, y a "0" cuando se hace para entrar en la red. El punto de

empalme se obtiene de la estructura de `splice_time()` en la `splice_info_section()`. Cuando el empalme se hace para salir de la red, el mensaje de aviso DPI debe llegar al servidor/empalmador al menos cuatro segundos antes del tiempo de empalme; dicho de otra manera, se trata de un "rebobinado previo" de cuatro segundos. El servidor/empalmador decide cuál es el mejor punto de empalme con respecto al tiempo de empalme.

Un mensaje de aviso que indica un empalme para salir de la red que todavía está fuera de la ventana de rebobinado previo especificada, se puede cancelar. La orden "cancelar" consiste en enviar un mensaje de aviso DPI con la bandera indicador cancelar evento, `cancel_event_indicator`, puesta a "1". Si la orden se recibe durante el "rebobinado previo" o después de iniciada la pausa, no se cumple.

Se puede actualizar un evento empalme para salir de la red mediante un mensaje de actualización, sin tener que cancelar un mensaje transmitido anteriormente para el mismo evento empalme. Se puede actualizar un evento varias veces. El empalmador considerará válido el último mensaje que cumple las restricciones de temporización de "rebobinado previo", anulando todos los mensajes anteriores correspondientes al mismo evento empalme.

Cuando hay que parar una pausa y no se conoce el instante preciso, se admite el envío de una instrucción "cancelar" más otra "parar" o "terminar". El empalmador actuará sobre cualquiera de los dos mensajes que sea válido y descartará el otro. Las instrucciones "parar" y "terminar" se describen en la cláusula I.5.10.2.

#### **I.5.10.2 Terminar una pausa**

Hay dos maneras de terminar una pausa en curso. Cuando ya está en marcha o ha terminado el proceso de vuelta a la red, o la pausa no está en "rebobinado previo", el servidor/empalmador ignorará la instrucción "terminar" y/o "parar".

- La instrucción "terminar" se genera enviando un mensaje con la bandera empalme inmediato, `splice_immediate`, puesta a "1" y la bandera indicador salir de la red, `out_of_network_indicator`, puesta a "0" en `splice_info_section()`. El servidor/empalmador se vuelve a conectar a la red inmediatamente (lo antes posible).
- La instrucción "parar" se genera enviando un mensaje con la bandera `splice_immediate` puesta a "0" y la bandera `out_of_network_indicator` puesta a "0" en `splice_info_section()`. El `splice_time` indica el punto de empalme en la red.

#### **I.5.10.3 Uso compartido de una pausa por varios anuncios publicitarios**

El mensaje de aviso DPI puede indicar la oportunidad de insertar varios anuncios publicitarios a servidores distintos. En este caso, el empalmador recibe el mensaje de aviso DPI en el tren que genera la red y lo pasa a cada uno de los servidores, que se encargan de la programación y arbitraje. Cada servidor manejará el mensaje de aviso DPI para colocar cada anuncio publicitario en la posición correcta de la pausa. Entonces, la pausa queda dividida, de forma que un servidor ocupa la primera posición y el otro, la posición siguiente, o cualquier otro tipo de combinación de servidores y posiciones en la pausa.

### **I.5.11 Creación y uso de descriptores de empalme privados**

#### **I.5.11.1 ¿Qué es el descriptor?**

"Descriptor" es un término de la norma MPEG-2. El descriptor se utiliza para introducir sintaxis nueva en una norma existente, pero de forma que los equipos existentes puedan saltársela. Como opción, el descriptor puede ir en una sección de información. Puesto que su formato es conocido, el equipo receptor puede obviarlo sin causar pérdidas de sincronización durante el proceso de análisis.

Otro uso del descriptor es proporcionar sintaxis opcional. Al igual que con la sintaxis nueva, cualquier equipo existente que no entienda la información opcional puede obviarla.

#### **I.5.11.1.1 El problema**

Los descriptores normalizados tienen un problema cuando los usuarios crean descriptores privados para uso propio, porque los diferentes usuarios pueden tener el mismo número en una etiqueta de descriptor, pero una sintaxis totalmente distinta en la cabida útil del descriptor. Esto no plantea problema si el equipo receptor no entiende el descriptor, pero sí es un problema cuando intenta entender el descriptor equivocado.

MPEG y DVB han solucionado este problema gracias a un descriptor específico. MPEG no describe formalmente su utilización, pero DVB sí lo hace. Básicamente, el descriptor de datos privados, `private_data_descriptor`, tiene un identificador de 32 bits que modifica la manera en que el receptor "entiende". Cuando el dispositivo se encuentra con un identificador que entiende, acepta las etiquetas de descriptores privados de usuario; cuando encuentra un identificador que no entiende, o antes de ver cualquier identificador, deja de aceptar etiquetas privadas de usuario.

Este método funciona, pero tiene un defecto, y es que en principio se previó que este descriptor fuese una opción, por lo que muchas empresas han fabricado dispositivos que no envían (ni reciben) datos privados, y al no aprovechar la solución, subsiste el problema. Tampoco se utiliza un MPEG, porque no existe una norma obligatoria en relación con el descriptor.

#### **I.5.11.1.2 La solución**

El identificador de 32 bits que contiene `splice_info_descriptors` cumple el mismo propósito, pero el estar colocado en el encabezamiento del descriptor lo torna obligatorio. El inconveniente es que triplica el tamaño de la cabecera. Puesto que es previsible que las `splice_info_sections` sean bastante pequeñas y quepan fácilmente en un solo paquete de transporte, los bytes que se añaden al encabezamiento no deben causar perjuicio.

