



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

J.181

(03/2001)

SERIE J: REDES DE CABLE Y TRANSMISIÓN DE
PROGRAMAS RADIOFÓNICOS Y TELEVISIVOS, Y DE
OTRAS SEÑALES MULTIMEDIOS

Varios

**Mensaje de aviso de inserción de programa
digital para sistemas de televisión por cable**

Recomendación UIT-T J.181

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE J

REDES DE CABLE Y TRANSMISIÓN DE PROGRAMAS RADIOFÓNICOS Y TELEVISIVOS, Y DE OTRAS SEÑALES MULTIMEDIOS

Recomendaciones generales	J.1–J.9
Especificaciones generales para transmisiones radiofónicas analógicas	J.10–J.19
Características de funcionamiento de los circuitos radiofónicos	J.20–J.29
Equipos y líneas utilizados para circuitos radiofónicos analógicos	J.30–J.39
Codificadores digitales para señales radiofónicas analógicas	J.40–J.49
Transmisión digital de señales radiofónicas	J.50–J.59
Circuitos para transmisiones de televisión analógica	J.60–J.69
Transmisiones de televisión analógica por líneas metálicas e interconexión con radioenlaces	J.70–J.79
Transmisión digital de señales de televisión	J.80–J.89
Servicios digitales auxiliares para transmisiones de televisión	J.90–J.99
Requisitos operacionales y métodos para transmisiones de televisión	J.100–J.109
Sistemas interactivos para distribución de televisión digital	J.110–J.129
Transporte de señales MPEG-2 por redes de transmisión de paquetes	J.130–J.139
Mediciones de la calidad de servicio	J.140–J.149
Distribución de televisión digital por redes locales de abonados	J.150–J.159
IPCablecom	J.160–J.179
Varios	J.180–J.199
Aplicación para televisión digital interactiva	J.200–J.209

Para más información, véase la Lista de Recomendaciones del UIT-T.

Recomendación UIT-T J.181

Mensaje de aviso de inserción de programa digital para sistemas de televisión por cable

Resumen

Esta Recomendación trata del empalme de trenes de transporte MPEG-2 para la inserción de programas digitales, por ejemplo, la inserción de anuncios y de contenidos de otro tipo. Se define un mecanismo de mensajería en el tren para señalar las oportunidades de empalme e inserción. Se especifica una técnica de transmisión de notificaciones de la próxima llegada de puntos de empalme en el tren de transporte.

Orígenes

La Recomendación UIT-T J.181, preparada por la Comisión de Estudio 9 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 9 de marzo de 2001.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2001

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

ÍNDICE

Página

1	Alcance	1
2	Referencias.....	1
2.1	Referencias normativas.....	1
3	Definiciones de términos	2
4	Abreviaturas	3
5	Introducción	4
5.1	Puntos de empalme.....	4
5.2	Puntos de empalme de programa	4
5.3	Evento de empalme	5
6	Descriptores de PMT.....	5
6.1	Descriptor de registro	5
6.1.1	Definición semántica de los campos del descriptor de registro	6
6.2	Descriptor de identificador de tren.....	6
6.2.1	Definición de semántica de los campos del descriptor de identificador de tren.....	6
7	Tabla de información de empalme	7
7.1	Visión general	7
7.1.1	Discontinuidades de la base de tiempo	7
7.2	Sección de información de empalme.....	8
7.2.1	Definición semántica de los campos de splice_info_section().....	9
7.3	Instrucción de empalme	10
7.3.1	splice_null().....	10
7.3.2	splice_schedule()	11
7.3.3	splice_insert()	12
7.4	Tiempo.....	14
7.4.1	splice_time().....	14
7.4.2	break_duration()	15
7.5	Constricciones	15
7.5.1	Constricciones impuestas a splice_info_section().....	15
7.5.2	Constricciones impuestas a la interpretación de tiempo	16
8	Descriptores de empalmes	18
8.1	Visión general	18
8.2	Descriptor de empalme	18
8.2.1	Definición semántica de los campos de splice_descriptor().....	18

	Página
8.3	Descriptores de empalmes específicos 19
	8.3.1 avail_descriptor()..... 19
9	Criptación..... 19
9.1	Visión general 19
9.2	Criptación de clave fija..... 20
9.3	Algoritmos de criptación 20
	9.3.1 Modo DES-ECB..... 20
9.4	Modo DES-CBC 20
9.5	Modo DES EDE3-ECB triple 21
9.6	Algoritmos privados de usuario 21
Apéndice I – Bibliografía 21	

Recomendación UIT-T J.181

Mensaje de aviso de inserción de programa digital para sistemas de televisión por cable

1 Alcance

La presente Recomendación se refiere al empalme de trenes MPEG-2 para la inserción de programas digitales, lo que incluye la inserción de anuncios y la inserción de otros tipos de contenido. Se define un mecanismo de mensajería en el tren para señalar las oportunidades de empalme e inserción.

Se supone la presencia de un tren de transporte que cumple plenamente la norma MPEG-2 (un tren de transporte multiprograma o un tren de transporte de programa único). En el tren 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, según pretenden UIT-T H.222.0 | ISO/CEI 13818-1 y SMPTE 312M, no será suficiente en las plantas de cable debido a la aplicación de métodos actuales, y duraderos, tales como el de multiplexión estadística con velocidad binaria variable (VBR, *variable bit rate*) y el de renovación progresiva (sin tramas I).

Esta Recomendación especifica una técnica de transmisión de notificaciones de la próxima llegada de puntos de empalme en el tren de transporte. Se define una tabla de información de empalme para informar a los dispositivos situados más adelante sobre eventos de empalme tales como la interrupción de la red o el retorno tras una interrupción de la red. La tabla de información de empalme, perteneciente a un programa dado, se lleva en un identificador de paquete (PID, *packet identifier*) aparte al que hace referencia la tabla de correspondencia de programa (PMT, *program map table*). De esa manera, la notificación del evento de empalme puede pasar a través de la mayoría de los remultiplexores de trenes de transporte sin necesidad de un procesamiento especial.

La presente Recomendación no se ocupa de las restricciones impuestas a los dispositivos de empalme.

2 Referencias

2.1 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- 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*.
- 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*.
- ISO/CEI 13818-4:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 4: Conformance testing*, plus Corrigendum 2 (1998).

3 Definiciones de términos

A lo largo de la presente Recomendación, los términos indicados más abajo tienen significados específicos. Puesto que el significado de algunos de los términos definidos en ISO/CEI 13818 es un significado técnico muy concreto, se remite al lector a la fuente original para su definición. A continuación se da una breve definición de los términos utilizados en esta Recomendación.

3.1 unidad de acceso: Representación codificada de una imagen de vídeo o una trama de audio (véase UIT-T H.262 | ISO/CEI 13818-2).

3.2 tono de aviso analógico: En un sistema analógico, una señal que normalmente es una secuencia de multifrecuencia bitono (DTMF, *dual tone multifrequency*) o un cierre de contacto que indica al equipo de inserción de anuncios que va a empezar o terminar una cuña publicitaria.

3.3 cuña: Espacio de tiempo proporcionado a los operadores de cable por los servicios de programación por cable durante un programa para que lo utilice el operador del sistema de televisión por antena colectiva (CATV, *community antenna television*); el tiempo se vende normalmente a los anunciantes locales o se utiliza para la promoción del propio canal.

3.4 interrupción: Cuña o inserción efectiva en curso.

3.5 modo empalme de componente: Modo del mensaje de aviso por el cual la `program_splice_flag` (bandera de empalme de programa) se pone a "0" e indica que cada PID/componente que se pretende empalmar será indicado por separado por la sintaxis que sigue. Los componentes no indicados en el mensaje no han de ser empalmados.