#### **I.5.11.2 Registro**

El identificador que va en el encabezamiento del descriptor es un valor autorregistrado. Se supone que con  $2^{32}$  combinaciones, es muy raro que dos empresas elijan al azar el mismo valor. La empresa que quiera utilizar un descriptor privado crea un valor de 32 bits para uso propio. Un sencillo mecanismo "aleatorio" es utilizar el número de 32 bits como cadena de 4 caracteres, utilizando, el acrónimo del nombre de la empresa para obtener esos caracteres. Por ejemplo, el identificador `registration_descriptor` normal es `0x43554549` ("CUEI" en ASCII).

Evidentemente, si todas las empresas que crean descriptores privados utilizan el mismo identificador (como `0x00000001`), el problema seguirá existiendo.

Una vez elegido un identificador, puede utilizarse la etiqueta de descriptor, con lo que la empresa dispone de hasta 256 descriptores privados antes de tener que crear un identificador nuevo.

#### **I.5.11.3 Creación de descriptores privados compatibles**

Suponiendo que se ha elegido un identificador, se puede definir un descriptor privado. Todos los descriptores, sean privados o nuevos que estén incluidos en la Recomendación, tienen los mismos seis bytes en el encabezamiento. En el cuadro I.2 se muestra el esquema de todos los descriptores incluidos en la `splice_info_section`; se incluyen todos los descriptores presentes y futuros que define la norma.

**Cuadro I.2/J.181 – Splice\_descriptor()**

Sintaxis	Bits	Descripción
<code>splice_descriptor() {</code>		
<b>splice_descriptor_tag</b>	8	<b>uimsbf</b>
<b>descriptor_length</b>	8	<b>uimsbf</b>
<b>identifier</b>	32	<b>uimsbf</b>
for(i=0; i<N; i++) {		
<b>private_byte</b>	8	<b>uimsbf</b>
}		
<code>}</code>		

Este splice\_descriptor no es un descriptor real, sino un modelo para todos los descriptores concretos. La norma sólo define un descriptor real el avail\_descriptor (descriptor de espacio publicitario), que se reproduce en el cuadro I.3 para mostrar cómo se crea un nuevo descriptor.

**Cuadro I.3/J.181 – Avail\_descriptor()**

Sintaxis	Bits	Descripción
<code>avail_descriptor() {</code>		
<b>splice_descriptor_tag</b>	8	<b>uimsbf</b>
<b>descriptor_length</b>	8	<b>uimsbf</b>
<b>identifier</b>	32	<b>uimsbf</b>
<b>provider_avail_id</b>	32	<b>uimsbf</b>
<code>}</code>		

Paso 1: Elegir un identificador. En este ejemplo se usa "CUEI".

Paso 2: Elegir una splice\_descriptor\_tag. Dado que éste fue el primer descriptor definido, se eligió el valor cero.

Paso 3: Definir la sintaxis privada, que constituye el cuerpo del descriptor. En la plantilla, la cabida útil del descriptor se representa como un bucle genérico que contiene bytes de datos. Esto es exactamente lo que ve cualquier dispositivo que no entiende el descriptor. Utilizando el campo descriptor\_length, el dispositivo puede obviar la carga y seguir procesando el siguiente descriptor.

En este ejemplo, la cabida útil tiene exactamente un campo formado por 32 bits.

La única restricción impuesta a la cabida útil de un descriptor es que tenga un número exacto de bytes (múltiplo de 8 bits), y un total inferior a 250 bytes<sup>3</sup>. Los campos de la cabida útil del descriptor pueden contener cualquier número de bits. Si la sintaxis necesaria no tiene un múltiplo de 8 bits, hay que insertar bits reservados para llegar con el relleno a un múltiplo de 8.

#### **I.5.11.4 Uso de avail\_descriptor**

El campo de identificación de proveedor de espacio publicitario, provider\_avail\_id, es un valor uimsbf de 32 bits que se puede usar de varias maneras. En los actuales sistemas DTMF analógicos se utilizan mensajes de aviso de 4 caracteres con un código de tres dígitos para identificar la pausa. En general, el cuarto carácter es '\*' para un tono de inicio, que indica la duración de un rebobinado previo predeterminado, o '#' si se trata de un tono de parada inmediata. Un ejemplo es "635\*".

Los actuales sistemas analógicos utilizan estos mensajes para identificar los distintos tipos de información de aviso, como duración, hora del día, avance informativo o pausa privada.

Se recomienda utilizar las demás características de esta Recomendación que identifican el inicio o el fin del mensaje de aviso y luego insertar simplemente el código de identificación como un

<sup>3</sup> El convenio en MPEG, DVB y ATSC es que la longitud total del descriptor debe ser de 256 bytes, por lo que se limita la longitud del descriptor para que esto se cumpla.

número entero de 32 bits. Por ejemplo, si una red utiliza la secuencia de caracteres "017\*" como ID del aviso analógico, se inserta el valor 17 en el campo `provider_avail_id` y se fija el bit `out_of_network_indicator` en el mensaje de aviso. El mensaje fin tiene que tener el mismo `provider_avail_id` que el mensaje inicio.

### **1.5.12 Manejo de las discontinuidades de la base de tiempo**

En I.6.1.1 se indica que un mensaje de aviso que se envíe justo antes de una discontinuidad de la base de tiempos (TDB, *time base discontinuity*) no puede incluir un `splice_time` expresado en la nueva base de tiempo que sigue a la TDB. Una de las razones que fundamentan este requisito es que, si no se cumple, al hacer la remultiplexación resultará muy difícil preservar la validez del campo `splice_time`. Si no se cumple este requisito, puede suceder que el dispositivo de remultiplexación (remux) o de resellado temporal necesite el valor de un sello de tiempo de presentación, PTS, que le llegará varios cientos de milisegundos más tarde.

Por ello, se decidió que el campo `splice_time` se debe expresar siempre en la base de tiempo que se está usando, es decir la correspondiente al lugar en que el paquete mensaje de aviso que contiene este campo se inserta en el tren de señales. Por consiguiente, el insertador de mensaje de aviso, incluso en los pocos casos en que sepa que puede hacer otra cosa, tiene que comportarse como si no hubiese una TBD durante el "tiempo de armado". El "tiempo de armado" es el que transcurre entre el primer mensaje de aviso que se refiere a un evento y el evento propiamente dicho.

Es probable que el campo `splice_time` tenga que apuntar a una imagen situada después de la TBD, imagen que está asociada con una PTS expresada en la base de tiempo nueva. El mensaje de aviso no está autorizado a usar dicho valor de PTS y, en cambio, debe utilizar el valor PTS al que se habría asociado la imagen de no existir la TBD, o si se hubiese eliminado, por ejemplo, cambiando todos los sellos de tiempo que siguen a la TBD a la base de tiempo antigua.

Esto simplifica el trabajo de un remux simple, que transfiere las marcas temporales que entran hasta la salida (sin cambiar los sellos de tiempo PTS y PCR y, por tanto, sin eliminar las discontinuidades de la base de tiempo presentes a la entrada). Los remultiplexadores simples se encargan de preservar la validez del tiempo de empalme que lleva el mensaje de aviso que pasa por ellos. Por consiguiente, tiene que garantizar que el mensaje de aviso no puede cruzar la frontera de TBD, ya que ello invalidará el tiempo de empalme del mensaje de aviso.

Los remultiplexadores que cambian continuamente la PCR y el PTS a la salida, con lo que eliminan automáticamente las discontinuidades de la base de tiempo, tienen que preservar también la validez del `splice_time` del mensaje de aviso. Deben saber en qué lado de la TBD que llega está el mensaje de aviso para hacer corresponder correctamente el mensaje de aviso a la salida del dispositivo (y poner el sello de tiempo correspondiente en el campo `splice_time`).

Para los remultiplexadores, tanto simples como de resellado, se recomienda vivamente que nunca se reduzca el tiempo de armado al procesar el tren de señales y los mensajes de aviso que contiene (es decir, que nunca se añada un retardo al mensaje de aviso con respecto al reloj del sistema del tren). Si se reduce el tiempo de armado de un mensaje de aviso que ya se encuentra en el límite más bajo del tiempo de armado recomendado u obligatorio, se infringirán estos requisitos o recomendaciones.

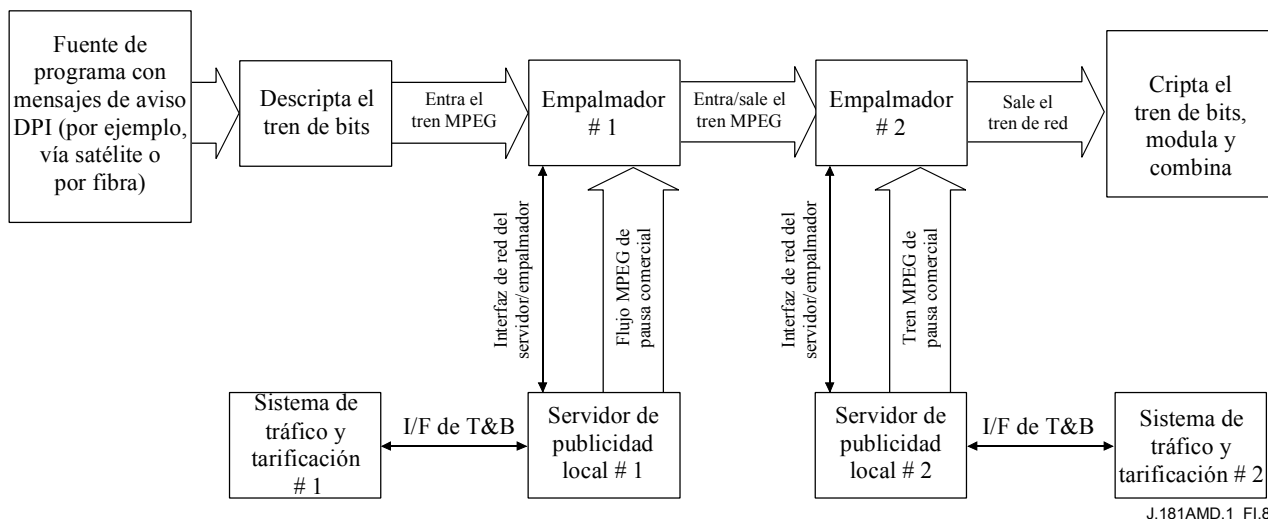
El campo `splice_time` va en la parte posiblemente criptada del mensaje de aviso, lo que lo hace potencialmente inaccesible para el dispositivo de resellado. Si se utiliza la criptación, éste debe actualizar el campo `pts_adjustment`, que se encuentra en la parte no criptada de la sintaxis justamente por este motivo. Cuando no se utiliza la criptación, el resellador puede actualizar el campo `splice_time` directamente o modificar el campo `pts_adjustment`.

Cuando hay una TBD durante el tiempo de armado, el dispositivo que recibe y actúa sobre el mensaje de aviso, por ejemplo, el empalmador/servidor, tiene que traducir la antigua base de tiempo a la nueva, es decir traducir el `splice_time` del mensaje de aviso (ajustado según el valor del campo

pts\_adjustment) a la base de tiempo nueva para obtener la PTS del cuadro en el que se efectuará el empalme.

### 1.5.13 Dispositivos empalmadores en cascada

Esta Recomendación permite utilizar empalmadores/servidores en cascada, puesto que un empalmador es sólo una forma de remultiplexador. En la figura I.8 se muestra el caso en que dos empalmadores/servidores en cascada realizan la inserción. Las fronteras externas del sistema son las mismas que en el caso de un único empalmador. No es necesario que haya comunicación directa entre los dos servidores.