3.6 mensaje de aviso: Véase "mensaje".

3.7 evento: Evento de empalme o evento de observación.

3.8 punto de entrada: Punto del tren, adecuado para la entrada, que se halla en la frontera de una unidad de acceso elemental.

3.9 mensaje: En el contexto de esta Recomendación, un mensaje es el contenido de cualquier `splice_info_section` (sección de información de empalme).

3.10 punto de salida: Punto del tren, adecuado para la salida, que se halla en la frontera de una unidad de acceso elemental.

3.11 identificador de paquete (PID, *packet identifier*): Valor único de 13 bits utilizado para identificar el tipo de datos almacenados en la cabida útil del paquete (véase UIT-T H.222.0 | ISO/CEI 13818-1).

3.12 `payload_unit_start_indicator` (indicador de comienzo de unidad de cabida útil): Bit del encabezamiento del paquete de transporte que señala, entre otras cosas, que empieza una señal en la cabida útil que sigue (véase UIT-T H.222.0 | ISO/CEI 13818-1).

3.13 tren de identificador de paquete: Todos los paquetes con el mismo PID dentro de un tren de transporte.

3.14 `pointer_field` (campo de puntero): El primer byte de la cabida útil de un paquete de transporte, necesario cuando en ese paquete empieza una sección (véase UIT-T H.222.0 | ISO/CEI 13818-1).

3.15 tiempo de presentación: Tiempo o momento en que una unidad de presentación es presentada en el decodificador objetivo del sistema (véase UIT-T H.222.0 | ISO/CEI 13818-1).

3.16 programa: Conjunto de trenes de PID de vídeo, audio y datos que comparten un número de programa común dentro de un MPTS (véase UIT-T H.222.0 | ISO/CEI 13818-1).

3.17 punto de entrada en programa: Grupo de puntos de entrada de tren de PID que coinciden en tiempo de presentación.

- 3.18 punto de salida de programa:** Grupo de puntos de salida de tren de PID que coinciden en tiempo de presentación.
- 3.19 modo empalme de programa:** Modo del mensaje de aviso por el cual la `program_splice_flag` (bandera de empalme de programa) se pone a "1" e indica que el mensaje se refiere a un punto de empalme de programa y que todos los PID/componentes del programa han de ser empalmados.
- 3.20 punto de empalme de programa:** Punto de entrada en programa o punto de salida de programa.
- 3.21 descriptor de registro:** El descriptor de registro se lleva en la PMT de un programa para indicar que, cuando se señalen eventos de empalme, deberán llevarse las `splice_info_sections` en un tren de PID dentro de ese programa. La presencia del descriptor de registro significa que el programa cumple con la presente Recomendación.
- 3.22 reservado:** El término "reservado", cuando se utiliza en las cláusulas en que se define el tren de bits codificado, indica que el valor puede ser utilizado en futuras ampliaciones de la Recomendación. A menos que se indique otra cosa en esta Recomendación, todos los bits reservados se pondrán a "1".
- 3.23 evento de empalme:** Una oportunidad de empalmar uno o más trenes PID.
- 3.24 modo empalme inmediato:** Modo del mensaje de intercalación por el cual el dispositivo empalmador deberá aprovechar la primera oportunidad en el tren, con respecto a la `splice_info_table` (tabla de información de empalme), para empalmar. Cuando no está en este modo, el mensaje indica un "pts_time", que es un tiempo de presentación, para el momento del empalme pretendido.
- 3.25 punto de empalme:** Punto de un tren PID que es un punto de salida o un punto de entrada.
- 3.26 evento de observación:** Programa de televisión o tramo de material comprimido dentro de un servicio; distinto de un evento de empalme, que es un punto en el tiempo.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

- ATSC Advanced Television Systems Committee.
- bslbf Cadena de bits, bit izquierdo primero (*bit string, left bit first*), izquierdo implica el orden en el que se escriben las cadenas de bits
- MPTS Tren (flujo) de transporte multiprograma (*multi program transport stream*)
- PMT Tabla de correspondencia de programa (*program map table*) (véase UIT-T H.222.0 | ISO/CEI 13818-1)
- PTS Indicación de tiempo de presentación (*presentation time stamp*) (véase UIT-T H.222.0 | ISO/CEI 13818-1)
- rpchof Coeficientes de polinomio residual, orden más alto primero (*remainder polynomial coefficients, highest order first*)
- SPTS Tren (flujo) de transporte de un solo programa (*single program transport stream*)
- uimsbf Entero sin signo, bit más significativo primero (*unsigned integer, most significant bit first*)

5 Introducción

5.1 Puntos de empalme

La presente Recomendación define puntos de empalme, cuya finalidad es permitir el empalme de trenes de bits comprimidos. Los puntos de empalme de un tren de transporte MPEG-2 hacen posible conmutar de un programa a otro. Indican un lugar en el que conmutar o un lugar en el tren de bits en donde se puede llevar a cabo una conmutación. La realización de empalmes en esos puntos puede que dé como resultado, o puede que no, una buena calidad visual y de audio. Esto es algo que depende del comportamiento del dispositivo empalmador.

Los trenes de transporte se crean multiplexando trenes de PID. En esta Recomendación, se definen dos tipos de puntos de empalme para trenes de PID: puntos de salida y puntos de entrada. Los puntos de entrada son lugares de los trenes de bits en donde parece conveniente entrar, desde el punto de vista de la realización de un empalme. Los puntos de salida son lugares de donde parece conveniente que salga el tren de bits. Se ha definido la agrupación de los puntos de entrada de cada uno de los trenes de PID en puntos de entrada al programa para permitir la conmutación de los programas en su totalidad (vídeo con audio). También se han definido puntos de salida del programa para la salida de un programa.

Los puntos de salida y los puntos de entrada son puntos imaginarios del tren de bits situados entre dos unidades de acceso de tren elemental. Los puntos de salida y los puntos de entrada no necesariamente están alineados en cuanto a paquetes de transporte ni en cuanto a paquetes de tren elemental paquetizado (PES, *packetized elementary streams*). Un punto de salida puede estar coincidente con un punto de entrada; es decir, una sola frontera de unidad de acceso puede servir como lugar seguro para salir de un tren de bits y como lugar seguro para entrar en él.

El resultado de una operación de conmutación simple contendrá datos de la unidad de acceso de un tren hasta su punto de salida seguidos por los datos de otro tren que empiece en la primera unidad de acceso posterior a un punto de entrada. Pueden existir operaciones de empalme más complejas en las que datos anteriores a un punto de salida o datos posteriores a un punto de entrada sean modificados por un dispositivo empalmador. Los dispositivos empalmadores pueden insertar también datos entre el punto de salida de un tren y el punto de entrada de otro tren. El comportamiento de los dispositivos empalmadores no es especificado ni constreñido en modo alguno por la presente Recomendación.

5.2 Puntos de empalme de programa

Los puntos de entrada a programa y los puntos de salida de programa son conjuntos de puntos de entrada o puntos de salida de un tren de PID que coinciden en el tiempo de presentación.

Aunque los puntos de empalme de un punto de empalme de programa coinciden en el tiempo de presentación, normalmente no aparecen unos cerca de otros en el tren de transporte. Puesto que la decodificación de vídeo comprimido dura mucho más que la de audio, los puntos de empalme de audio pueden retrasarse con respecto a los puntos de empalme de vídeo incluso hasta en cientos de milisegundos, y con una demora que puede variar durante el programa.