**Figura I.8/J.181 – Dispositivos empalmadores/servidores en cascada**

Hay que tener en cuenta algunas cuestiones cuando se hacen trabajar los dispositivos en cascada, que se exponen a continuación.

#### 1.5.13.1 Resellado temporal de mensajes de aviso

Los empalmadores pueden manejar sus tiempos de salida (PCR, PTS) de diversas maneras. Si un empalmador siempre transfiere el dominio de tiempo de la producción de red a su salida, incluso cuando un anuncio sustituye a la producción de red, no tiene que resellar el mensaje de aviso que ha pasado. Pero si el empalmador siempre o momentáneamente modifica los sellos de PCR y PTS (en el momento del mensaje de aviso), tiene que devolver el mensaje de aviso pasado a su condición verdadera. Esta Recomendación tiene un campo no criptado, pts\_adjustment, que permite que se vuelva a sellar el paquete de aviso mediante un cambio de este campo de adaptación, en vez de tener que ir a la variable o variables pts\_time() y volver a sellarlas una por una. Al principio, esta función se incluyó para la operación de criptado cuando un multiplexador/empalmador intermedio no era capaz de describir el mensaje, pero sí podía volver a sellar los paquetes de aviso. Ahora bien, este campo facilita el trabajo de los mux intermedios en todos los casos. El empalmador tiene que añadir el campo pts\_adjustment a todos los campos pts\_time() para obtener el tiempo de empalme real.

#### 1.5.13.2 Propagación de mensajes de aviso

Los empalmadores tienen que tener algún tipo de configuración variable para permitir el paso de un mensaje de aviso con el tiempo corregido, y también la opción de impedir que el mensaje de aviso se siga propagando (bloqueo). Corresponde a la red y al operador decidir hasta dónde se puede propagar el mensaje de aviso. El mensaje de aviso puede recorrer todo el camino hasta el adaptador multimedia, ya que muy probablemente la red de cable lo criptará. Entonces, el adaptador

multimedia lo podrá describir como parte de un sistema cerrado, con lo que, casi con toda certeza, impedirá el funcionamiento de ciertas formas de bloqueadores de anuncios publicitarios (comercial killers).

### **I.5.13.3 Retardo**

Los empalmadores suelen causar cierto retardo en el tren MPEG. Esto es más preocupante cuando los empalmadores funcionan en cascada, puesto que los retardos se acumulan. Las únicas aplicaciones a las que el retardo afecta considerablemente son las operaciones de distribución simultánea y las variables para las que el tiempo es crítico (transmisión de las cotizaciones en bolsa en tiempo real). Con los actuales sistemas de audio de gran calidad, se piensa que la distribución simultánea no es realmente necesaria. Hoy día, los datos que se transmiten por mecanismos de difusión de televisión ya están diferidos, y es habitual indicarlo claramente en la pantalla junto con los datos. En cualquiera de las dos situaciones, los actuales empalmadores suelen añadir uno o dos segundos de retardo, cosa que, por el momento, no parece preocupar a los operadores de cable.

### **I.5.13.4 Cascada lógica**

La API de servidor/empalmador de publicidad [1] permite solucionar los inconvenientes de los empalmadores que trabajan en cascada, pues un empalmador físico puede manejar múltiples servidores que atienden la publicidad local/regional/nacional. Esta cascada lógica elimina la necesidad de disponer de tres empalmadores físicos en cascada y el retardo adicional que introducen.

## **I.6 Información adicional**

### **I.6.1 Consideraciones para evaluar los dispositivos empalmadores MPEG-2**

#### **I.6.1.1 Generalidades**

El objetivo de esta cláusula es proporcionar información para entender el funcionamiento de los empalmadores MPEG-2 conforme a las normas MPEG-2 publicadas. La única finalidad de esta cláusula informativa es aclarar el tema.

Los empalmadores MPEG-2 son una tecnología emergente, cuyas capacidades y limitaciones tienden a ser muy dependientes del entorno en que funcionan, y en general no se entienden correctamente. Para especificar las capacidades que debe tener un empalmador y qué nivel de calidad de funcionamiento debe alcanzar, es necesario conocer los factores que afectan a su comportamiento e identificar el contexto en el que funcionará. Por ejemplo, no tiene sentido analizar cuestiones como la exactitud de cuadro si no se estipulan claramente las condiciones de medición. También resulta útil conocer el enfoque tecnológico general que usa el empalmador, ya que puede dar una idea de sus capacidades y limitaciones generales.

#### **I.6.1.2 Tecnología de empalme**

Actualmente, hay varios enfoques para el empalme MPEG-2, cada uno de ellos con sus ventajas y desventajas. A continuación, se intenta categorizarlos.

##### **I.6.1.2.1 Empalme de tren de transporte (TS, *transport stream*)**

Los empalmadores TS (flujo de transporte) funcionan en el nivel de tren de transporte MPEG-2 y simplemente conmutan de un tren de paquete de transporte a otro. Es habitual que el empalmador de TS vuelva a mapear los PID, pero sin modificar el estado VBV del tren asociado con el resellado de PTS/DTS.

Un empalmador TS no conoce el estado de los trenes elementales que empalma. Para obtener un resultado que mantenga la integridad del decodificador en todo momento, el empalmador debe operar sobre trenes que hayan sido condicionados para garantizar que los puntos de empalme elegidos cumplen determinados requisitos (por ejemplo, la norma SMPTE 312M [5]).

Puesto que sólo operan sobre los datos del TS, y suponen que el tren se ha condicionado adecuadamente (y que tienen instrucción de empalmar únicamente en los puntos válidos), los empalmadores de TS son el tipo de empalmador de más fácil implementación.