Esta Recomendación define dos maneras de señalar los puntos de empalme que han de ser empalmados dentro de un programa. Una `program_splice_flag` (bandera de empalme de programa), cuando es verdadero, indica que el modo empalme de programa está activo y que todos los PID de un programa pueden ser empalmados (el PID de la tabla de información de empalme es una excepción; el empalme o la transmisión de estos mensajes queda fuera del alcance de la presente Recomendación). Una bandera falso indica que el modo empalme de componente está activo y que el mensaje especificará de manera inequívoca qué PID han de ser empalmados e indicará un tiempo de empalme único para cada uno de ellos. De esta manera, el dispositivo empalmador sabrá si tiene que empalmar diversos tipos de datos no especificados así como vídeo y audio.

Aunque esta Recomendación permite que se dé un tiempo de empalme único a cada componente de un programa, la mayoría de los mensajes del modo empalme de componente utilizarán previsiblemente un tiempo de empalme (un tiempo de empalme por defecto) para todos los componentes descritos en la cláusula 7. La posibilidad de especificar facultativamente un tiempo de empalme diferente para cada componente está previsto que se utilice cuando el tiempo de comienzo o el de parada de uno o más componentes difieran de manera significativa de los de otros componentes dentro del mismo mensaje. Un ejemplo al respecto sería el de una miniaplicación telecargada que debe llegar a un adaptador multimedios varios segundos antes de un anuncio.

5.3 Evento de empalme

Esta Recomendación proporciona un método de señalización dentro de banda de mensajes splice_schedule (plan de empalme) y splice_insert (inserción de empalme) al equipo empalmador situado más adelante. La señalización de un evento de empalme identifica qué punto de empalme dentro de un tren hay que utilizar para el empalme. Un dispositivo empalmador puede optar por actuar o no tras un evento señalado (un evento señalado deberá interpretarse como una oportunidad de empalmar; no como una instrucción). Una tabla de información de empalme lleva la notificación de las oportunidades de eventos de empalme. Cada evento de empalme señalado es análogo a un tono de aviso. La tabla de información de empalmes incorpora la funcionalidad de tonos de aviso y la amplía para hacer posible la planificación de eventos de empalme por adelantado.

La presente Recomendación establece que la tabla de información de evento se lleve programa por programa en un tren de PID con un stream_type (tipo de tren) designado. El PID de información de empalme del programa está indicado en la tabla de correspondencia de programa del programa. De esta manera, la tabla de información de empalme cambia con el programa a medida que pasa a través de operaciones de remultiplexión. Un stream_type común identifica todos los trenes de PID que llevan tablas de información de empalme. Los remultiplexores pueden utilizar este campo stream_type para retirar información de empalme antes de enviar el tren de transporte al dispositivo del usuario de destino.

6 Descriptores de PMT

6.1 Descriptor de registro

El descriptor de registro (véase el cuadro 2-45 "Descriptor de registro" en 2.6.8/UIT-T H.222.0 | ISO/CEI 13818-1) se define para identificar de manera inequívoca los programas que cumplen esta Recomendación. El descriptor de registro deberá llevarse en el bucle program_info (información de programa) de la tabla de correspondencia de programa (PMT) para cada programa que se atenga a esta Recomendación. Ha de estar presente en todas las PMT de todos los programas conformes dentro de un multiplex. La presencia del descriptor de registro indica además que, cuando se señalen eventos de empalme, las splice_info_section se llevarán en un tren de PID único dentro del programa. El PID elegido para esta tabla de información de empalme deberá ser un PID exclusivo, distinto del PID de otras tablas de información de empalme para otros programas dentro del multiplex y distinto asimismo del PID elegido para otras tablas de información de empalme definidas por otras normas (a saber, SMPTE 312M).

Se señala que este descriptor se aplica al programa indicado y no al multiplex en su totalidad. El contenido del descriptor de registro se especifica en el cuadro 6-1 que sigue:

Cuadro 6-1/J.181 – Descriptor de registro

Sintaxis	Bits	Mnemónica
registration_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
SCTE_splice_format_identifier	32	uimsbf
}		

6.1.1 Definición semántica de los campos del descriptor de registro

descriptor_tag (rótulo de descriptor): El descriptor_tag es un campo de 8 bits que identifica a cada descriptor. A efectos de registro, este campo deberá fijarse en 0x05.

descriptor_length (longitud de descriptor): El descriptor_length es un campo de 8 bits que especifica el número de bytes del descriptor que sigue inmediatamente al campo descriptor_length. Para este descriptor de registro, el descriptor_length deberá fijarse en 0x04.

SCTE_splice_format_identifier (identificador de formato de empalme de SCTE): SCTE ha asignado un valor de 0x43554549 (ASCII "CUEI") a este campo de 4 bytes para identificar el programa (dentro de un múltiplex) en el que se lleva como cumplidor de la presente Recomendación.

6.2 Descriptor de identificador de tren

El descriptor de identificador de tren se puede utilizar en la PMT para etiquetar trenes de componentes de un servicio de manera que puedan ser diferenciados. El descriptor de identificador de tren deberá estar situado en el bucle de descriptor elemental que sigue al campo ES_info_length (longitud de información de tren elemental) pertinente. El descriptor de identificador de tren se utilizará si la program_splice_flag es cero. Si se utilizan descriptores de identificador de tren, deberá estar presente uno de ellos cada vez que aparezca el bucle de tren elemental dentro de la PMT y deberá tener un rótulo de componente único dentro del programa de que se trate.

6.2.1 Definición de semántica de los campos del descriptor de identificador de tren

Véase el cuadro 6-2.

Cuadro 6-2/J.181 – Stream_identifier_descriptor

Sintaxis	Bits	Mnemónica
stream_identifier_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
component_tag	8	uimsbf
}		

descriptor_tag: El descriptor_tag es un campo de 8 bits que identifica a cada descriptor. Para el stream_identifier_descriptor, este campo deberá fijarse en 0x52 según lo especificado por el grupo DVB.

descriptor_length: El descriptor_length es un campo de 8 bits que especifica el número de bytes del descriptor que sigue inmediatamente al campo descriptor_length. Para este descriptor de registro, descriptor_length deberá fijarse en 0x01.

component_tag (rótulo de componente): Este campo de 8 bits identifica el tren de componentes para asociarlo con una descripción dada en un descriptor de componente. Dentro de una sección de correspondencia de programas, cada descriptor de identificador de tren deberá tener un valor diferente para este campo.

7 Tabla de información de empalme

7.1 Visión general

La tabla de información de empalme proporciona información de instrucción y control al empalmador. Le informa por adelantado sobre eventos de empalme antes de que se produzcan. Está diseñada para acomodar inserciones de anuncios en las contribuciones de procedencia múltiple que nutren la red. En este entorno, ejemplos de eventos de empalme serían:

- 1) un empalme de material de programas de red en un anuncio; o
- 2) el empalme de un anuncio de vuelta al material de programas de red.

La tabla de información de empalme se puede enviar múltiples veces y los eventos de empalme pueden ser cancelados. Se ha definido la sintaxis de una `splice_info_section` para llevar la tabla de información de empalme. La `splice_info_section` se transporta en su propio tren de PID con el PID declarado en la PMT de ese programa.

Un evento de empalme indica la oportunidad de empalmar uno o más trenes elementales dentro de un programa. Cada evento de empalme se identifica de manera exclusiva con un `splice_event_id` (identificador de evento de empalme). Los eventos de empalme se pueden comunicar de tres maneras: planificándolos por adelantado, dando un aviso de puesta en funcionamiento o dando una instrucción de ejecución del evento de empalme en los puntos de empalme especificados. Estos tres tipos de mensaje se envían por medio de la `splice_info_section`. El campo `splice_command_type` (tipo de instrucción de empalme) especifica el mensaje que se envía. Dependiendo del valor de este campo, se pueden aplicar diferentes constricciones a la sintaxis restante.