#### **I.6.1.2.2 Empalme de tren elemental (ES, *elementary stream*)**

Los empalmadores de tren elemental de paquetes (PES) funcionan en el nivel de tren elemental MPEG-2 y modifican los datos del tren elemental según sea necesario para realizar el empalme. Esto les permite modificar el estado VBV de los trenes de vídeo según convenga y manejar adecuadamente ciertas situaciones, como el empalme de material a 3:2. También pueden silenciar los trenes de audio en los puntos de empalme para evitar los chasquidos característicos de los programas mal editados.

Puesto que tienen la capacidad de modificar los datos del tren elemental instantáneamente, los empalmadores PES no tienen las restricciones de contenido impuestas a los empalmadores TS para alcanzar el mismo nivel de calidad de funcionamiento.

#### **I.6.1.2.3 Empalme a nivel de imagen**

##### **I.6.1.2.3.1 Recodificación parcial**

Los empalmadores de imagen funcionan a un nivel más profundo que los empalmadores PES, pues operan sobre los datos de imagen que transporta el tren elemental de vídeo. Los datos de imagen no se descomprimen y recomprimen, pero se pueden hacer otras operaciones, como la recuantización, para manejar la velocidad binaria y el estado VBV del flujo. Este tipo de empalmador es más complejo que el empalmador PES, pero puede funcionar mejor en determinadas situaciones, pues su control del tren de bits es más fino. Además, hay que poner especial cuidado en evitar las pérdidas de calidad de imagen.

##### **I.6.1.2.3.2 Recodificación total**

Los empalmadores de recodificación decodifican los datos MPEG, hacen el empalme en la banda de base y recodifican el resultado. Estos empalmadores llevan incorporado un decodificador y codificador MPEG por canal, y su implementación es la más costosa. Además, hay que poner especial cuidado en evitar las pérdidas de calidad de imagen.

##### **I.6.1.2.3.3 Generación de cuadros I**

La generación de cuadros I no es un tipo de empalmador si no más bien una posible característica del empalmador. Los empalmadores que tienen esta capacidad pueden producir cuadros I en cualquier punto del flujo. Los empalmadores TS no poseen esta capacidad, mientras que los recodificadores sí la tienen. Para los empalmadores PES y los que trabajan en el nivel de imagen esta capacidad es una opción.

#### **I.6.1.3 Entorno**

Para caracterizar el comportamiento del empalmador, es necesario identificar el o los entornos en que se supone que va a funcionar. Las distintas arquitecturas e implementaciones de empalmador tendrán distintos niveles de desempeño en situaciones distintas.

En el contexto de este análisis, el entorno describe el tipo de contenido de programa MPEG-2 que el empalmador tiene que procesar, y cómo se lo entrega al empalmador y éste lo procesa.

A pesar de que es probable que el mecanismo utilizado para controlar el empalmador tenga un efecto directo en determinadas esferas de su desempeño (exactitud de cuadro), las cuestiones relativas al control quedan fuera del alcance de este apéndice.



### **1.6.1.3.1 Tren elemental de vídeo**

Hay distintos temas relativos a la composición de un tren elemental de vídeo que pueden tener un impacto considerable sobre el comportamiento del empalmador.

#### **1.6.1.3.1.1 Subconjuntos jerárquicos de flujos MPEG-2 – Perfiles y niveles**

El perfil y el nivel de un tren elemental de vídeo determinan cómo se comportarán los distintos tipos de empalmadores. Puesto que no operan sobre trenes elementales, los empalmadores TS no tienen en cuenta el perfil ni el nivel del tren que están empalmando, pero otros tipos de empalmadores sí tienen en cuenta la información que transmiten los perfiles y niveles. Por ejemplo, un empalmador PES puede pasar de perfil principal/nivel principal a perfil principal/nivel bajo sin que esto tenga un impacto significativo en el soporte físico, mientras que un empalmador de imagen o recodificador puede resultar considerablemente más caro.

Es muy probable que las combinaciones de perfiles y niveles más interesantes sean nivel principal/perfil principal, nivel principal/perfil 4:2:2, y nivel alto/perfil principal.

Es importante señalar que la resolución espacial puede cambiar entre las secuencias de imágenes o *clips* que se empalman, y aunque todo hace pensar que esto no plantea problemas en un buen empalmador, es probable que algunos decodificadores no sean capaces de conmutar adecuadamente, lo que produce un empalme con perturbaciones (no liso).

#### **1.6.1.3.1.2 Estructura del tren de datos – Estructura del GOP**

Hay tres factores relacionados con la estructura del grupo de imágenes, GOP, que repercuten en el comportamiento del empalmador. El primero es si el GOP es abierto o cerrado. Es probable que un GOP abierto cause problemas al empalmador porque comienza con cuadros B que referencian al GOP previo; cuando el empalme se hace en la frontera GOP, el cuadro ancla referenciado por los cuadros B no existe en el tren de salida. Esto puede ser importante para determinados tipos de empalmadores.

El segundo factor es el número de cuadros B que se usa entre los cuadros ancla. Si se ordena al empalmador que realice un empalme entre cuadros ancla, este número determina a qué distancia del punto de empalme se encuentra el cuadro ancla más próximo. Algunos empalmadores necesitan que el cuadro ancla tenga un punto de entrada o un punto de salida (o ambos), de allí que este factor afecte su comportamiento.

Otro factor es el uso de "refresco progresivo", técnica con la que el tren no tiene cuadros I (si no macrobloques I). La ausencia de cuadros I supone un serio problema para los empalmadores que no son capaces de generar cuadros I bajo demanda.

#### **1.6.1.3.1.3 Velocidad binaria**

Los trenes se pueden codificar a velocidad binaria constante (CBR, *constant bit rate*) o a velocidad binaria variable (VBR, *variable bit rate*). Los empalmadores tienen que hacer empalmes entre fuentes CBR con la misma velocidad binaria; entre fuentes CBR con velocidad diferente; o entre fuentes VBR.