Son posibles los siguientes tipos de instrucción: `splice_null()` (empalme nulo), `splice_schedule()` (plan de empalme) y `splice_insert()` (inserción de empalme).

La instrucción `splice_null()` se da a efectos de extensibilidad.

La instrucción `splice_schedule()` permite transmitir por adelantado un plan de eventos de empalme.

La instrucción `splice_insert()` deberá enviarse al menos una vez antes de cada punto de empalme. Los paquetes que contengan la `splice_info_table` (tabla de información de empalme) en su totalidad precederán siempre al paquete que contenga el punto de empalme conexo (es decir, el primer paquete que contiene el primer byte de una unidad de acceso cuyo tiempo de presentación concuerda con el especificado en la `splice_info_section`).

Para dar un aviso por adelantado de un empalme inmediato (una función de puesta en funcionamiento) podría enviarse la instrucción `splice_insert()` múltiples veces antes del punto de empalme. Por ejemplo, la instrucción `splice_insert()` podría enviarse 8, 5, 4 y 2 segundos antes del paquete que contiene el punto de empalme conexo.

7.1.1 Discontinuidades de la base de tiempo

Cuando haya una discontinuidad de la base de tiempo (TBD, *time base discontinuity*) del sistema, los paquetes que contengan una instrucción `splice_insert()` con el tiempo expresado en la nueva base de tiempo no deberán llegar antes de que se produzca la TBD. Los paquetes que contengan una instrucción `splice_insert()` con el tiempo expresado en la base de tiempos anterior no deberán llegar después que se produzca la TBD. Véase ISO/CEI 13818-4, Information technology – Generic coding of moving pictures and associated audio information – Part 4: Conformance testing.

Más adelante se presenta la sintaxis completa, seguida por la definición de términos y por las constricciones.

7.2 Sección de información de empalme

La splice_info_section (véase cuadro 7-1) deberá llevarse en paquetes de transporte de tal manera que en cualquier paquete de transporte sólo esté presente una sección, o una sección parcial. Las splice_info_section deben empezar siempre en el comienzo de la cabida útil del paquete de transporte. Cuando una sección empieza en un paquete de transporte, debe estar presente un pointer_field (campo de puntero) y ser igual a 0x00 y el bit payload_unit_start_indicator debe ser igual a uno (de acuerdo con los requisitos de la utilización de la sintaxis de sección de UIT-T H.222.0 | ISO/CEI 13818-1).

Cuadro 7-1/J.181 – Sección de información de empalme

Sintaxis	Bits	Mnemónica	Criptado
splice_info_section() {			
table_id	8	uimsbf	
section_syntax_indicator	1	bslbf	
private_indicator	1	bslbf	
reserved	2	bslbf	
section_length	12	uimsbf	
protocol_version	8	uimsbf	
encrypted_packet	1	bslbf	
encryption_algorithm	6	uimsbf	
pts_adjustment	33	uimsbf	
cw_index	8	uimsbf	
reserved	24	bslbf	
splice_command_type	8	uimsbf	E
if (splice_command_type = 0x00)			
splice_null()			E
if (splice_command_type = 0x04)			
splice_schedule()			E
if (splice_command_type = 0x05)			
splice_insert()			E
descriptor_loop_length	16	uimsbf	E
for (I = 0; I < N; I++)			
splice_descriptor()			E
for (I = 0; I < N; I++)			
alignment_stuffing	8	bslbf	E
if (encrypted_packet)			
E_CRC_32	32	rpchof	E
CRC_32	32	rpchof	
}			

7.2.1 Definición semántica de los campos de splice_info_section()

table_id (identificador de tabla): El table_id es un campo de 8 bits. Su valor deberá ser 0xFC.

section_syntax_indicator (indicador de sintaxis de sección): El section_syntax_indicator es un campo de 1 bit que deberá estar siempre puesto a "0" indicando que se han de utilizar secciones cortas de MPEG.

private_indicator (indicador privado): El private_indicator es una bandera de 1 bit que deberá ponerse a "0".

section_length (longitud de sección): El section_length es un campo de 12 bits que especifica el número de bytes restantes en la splice_info_section que sigue inmediatamente al campo section_length hasta el final del splice_info_section. El valor de este campo no deberá exceder de 4093.

protocol_version (versión de protocolo): Es un entero sin signo de 8 bits que indica el número de la versión del segmento de la tabla completa entregada con esta sección. El valor de protocol_version deberá ser 0x00.

encrypted_packet (paquete criptado): Cuando este bit está puesto a "1", indica qué porciones de la splice_info_section, empezando con splice_command_type y terminando con E_CRC_32, son criptadas. Cuando este bit se pone a "0", no se cripta ninguna parte de este mensaje. Las porciones potencialmente criptadas de la splice_info_table se indican mediante una E en la columna Criptado del cuadro 7-1.

encryption_algorithm (algoritmo de criptación): Este entero sin signo de 6 bits indica qué algoritmo de criptación se utilizó para criptar el mensaje en curso. Cuando encrypted_packet es cero, este campo está presente pero no está definido. Para los detalles de la utilización de este campo, véase la cláusula 9 y, en concreto, el cuadro 9-1.

pts_adjustment (ajuste de indicación de tiempo de presentación): Entero sin signo de 33 bits que aparece en claro y que deberá ser utilizado por un dispositivo empalmador como un desplazamiento a añadir al o a los campos pts_time (tiempo de indicación de tiempo de presentación) criptados (algunas veces) mediante este mensaje para obtener el tiempo o los tiempos de empalme pretendidos. Cuando este campo tiene un valor de 0, el o los campos pts_time deberán ser utilizados sin desplazamiento alguno. Normalmente, el creador de un mensaje de aviso pondrá un valor de cero en este campo. Este valor de ajuste es el medio por el cual un dispositivo situado hacia el origen que reindica pcr/pts/dts, puede llevar al dispositivo empalmador la manera de convertir el campo pts_time del mensaje a un dominio de tiempo recién impuesto.

Se pretende que el primer dispositivo que reindique pcr/pts/dts y que pase el mensaje de aviso inserte un valor en el campo pts_adjustment que sea la diferencia de tiempo entre el dominio de tiempo de la entrada a este dispositivo y el dominio de tiempo a su salida. Todos los dispositivos subsiguientes, que reindiquen también pcr/pts/dts, pueden modificar aún el campo pts_adjustment añadiendo su diferencia de tiempo a la diferencia de tiempo existente del campo y situando el resultado de nuevo en el campo pts_adjustment. Tras cada alteración del campo pts_adjustment, el dispositivo modificador debe calcular de nuevo y actualizar el campo CRC_32.

El campo pts_adjustment deberá en todo momento tener el valor adecuado que se ha de utilizar para la conversión del campo pts_time a la base de tiempo actual. La conversión se efectúa agregando los dos campos. En presencia de una condición de reiniciación o desbordamiento, se pasará por alto el acarreo.

cw_index (índice de palabra de control): Entero sin signo de 8 bits que lleva la palabra de control (clave) que se ha de utilizar para descriptar el mensaje. El dispositivo empalmador puede almacenar hasta 256 claves proporcionadas previamente a tal efecto. Cuando el bit encrypted_packet es cero, este campo está presente pero no definido.

splice_command_type: Entero sin signo de 8 bits al que se ha asignado unos de los valores mostrados en el cuadro 7-2.