El empalme entre fuentes de velocidad binaria variable es significativamente más complejo que el de fuentes CBR de igual velocidad. Un empalmador diseñado para manejar sólo fuentes CBR puede experimentar problemas para manejar datos VBR.

#### **1.6.1.3.1.4 Puntos de empalme**

Diferentes arquitecturas de empalmador imponen restricciones diferentes sobre la ubicación de los puntos de empalme. Un empalmador de tren de transporte puede exigir que los trenes cumplan SMPTE 312M [5], mientras que otros pueden necesitar que los puntos de entrada y de salida aparezcan en los cuadros I o en cuadros ancla.

Cuando se le da la instrucción de empalmar en un punto que no cumple sus requisitos, los distintos empalmadores pueden comportarse de manera muy diferente. Algunos no tendrán problemas para funcionar en cualquier circunstancia; otros quizás tengan que ajustar la posición del punto de empalme en el tren, insertar una secuencia de transición entre los clips o crear un empalme con perturbaciones (no liso).

#### **I.6.1.3.2 Trenes elementales de audio**

Los trenes elementales de audio son considerablemente más simples que los trenes elementales de vídeo, pero puede resultar difícil hacer un empalme audio sin perturbaciones y correctamente sincronizado con el vídeo. Distintos factores, como el tipo de codificación, la velocidad binaria y la velocidad de muestreo afectan el funcionamiento de los empalmadores (y de los adaptadores multimedios).

##### **I.6.1.3.2.1 Tipo de tren**

Los dos tipos de audio más utilizados son MPEG-1 capa II y Dolby AC-3. El AC-3 utiliza una duración de cuadro superior, lo que probablemente afectará la sincronización audio-vídeo de algunos empalmadores. El empalme entre tipos de trenes distintos puede resultar problemático.

##### **I.6.1.3.2.2 Velocidad binaria/velocidad de muestreo**

Los clips pueden tener una velocidad de muestreo y una velocidad binaria diferentes. Esto incide en la selección del punto de empalme, y el empalmador debe tenerlo debidamente en cuenta.

#### **I.6.1.3.3 Trenes de datos**

La manipulación de los trenes de datos asociados con un programa puede resultar difícil. En general, el empalmador puede decidir simplemente conmutar un tren de datos en la frontera PES más cercana al punto de empalme, o no hacer nada. Pero ciertos tipos de trenes de datos pueden estar relacionados con el contenido audio-vídeo de tal forma que no sea conveniente hacer un "corte limpio" sin aplicar al tren un procesamiento según el tipo.

##### **I.6.1.3.4 Tipo de multiplexador**

Una vez que el empalmador identifica las características de los programas que tiene que procesar, se examinan las características del multiplexador que contiene los programas, puesto que normalmente tiene que trabajar con uno o más programas de un tren multiplexado.

##### **I.6.1.3.4.1 Multiplexación estadística**

Los multiplexadores pueden implementarse con asignación de velocidad binaria fija para los programas que contienen, o implementarse con multiplexado estadístico, en el que se permite que los programas tengan velocidad binaria variable. En este último caso, el multiplexador estadístico se utiliza para garantizar que la velocidad binaria combinada de todos los programas que contiene el multiplexador no excede un valor determinado.

En el lado de entrada, si el empalmador puede manejar trenes VBR, puede manejar las entradas multiplexadas estadísticamente. En el lado de salida, el multiplexado estadístico puede significar un reto importante para el empalmador.

La remultiplexación estadística puede ser una operación compleja, cuyo comportamiento y calidad de funcionamiento es posiblemente más difícil de caracterizar que el empalme mismo. Las distintas arquitecturas del multiplexador estadístico exigen técnicas distintas para limitar la velocidad binaria combinada del múltiplex, y muy probablemente el dispositivo empleará técnicas diferentes según el estado de multiplexación que tiene que procesar.

Hay multiplexadores estadísticos diseñados para limitar los excesos transitorios de la velocidad binaria combinada, mientras que otros tienen un diseño que les permite funcionar bien con excesos de larga duración. No hay duda de que la calidad de la imagen en distintas situaciones es un factor

fundamental, dado que los multiplexadores estadísticos tienden a introducir distorsiones espaciales o temporales (o ambas) que varían con el tiempo cuando se enfrentan con situaciones transitorias de mayor velocidad binaria.

La clasificación de los tipos de multiplexadores estadísticos queda fuera del alcance de este apéndice. No obstante, resulta como mínimo útil señalar que conviene saber si el empalmador es capaz de producir una salida con multiplexado estadístico; gracias a un multiplexador estadístico externo, un empalmador que no produce directamente una salida con multiplexado estadístico puede trabajar en un entorno de multiplexación estadística.

#### **I.6.1.3.4.2 Multicanal**

Los multiplexadores que llevan señales HD pueden contener una mezcla de programas HD y SD. Además del empalme entre programas HD, se puede pedir al empalmador que realice empalmes entre un único programa HD y múltiples programas SD. En este caso, el empalmador condiciona los programas, de manera que el decodificador siguiente pueda conmutar entre un programa HD y uno de los diversos programas SD. Este tipo de comportamiento también resulta útil cuando en un entorno sólo SD hay que manejar esquemas publicitarios con objetivos muy definidos.

Además de especificar los perfiles y niveles MPEG-2 que el empalmador tiene que procesar, también resulta útil saber si el empalmador soporta el empalme entre perfiles y/o el empalme multicanal.

#### **I.6.1.4 Desempeño del empalmador**

Hay muchas métricas con las que evaluar el desempeño de los procesadores MPEG-2. Muchas de ellas se aplican normalmente a los dispositivos como los codificadores y los remultiplexadores, y también se pueden aplicar a los empalmadores (por ejemplo, retardo de fase PCR, etc.). Esta cláusula se centra en las mediciones que tienen una importancia especial para los empalmadores.

##### **I.6.1.4.1 Empalme liso o sin perturbaciones**

El empalme liso puede significar distintas cosas según quien lo interprete. Por ello, es importante definir "liso" más claramente y reconocer que, de hecho, hay distintos niveles de comportamiento de empalme "liso". En este apéndice se recomiendan las siguientes definiciones:

###### **I.6.1.4.1.1 empalme casi liso n: sinónimo de empalme no liso**

**I.6.1.4.1.2 empalme no liso n:** empalme que se efectúa con perturbaciones: es no **liso visual** o **sintácticamente**. Puede causar la congelación momentánea de la imagen, una pantalla blanca, o "temblor o vibración" del movimiento. Los espectadores no detectan muchos empalmes no lisos, por lo que la evaluación de su calidad es subjetiva. La apariencia del empalme generalmente depende de la relación entre los dos flujos empalmados en el instante de empalme.

###### **I.6.1.4.1.3 empalme liso n:** empalme que es **sintáctica** y **visualmente liso**.

**I.6.1.4.1.4 empalme** (1) n: proceso de salir de un tren de bits y unirse a otro. (2) v. Acción de dicha unión. (3) n. La posición en el tren de bits resultante donde se ha realizado la unión.

**I.6.1.4.1.5 empalme sintácticamente liso n:** empalme que resulta en un tren de bits que cumple los requisitos sintácticos y semánticos de las normas de los sistemas MPEG-2 [2] y de vídeo [3]. No es necesariamente un **empalme visualmente liso** (posiblemente debido a la inserción de **cuadros de transición**).

**I.6.1.4.1.6 secuencia de transición, cuadros de transición n:** secuencia corta del tren de bits de los cuadros que sintetiza el **empalmador** para intercalarla entre el fin del **tren previo** y el comienzo del **tren nuevo**, generalmente para controlar los niveles de la memoria tampón.

**I.6.1.4.1.7 empalme visualmente liso** n: empalme que resulta en una secuencia ininterrumpida de cuadros decodificados, de forma que el último cuadro del tren previo va seguido del primer cuadro del tren nuevo sin intervención.

**I.6.1.4.1.8 cuadros de transición:** este tipo de empalme puede o no ser un **empalme sintácticamente liso**. El desempeño visual depende de las características del codificador, que no se definen en MPEG-2.

Es muy probable que el entorno en el que funciona el empalmador tenga un efecto importante en su capacidad de hacer empalmes lisos. Incluso un empalmador TS puede realizar empalmes lisos en un flujo adaptado a SMPTE 312M [5] cuando se le ordena que empalme exactamente en el punto de empalme adaptado. No obstante, muchos tipos de empalmadores pueden experimentar problemas para producir empalmes visualmente lisos entre programas que no contienen cuadros I. Algunos empalmadores pueden aprovechar la codificación de velocidad variable o el retardo para producir empalmes visualmente lisos. Por consiguiente, es importante señalar en qué circunstancias un determinado empalmador producirá un empalme con un cierto nivel de perturbaciones.

#### **I.6.1.4.2 Exactitud de cuadro**

La exactitud de cuadro describe si la transición entre las secuencias ocurre exactamente en el cuadro especificado. Suponiendo que un empalmador dado puede recibir la instrucción de empalmar en un cuadro exacto, la cuestión consiste en saber si la puede cumplir exactamente en el punto deseado.

Como se ha expuesto en las cláusulas anteriores, los empalmadores suelen imponer algunas restricciones en los tipos de cuadro que pueden servir como puntos de entrada y de salida. Por tanto, al estudiar la exactitud de cuadro de un empalmador, es importante especificar la situación bajo la que se determinará la exactitud. Por ejemplo, prácticamente todos los empalmadores pueden ejecutar la exactitud de cuadro cuando se les ordena empalmar en puntos de empalme precondicionados, mientras que sólo los empalmadores que producen cuadros I por demanda pueden tener exactitud de cuadro cuando se les ordena empalmar en una situación en la que ni el punto de entrada ni el punto de salida son cuadros ancla. Es probable que otros empalmadores tengan exactitud de cuadro cuando se les ordena empalmar en un punto de salida que es un cuadro ancla y el punto de entrada es un cuadro I.

Desafortunadamente, los fabricantes tienen opiniones divergentes sobre lo que es la "exactitud de cuadro". Por consiguiente, es importante establecer la gama de posibles circunstancias y saber cómo poder ajustar el punto de empalme en dichas circunstancias.

#### **I.6.1.4.3 Retardo**

Si bien es importante diseñar las instalaciones de retransmisión con un retardo fijo mínimo, algunos empalmadores pueden aprovechar el retardo variable para mejorar la calidad de sus empalmes cuando los puntos de entrada y de salida apropiados no aparecen en los tiempos o instantes oportunos.

Es útil identificar el límite máximo de retardo aceptable para un determinado tipo de instalación, así como saber si es aceptable un retardo variable, y hasta qué punto. Las instalaciones en distintos puntos de la cadena de difusión tienen requisitos diferentes; por ejemplo, una instalación de origen puede ser muy sensible a las variaciones del retardo, mientras que éste hace poca mella en la última instalación de retransmisión antes de los hogares.