Cuadro 7-2/J.181 – Valores de splice_command_type

Valor de splice_command_type	Instrucción
0x00	Nulo
0x01	Reservado
0x02	Reservado
0x03	Reservado
0x04	Plan
0x05	Inserción
0x06-0xff	Reservado

descriptor_loop_length (longitud de bucle de descriptor): Entero sin signo de 16 bits que especifica el número de bytes utilizados en el bucle del descriptor de empalme que sigue inmediatamente.

alignment_stuffing (relleno de alineación): Este campo es una función del algoritmo de criptación que, en concreto, se haya elegido. Puesto que algunos algoritmos de criptación requieren una longitud específica para los datos criptados, es necesario prever la inserción de bytes de relleno. Por ejemplo, DES necesita la presencia de un múltiplo de 8 bytes para criptar al final del paquete. Así se hace posible la utilización de la norma DES, en vez de requerir una versión especial del algoritmo de criptación.

E_CRC_32: Este campo de 32 bits contiene el valor de verificación por redundancia cíclica (CRC, *cyclic redundancy check*) que da una salida de cero de los registros del decodificador definido en UIT-T H.222.0 | ISO/CEI 13818-1 después de procesar toda la porción descrita de la splice_info_section. La finalidad de este campo es dar una indicación de que se ha llevado a cabo la descripción de manera satisfactoria. Por ello, la salida cero se obtiene tras la descripción y el procesamiento de los campos splice_command_type en todo el E_CRC_32.

CRC_32: Este campo de 32 bits contiene el valor CRC que da una salida de cero de los registros del decodificador definido en UIT-T H.222.0 | ISO/CEI 13818-1 después de procesar toda la splice_info_section (sección de información de empalme) que incluye el campo table_id (identificador de tabla) a lo largo del campo CRC_32. El procesamiento de CRC_32 deberá producirse antes de la descripción de los campos criptados y se utilizarán los campos criptados en su estado criptado.

7.3 Instrucción de empalme

7.3.1 splice_null()

La instrucción splice_null() se da para la ampliación de la Recomendación. La instrucción splice_null() permite el envío de una splice_info_table que lleva nuevos descriptores sin tener que enviar una de las instrucciones existentes [splice_insert() o splice_schedule()]. (Véase el cuadro 7-3.)

Cuadro 7-3/J.181 – splice_null()

Sintaxis	Bits	Descripción
splice_null() { }		

7.3.2 splice_schedule()

La instrucción splice_schedule() se da para hacer posible el transporte por adelantado de un plan de eventos de empalme.

7.3.2.1 Definición semántica de los campos de splice_schedule()

Véase el cuadro 7-4.

Cuadro 7-4/J.181 – splice_schedule()

Sintaxis	Bits	Mnemónica
splice_schedule() {		
splice_count	8	uimsbf
for (i = 0; i < splice_count; i++) {		
splice_event_id	32	bslbf
splice_event_cancel_indicator	1	bslbf
reserved	7	bslbf
if (splice_event_cancel_indicator = "0") {		
out_of_network_indicator	1	bslbf
program_splice_flag	1	bslbf
duration_flag	1	bslbf
reserved	5	bslbf
if (program_splice_flag = "1")		
utc_splice_time	32	uimsbf
if (program_splice_flag = "0") {		
component_count	8	uimsbf
for (j = 0; j < component_count; j++) {		
component_tag	8	uimsbf
utc_splice_time	32	uimsbf
}		
}		
if (duration_flag)		
break_duration()		
unique_program_id	16	uimsbf
avail	8	uimsbf
avail_count	8	uimsbf
}		
}		

splice_count (cuenta de empalmes): Entero sin signo de 8 bits que indica el número de eventos de empalme especificados en el bucle que sigue.

splice_event_id: Identificador de evento de empalme único de 32 bits.

splice_event_cancel_indicator (indicador de cancelación de evento de empalme): Bandera de 1 bit que, cuando está puesto a "1", indica que ha sido cancelado un evento de empalme enviado previamente, identificado mediante splice_event_id.

out_of_network_indicator (indicador de fuera de red): Bandera de 1 bit. Cuando está puesto a "1", indica que el evento de empalme es una oportunidad para salir del programa de la red y que el valor de splice_time() (tiempo de empalme) se referirá a un punto de salida o un punto de salida de programa pretendido. Cuando está puesto a "0", la bandera indica que el evento de empalme es una oportunidad para volver al programa de la red y que el valor de splice_time() se referirá a un punto de entrada o un punto de entrada en programa pretendido.

program_splice_flag: Bandera de 1 bit que, cuando está puesto a "1", indica que el mensaje se refiere a un punto de empalme de programa y que el modo es el modo empalme de programa según el cual todos los PID/componentes del programa han de ser empalmados. Cuando está puesto a "0", este campo indica que el modo es el modo empalme de componente según el cual cada componente que haya de ser empalmado será indicado de manera separada por la sintaxis que sigue.

duration_flag (bandera de duración): Bandera de 1 bit que indica la presencia del campo break_duration() (duración de interrupción).

utc_splice_time (tiempo de empalme UTC): Entero sin signo de 32 bits que representa el tiempo o el momento del evento de empalme señalado como el número de segundos a partir de las 00 horas UTC del 6 de enero de 1980, incluida la cuenta de segundos de arrastre. El utc_splice_time se puede convertir en UTC sin utilizar el valor GPS.UTC_offset dado por la tabla tiempo de sistema. El campo **utc_splice_time** se utiliza solamente en la instrucción splice_schedule().

component_count (cuenta de componentes): Entero sin signo de 8 bits que especifica el número de ejemplares de datos de tren de PID elemental en el bucle que sigue. Los componentes equivalen a trenes PID elementales.

component_tag: Valor de 8 bits que identifica el tren de PID elemental que contiene el punto de empalme especificado por el valor splice_time() que sigue. El valor deberá ser el mismo que el utilizado en el stream_identification_descriptor() para identificar el tren de PID elemental.

unique_program_id (identificador de programa único): Este valor deberá proporcionar una identificación única para un evento de observación dentro del servicio. Puesto que esta Recomendación se refiere a toda norma aplicable, es preciso un valor independiente de cualquier norma. Realiza la misma función que el campo event_id en la norma ATSC o la norma DVB.

avail (cuña): Este campo proporciona la identificación de una cuña específica dentro de un unique_program_id. Está previsto que este valor aumente con cada nueva cuña de un evento de observación, y que vuelva a uno con la primera cuña de un evento de observación nuevo. Este campo aumentará, según lo previsto, con cada nueva cuña. Puede llevar, facultativamente, el valor de cero para indicar su no utilización.

avail_count (cuenta de cuñas): Este campo proporciona la cuenta del número esperado de cuñas individuales en el evento de observación en curso. Cuando este campo es cero, indica que el campo cuña carece de significado.

7.3.3 splice_insert()

La instrucción splice_insert() (inserción de empalme) deberá enviarse al menos una vez por cada evento de empalme. (Véase el cuadro 7-5.)