#### **I.6.1.4.4 El tren de salida del empalmador debe cumplir MPEG-2**

Así como el tren de entrada en la red debe cumplir la norma de transporte MPEG-2 [2], el tren de transporte de salida del empalmador/cabecera (después del procesado de los mensajes de aviso y de la inserción publicitaria apropiada) que procesa el empalmador siguiente o el adaptador multimedio tiene que ser conforme con MPEG-2 [4] y cumplir la norma de interfaz de red SCTE 40 2001 [6]. Aunque los trenes de red no tengan demasiadas discontinuidades (como diferentes códigos de fin de

secuencia o cambios en la base de tiempo), los trenes de salida de los empalmadores pueden incluir una o más discontinuidades que permite la norma MPEG, por ejemplo terminar una secuencia de vídeo con el código de fin de secuencia seguido de una nueva secuencia con un tamaño de cuadro o velocidad binaria con codificación diferente, o cambios desde y hacia el modo película, además de las discontinuidades de la base de tiempo señaladas por la bandera de discontinuidad PCR. Los empalmadores y adaptadores multimedios que pasan las pruebas de conformidad MPEG-2 procesan las discontinuidades en los trenes de transporte que se describen a continuación. El siguiente texto en cursiva sobre la manera en que el adaptador multimedia maneja las discontinuidades pertenece a ISO/CEI 13818-4 [4] y se ofrece como guía para la conformidad del tren de salida del empalmador después de un empalme:

*Manejo de las discontinuidades en el decodificador (concatenación de secuencias; discontinuidades de decodificación; empalme; cambios de formato)*

*En los trenes de transporte que cumplen la norma, pueden ocurrir las siguientes discontinuidades en los parámetros del proceso de decodificación en cualquier frontera de la unidad de acceso de audio o cualquier frontera de la secuencia de vídeo:*

- *Vídeo: cualquier parámetro configurado en el encabezamiento de secuencia o en el encabezamiento de capa inferior, como perfil/nivel, velocidad de cuadros, velocidad binaria, parámetros GOP, formato de cuadro, etc.*
- *Audio: cualquier parámetro, como capa de audio, velocidad binaria, velocidad de muestreo, etc.*
- *Vídeo y audio: el tiempo de decodificación de la primera unidad de acceso después de la frontera puede ser superior a lo esperado de no existir la frontera. Esto puede ocurrir independientemente para todos, algunos, o un solo tren elemental de programa. Puede o no estar indicado por la presencia de información adicional referida a un punto de empalme liso o no liso.*

*Suponiendo cualquier combinación de modificaciones de los parámetros del proceso de decodificación que producen valores de parámetros que soporta el decodificador objeto de prueba, éste debe:*

- *mantener una sincronización de presentación correcta entre los distintos trenes elementales del programa;*
- *no producir perturbaciones inaceptables de audio o vídeo, como chisporroteo, bloqueo, etc. Sin embargo, cuando se produce una discontinuidad de decodificación puede que no haya ningún dato que presentar durante un intervalo de tiempo. En esos instantes, se recomienda silenciar los decodificadores de audio y que los decodificadores de vídeo congelen el cuadro/la trama.*
- *"Además, cuando el PTS indica un desplazamiento de fase en la temporización de la visualización de vídeo después de la discontinuidad (es decir, la diferencia entre el PTS actual y previsto no es un número entero exacto de periodos de cuadro) los decodificadores pueden continuar el proceso de visualización sin producir discontinuidades en la temporización vertical. Esto supone volver a mapear el vídeo decodificado durante el proceso de visualización, lo que puede exigir más memoria (que la especificada en el modelo T-STD). Por otra parte, los decodificadores también pueden resincronizarse (genlock) ajustando directamente la temporización de visualización vertical con la temporización de decodificación, lo que admite una discontinuidad de fase de la temporización de visualización vertical. Esto puede originar un efecto de resincronización visible en los dispositivos de visualización. Se permiten las dos implementaciones en los decodificadores que son conformes."*

Se dan a continuación algunos ejemplos de comportamiento del adaptador multimedio en condiciones de empalme diferentes:

- 1) Si el tren de entrada (de salida de la cabecera o del empalmador hacia el adaptador multimedio) cumple íntegramente la sintaxis de vídeo, el transporte T-STD y la continuidad de la base de tiempo (es decir, no hay discontinuidades PCR y los PTS continúan en fase) antes y después del punto donde se produce la inserción, el adaptador multimedio debe cambiar a la nueva secuencia sin perturbaciones. Esto significa también cambios en el encabezamiento en los puntos de inserción donde cambian el tamaño de cuadro codificado, la velocidad binaria o las matrices de cuantización. Los cambios de velocidad de cuadro exigen la resincronización y, posiblemente, la reinicialización de algunos adaptadores multimedios.
- 2) Si una discontinuidad PCR aparece antes del comienzo de una secuencia insertada (y sigue habiendo conformidad con T-STD), el adaptador multimedio adquirirá la nueva secuencia con algún dispositivo de visualización que resuelve el cambio de fase de la base de tiempo.
- 3) Si el tren de entrada rompe el T-STD en el punto en que cambia la secuencia (es decir, cuando la secuencia insertada rebasa la memoria tampón T-STD), el adaptador multimedio se puede volver a reinicializar puesto que la entrada ya no es "conforme".
- 4) El caso en el que no se señalizan discontinuidades de la base de tiempo y la secuencia insertada empieza con una nueva base de tiempo, también es no conforme y el adaptador multimedio podrá reinicializarse.

Es obligatorio en las normas del mensaje de aviso que el tren de transporte que va al empalmador siguiente o al adaptador multimedio observe las especificaciones de transporte y vídeo MPEG-2 y que no se produzcan errores en la salida del adaptador multimedio. Aunque estos decodificadores pueden manejar errores, con toda probabilidad no podrán manejar con facilidad trenes "no conformes". Todos los adaptadores multimedios conformes con MPEG deben manipular los cambios de resolución de vídeo después de un código de fin de secuencia (`seq_end_code`) y los que se produzcan en los parámetros del encabezamiento de secuencia (`seq_header`) que sigue (como las matrices de cuantización, velocidad binaria, tamaño de cuadro, relación de aspecto, retardo bajo, etc.). Procesar los cambios de la tabla de correspondencia de programas (PMT) puede causar más retardo que realizar exclusivamente cambios en el tren de vídeo, porque el procesado de la PMT se suele hacer con soporte lógico propietario, mientras que los cambios de vídeo se suelen procesar con ASIC.