Cuadro 7-5/J.181 – splice_insert()

Sintaxis	Bits	Mnemónica
splice_insert() {		
splice_event_id	32	bslbf
splice_event_cancel_indicator	1	bslbf
reserved	7	bslbf
if (splice_event_cancel_indicator = "0") {		
out_of_network_indicator	1	bslbf
program_splice_flag	1	bslbf
duration_flag	1	bslbf
splice_immediate_flag	1	bslbf
reserved	4	bslbf
if ((program_splice_flag = "1") && (splice_immediate_flag = "0"))		
splice_time()		
if (program_splice_flag = "0") {		
component_count	8	uimsbf
for (I = 0; I < component_count; I++) {		
component_tag	8	uimsbf
if (splice_immediate_flag = "0")		
splice_time()		
}		
}		
if (duration_flag = "1")		
break_duration()		
unique_program_id	16	uimsbf
avail	8	uimsbf
avail_count	8	uimsbf
}		
}		

7.3.3.1 Definición semántica de los campos de splice_insert()

splice_event_id: Identificador de evento de empalme único de 32 bits.

splice_event_cancel_indicator: Bandera de 1 bit que, cuando está puesto a "1", indica que ha sido cancelado un evento de empalme enviado previamente, identificado por splice_event_id.

out_of_network_indicator: Bandera de 1 bit. Cuando está puesto a "1", indica que el evento de empalme es una oportunidad para salir del programa de la red y que el valor de splice_time() se referirá a un punto de salida o punto de salida de programa pretendido. Cuando está puesto a "0", la bandera indica que el evento de empalme es una oportunidad para volver al programa de red y que el valor de splice_time() se referirá a un punto de entrada o un punto de entrada en programa pretendido.

program_splice_flag: Bandera de 1 bit que, cuando está puesto a "1", indica que el mensaje se refiere a un punto de empalme de programa y que el modo es el modo empalme de programa según el cual todos los PID/componentes del programa han de ser empalmados. Cuando está puesto a "0",

este campo indica que el modo es el modo empalme de componente según el cual cada componente que haya de ser empalmado será indicado de manera separada por la sintaxis que sigue.

duration_flag: Bandera de 1 bit que, cuando está puesto a "1", indica la presencia del campo `break_duration()`.

splice_immediate_flag (bandera de empalme inmediato): Cuando esta bandera es "1", indica la ausencia del campo `splice_time()` y que el modo de empalme deberá ser el modo empalme inmediato, según el cual el dispositivo empalmador deberá aprovechar la primera oportunidad en el tren, relativa al paquete de información de empalme, para empalmar. Cuando esta bandera es "0", indica la presencia del campo `splice_time()` en al menos un lugar dentro de la instrucción `splice_insert()`.

component_count: Entero sin signo de 8 bits que especifica el número de ejemplares de datos del tren PID elemental en el bucle que sigue. Los componentes equivalen a trenes de PID elementales.

component_tag: Valor de 8 bits que identifica el tren de PID elemental que contiene el punto de empalme especificado por el valor de `splice_time()` que sigue. El valor deberá ser el mismo que el utilizado en el `stream_identification_descriptor()` (descriptor de indicación de trenes) para identificar el tren de PID elemental.

unique_program_id: Este valor deberá proporcionar una identificación única para un evento de observación dentro del servicio. Puesto que esta Recomendación se refiere a toda norma global, es preciso un valor independiente de cualquier norma. Realiza la misma función que el campo `event_id` (identificador de evento) en la norma ATSC o en la norma DVB.

avail: Este campo proporciona la identificación de una cuña específica dentro de un `unique_program_id`. Está previsto que este valor aumente con cada nueva cuña de un evento de observación, y que vuelva a uno con la primera cuña de un evento de observación nuevo. Este campo aumentará, según lo previsto, con cada nueva cuña. Puede llevar, facultativamente, un valor de cero para indicar su no utilización.

avail_count: Este campo proporciona la cuenta del número esperado de cuñas individuales en el evento de observación en curso. Cuando este campo es cero, indica que el campo cuña carece de significado.

7.4 Tiempo

7.4.1 `splice_time()`

La estructura de `splice_time()` especifica el tiempo o momento del evento de empalme. (Véase el cuadro 7-6.)

Cuadro 7-6/J.181 – `splice_time()`

Sintaxis	Bits	Mnemónica
<code>splice_time() {</code>		
time_specified_flag	1	bslbf
if (time_specified_flag = 1) {		
reserved	6	bslbf
pts_time	33	uimsbf
}		
else		
reserved	7	bslbf
}		

7.4.1.1 Definición semántica de los campos de splice_time()

time_specified_flag (bandera de tiempo especificado): Bandera de 1 bit que, cuando está puesto a "1", indica la presencia del campo pts_time y los bits reservados asociados.

pts_time: Campo de 33 bits que indica el número de "ticks" del reloj del programa a 90 kHz. Este campo representa el tiempo de presentación del punto de empalme pretendido, que es equivalente al tiempo de presentación de la unidad de acceso que sigue al punto de empalme pretendido.

7.4.2 break_duration()

La estructura break_duration() especifica la duración de la interrupción o las interrupciones comerciales. Se puede utilizar para dar al empalmador una indicación de cuándo concluirá la interrupción y cuándo se volverá a producir el punto de entrada en la red. (Véase el cuadro 7-7.)

Cuadro 7-7/J.181 – break_duration()

Sintaxis	Bits	Mnemónica
break_duration() {		
auto_return	1	bslbf
reserved	6	bslbf
duration	33	uimsbf
}		

7.4.2.1 Definición semántica de los campos de break_duration()

auto_return (retorno autónomo): Bandera de 1 bit que, cuando está puesto a "1", indica que la duración será utilizada por el dispositivo empalmador para saber cuándo tendrá lugar el retorno al programa de la red (fin de la interrupción). No está previsto enviar una instrucción splice_insert() con el out_of_network_indicator puesto a "0" para terminar la interrupción. Cuando esta bandera es "0", no es preciso que el campo de duración, si está presente, termine la interrupción porque para concluirla se enviará una nueva instrucción splice_insert(). En este caso, la presencia del campo break_duration actúa como mecanismo de seguridad si se pierde la instrucción splice_insert() al final de una interrupción.

duration (duración): Campo de 33 bits que indica el tiempo transcurrido en términos de ticks del reloj del programa a 90 kHz.

7.5 Constricciones

7.5.1 Constricciones impuestas a splice_info_section()

La splice_info_section se llevará en un tren de PID que es específico de un programa y al que se hace referencia en la PMT. El PID de splice_info_section deberá estar identificado en la PMT por un stream_type (tipo de tren) igual a 0x86.

La splice_info_section llevada en un tren de PID al que se hace referencia en la PMT de un programa deberá contener solamente información sobre los eventos de empalme que ocurran en ese programa.

Un evento de empalme se definirá mediante un solo valor de splice_event_id.

Si se va a utilizar el modo empalme de componente, cada tren de PID elemental deberá ser identificado mediante un stream_identifier_descriptor llevado en el circuito de la PMT, uno por cada PID. El stream_identifier_descriptor llevará un component_tag, que corresponda exclusivamente a un tren de PID entre los contenidos dentro de un programa e indicados en la PMT para ese programa.

Cualquier `splice_event_id` que se envíe en una `splice_info_section` utilizando una instrucción `splice_schedule()` deberá ser enviado de nuevo antes del evento mediante una instrucción `splice_insert()`. Por ello, habrá una correspondencia entre los valores de `splice_event_id` elegidos para determinados eventos señalados por la instrucción `splice_schedule()` (futuro lejano) y los valores de `splice_event_id` utilizados en la instrucción `splice_insert()` (futuro cercano) para indicar los mismos eventos.

No es necesario enviar los valores de `splice_event_id` en un orden creciente en mensajes subsiguientes ni han de ser incrementados cronológicamente. Los valores de `splice_event_id` se pueden elegir de manera aleatoria. Cuando se utilice la expresión `splice_schedule()`, los valores de `splice_event_id` deberán ser únicos durante el periodo de la instrucción `splice_schedule()`. Un valor de `splice_event_id` puede ser reutilizado cuando haya pasado su `splice_time` asociado.

Si la `splice_immediate_flag` está puesta a "1", se interpretará que el tiempo de empalme es el momento presente. Esto es lo que se llama "modo empalme inmediato". Cuando se utiliza esta forma con la instrucción `splice_insert()`, el empalme se puede producir en la oportunidad más próxima (anterior o posterior) detectada por el empalmador. El "modo empalme inmediato" puede ser utilizado para empalmar puntos de entrada o de salida, es decir, para los dos estados del `out_of_network_indicator`.

Habrá que prever que cualquier cuña pueda ser terminada con un mensaje de modo empalme de programa, un mensaje de modo empalme de componente o sin mensaje alguno (con lo que se alcanza la `break_duration`) con independencia de la naturaleza del mensaje al comienzo de la cuña.

7.5.2 Constricciones impuestas a la interpretación de tiempo

7.5.2.1 Constricciones impuestas a `splice_time()` para `splice_insert()`

Si `splice_command_type` es igual a 0x05 (inserción), se impondrán las siguientes constricciones a `splice_time()`:

Para especificar un punto de salida de programa, esto es, cuando la `program_splice_flag` sea igual a 1 y el `out_of_network_indicator` sea igual a 1, el valor de `pts_time` deberá ser igual al tiempo de presentación del punto de salida de programa pretendido.

Para especificar un punto de salida en un tren de PID elemental, esto es, cuando la `program_splice_flag` sea igual a 0 y el `out_of_network_indicator` sea igual a 1, el valor de `pts_time` deberá ser igual al tiempo de presentación del punto de salida pretendido del tren de PID elemental que corresponde al valor de `component_tag`.

Un punto de salida se definirá de manera que tenga el mismo tiempo de presentación que la unidad de acceso que le sigue inmediatamente en el tren. Así pues, el punto de salida pretendido de un evento de empalme señalado deberá ser el punto de salida inmediatamente anterior a la unidad de acceso cuyo tiempo de presentación concuerda con el `splice_time()` señalado.

Para especificar un punto de entrada en programa, esto es, cuando la `program_splice_flag` sea igual a 1 y el `out_network_indicator` sea igual a 0, el valor de `pts_time` deberá ser igual al tiempo de presentación del punto de entrada en programa pretendido.

Para especificar un punto de entrada en un tren de PID elemental, esto es, cuando la `program_splice_flag` sea igual a 0 y el `out_of_network_indicator` sea igual a 0, el valor de `pts_time` deberá ser igual al tiempo de presentación del punto de entrada pretendido del tren de PID elemental que corresponde al valor de `component_tag`.

Un punto de entrada se definirá de manera que tenga el mismo tiempo de presentación que la unidad de acceso que le sigue inmediatamente en el tren. Así pues, el punto de entrada pretendido de un evento de empalme señalado deberá ser el punto de entrada inmediatamente anterior a la unidad de acceso cuyo tiempo de presentación concuerda con el `splice_time()` señalado.

Cuando esté en vigor el modo empalme de componente y el `out_of_network_indicator` sea "1" (comienzo de una interrupción), cada componente indicado en el bucle de componentes de `splice_insert()` deberá ser cambiado del componente de la red al componente suministrado por el empalmador en el momento indicado. Los componentes no indicados en el bucle de componentes del mensaje permanecerán inalterados: si un componente de salida de empalmador fuese a la sazón el componente de la red, seguirá siendo el componente de la red; si un componente de salida de empalmador fuese a la sazón el componente suministrado por el empalmador, seguirá siendo el componente suministrado por el empalmador.

Cuando esté en vigor el modo empalme de componentes y el `out_of_network_indicator` sea "0" (fin de una interrupción), cada componente indicado en el bucle de componentes de `splice_insert()` deberá ser cambiado del componente suministrado por el empalmador al componente de la red en el momento indicado. Los componentes no indicados en el bucle de componentes del mensaje permanecerán inalterados: si un componente de salida de empalmador fuese a la sazón el componente de la red, seguirá siendo el componente de la red; si un componente de salida de empalmador fuese a la sazón el componente suministrado por el empalmador, seguirá siendo el componente suministrado por el empalmador.

Cuando esté en vigor el modo empalme de componente y no lo esté en cambio el modo empalme inmediato, será necesario que el primer componente indicado en el bucle de componentes de la instrucción `splice_insert()` tenga un `pts_time` válido en su `splice_time()` asociado y deberá hacerse referencia a este `pts_time` como el `pts_time` por defecto. Los componentes subsiguientes indicados en el bucle de componentes del mismo mensaje, que no tengan un `pts_time` asociado, utilizarán este `pts_time` por defecto. Se permitirá que cualquiera de los componentes, y todos ellos, a continuación del primer componente indicado de una instrucción `splice_insert()` contenga(n) un `pts_time` único diferente del `pts_time` por defecto.

7.5.2.2 Constricciones impuestas a `break_duration()` para `splice_insert()`

Si `splice_command_type` es igual a 0x05 (inserción), se impondrán las siguientes constricciones a `break_duration()`:

El valor dado en `break_duration()` se interpreta como la duración pretendida de la interrupción comercial. Es un campo facultativo que se ha de utilizar cuando el `out_of_network_indicator` sea igual a 1. Se puede utilizar en la misma instrucción `splice_insert()` que especifica el tiempo de comienzo de la interrupción, con lo que el empalmador puede calcular el momento en que se acabará la interrupción.

Las interrupciones pueden ser terminadas emitiendo una instrucción `splice_insert()` con el `out_of_network_indicator` puesto a 0. Se puede dar un `splice_time()` o se puede utilizar el modo empalme inmediato. Cuando se haya dado una `break_duration` al comienzo de la interrupción estando `auto_return` puesto a 0, se podrá utilizar el valor de `break_duration` como mecanismo de reserva para asegurar que se produce realmente el retorno a la red en caso de pérdida de un paquete de aviso.

Las interrupciones se pueden terminar también dando una duración de la interrupción al comienzo de la misma y haciendo depender del dispositivo empalmador el retorno al programa de la red en el momento apropiado. La bandera `auto_return` debe ser 1. A este modo se le denominará retorno autónomo. Las interrupciones del modo autónomo pueden, no obstante, terminarse pronto. Para terminar una interrupción prematuramente puede darse una segunda instrucción `splice_insert()`, en la que el `out_of_network_indicator` sea igual a 0. El nuevo momento del empalme de vuelta a la red lo puede dar un `splice_time()` actualizado, o bien se puede utilizar el modo empalme inmediato.

8 Descriptores de empalmes

8.1 Visión general

El `splice_descriptor` (descriptor de empalme) es un prototipo para la adición de nuevos campos a la `splice_info_section`. Todos los descriptores incluidos utilizan la misma sintaxis con los seis primeros bytes. Para hacer posible la adición de información privada, se ha incluido el código "identificador". Así se elimina la necesidad de un descriptor de registro en el bucle del descriptor.

Cualquier equipo receptor deberá pasar por alto los descriptores con identificadores desconocidos o rótulos de descriptor desconocidos. En el caso de descriptores con identificadores conocidos, el equipo receptor deberá hacer caso omiso de los descriptores cuyo `splice_descriptor_tag` (rótulo de descriptor de empalme) no se conozca.

En la `splice_info_section` pueden existir descriptores `splice_descriptor` para ampliaciones específicas de las diversas instrucciones. (Véase el cuadro 8-1.)

Cuadro 8-1/J.181 – Rótulos de descriptores de empalme

Rótulo	Descriptores del identificador "CUEI"
0x00	avail_descriptor
0x01-0xFF	Reservado para futuros <code>splice_descriptor</code> de SCTE

8.2 Descriptor de empalme

La sintaxis del descriptor de empalme que se indica en esta cláusula se ha de utilizar como plantilla para implementaciones específicas de un descriptor destinado a la `splice_info_section`. Se señala que los descriptores de empalme sólo se utilizan dentro de una `splice_info_section`. No se utilizan en la sintaxis de MPEG, por ejemplo la PMT, o en la sintaxis de cualquier otra norma. De esta manera es posible aprovechar toda la gama de rótulos de descriptor cuando se definen nuevos descriptores. (Véase el cuadro 8-2.)

Cuadro 8-2/J.181 – `splice_descriptor()`

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

8.2.1 Definición semántica de los campos de `splice_descriptor()`

splice_descriptor_tag: Este número de 8 bits define la sintaxis de los bytes privados que forman el cuerpo del descriptor. Los rótulos del descriptor los define el propietario del descriptor, registrado como utilizador del identificador.

descriptor_length: Este número de 8 bits da la longitud, en bytes, del descriptor que sigue a este campo. Los descriptores están limitados a 256 bytes, por lo que este valor está limitado a 254.

identifier (identificador): Este número de 32 bits se utiliza para identificar al propietario del descriptor. El SCTE reserva el código 0x43554549 (ASCII "CUEI").

private_byte (byte privado): El resto del descriptor se dedica a los campos de datos que requiera el descriptor que se define.

8.3 Descriptores de empalmes específicos

8.3.1 avail_descriptor()

El `avail_descriptor` (descriptor de cuña) es una implementación de un `splice_descriptor`. Proporciona una ampliación facultativa a la instrucción `splice_insert()` que permite el envío de un identificador de autorización para una cuña. Se pueden incluir múltiples copias de este descriptor utilizando el mecanismo de bucle proporcionado. Este identificador tiene por objeto reproducir la funcionalidad del sistema de tono de aviso utilizado en sistemas analógicos para la inserción de anuncios. Sólo se habrá de utilizar con una instrucción `splice_insert()`, dentro de una `splice_info_section`. (Véase el cuadro 8-3.)

Cuadro 8-3/J.181 – avail_descriptor()

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

8.3.1.1 Definición semántica de los campos de `avail_descriptor()`

splice_descriptor_tag: Este número de 8 bits define la sintaxis de los bytes privados que forman el cuerpo del descriptor. El `splice_descriptor_tag` tendrá un valor de 0x00.

descriptor_length: Este número de 8 bits da la longitud, en bytes, del descriptor que sigue a este campo. El campo `descriptor_length` tendrá un valor de 0x08.

identifier: Este número de 32 bits se utiliza para identificar al propietario del descriptor. El identificador tendrá un valor de 0x43554549 (ASCII "CUEI").

provider_avail_id (identificador de cuña de proveedor): Este número de 32 bits proporciona la información que un dispositivo receptor puede utilizar para alterar su comportamiento durante una cuña, o fuera de la misma. Se puede utilizar de manera similar a los tonos de aviso analógicos. Un ejemplo sería el de una red que indicara a un afiliado o extremo de cabecera que oscureciera un evento deportivo.

9 Criptación

9.1 Visión general

La `splice_info_section` soporta la criptación de un tramo de la sección para poder impedir que accedan a las cuñas los receptores no autorizados. En esta cláusula se describen los diversos algoritmos de criptación que se pueden utilizar. La criptación de la sección es facultativa, como lo es la implementación de la criptación por el creador del mensaje o por cualquier dispositivo de recepción. La utilización de la criptación se considera facultativa para que un fabricante pueda enviar

sistemas "en claro" sin preocuparse respecto a la exportación de la tecnología de la criptación. Si la criptación está incluida en el sistema, cualquier dispositivo receptor implementará todos los algoritmos indicados en esta Recomendación, lo que permite al creador de una tabla de información de empalme utilizar cualquiera de los algoritmos en una transmisión. La utilización de tecnología de criptación privada es opcional, y queda fuera del alcance de la presente Recomendación.

9.2 Criptación de clave fija

La criptación utilizada con esta Recomendación supone que se va a utilizar una clave fija. Se proporciona la misma clave al transmisor y al receptor. El método de entrega de la clave a todas las partes no se especifica. Esta Recomendación permite disponer de hasta 256 claves diferentes para la descripción. El campo `cw_index` se utiliza para determinar qué clave deberá emplearse cuando se describe una sección. La longitud de la clave fija depende del tipo del algoritmo utilizado. Se supone que la clave fija entregada a todas las partes tendrá la longitud correcta para el algoritmo que se pretende utilizar.

9.3 Algoritmos de criptación

El campo `encryption_algorithm` (algoritmo de criptación) de la `splice_info_section` (sección de información de empalme) es un valor de 6 bits, que puede contener uno de los valores mostrados en el cuadro 9-1. Todas las variantes de la norma de criptación de datos utilizan una clave de 64 bits (de hecho, 56 bits más una suma de control) para criptar o descripar un bloque de 8 bytes. En el caso de DES triple, se necesitarán tres claves de 64 bits, una por cada uno de los tres pasos del algoritmo DES. El DES triple "normalizado" utiliza en realidad dos claves, porque la primera y la tercera son idénticas.

Cuadro 9-1/J.181 – Algoritmo de criptación

Valor	Algoritmo de criptación	Norma
0	Sin criptación	
1	Modo DES-ECB	ANSI X9.32
2	Modo DES-CBC	ANSI X9.32
3	Modo DES EDE3-ECB triple	ANSI X9.52
4-31	Reservado	
32-63	Privado de usuario	

9.3.1 Modo DES-ECB

Este algoritmo utiliza la "Norma de criptación de datos", en el modo libro de códigos electrónico.

Para utilizar este tipo de criptación, los datos criptados deben contener un múltiplo de 8 bytes de datos, del campo `splice_command_type` al campo `E_CRC_32`. El bucle de `alignment_stuffing` se puede utilizar para rellenar cualesquiera bytes adicionales que pudieran necesitarse.

9.4 Modo DES-CBC

Este algoritmo utiliza la "Norma de criptación de datos", en el modo encadenamiento de bloques de cifras. El algoritmo básico es idéntico al DES ECB. A cada bloque de texto en claro de 64 bits se le aplica un operador lógico OR exclusivo binario con el bloque de texto cifrado previo antes de ser criptado con la clave DES. Al primer bloque se le aplica un operador lógico OR exclusivo con un vector inicial. Para los fines de esta Recomendación, el vector inicial tendrá un valor fijo de cero.

Para utilizar este tipo de criptación, los datos criptados deben contener un múltiplo de 8 bytes de datos, del campo `splice_command_type` al campo `E_CRC_32`. El bucle de `alignment_stuffing` se puede utilizar para rellenar cualesquiera bytes adicionales que pudieran necesitarse.

9.5 Modo DES EDE3-ECB triple

Este algoritmo utiliza tres claves de 64 bits, empleando cada una de ellas en uno de los pasos del algoritmo DES-ECB. En el dispositivo de transmisión, cada bloque de datos se cripta primero con la primera clave, se descripta con la segunda clave y finalmente se cripta con la tercera clave. En el lugar de recepción, cada bloque es descriptado primero con la tercera clave, criptado con la segunda y finalmente descriptado con la primera.

Para utilizar este tipo de criptación, los datos criptados deben contener un múltiplo de 8 bytes de datos, del campo `splice_command_type` al campo `E_CRC_32`. El bucle de `alignment_stuffing` se puede utilizar para rellenar cualesquiera bytes adicionales que pudieran necesitarse.

9.6 Algoritmos privados de usuario

Esta Recomendación permite la utilización de algoritmos de criptación privados. No se especifica cómo se ponen de acuerdo los dispositivos de transmisión y recepción sobre el algoritmo a utilizar con cualquier código privado de usuario. Tampoco se especifica cómo debe registrarse o administrarse la coordinación de valores privados del campo `encryption_algorithm`.

APÉNDICE I

Bibliografía

- SMPTE 312M-1999, *Television – Splice Points for MPEG-2 Transport Streams*.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación