

Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

H.222.0

(05/2006)

SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIA

Infraestructura de los servicios audiovisuales –
Multiplexación y sincronización en transmisión

**Tecnología de la información – Codificación
genérica de imágenes en movimiento e
información de audio asociada: Sistemas**

Recomendación UIT-T H.222.0

RECOMENDACIONES UIT-T DE LA SERIE H
SISTEMAS AUDIOVISUALES Y MULTIMEDIA

CARACTERÍSTICAS DE LOS SISTEMAS VIDEOTELEFÓNICOS	H.100–H.199
INFRAESTRUCTURA DE LOS SERVICIOS AUDIOVISUALES	
Generalidades	H.200–H.219
Multiplexación y sincronización en transmisión	H.220–H.229
Aspectos de los sistemas	H.230–H.239
Procedimientos de comunicación	H.240–H.259
Codificación de imágenes vídeo en movimiento	H.260–H.279
Aspectos relacionados con los sistemas	H.280–H.299
Sistemas y equipos terminales para los servicios audiovisuales	H.300–H.349
Arquitectura de servicios de directorio para servicios audiovisuales y multimedia	H.350–H.359
Arquitectura de la calidad de servicio para servicios audiovisuales y multimedia	H.360–H.369
Servicios suplementarios para multimedia	H.450–H.499
PROCEDIMIENTOS DE MOVILIDAD Y DE COLABORACIÓN	
Visión de conjunto de la movilidad y de la colaboración, definiciones, protocolos y procedimientos	H.500–H.509
Movilidad para los sistemas y servicios multimedia de la serie H	H.510–H.519
Aplicaciones y servicios de colaboración en móviles multimedia	H.520–H.529
Seguridad para los sistemas y servicios móviles multimedia	H.530–H.539
Seguridad para las aplicaciones y los servicios de colaboración en móviles multimedia	H.540–H.549
Procedimientos de interfuncionamiento de la movilidad	H.550–H.559
Procedimientos de interfuncionamiento de colaboración en móviles multimedia	H.560–H.569
SERVICIOS DE BANDA ANCHA Y DE TRÍADA MULTIMEDIA	
Servicios multimedia de banda ancha sobre VDSL	H.610–H.619

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

Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Sistemas

Resumen

En esta Recomendación | Norma Internacional se especifica la capa de sistema de la codificación. Elaborada en 1994, es el principal soporte de la combinación y sincronización de los métodos de codificación de vídeo y audio definidos en las Partes 2 y 3 de ISO/CEI 13818. Desde 1994, esta norma se ha ampliado para soportar otras especificaciones de codificación de vídeo (ISO/CEI 14496-2 e ISO/CEI 14496-10), especificaciones de codificación de audio (ISO/CEI 13818-7 e ISO/CEI 14496-3), trenes de sistema (ISO/CEI 14496-1 e ISO/CEI 15938-1), IPMP (ISO/CEI 13818-11), así como metadatos genéricos. La capa de sistema soporta seis funciones básicas:

- 1) la sincronización de múltiples trenes comprimidos en la decodificación;
- 2) el entrelazado de múltiples trenes comprimidos en un solo tren;
- 3) la inicialización de almacenamiento en memoria tampón para comenzar la decodificación;
- 4) la gestión continua de la memoria tampón;
- 5) la identificación de tiempo; y
- 6) la multiplexación y señalización de varios componentes en un tren de sistema.

Un tren de bits multiplexado de la Rec. UIT-T H.222.0/ISO | CEI 13818-1 es un **tren de transporte** o un **tren de programa**. Ambos se construyen a partir de **paquetes PES** y paquetes que contienen otra información necesaria. Los dos tipos de tren soportan la multiplexación de trenes comprimidos de vídeo y de audio de un programa con una base de tiempo común. Además, el **tren de transporte** soporta la multiplexación de trenes comprimidos de vídeo y audio de múltiples programas con bases de tiempo independientes. En los entornos casi sin errores, suele resultar más adecuado el **tren de programa** que soporta el procesamiento informático de la información de programa. El **tren de transporte** es más conveniente en los entornos donde es probable que haya errores.

Un tren de bits multiplexado de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, ya sea un tren de transporte o un tren de programa, está formado por dos capas: la capa externa es la capa de sistema, y la capa interna es la capa de compresión. La capa de sistema proporciona las funciones necesarias para utilizar uno o más trenes de datos comprimidos en un sistema. Las partes de esta especificación relativas al vídeo y el audio definen la capa de codificación de compresión para los datos de audio y vídeo. No se define en esta Recomendación | Norma Internacional la codificación de otro tipo de datos, aunque la soporta la capa de sistema, siempre y cuando los otros tipos de datos se ajusten a las restricciones definidas en esta Recomendación | Norma Internacional.

Orígenes

La Recomendación UIT-T H.222.0 fue aprobada el 29 de mayo de 2006 por la Comisión de Estudio 16 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8. Se publica también un texto idéntico como Norma Internacional ISO/CEI 13818-1.

PREFACIO

La Unión Internacional de Telecomunicaciones (UIT) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones y de las tecnologías de la información y la comunicación. El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) 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.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

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 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 en la dirección <http://www.itu.int/ITU-T/ipr/>.

© UIT 2008

Reservados todos los derechos. Ninguna parte de esta publicación puede reproducirse por ningún procedimiento sin previa autorización escrita por parte de la UIT.

ÍNDICE

	<i>Página</i>
SECCIÓN 1 – GENERALIDADES	1
1.1 Alcance	1
1.2 Referencias normativas	1
SECCIÓN 2 – ELEMENTOS TÉCNICOS	2
2.1 Definiciones	2
2.2 Símbolos y abreviaturas	6
2.3 Método para describir la sintaxis del tren de bits	8
2.4 Requisitos del tren de bits del tren de transporte	9
2.5 Requisitos del tren de bits del tren de programa	54
2.6 Descriptores de programas y de elementos de programa	68
2.7 Restricciones impuestas a la semántica del tren multiplexado	99
2.8 Compatibilidad con ISO/CEI 11172	104
2.9 Registro de identificadores de derechos de autor	104
2.10 Registro de formato de datos privados	104
2.11 Transporte de datos ISO/CEI 14496	105
2.12 Transporte de metadatos	117
2.13 Transporte de datos ISO 15938	126
2.14 Transporte de vídeo de la Rec. UIT-T H.264 ISO/CEI 14496-10	126
Anexo A – Modelo de decodificador CRC	131
A.0 Modelo de decodificador CRC	131
Anexo B – Instrucciones y control del medio de almacenamiento digital (DSM-CC)	132
B.0 Introducción	132
B.1 Elementos generales	133
B.2 Elementos técnicos	135
Anexo C – Información específica de programa	141
C.0 Explicación de la información específica de programa en trenes de transporte	141
C.1 Introducción	141
C.2 Mecanismo funcional	141
C.3 Correspondencia de secciones con paquetes del tren de transporte	142
C.4 Velocidades de repetición y acceso aleatorio	143
C.5 ¿Qué es un programa?	143
C.6 Atribución de número de programa	143
C.7 Utilización de PSI en un sistema típico	144
C.8 Relaciones de estructuras PSI	144
C.9 Utilización de anchura de banda y tiempo de adquisición de señal	147
Anexo D – Modelo de temporización de sistemas de la presente Recomendación Norma Internacional y repercusiones de aplicación	149
D.0 Introducción	149
Anexo E – Aplicaciones de transmisión de datos	160
E.0 Consideraciones	160
E.1 Sugerencia	160
Anexo F – Gráficos de sintaxis para la presente Recomendación Norma Internacional	161
F.0 Introducción	161
Anexo G – Información general	166
G.0 Información general	166
Anexo H – Datos privados	167
H.0 Datos privados	167
Anexo I – Conformidad de sistemas e interfaz en tiempo real	168
I.0 Conformidad de sistemas e interfaz en tiempo real	168

Anexo J – Interfaz entre redes que inducen fluctuación de fase y decodificadores MPEG-2	169
J.0 Introducción.....	169
J.1 Modelos de conformidad de red	169
J.2 Especificación de red para suprimir la fluctuación de fase	170
J.3 Ejemplo de implementaciones de decodificador	171
Anexo K – Empalme de trenes de transporte	172
K.0 Introducción.....	172
K.1 Diferentes tipos de punto de empalme	173
K.2 Comportamiento del codificador en caso de empalmes	173
Anexo L – Procedimiento de registro (véase 2.9).....	175
L.1 Procedimiento para solicitar un identificador registrado (RID)	175
L.2 Responsabilidades de la autoridad de registro	175
L.3 Responsabilidades de las partes que soliciten un RID	175
L.4 Procedimiento de apelación contra denegaciones de solicitudes.....	176
Anexo M – Formulario de solicitud de registro (véase 2.9).....	177
M.1 Datos de referencia de la organización que solicita un identificador registrado (RID)	177
M.2 Declaración de intención de aplicar el RID asignado	177
M.3 Fecha prevista de aplicación del RID	177
M.4 Representante autorizado	177
M.5 Sólo para uso oficial de la autoridad de registro.....	177
Anexo N	178
Anexo O – Procedimiento de registro (véase 2.10).....	179
O.1 Procedimiento para solicitar un identificador registrado (RID)	179
O.2 Responsabilidades de la autoridad de registro	179
O.3 Datos de referencia para la autoridad de registro.....	179
O.4 Responsabilidades de las partes que solicitan un RID	179
O.5 Procedimiento de apelación contra denegaciones de solicitudes.....	179
Anexo P – Formulario de solicitud de registro	181
P.1 Datos de referencia de la organización que solicita un identificador registrado (RID)	181
P.2 Solicitud de un RID específico.....	181
P.3 Breve descripción del RID en uso y fecha en que se estableció el sistema.....	181
P.4 Declaración de intención de aplicar el RID asignado	181
P.5 Fecha prevista de aplicación del RID	181
P.6 Representante autorizado	181
P.7 Para uso oficial de la autoridad de registro	181
Anexo Q – Modelos de memoria tampón de T-STD y P-STD para ADTS de ISO/CEI 13818-7.....	182
Q.1 Introducción.....	182
Q.2 Velocidad de fuga de la memoria tampón de transporte	182
Q.3 Tamaño de memoria tampón	182
Q.4 Conclusión.....	183
Anexo R – Transporte de escenas ISO/CEI 14496 en trenes de la Rec. UIT-T H.222.0 ISO/CEI 13818-1	184
R.1 Procedimiento de acceso de contenido para componentes de programa de ISO/CEI 14496 dentro de un tren de programa	184
R.2 Procedimiento de acceso de contenido para componentes de programa de ISO/CEI 14496 dentro de un tren de transporte	185

Introducción

La parte relativa a sistemas de la presente Recomendación | Norma Internacional trata de la combinación de uno o más trenes elementales de vídeo y de audio y de otros datos en un solo tren o múltiples trenes adecuados para almacenamiento o transmisión. La codificación de sistemas sigue las reglas sintácticas y semánticas impuestas por esta Especificación y proporciona información para permitir la decodificación sincronizada de las memorias tampón del decodificador en una amplia gama de condiciones de extracción o recepción.

La codificación de sistemas se especificará en dos formas: el **tren de transporte** (TS, *transport stream*) y el **tren de programa** (PS, *program stream*), cada uno de los cuales está optimizado para un conjunto diferente de aplicaciones. El tren de transporte y el tren de programa definidos en esta Recomendación | Norma Internacional proporcionan la sintaxis de codificación necesaria y suficiente para sincronizar la decodificación y presentación de la información de audio y de vídeo, a la vez que asegura que las memorias tampón de datos en los decodificadores no están desbordadas ni subutilizadas. La información se codifica en la sintaxis utilizando indicaciones de tiempo relativas a la decodificación y presentación de datos de audio y visuales codificados e indicaciones de tiempo relativas a la entrega del propio tren de datos. Ambos trenes se definen como múltiplex de paquetes.

El método de multiplexación básico para trenes elementales de vídeo y de audio se ilustra en la figura Intro. 1. Los datos de vídeo y de audio se codifican como se describe en la Rec. UIT-T H.262 | ISO/CEI 13818-2 y en ISO/CEI 13818-3. Los trenes elementales comprimidos resultantes son paquetizados para producir **paquetes de tren elemental paquetizado** (PES, *packetized elementary streams*). Se puede añadir la información necesaria para utilizar los paquetes PES con independencia de los trenes de transporte o de los trenes de programa cuando se forman paquetes PES. Esta información no es necesaria y no hay que añadirla cuando los paquetes PES se combinan después con información de nivel de sistemas para formar **trenes de transporte** o **trenes de programa**. Esta norma de sistemas trata de los procesos situados a la derecha de la línea vertical de trazo interrumpido.

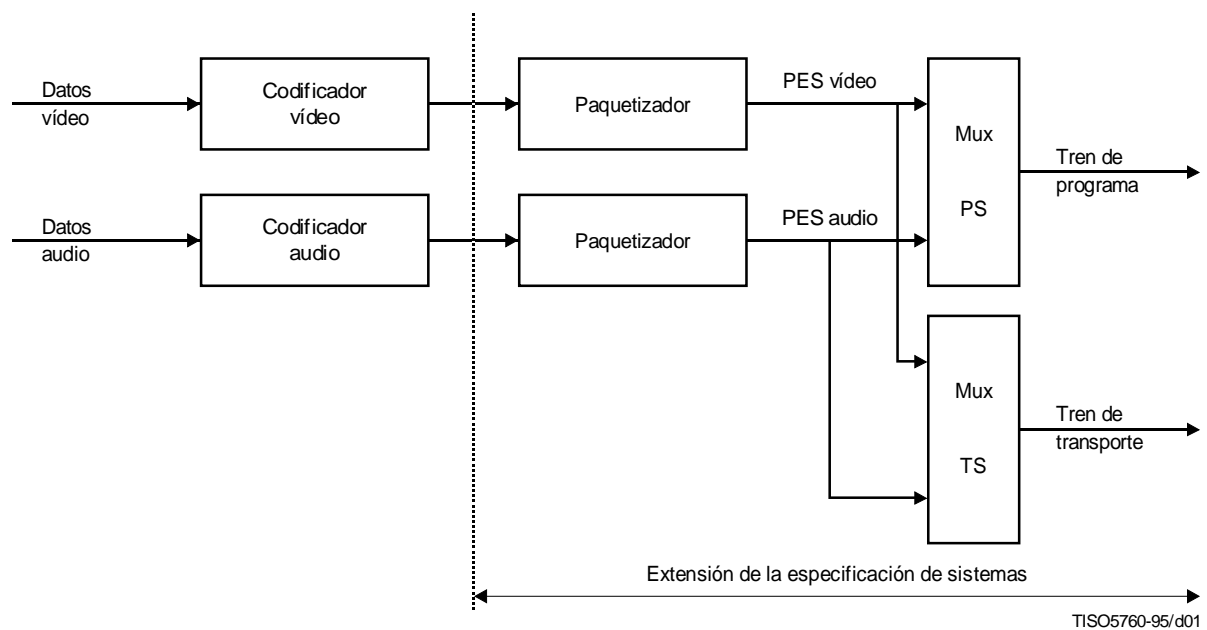


Figura Intro. 1 – Visión de conjunto simplificada del alcance de esta Recomendación | Norma Internacional

El **tren de programa** es análogo y similar a la capa de sistemas de ISO/CEI 11172. Resulta de la combinación en un solo tren de uno o más trenes de paquetes PES que tienen una base de tiempos común.

Para las aplicaciones que requieren que los trenes elementales que comprenden un solo programa estén en trenes separados que no estén multiplexados, los trenes elementales pueden ser codificados también con trenes de programa, separados uno por cada tren elemental, con una base de tiempos común. En este caso, los valores codificados en los campos de referencia de reloj de sistema de los diversos trenes deberán concordar.

Al igual que el tren de programa, todos los trenes elementales pueden ser decodificados con sincronización.

El tren de programa está diseñado para utilización en entornos relativamente libres de errores y es adecuado para aplicaciones que pueden conllevar procesamiento de soporte lógico de información de sistemas, tales como aplicaciones multimedia interactivas. Los paquetes del tren de programa pueden ser de longitud variable y relativamente grande.

El **tren de transporte** combina uno o más programas con una o más bases de tiempos independientes en un solo tren. Los paquetes PES compuestos de trenes elementales que forman un programa comparten una base de tiempos común. El tren de transporte está diseñado para utilización en entornos donde es probable que haya errores, tales como almacenamiento o transmisión en medios propensos a pérdidas y con alto nivel de ruido. Los paquetes del tren de transporte tienen una longitud de 188 bytes.

Los trenes de programa y de transporte están diseñados para diferentes aplicaciones y sus definiciones no siguen estrictamente un modelo estratificado. Es posible y razonable convertirlos entre sí, aunque uno no es un subconjunto o superconjunto del otro. En particular, es posible la extracción del contenido de un programa de un tren de transporte y la creación de un tren de programa válido y se realiza mediante el formato de intercambio común de paquetes PES, pero no todos los campos necesarios en un tren de programa están contenidos en el tren de transporte, por lo que algunos deben ser derivados. El tren de transporte se puede utilizar para abarcar una gama de capas en un modelo estratificado, y está diseñado con miras a la implementación eficaz y fácil en aplicaciones de gran anchura de banda.

El alcance de las reglas sintácticas y semánticas expuestas en la especificación de sistemas difiere: las reglas sintácticas se aplican solamente a la codificación de la capa de sistemas y no se extienden a la codificación de la capa de compresión de las especificaciones de vídeo y de audio; en cambio, las reglas semánticas se aplican a todo el tren combinado.

La especificación de sistemas no establece la arquitectura o la implementación de codificadores o decodificadores ni de multiplexores o demultiplexores. Sin embargo, las propiedades de tren de bits imponen requisitos funcionales y de calidad a los codificadores, decodificadores, multiplexores y demultiplexores. Por ejemplo, los codificadores deben satisfacer requisitos mínimos de tolerancia de reloj. A pesar de éste y otros requisitos, existe un grado considerable de libertad para el diseño y la implementación de los codificadores, decodificadores, multiplexores y demultiplexores.

Intro. 1 Tren de transporte

El tren de transporte se define como un tren adaptado para comunicar o almacenar uno o más programas de datos codificados de acuerdo con la Rec. UIT-T H.262 | ISO/CEI 13818-2 e ISO/CEI 13818-3 así como otros datos en entornos en los cuales pueden producirse errores importantes. Estos errores se pueden manifestar como errores de valor de bit o pérdida de paquetes.

Los trenes de transporte pueden ser de velocidad fija o variable. En cualquiera de los dos casos, los trenes elementales constituyentes pueden ser de velocidad fija o variable. Las constricciones impuestas por la sintaxis y la semántica al tren son idénticas en cada uno de estos casos. La velocidad del tren de transporte es definida por los valores y ubicaciones de los campos de referencia de reloj de programa (PCR, *program clock reference*) que en general son campos PCR separados para cada programa.

La construcción y entrega de un tren de transporte que contenga múltiples programas con bases de tiempo independientes plantea algunas dificultades, entre otras razones, porque la velocidad binaria global es variable. Véase 2.4.2.2.

El tren de transporte se puede construir mediante cualquier método que dé como resultado un tren válido. Es posible construir trenes de transporte que contienen uno o más programas a partir de trenes de datos codificados elementales, de trenes de programa o de otros trenes de transporte que contienen uno o más programas.

El tren de transporte está diseñado de manera que sea posible realizar varias operaciones en el mismo, con mínimo esfuerzo, como sigue:

- 1) Extraer los datos codificados de un programa dentro del tren de transporte, decodificarlos y presentar los resultados decodificados como se muestra en la figura Intro. 2.
- 2) Extraer los paquetes del tren de transporte de un programa dentro del tren y producir como salida un tren de transporte diferente con ese programa solamente, como se muestra en la figura Intro. 3.
- 3) Extraer los paquetes del tren de transporte de uno o más programas de uno o más trenes de transporte y producir como salida un tren de transporte diferente (no ilustrado).
- 4) Extraer el contenido de un programa del tren de transporte y producir como salida un tren de programa que contiene ese programa, como se muestra en la figura Intro. 4.
- 5) Tomar un tren de programa, convertirlo en un tren de transporte para transmitirlo por un entorno con pérdida y después recuperar un tren de programa válido y, en algunos casos, idéntico.

Las figuras Intro. 2 e Intro. 3 ilustran prototipos de sistemas de demultiplexación y decodificación, que toman como entrada un tren de transporte. La figura Intro. 2 ilustra el primer caso, cuando un tren de transporte es demultiplexado y decodificado directamente. Los trenes de transporte se construyen en dos capas:

- una capa de sistemas; y
- una capa de compresión.

El tren de entrada al decodificador de tren de transporte tiene una capa de sistemas que envuelve una capa de compresión. Los trenes de entrada a los decodificadores de vídeo y audio sólo tienen la capa de compresión.

Las operaciones realizadas por el prototipo de decodificador que acepta trenes de transporte se aplican a todo el tren de transporte ("operaciones de múltiplex"), o a trenes elementales ("operaciones específicas de tren"). La capa de sistemas del tren de transporte se divide en dos subcapas, una para operaciones múltiplex (la capa de paquete del tren de transporte) y una para operaciones específicas del tren (la capa de paquete PES).

En la figura Intro. 2 se incluye también un prototipo de decodificador para trenes de transporte, que incluye audio y vídeo, con el fin de ilustrar la función de un decodificador. La arquitectura no es única – algunas funciones de decodificador de sistemas, tales como el control de temporización del decodificador, pudieran estar también distribuidas entre decodificadores de trenes elementales y el decodificador específico de canal – pero esta figura es útil para el análisis. Del mismo modo, la indicación de errores detectados por el decodificador específico de canal a los decodificadores individuales de audio y de vídeo se puede realizar de diversas maneras y estos trayectos de comunicación no se muestran en el diagrama. El diseño del prototipo de decodificador no conlleva ningún requisito normativo para el diseño de un decodificador de tren de transporte. En realidad, se permiten también datos que no sean de audio ni de vídeo, pero no se muestran.

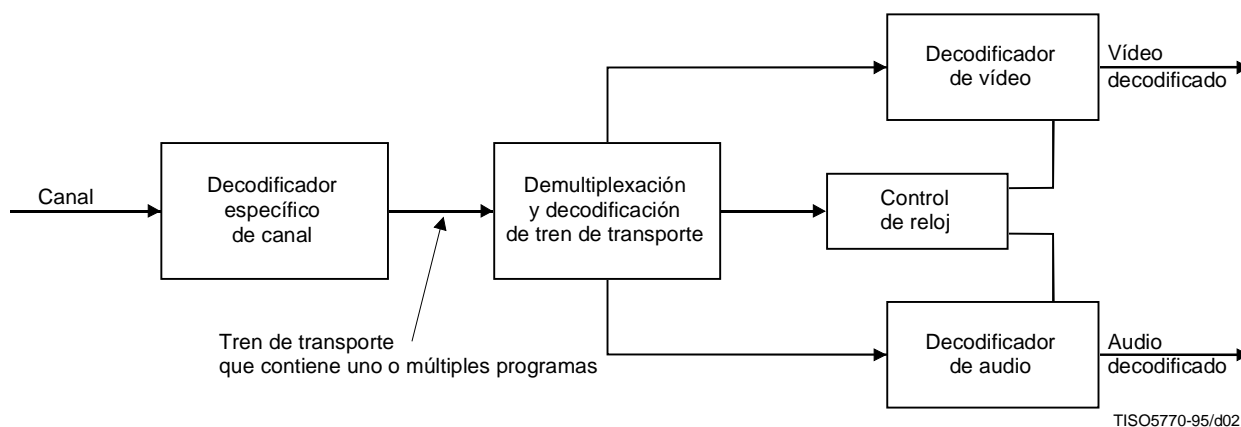


Figura Intro. 2 – Ejemplo de prototipo de demultiplexación y decodificación de tren de transporte

La figura Intro. 3 ilustra el segundo caso, en el que un tren de transporte que contiene múltiples programas se convierte en un tren de transporte que contiene un solo programa. En este caso, la operación de remultiplexación puede necesitar la corrección de valores de la referencia de reloj de programa (PCR) para tener en cuenta los cambios en las ubicaciones de la PCR en el tren de bits.



Figura Intro. 3 – Ejemplo de prototipo de multiplexación de tren de transporte

La figura Intro. 4 ilustra el caso cuando un tren de transporte multiprograma es demultiplexado primero y convertido después en un tren de programa.

La figura Intro. 3 y la figura Intro. 4 indican que es posible y razonable la conversión entre diferentes tipos y configuraciones de trenes de transporte. Hay campos específicos, definidos en las sintaxis de **tren de transporte** y de **tren de programa**, que facilitan las conversiones ilustradas. Las implementaciones específicas de demultiplexores o decodificadores no tienen que incluir todas estas funciones.

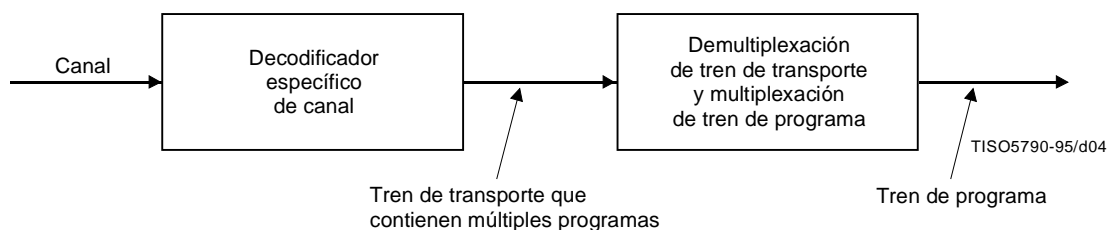


Figura Intro. 4 – Prototipo de conversión de tren de transporte a tren de programa

Intro. 2 Tren de programa

El tren de programa se define como un tren adaptado para comunicación o almacenamiento de un programa de datos codificados y otros datos en entornos en los cuales los errores son muy improbables, y cuando el procesamiento de codificación de sistemas, por ejemplo mediante soporte lógico, es una consideración importante.

Los trenes de programa pueden ser de velocidad fija o variable. En cualquiera de los dos casos, los trenes elementales constituyentes pueden ser de velocidad fija o variable. Las constricciones sintácticas y semánticas impuestas al tren son idénticas en cada caso. La velocidad del tren del programa es definida por los valores y ubicaciones de la referencia de reloj de sistema (SCR, *system clock reference*) y los campos de velocidad múltiplex.

En la figura Intro. 5 se muestra un prototipo de sistema de decodificador de tren de programa de audio/vídeo. La arquitectura no es única -las funciones de decodificador de sistemas que incluyen control de temporización de decodificador pudieran también estar distribuidas entre decodificadores de trenes elementales y el decodificador específico de canal- pero esta figura es útil para el análisis. El diseño del prototipo de decodificador no supone ningún requisito normativo para el diseño de un decodificador de tren de programa. En realidad, se permiten también datos que no son de audio ni de vídeo, pero no se muestran.

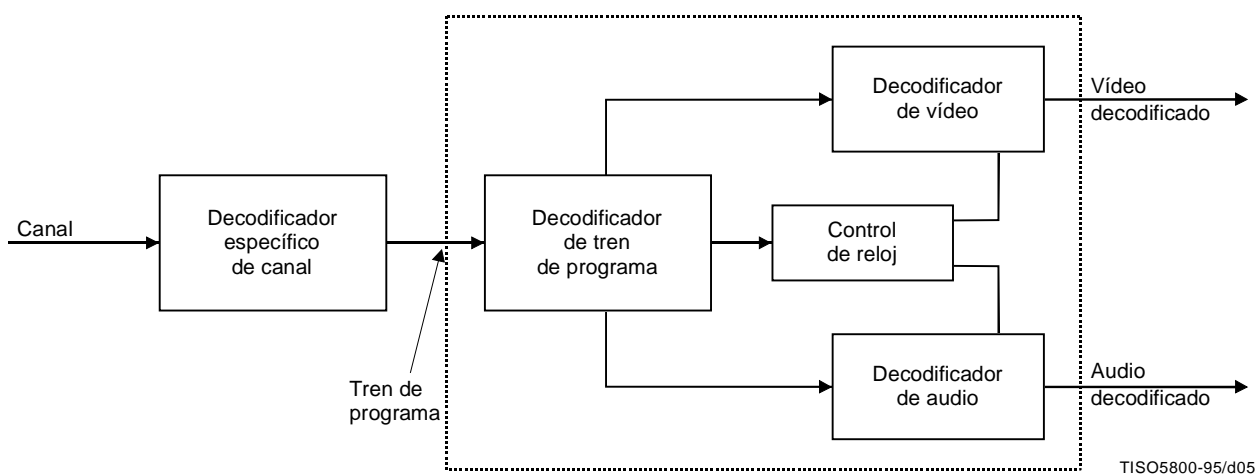


Figura Intro. 5 – Prototipo de decodificador para trenes de programa

El prototipo de decodificador para trenes de programa mostrado en la figura Intro. 5 se compone de decodificadores de sistemas, de vídeo y de audio conformes a las partes 1, 2 y 3, respectivamente, de ISO/CEI 13818. En este decodificador se supone que la representación codificada multiplexada de uno o más trenes de audio y/o de vídeo se almacena o comunica por algún canal en algún formato específico del canal. Este formato específico de canal no está regido por la presente Recomendación | Norma Internacional ni es la parte de decodificación específica de canal del prototipo de decodificador.

El prototipo de decodificador acepta como entrada un tren de programa y utiliza un decodificador de tren de programa para traer información de temporización del tren. El decodificador de tren de programa demultiplexa el tren y los trenes elementales así producidos sirven como entradas a decodificadores de vídeo y de audio, cuyas salidas son señales de vídeo y de audio decodificadas. Aunque no se muestra en la figura, en el diseño se incluye el flujo de información de temporización entre el decodificador de tren de programa, los decodificadores de vídeo y audio y el decodificador específico de canal. Los decodificadores de vídeo y audio están sincronizados entre sí y con el canal que utiliza esta información de temporización.

Los trenes de programa se construyen en dos capas: una capa de sistemas y una capa de compresión. El tren de entrada al decodificador de tren de programa tiene una capa de sistemas que envuelve una capa de compresión. Los trenes de entrada a los decodificadores de vídeo y audio sólo tienen la capa de compresión.

Las operaciones realizadas por el prototipo de decodificador se aplican a todo el tren de programa ("operaciones de múltiplex") o a trenes elementales individuales ("operaciones específicas de tren"). La capa de sistemas del tren de programa se divide en dos subcapas, una para operaciones de múltiplex [la capa de paquete (*pack layer*)] y una para operaciones específicas de tren (la capa de paquete PES).

Intro. 3 Conversión entre tren de transporte y tren de programa

Puede ser posible y razonable la conversión entre **trenes de transporte** y **trenes de programa** por medio de paquetes PES. Esto resulta de la especificación de **tren de transporte** y de **tren de programa** que figura en 2.4.1 y 2.5.1 de los requisitos normativos de la presente Recomendación | Norma Internacional. Los paquetes PES pueden corresponder directamente, con algunas constricciones, de la cabida útil de un tren de bits multiplexado con la cabida útil de otro tren de bits multiplexado. Para facilitar esto, es posible identificar el orden correcto de los paquetes PES en un programa, si el contador de la secuencia de paquetes de programa está presente en todos los paquetes PES.

Alguna otra información necesaria para la conversión, por ejemplo, la relación entre trenes elementales, está disponible en tablas y encabezamientos en ambos trenes. Estos datos, si están disponibles, deberán ser correctos en cualquier tren antes y después de la conversión.

Intro. 4 Tren elemental paquetizado

Cada **tren de transporte** y cada **tren de programa** está construido lógicamente a partir de paquetes PES, como se indica en las definiciones de sintaxis de 2.4.3.6. Los paquetes PES se utilizarán para la conversión entre trenes de transporte y trenes de programa; en algunos casos, los paquetes PES no tienen que ser modificados cuando se realizan estas conversiones. Los paquetes PES pueden ser mucho mayores que el tamaño de un paquete de tren de transporte.

Se puede utilizar una secuencia continua de paquetes PES de un tren elemental con un identificador (ID) de tren para construir un tren PES. Cuando se utilizan paquetes PES para formar un tren PES, incluirán campos de referencia de reloj de tren elemental (ESCR, *elementary stream clock reference*) y campos de velocidad de tren elemental (ES_Rate, *elementary stream rate*), con las constricciones definidas en 2.4.3.8. Los datos del tren PES serán bytes contiguos del tren elemental en su orden original. Los trenes PES no contienen cierta información de sistema necesaria que está contenida en trenes de programa y trenes de transporte. Como ejemplo cabe citar la información en el encabezamiento de saca, encabezamiento de sistemas, correspondencia de tren de programa, directorio de tren de programa, tabla de correspondencia de programas y elementos de la sintaxis de paquete de tren de transporte.

El tren PES es una construcción lógica que puede ser útil en implementaciones de esta Recomendación | Norma Internacional; sin embargo, no se ha definido como un tren para intercambio e interoperabilidad. Las aplicaciones que requieren trenes con un solo tren elemental pueden utilizar trenes de programa o trenes de transporte, cada uno con un tren elemental solamente. Estos trenes contienen toda la información de sistemas necesaria. Se pueden construir múltiples trenes de programa o trenes de transporte, cada uno con un solo tren elemental, con una base de tiempos común y que, por tanto, transportan un programa completo, es decir, con audio y vídeo.

Intro. 5 Modelo de temporización

Sistemas, vídeo y audio tienen un modelo de temporización en el cual el retardo de extremo a extremo de la entrada de la señal a un codificador a la salida de la señal de un decodificador es una constante. Este retardo es la suma de la codificación, almacenamiento en memoria tampón de codificador, multiplexación, comunicación o almacenamiento, demultiplexación, almacenamiento en memoria tampón de decodificador, decodificación y presentación. Como parte de este modelo de temporización, todas las imágenes vídeo y muestras audio son presentadas exactamente una vez, a menos que estén codificadas específicamente para lo contrario, y el intervalo entre imágenes y la velocidad de muestra de audio son iguales en el decodificador y en el codificador. La codificación del tren de sistema contiene información de temporización que se puede utilizar para implementar sistemas con retardo de extremo a extremo constante. Es posible implementar decodificadores que no sigan este modelo exactamente, pero en tales casos, es responsabilidad del decodificador funcionar de una manera aceptable. La temporización está incluida en las especificaciones normativas de esta Recomendación | Norma Internacional, que deben ser cumplidas por todos los trenes de bits válidos, con independencia del medio de crearlos.

Toda la temporización se define desde el punto de vista de un reloj de sistema común, denominado reloj de tiempo de sistema (STC, *system time clock*). En el tren de programa, este reloj tiene una relación especificada exactamente con los relojes de muestras de vídeo o de audio, o puede tener una frecuencia de trabajo que difiera ligeramente de la relación exacta mientras proporcione temporización de extremo a extremo precisa y recuperación por reloj.

En el tren de transporte, la frecuencia de reloj de sistema está constreñida a tener la relación especificada exactamente con los relojes de muestra de audio y vídeo en todos los momentos; el efecto de esta restricción es simplificar la recuperación de la velocidad de muestra en los decodificadores.

Intro. 6 Acceso condicional

La criptación y la aleatorización para acceso condicional a programas codificados en los trenes de programa y de transporte son soportadas por las definiciones de trenes de datos de sistemas. En esta Recomendación | Norma Internacional no se especifican los mecanismos de acceso condicional. Los trenes se definen de modo que en la práctica la implementación de sistemas de acceso condicional sea razonable, y hay algunos elementos sintácticos especificados que proporcionan soporte específico para estos sistemas.

Intro. 7 Operaciones de múltiplex

Las operaciones de múltiplex incluyen la coordinación de extracción de datos del canal, el ajuste de relojes y la gestión de memorias tampón. Las tareas están íntimamente relacionadas. Si la velocidad de extracción de datos del canal es controlable, la entrega de datos puede ajustarse de modo que las memorias tampón del decodificador no estén desbordadas ni subutilizadas, pero si la velocidad de datos no es controlable, los decodificadores de trenes elementales deben subordinar su temporización a los datos recibidos del canal para evitar desbordamiento o subutilización.

Los trenes de programa se componen de pacas (*packs*) cuyos encabezamientos facilitan las tareas mencionadas anteriormente. Los encabezamientos de pacas especifican tiempos previstos en los cuales cada byte ha de pasar del canal al decodificador de tren de programa, y este plan de llegada fijado como objetivo sirve como una referencia para la corrección del reloj y la gestión de la memoria tampón. El plan no tiene que ser seguido exactamente por los decodificadores, pero éstos deben compensar las desviaciones con respecto al mismo.

De manera similar, los trenes de transporte se componen de paquetes de tren de transporte con encabezamientos que contienen información que especifica los tiempos en los cuales está previsto que cada byte pase del canal a un decodificador de tren de transporte. Este plan proporciona exactamente la misma función que está especificada en el tren de programa.

Otra operación de múltiplex es la capacidad de un decodificador de establecer los recursos que se requieren para decodificar un tren de transporte o un tren de programa. La primera paca de cada tren de programa transporta parámetros para asistir a los decodificadores en esta tarea. Se incluye, por ejemplo, la velocidad de datos máxima del tren y el número más alto de canales vídeo simultáneos. Del mismo modo, el tren de transporte contiene información globalmente útil.

Cada tren de transporte y cada tren de programa contiene información que identifica las características pertinentes de los trenes elementales que constituyen cada programa y las relaciones entre éstos. Esta información puede incluir el idioma hablado en los canales de audio, así como la relación entre trenes de vídeo cuando se aplica codificación vídeo multicapa.

Intro. 8 Operaciones de tren (capa de paquete PES)

Las principales operaciones específicas de tren son:

- 1) demultiplexación; y
- 2) sincronización de repetición de múltiples trenes elementales.

Intro. 8.1 Demultiplexación

En la codificación, los trenes de programa se forman multiplexando trenes elementales, y los trenes de transporte se forman multiplexando trenes elementales, trenes de programa o el contenido de otros trenes de transporte. Los trenes elementales pueden incluir trenes privados, reservados y de relleno, además de trenes de audio y vídeo. Los trenes se subdividen temporalmente en paquetes y los paquetes se ponen en serie. Un paquete PES contiene bytes codificados de un tren elemental solamente.

En el tren de programa, los paquetes pueden tener longitud fija y variable, a reserva de las constricciones especificadas en 2.5.1 y 2.5.2. Para los trenes de transporte, la longitud de paquete es 188 bytes. Los paquetes PES pueden tener longitud fija y variable, pero serán relativamente largos en la mayoría de las aplicaciones.

En la decodificación, se requiere la demultiplexación para reconstituir trenes elementales a partir del tren de programa o del tren de transporte multiplexados, lo que es posible gracias a los códigos de ID de tren en los encabezamientos de paquete de tren de programa y a los códigos de ID de paquete en el tren de transporte.

Intro. 8.2 Sincronización

La sincronización entre múltiples trenes elementales se realiza con indicaciones de tiempo de presentación (*PTS, presentation time stamps*) en el tren de programa y en los trenes de transporte. Las indicaciones de tiempo están generalmente en unidades de 90 kHz, pero la referencia de reloj de sistema (*SCR*), la referencia de reloj de programa (*PCR*) y la referencia de reloj de tren elemental (*ESCR*) tienen extensiones con una resolución de 27 MHz. La decodificación de N trenes elementales se sincroniza ajustando la decodificación de trenes a una base de tiempos

principal común en vez de ajustar la decodificación de un tren para que concuerde con la de otro. La base de tiempos principal puede ser uno de los relojes de N decodificadores, el reloj de la fuente de datos o puede ser algún reloj externo.

Cada programa en un tren de transporte, que puede contener múltiples programas, puede tener su propia base de tiempos. Las bases de tiempos de diferentes programas dentro de un tren de transporte pueden ser diferentes.

Como las PTS se aplican a la decodificación de trenes elementales individuales, residen en la capa de paquete PES de los trenes de transporte y de los trenes de programa. La sincronización de extremo a extremo se produce cuando los codificadores ahorran indicaciones de tiempo en tiempo de captura, cuando las indicaciones de tiempo se propagan con datos codificados asociados a los decodificadores y cuando los decodificadores utilizan estas indicaciones de tiempo para planificar presentaciones.

La sincronización de un sistema de decodificación con un canal se logra mediante la utilización de la SCR en el tren de programa y por su análoga, la PCR, en el tren de transporte. La SCR y la PCR son indicaciones de tiempo que codifican la temporización del propio tren de bits y se derivan de la misma base de tiempos utilizada para los valores PTS de audio y vídeo del mismo programa. Como cada programa puede tener su propia base de tiempos, hay campos PCR separados para cada programa en un tren de transporte que contiene múltiples programas. En algunos casos, puede ser posible que los programas compartan campos PCR. Para el método de identificar la PCR que está asociada con un programa, véase 2.4.4, Información específica de programa (PSI, *program specific information*). Un programa sólo tendrá una base de tiempos PCR asociada con él.

Intro. 8.3 Relación con la capa de compresión

La capa de paquete PES es independiente de la capa de compresión en algunos sentidos, pero no en todos. Es independiente en el sentido de que las cabidas útiles de paquetes PES no tienen que comenzar en los códigos de comienzo de capa compresión, como se define en las partes 2 y 3 de ISO/CEI 13818. Por ejemplo, los códigos de comienzo de vídeo pueden aparecer en cualquier lugar dentro de la cabida útil de un paquete PES y los códigos de comienzo pueden ser divididos por un encabezamiento de paquete PES. Sin embargo, las indicaciones de tiempo codificadas en encabezamientos de paquetes PES se aplican a tiempos de presentación de construcciones de la capa de compresión (a saber, unidades de presentación). Además, cuando los datos del tren elemental son conformes a la Rec. UIT-T H.262 | ISO/CEI 13818-2 o a ISO/CEI 13818-3, los bytes de datos de paquetes PES serán alineados, byte por byte, con los bytes de los datos conformes a la presente Recomendación | Norma Internacional.

Intro. 9 Decodificador de referencia de sistemas

La parte 1 de ISO/CEI 13818 emplea un "decodificador-objetivo de sistemas" (STD, *system target decoder*) que se considera como un modelo u objetivo, uno para trenes de transporte (véase 2.4.2) denominado "decodificador-objetivo de sistemas de transporte" (T-STD, *transport system target decoder*) y uno para trenes de programa (véase 2.5.2) denominado "decodificador-objetivo de sistemas de programa" (P-STD, *program system target decoder*), para proporcionar un formalismo para las relaciones de temporización y almacenamiento en memoria tampón. Como el STD está parametrizado desde el punto de vista de los campos de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (por ejemplo, tamaños de memoria tampón), cada tren elemental lleva su propia parametrización del STD. Los codificadores producirán trenes de bits que satisfagan las constricciones del STD apropiadas. Los decodificadores físicos pueden suponer que un tren funciona adecuadamente en su STD; el decodificador físico debe compensar las maneras en las cuales su diseño difiere del diseño del STD.

Intro. 10 Aplicaciones

Los trenes definidos en esta Recomendación | Norma Internacional están destinados a ser lo más útiles posible a una gran variedad de aplicaciones. Los diseñadores de aplicaciones seleccionarán sencillamente el tren más apropiado.

Las modernas redes de comunicaciones de datos son capaces de soportar el vídeo de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y el audio de ISO/CEI 13818. Se requiere un protocolo de transporte en tiempo real. El tren de programa puede ser adecuado para transmisión por estas redes.

El tren de programa es también adecuado para aplicaciones multimedia en CD-ROM. El procesamiento del soporte lógico del tren de programa puede ser apropiado.

El tren de transporte puede ser más adecuado para entornos propensos a errores, como los utilizados para distribuir trenes de bits comprimidos por redes de larga distancia y en sistemas de radiodifusión.

Muchas aplicaciones requieren almacenamiento y extracción de trenes de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 en varios medios de almacenamiento digital (DSM, *digital storage media*). En el anexo B y en la Parte 6 de ISO/CEI 13818 se especifica un protocolo de instrucciones y control de medios de almacenamiento digital (DSM-CC, *digital storage media command and control*) para facilitar el control de estos medios.

**NORMA INTERNACIONAL
RECOMENDACIÓN UIT-T**

**Tecnología de la información – Codificación genérica de imágenes
en movimiento e información de audio asociada: Sistemas**

SECCIÓN 1 – GENERALIDADES

1.1 Alcance

La presente Recomendación | Norma Internacional especifica la capa de sistemas de la codificación. Se ha elaborado principalmente para soportar la combinación de los métodos de codificación de vídeo y audio definidos en las partes 2 y 3 de ISO/CEI 13818. La capa de sistemas soporta seis funciones básicas:

- 1) la sincronización de múltiples trenes comprimidos en la decodificación;
- 2) el entrelazado de múltiples trenes comprimidos en un solo tren;
- 3) la inicialización de almacenamiento en memoria tampón para comenzar la decodificación;
- 4) la gestión continua de la memoria tampón;
- 5) la identificación de tiempo; y
- 6) la multiplexación y señalización de varios componentes en un tren de sistema.

Un tren de bits multiplexado de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 es un **tren de transporte** o un **tren de programa**. Ambos tipos de trenes se construyen con **paquetes PES** y paquetes que contienen otra información necesaria y soportan la multiplexación de trenes comprimidos de vídeo y audio de un programa con una base de tiempos común. El **tren de transporte** soporta además la multiplexación de trenes comprimidos de vídeo y audio de múltiples programas con bases de tiempos independientes. Para entornos casi sin errores, en general el **tren de programa** es más apropiado, porque soporta el procesamiento del soporte lógico de información de programa. El **tren de transporte** es más adecuado para utilización en entornos donde es probable que haya errores.

Un tren de bits multiplexado de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, sea un tren de transporte o un tren de programa, se construye en dos capas: la capa externa es la capa de sistemas y la capa interna es la capa de compresión. La capa de sistemas proporciona las funciones necesarias para utilizar uno o más trenes de datos comprimidos en un sistema. Las partes de esta Especificación relativas a vídeo y audio definen la capa de codificación de compresión para datos de audio y de vídeo. La codificación de otros tipos de datos no se define en esta Especificación, pero es soportada por la capa de sistemas a condición de que los otros tipos de datos cumplan las constricciones definidas en 2.7.

1.2 Referencias normativas

Las siguientes Recomendaciones y Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación | Norma Internacional. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones, por lo que se preconiza que los participantes en acuerdos basados en la presente Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y las Normas enumeradas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

1.2.1 Recomendaciones | Normas Internacionales idénticas

- 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*.

1.2.2 Recomendaciones | Normas Internacionales asociadas, equivalentes en contenido técnico

- Recomendación UIT-T H.264 (2005), *Codificación de vídeo avanzada para los servicios audiovisuales genéricos*.

- ISO/CEI 14496-10:2005, *Information technology – Coding of audio-visual objects – Part 10: Advanced video coding.*
- Recomendación UIT-T T.171 (1996), *Protocolos para servicios audiovisuales interactivos: Representación codificada de objetos multimedia e hipermédios.*
- ISO/CEI 13522-1:1997, *Information technology – Coding of Multimedia and Hypermedia information – Part 1: MHEG object representation – Base notation (ASN.1).*

1.2.3 Referencias adicionales

- ISO 639-2:1998, *Codes for the representation of names of languages – Part 2: Alpha-3 code.*
- ISO 8859-1:1998, *Information technology – 8 bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1.*
- ISO 15706:2002, *Information and documentation – International Standard Audiovisual Number (ISAN).*
- ISO/PRF 15706-2, *Information and documentation – International Standard audiovisual number (ISAN) – Part 2: Version identifier.*
- ISO/CEI 11172-1:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 1: Systems.*
- ISO/CEI 11172-2:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video.*
- ISO/CEI 11172-3:1993, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio.*
- ISO/CEI 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio.*
- ISO/CEI 13818-6:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC.*
- ISO/CEI 13818-7:2006, *Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC).*
- ISO/CEI 13818-11:2004, *Information technology – Generic coding of moving pictures and associated audio information – Part 11: IPMP on MPEG-2 systems.*
- ISO/CEI 14496-1:2004, *Information technology – Coding of audio-visual objects – Part 1: Systems.*
- ISO/CEI 14496-2:2004, *Information technology – Coding of audio-visual objects – Part 2: Visual.*
- ISO/CEI 14496-3:2005, *Information technology – Coding of audio-visual objects – Part 3: Audio.*
- Recomendación UIT-R BT.601-6 (2007), *Parámetros de codificación de televisión digital para estudios con formatos de imagen normal 4:3 y de pantalla ancha 16:9.*
- Recomendación UIT-R BT.470-7 (2005), *Sistemas de televisión analógica convencional.*
- Recomendación UIT-R BR.648, *Grabación digital de señales de audio.*
- Recomendación UIT-T J.17 (1988), *Preacentuación utilizada en los circuitos radiofónicos.*
- IEC Publication 60908:1999, *Audio recording – Compact disc digital audio system.*

SECCIÓN 2 – ELEMENTOS TÉCNICOS

2.1 Definiciones

A los efectos de la presente Recomendación | Norma Internacional, se aplican las siguientes definiciones. Si la definición es específica de una parte de la Especificación esto se indica entre paréntesis.

2.1.1 unidad de acceso (sistema): Una representación codificada de una unidad de presentación. En el caso de audio, una unidad de acceso es la representación codificada de una trama de audio.

En el caso de vídeo, una unidad de acceso incluye todos los datos codificados para una imagen, y cualquier relleno que le siga hasta, pero sin incluir, el comienzo de la siguiente unidad de acceso. Si una imagen no va precedida de un código de comienzo de grupo o de un código de encabezamiento de secuencia, la unidad de acceso empieza con el código de comienzo de imagen. Si una imagen va precedida por un código de comienzo de grupo y/o un código de encabezamiento de secuencia, la unidad de acceso empieza con el primer byte del primero de estos códigos de

comienzo. Si se trata de la última imagen que precede a un código de fin de secuencia en el tren de bits, todos los bytes entre el último byte de la imagen codificada y el código de fin de secuencia (incluido el código de fin de secuencia) pertenecen a la unidad de acceso.

Para consultar la definición de unidad de acceso en el contexto de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10, véase la definición de unidad de acceso AVC en 2.1.3.

2.1.2 imagen de 24 horas AVC (sistema): Unidad de acceso AVC cuya hora de presentación es posterior a las siguientes 24 horas. En esta definición, se entiende que la hora de presentación de la unidad de acceso AVC n es posterior a las siguientes 24 horas si la diferencia entre la hora de llegada inicial $t_{ai}(n)$ y la hora de salida DPB $t_{o,dpb}(n)$ es mayor que 24 horas.

2.1.3 unidad de acceso AVC (sistema): Unidad de acceso definida para trenes de bytes en la Rec. UIT-T H.264 | ISO/CEI 14496-10 con las restricciones especificadas en 2.14.1.

2.1.4 sector AVC (sistema): `byte_stream_nal_unit` (unidad nal de trenes de bytes) definida en la Rec. UIT-T H.264 | ISO/CEI 14496-10 con valores `nal_unit_type` de 1 ó 5, o estructura de datos `byte_stream_nal_unit` con valor `nal_unit_type` de 2 y cualquier estructura de datos `byte_stream_nal_unit` asociada con `nal_unit_type` igual a 3 y/o 4.

2.1.5 imagen fija AVC (sistema): Una imagen fija AVC consta de una unidad de acceso AVC que contiene una imagen IDR, precedida por unidades NAL SPS y PPS que transportan información suficiente para decodificar correctamente la imagen IDR. Antes de la imagen AVC, deberá haber otra imagen fija AVC o una unidad NAL de fin de secuencia que termina la secuencia de vídeo codificado anterior a no ser que la imagen fija AVC sea la primera unidad de acceso del tren de vídeo.

2.1.6 secuencia de vídeo AVC (sistema): Secuencia de vídeo codificado definida en la cláusula 3.30 de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

2.1.7 tren de vídeo AVC (sistema): Tren de la Rec. UIT-T H.264 | ISO/CEI 14496-10. Un tren de vídeo AVC consta de una o varias secuencias de vídeo AVC.

2.1.8 velocidad binaria: Velocidad a la cual el tren de bits comprimido pasa del canal a la entrada de un decodificador.

2.1.9 alineado en byte: Un bit en un tren de bits codificado está alineado en bytes si su posición es un múltiplo de 8 bits del primer bit del tren.

2.1.10 canal: Un medio digital que almacena o transporta un tren de la Rec. UIT-T H.222.0/ISO/CEI 13818-1.

2.2.11 Cuadro-B codificado: Una imagen de cuadro-B o un par de imágenes de campo-B.

2.1.12 Cuadro codificado: Un cuadro-I codificado, un cuadro-B codificado o un cuadro-P codificado.

2.1.13 Cuadro-I codificado: Una imagen de cuadro-I o un par de imágenes de campo siendo la primera imagen de campo una imagen-I y la segunda una imagen-I o una imagen-P.

2.1.14 Cuadro-P codificado: Una imagen de cuadro-P o un par de imágenes de campo- P.

2.1.15 representación codificada: Un elemento de datos representado en su forma codificada.

2.1.16 compresión: Reducción del número de bits utilizados para representar un ítem de datos.

2.1.17 velocidad binaria constante: Funcionamiento en el cual la velocidad binaria es constante desde el principio al fin del tren de bits comprimido.

2.1.18 tren de parámetros de sistema constreñido; CSPS (sistema): (CSPS, *constrained system parameter stream*): Un tren de transporte o un tren de programa para el cual se aplican las constricciones definidas en 2.7.9.

2.1.19 verificación por redundancia cíclica (CRC, cyclic redundancy check): La verificación por redundancia cíclica para verificar la corrección de los datos.

2.1.20 elemento de datos: Un ítem de datos como se representa antes de la codificación y después de la decodificación.

2.1.21 tren decodificado: La reconstrucción decodificada de un tren de bits comprimido.

2.1.22 decodificador: Una materialización de un proceso de decodificación.

2.1.23 decodificación (proceso): El proceso definido en esta Recomendación | Norma Internacional que lee un tren de bits codificado de entrada y produce como salida imágenes o muestras de audio decodificadas.

- 2.1.24 indicación de tiempo de decodificación DTS (sistema): (DTS, *decoding time-stamp*):** Un campo que puede estar presente en un encabezamiento de paquete PES que indica el instante en que una unidad de acceso es decodificada en el decodificador-objetivo del sistema.
- 2.1.25 medio de almacenamiento digital (DSM, *digital storage media*):** Un dispositivo o sistema de almacenamiento o transmisión digital.
- 2.1.26 DSM-CC:** Instrucciones y control de medios de almacenamiento digital (*digital storage media command and control*).
- 2.1.27 mensaje de control de permisos (ECM, *entitlement control messages*):** El mensaje de control de permisos es una información de acceso condicional privado que especifica palabras de control y posiblemente otros parámetros típicamente específicos del tren, de aleatorización y/o de control.
- 2.1.28 mensaje de gestión de permisos (EMM, *entitlement management messages*):** El mensaje de gestión de permisos es información de acceso condicional privada que especifica los niveles de autorización o los servicios de decodificadores específicos. Pueden estar dirigidos a un decodificador o a grupos de decodificadores.
- 2.1.29 edición:** Proceso por el cual uno o más trenes de bits comprimidos son manipulados para producir un nuevo tren de bits comprimidos. Los trenes de bits editados conformes deben satisfacer los mismos requisitos que los trenes que no están editados.
- 2.1.30 tren elemental; ES (sistema) (ES, *elementary stream*):** Un término genérico que designa un tren de vídeo codificado, o un tren de audio codificado, u otros trenes binarios codificados en paquetes PES. Un tren elemental se transporta en una secuencia de paquetes PES con un solo identificador de tren.
- 2.1.31 referencia de reloj de tren elemental (sistema) (ESCR, *elementary stream clock reference*):** Una indicación de tiempo en el tren PES a partir de la cual el decodificador de trenes PES puede derivar temporización.
- 2.1.32 codificador:** Una materialización de un proceso de codificación.
- 2.1.33 codificación (proceso):** Un proceso, no especificado en esta Recomendación | Norma Internacional, que lee un tren de imágenes o muestras de audio de entrada y produce un tren de bits codificado conforme a esta Recomendación.
- 2.1.34 codificación de entropía:** Codificación sin pérdida de la longitud variable de la representación digital de una señal para reducir la redundancia.
- 2.1.35 evento:** Un evento se define como una colección de trenes elementales con una base de tiempos común, un tiempo de comienzo asociado y un tiempo de fin asociado.
- 2.1.36 reproducción rápida hacia adelante (vídeo):** Proceso de visualizar una secuencia, o partes de una secuencia de imágenes en un orden de visualización más rápido que el tiempo real.
- 2.1.37 prohibido:** El término "prohibido", cuando se utiliza en las cláusulas de esta Recomendación | Norma Internacional que definen el tren de bits codificado, indica que el valor especificado no se utilizará nunca.
- 2.1.38 metadatos:** Información que describe el contenido audiovisual y la esencia del contenido en datos, en un formato definido por la ISO o cualquier otra autoridad.
- 2.1.39 unidad de acceso de metadatos:** Estructura global dentro de los metadatos que define la fracción de metadatos que debe ser decodificada en un instante concreto. La estructura interna de la unidad de acceso de metadatos queda definida por el formato de los metadatos.
- 2.1.40 formato de aplicación de metadatos:** Identifica el formato de la aplicación que utiliza los metadatos. Contiene la información específica de la aplicación para el transporte de los metadatos.
- 2.1.41 información de configuración del decodificador de metadatos:** Son los datos que necesita el receptor para decodificar un determinado servicio de metadatos. La información de configuración del decodificador puede o no ser necesaria, dependiendo del formato de los metadatos.
- 2.1.42 formato de los metadatos:** Identifica el formato de codificación de los metadatos.
- 2.1.43 servicio de metadatos:** Conjunto coherente de metadatos con un mismo formato que se entrega a un receptor con un fin específico.
- 2.1.44 identificador de servicio de metadatos:** Identificador de un determinado servicio de metadatos. Se utiliza en algunos métodos de transporte de metadatos.
- 2.1.45 tren de metadatos:** Concatenación o serie de unidades de acceso de metadatos procedentes de uno o más servicios de metadatos.

- 2.1.46 tren (multiplexado) (sistema):** Un tren de bits compuesto de ninguno o varios trenes elementales combinados en una manera conforme a esta Recomendación | Norma Internacional.
- 2.1.47 capa (vídeo y sistemas):** Uno de los niveles en la jerarquía de datos de las especificaciones de vídeo y sistemas definidas en las Partes 1 y 2 de esta Recomendación | Norma Internacional.
- 2.1.48 saca (sistema):** Una saca está constituida por un encabezamiento de saca seguido de ninguno o más paquetes. Es una capa en la sintaxis de codificación de sistema descrita en 2.5.3.3.
- 2.1.49 datos de paquete (sistema):** Bytes de datos contiguos de un tren elemental presente en un paquete.
- 2.1.50 identificador de paquete; PID (sistema) (PID, *packet identifier*):** Un valor entero único utilizado para identificar trenes elementales de un programa en un tren de transporte monoprograma o multiprograma, según se describe en 2.4.3.
- 2.1.51 relleno (audio):** Método para ajustar la longitud media de una trama de audio al tiempo de duración de las muestras MIC correspondientes, añadiendo condicionalmente un intervalo a la trama de audio.
- 2.1.52 cabida útil:** La cabida útil se refiere a los bytes que siguen a los bytes de encabezamiento en un paquete. Por ejemplo, la cabida útil de algunos paquetes de tren de transporte incluye el encabezamiento de paquete PES y sus bytes de datos de paquete PES, o campo de puntero y secciones de información específica de programa, o datos privados, pero una cabida útil de paquete PES consiste solamente en bytes de datos de paquete PES. Los campos de encabezamiento y adaptación de paquete de tren de transporte no son cabida útil.
- 2.1.53 PES (sistema) (PES, *packetized elementary stream*):** Abreviatura de tren elemental paquetizado.
- 2.1.54 paquete PES (sistema):** La estructura de datos utilizada para transportar datos de tren elemental. Consiste en un encabezamiento de paquete PES seguido de varios bytes contiguos de un tren de datos elemental. Es una capa en la sintaxis de codificación de sistema descrita en 2.4.3.6.
- 2.1.55 encabezamiento de paquete PES (sistema):** Los campos iniciales de un paquete PES hasta, pero sin incluir, los campos de bytes de datos de paquete PES, cuando no se trata de un tren de relleno. En el caso de un tren de relleno, el encabezamiento de paquete PES se define de manera similar como los campos iniciales de un paquete PES hasta, pero sin incluir, los campos de bytes de relleno.
- 2.1.56 tren PES (sistema):** Un tren PES consiste en paquetes PES, cuyas cabidas útiles consisten en datos de un tren elemental, y todas las cuales tienen el mismo identificador de tren. Se aplican las constricciones semánticas específicas. Véase Intro. 4.
- 2.1.57 indicación de tiempo de presentación; PTS (sistema) (PTS, *presentation time-stamp*):** Un campo que puede estar presente en un encabezamiento de paquete PES e indica el tiempo en que una unidad de presentación es presentada en el decodificador-objetivo de sistema.
- 2.1.58 unidad de presentación; PU (sistema) (PU, *presentation unit*):** Una unidad de acceso de audio decodificado o una imagen decodificada.
- 2.1.59 programa (sistema):** Un programa es una colección de elementos de programa. Los elementos de programa pueden ser trenes elementales. No es necesario que los elementos de programa tengan una base de tiempo definida; los que la tienen, poseen una base de tiempo común y están destinados a la presentación sincronizada.
- 2.1.60 referencia de reloj de programa; PCR (sistema) (PCR, *program clock reference*):** Una indicación de tiempo en el tren de transporte a partir de la cual el decodificador deriva la temporización.
- 2.1.61 elemento de programa (sistema):** Un término genérico que designa un tren elemental u otro tren de datos que puede estar incluido en un programa.
- 2.1.62 información específica de programa; PSI (sistema) (PSI, *program specific information*):** PSI consiste en datos normativos necesarios para la demultiplexación de trenes de transporte y la regeneración satisfactoria de programas, y se describe en 2.4.4. Un ejemplo de PSI, la tabla de información de red no obligatoria, se define privadamente.
- 2.1.63 acceso aleatorio:** Proceso de comenzar a leer y decodificar el tren de bits codificado en un punto arbitrario.
- 2.1.64 reservado:** El término "reservado", cuando se utiliza en las cláusulas que definen el tren de bits codificado, indica que el valor puede ser utilizado en futuras extensiones definidas por la ISO. A menos que se especifique otra cosa en esta Recomendación | Norma Internacional, todos los bits reservados se pondrán a '1'.
- 2.1.65 aleatorización (sistema):** Alteración de las características de un tren de vídeo, de audio, o de datos codificados, para evitar recepción no autorizada de la información en forma clara. Esta alteración se obtiene mediante un proceso especificado bajo el control de un sistema de acceso condicional.

ISO/CEI 13818-1:2007 (S)

2.1.66 tren fuente: Un tren no multiplexado de muestras antes de una codificación de compresión.

2.1.67 empalme (sistema): La concatenación, realizada en el nivel de sistema, de dos trenes elementales diferentes. El tren de sistema resultante es totalmente conforme a esta Recomendación | Norma Internacional. El empalme puede resultar en discontinuidades en la base de tiempo, el contador de continuidad, la PSI y la decodificación.

2.1.68 códigos de comienzo (sistema): Códigos de 32 bits insertados en el tren de bits codificado. Se utilizan para varios fines que incluyen la identificación de algunas de las capas en la sintaxis de codificación. Los códigos de comienzo consisten en un prefijo de 24 bits (0x000001) y un identificador de tren de 8 bits, como se muestra en el cuadro 2-22.

2.1.69 memoria tampón de entrada STD (sistema): Una memoria tampón en la que lo primero que llega es lo primero que sale, a la entrada de un decodificador-objetivo de sistema para almacenamiento de datos comprimidos de trenes elementales antes de la decodificación.

2.1.70 imagen fija: Una imagen fija consta de una secuencia de vídeo codificada según la Rec. UIT-T H.262 | ISO/CEI 13818-2, ISO/CEI 11172-2 o ISO/CEI 14496-2, que contiene exactamente una imagen codificada que está intracodificada. Esta imagen tiene una PTS asociada y en caso de codificación de acuerdo con la ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 14496-2, la hora de presentación de las imágenes subsiguientes, si las hubiere, es posterior a la de la imagen fija con una diferencia de al menos dos periodos de imagen.

2.1.71 encabezamiento de sistema (sistema): El encabezamiento de sistema es una estructura de datos definida en 2.5.3.5 que transporta información que resume las características de sistemas del tren de programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.72 referencia de reloj de sistemas; SCR (sistema) (SCR, system clock reference): Una indicación de tiempo en el tren de programa de la cual se deriva la temporización del decodificador.

2.1.73 decodificador de sistemas objetivo (sistema); STD (sistema): (STD, system target decoder): Un modelo ficticio de referencia de un proceso de decodificación utilizado para definir la semántica de un tren de bits multiplexado de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

2.1.74 indicación de tiempo (sistema): Término que indica el tiempo de una acción específica, tal como la llegada de un byte o la presentación de una unidad de presentación.

2.1.75 encabezamiento de paquete de tren de transporte (sistema): Los campos iniciales de un paquete de tren de transporte hasta el campo de contador de continuidad inclusive.

2.1.76 velocidad binaria variable: Funcionamiento en el cual la llegada de los bytes transportados en paquetes del tren de transporte o en pacas del tren de programa, a la entrada de un decodificador, varía en función del tiempo.

2.2 Símbolos y abreviaturas

Los operadores matemáticos utilizados para describir esta Recomendación | Norma Internacional son similares a los empleados en el lenguaje de programación C. Sin embargo, las divisiones de enteros con truncamiento y redondeo se definen específicamente. Los operadores para bits se definen suponiendo representación de enteros con complemento de dos. Los bucles de numeración y cómputo comienzan generalmente en cero.

2.2.1 Operadores aritméticos

+	Adición
–	Sustracción (como un operador binario) o negación (como un operador unario)
++	Incremento
--	Decremento
* o ×	Multiplicación
^	Potencia
/	División con resultado entero con truncamiento del resultado hacia cero. Por ejemplo, 7/4 y –7/–4 son truncados a 1 y –7/4 y 7/–4 son truncados a –1.
//	División con resultado entero con redondeo al entero más próximo. Los valores de semienteros se redondean a partir de 0 a menos que se especifique otra cosa. Por ejemplo, 3//2 se redondea a 2 y –3//2 se redondea a –2.
DIV	División con resultado entero con truncamiento de resultado hacia –∞
%	Operador de módulo. Se define solamente para números positivos.

Signo()	Signo(x)	=	1	$x > 0$
			0	$x == 0$
			-1	$x < 0$

NINT() Operador entero más cercano. Devuelve el valor entero más cercano al argumento con valor real. Los valores de semienteros se redondean a partir de 0.

sin Seno

cos Coseno

exp Exponente

$\sqrt{\quad}$ Raíz cuadrada

\log_{10} Logaritmo con base diez

\log_e Logaritmo con base e

2.2.2 Operadores lógicos

|| OR lógico

&& AND lógico

! NOT lógico

2.2.3 Operaciones relacionales

> Mayor que

\geq Mayor o igual que

< Menor que

\leq Menor o igual que

== Igual a

!= No igual a

máx [...,] El valor máximo en la lista de argumentos

mín [...,] El valor mínimo en la lista de argumentos

2.2.4 Operadores para bits

& AND

| OR

>> Desplazamiento a la derecha con extensión de signo

<< Desplazamiento a la izquierda con relleno de 0

2.2.5 Asignación

= Operador de asignación

2.2.6 Mnemónicos

Se definen los mnemónicos para describir los diferentes tipos de datos utilizados en el tren de bits codificado.

bslbf	Cadena de bits, bit izquierdo primero, donde "izquierdo" es el orden en el cual las cadenas de bits se escriben en la Recomendación Norma Internacional. Las cadenas de bits se escriben como una cadena de unos y ceros entre comillas simples, por ejemplo '1000 0001'. Los blancos dentro de una cadena de bits son para facilitar la lectura y no tienen significado.
ch	Canal.
gr	Gránulo de 3 * 32 muestras de subbanda en la capa II de audio, 18 * 32 muestras de subbanda en la capa III de audio.
main_data	La porción main_data (datos principales) del tren de bits contienen los factores de escala, datos con codificación Huffman e información auxiliar.
main_data_beg	(comienzo de datos principales) Indica la ubicación en el tren de bits del comienzo de main_data para la trama. La ubicación es igual a la ubicación final de main_data de la trama anterior más un bit. Se calcula a partir del valor main_data_end (fin de datos principales) de la trama anterior.

ISO/CEI 13818-1:2007 (S)

part2_length	Este valor contiene el número de bits de main_data utilizados para factores de escala.
rpchof	Coefficientes del polinomio restante, orden más alto primero.
sb	Subbanda.
sfsi	Información de selector de factor de escala.
switch_point_l	Número de banda de factor de escala (banda de factor de escala de bloque largo) a partir de cuyo punto se utiliza la conmutación de ventana.
switch_point_s	Número de banda de factor de escala (banda de factor de escala de bloque corto) a partir de cuyo punto se utiliza la conmutación de ventana.
tcimsbf	Entero de complemento de dos, bit menos significativo (signo) primero.
uimsbf	Entero sin signo, bit más significativo primero.
vclcbf	Código de longitud variable, bit izquierdo primero, donde "izquierdo" se refiere al orden en el cual se escriben los códigos de longitud variable.
window	Número de intervalo de tiempo real en el caso de tipo de block_type = = 2, $0 \leq \text{window} \leq 2$.

El orden de byte de palabras multibytes es el byte más significativo primero.

2.2.7 Constantes

π	3,14159265359
e	2,71828182845

2.3 Método para describir la sintaxis del tren de bits

Los trenes de bits extraídos por el decodificador se describen en 2.4.1 y 2.5.1. Cada ítem de datos del tren de bits se presenta en el tipo de imprenta negritas. Se describe por su nombre, su longitud en bits y un mnemónico para su tipo y orden de transmisión.

La acción originada por un elemento de datos decodificados en un tren de bits depende del valor de ese elemento de datos y de los elementos de datos decodificados previamente. La decodificación de los elementos de datos y la definición de las variables de estado utilizadas en su decodificación se describen en las cláusulas que contienen la descripción semántica de la sintaxis. Se utilizan las siguientes construcciones para expresar las condiciones cuando están presentes elementos de datos, y están en tipo normal.

Obsérvese que esta sintaxis utiliza el convenio "C"-code de que una variable o expresión que evalúa a un valor no cero es equivalente a una condición que es verdadera:

<pre>while (condition) { data_element ... }</pre>	Si la condición es verdadera, el grupo de elementos de datos aparece a continuación en el tren de datos. Esto se repite hasta que la condición no es verdadera.
<pre>do { data_element ... } while (condition)</pre>	El elemento de datos siempre aparece por lo menos una vez. El elemento de datos se repite hasta que la condición no es verdadera.
<pre>if (condition) { data_element ... }</pre>	Si la condición es verdadera, el primer grupo de elementos de datos aparece a continuación en el tren de datos.
<pre>else { data_element ... }</pre>	Si la condición no es verdadera, el segundo grupo de elementos de datos aparece a continuación en el tren de datos.
<pre>for (i = 0; i < n; i++) { data_element ... }</pre>	El grupo de elementos de datos aparece n veces. Las construcciones condicionales dentro del grupo de elementos de datos pueden depender del valor de la variable de control de bucle i, que se pone a cero para la primera aparición, se incrementa a uno para la segunda aparición, y así sucesivamente.

Como se observa, el grupo de elementos de datos puede contener construcciones adicionales anidadas. Para compactación, se omiten los { } cuando solamente sigue un elemento de datos:

data_element []	data_element [] (elemento de datos) es una formación de datos. El número de elementos de datos es indicado por el contexto.
data_element [n]	data_element [n] es el n+1 elemento de una formación de datos.
data_element [m][n]	data_element [m] [n] es el m+1, n+1 elemento de una formación de datos bidimensional.
data_element [1][m][n]	data_element [1][m][n] es el 1+1, m+1, n+1 elemento de una formación de datos tridimensional.
data_element [m..n]	es la gama inclusiva de bits entre bit m y bit n en el elemento de datos.

Si bien la sintaxis se expresa en términos de procedimiento, no se debe suponer que en la figura 2-1 o la figura 2-2 se implemente un procedimiento de decodificación satisfactorio. En particular, estas figuras definen un tren de bits de entrada correcto y sin errores. Los decodificadores reales deben incluir un medio para buscar códigos de comienzo y bytes de sincronización (tren de transporte) para empezar a decodificar correctamente, e identificar errores, borraduras o inserciones durante la decodificación. Los métodos para identificar estas situaciones y las acciones que se han de ejecutar no están normalizados.

2.4 Requisitos del tren de bits del tren de transporte

2.4.1 Estructura y parámetros de codificación del tren de transporte

La capa de codificación del tren de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 permite combinar uno o más programas en un solo tren. Los datos de cada tren elemental se multiplexan junto con información que permite la presentación sincronizada de los trenes elementales dentro de un programa.

Un tren de transporte está constituido por uno o más programas. Los trenes elementales de audio y de vídeo están constituidos por unidades de acceso.

Los datos de tren elemental se transportan en paquetes PES. Un paquete PES consiste en un encabezamiento de paquete PES seguido por datos de paquete. Los paquetes PES están insertados en paquetes del tren de transporte. El primer byte de cada encabezamiento de paquete PES está colocado en la primera cabida útil disponible de un paquete de tren de transporte.

El encabezamiento de paquete PES empieza con un código de comienzo de 32 bits que identifica también al tren al cual pertenecen los datos de paquete. El encabezamiento de paquete PES puede contener indicaciones de tiempo de decodificación y de presentación (DTS y PTS). El encabezamiento de paquete PES contiene también otros campos facultativos. El campo de datos de paquete PES contiene un número variable de bytes contiguos de un tren elemental.

Los paquetes de tren de transporte empiezan con un prefijo de 4 bytes, que contiene un identificador de paquete (PID) de 13 bits, definido en el cuadro 2-2. El PID identifica, mediante tablas de información específicas de programa (PSI), el contenido de los datos en el paquete de tren de transporte. Los paquetes de tren de transporte de valor PID uno transportan datos de un tren elemental solamente.

Las tablas PSI se transmiten en el tren de transporte. Hay seis tablas PSI:

- tabla de asociación de programas;
- tabla de correspondencia de programas;
- tabla de acceso condicional;
- tabla de información de red;
- tabla de descripción de tren de transporte;
- tabla de información de control IPMP.

Estas tablas contienen la información necesaria y suficiente para demultiplexar y presentar programas. La tabla de correspondencia de programas del cuadro 2-33 especifica, entre otra información, los PID y, por tanto, los trenes elementales que están asociados para formar cada programa. Esta tabla indica también el PID de los paquetes del tren de transporte que llevan la PCR para cada programa. La tabla de acceso condicional estará presente si se emplea aleatorización. La tabla de información de red es facultativa y su contenido no se especifica en esta Recomendación | Norma Internacional. La tabla de información de control IPMP estará presente si cualquiera de los componentes del tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 utiliza la IPMP descrita en ISO/CEI 13818-11.

Los paquetes del tren de transporte pueden ser paquetes nulos. Los paquetes nulos están destinados a rellenar trenes de transporte. Se pueden insertar o suprimir mediante procesos de remultiplexación y, por consiguiente, no se puede suponer la entrega de la cabida útil de paquetes nulos al decodificador.

Esta Recomendación | Norma Internacional no especifica los datos codificados que se pueden utilizar como parte de sistemas de acceso condicional. Sin embargo, esta Especificación sí proporciona mecanismos para que los proveedores de servicios de programas transporten e identifiquen estos datos para el procesamiento del decodificador, y para referenciar correctamente datos que están indicados por esta Especificación. Este tipo de soporte se proporciona a través de estructuras de paquetes del tren de transporte y en la tabla de acceso condicional (véase el cuadro 2-32 de la PSI).

2.4.2 Decodificador-objetivo de sistemas del tren de transporte

La semántica del tren de transporte especificada en 2.4.3 y las constricciones de dicha semántica especificada en 2.7 requieren definiciones exactas de la llegada de bytes y eventos de decodificación y los instantes en que éstos se producen. Las definiciones necesarias se establecen en esta Recomendación | Norma Internacional utilizando un decodificador ficticio denominado decodificador-objetivo de sistema de tren de transporte (T-STD). El anexo D (informativo) contiene más explicaciones sobre el T-STD.

El T-STD es un modelo teórico utilizado para definir estos términos con precisión y para modelar el proceso de decodificación durante la construcción o verificación de trenes de transporte. El T-STD se define solamente con esta finalidad. Los decodificadores T-STD pueden ser de tres tipos: de vídeo, de audio y de sistemas. En la figura 2-1 se presenta un ejemplo. Ni la arquitectura del T-STD ni la temporización descritas excluyen la reproducción sincronizada e ininterrumpida de trenes de transportes procedentes de una variedad de decodificadores con diferentes arquitecturas o planes de temporización.

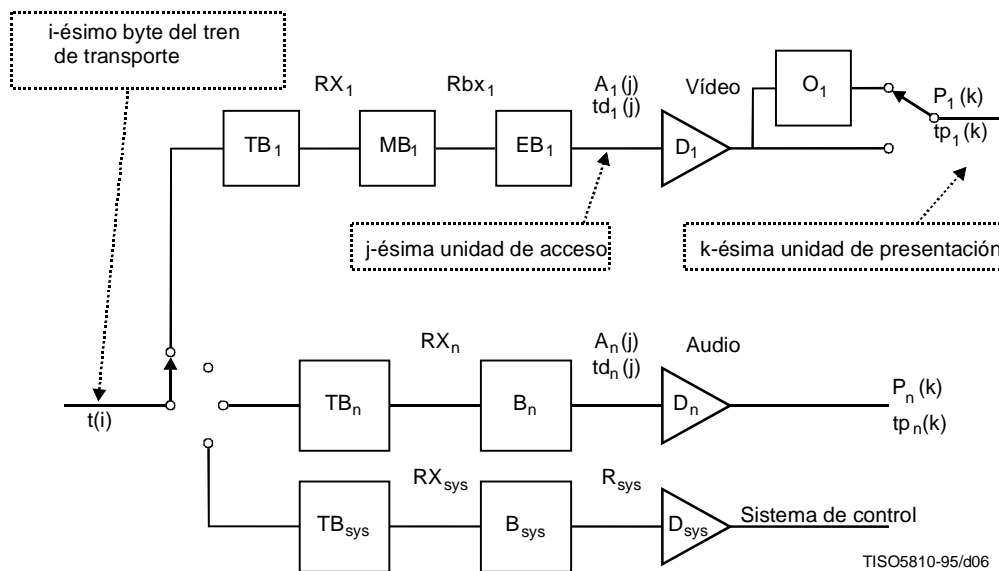


Figura 2-1 – Notación del decodificador-objetivo de sistemas del tren de transporte

La siguiente notación se utiliza para describir el decodificador-objetivo de sistemas del tren de transporte y se ilustra parcialmente en la anterior figura 2-1.

- i, i', i'' son índices de bytes en el tren de transporte. El primer byte tiene el índice 0.
- j es un índice de unidades de acceso en trenes elementales.
- k, k', k'' son índices de unidades de presentación en trenes elementales.
- n es un índice de trenes elementales.
- p es un índice de paquetes de tren de transporte en el tren de transporte.
- t(i) indica el tiempo en segundos en el cual el i-ésimo byte del tren de transporte entra en el decodificador-objetivo de sistema. El valor t(0) es una constante arbitraria.
- PCR(i) es el tiempo codificado en el campo PCR medido en unidades del periodo del reloj de sistema de 27 MHz, donde i es el índice de byte del byte final del campo de base de referencia de reloj de programa.

$A_n(j)$	es la j -ésima unidad de acceso en el tren elemental n . $A_n(j)$ está indexada según el orden de decodificación.
$td_n(j)$	es el tiempo de decodificación, medido en segundos, en el decodificador-objetivo de sistema de la j -ésima unidad de acceso en el tren elemental n .
$P_n(k)$	es la k -ésima unidad de presentación en el tren elemental n . $P_n(k)$ se obtiene de la decodificación de $A_n(j)$. $P_n(k)$ está indexada según el orden de presentación.
$tp_n(k)$	es el tiempo de presentación, medido en segundos, en el decodificador-objetivo de sistema de la k -ésima unidad de presentación en tren elemental n .
t	es el tiempo medido en segundos.
$F_n(t)$	es la ocupación total, medida en bytes, de la memoria tampón de entrada del decodificador-objetivo de sistema para el tren elemental n en el instante t .
B_n	es la memoria tampón principal para el tren elemental n . Sólo está presente en el caso de trenes elementales de audio.
BS_n	es el tamaño de la memoria tampón, B_n , medido en bytes.
B_{sys}	es la memoria tampón principal en el decodificador-objetivo de sistema para la información de sistema del programa que se está decodificando.
BS_{sys}	es el tamaño de B_{sys} , medido en bytes.
MB_n	es la memoria tampón de multiplexación para el tren elemental n . Sólo está presente en el caso de trenes elementales de vídeo.
MBS_n	es el tamaño de MB_n , medido en bytes.
EB_n	es la memoria tampón de tren elemental para el tren elemental n . Sólo está presente en el caso de trenes elementales de vídeo.
EBS_n	es el tamaño de la memoria tampón de tren elemental EB_n , medido en bytes.
TB_{sys}	es la memoria tampón de transporte de información de sistema para el programa que se está decodificando.
TBS_{sys}	es el tamaño de TB_{sys} , medido en bytes.
TB_n	es la memoria tampón de transporte para el tren elemental n .
TBS_n	es el tamaño de TB_n , medido en bytes.
D_{sys}	es el decodificador para información de sistema en el tren de programa n .
D_n	es el decodificador para el tren elemental n .
O_n	es la memoria tampón de reordenación para el tren elemental de vídeo n .
R_{sys}	es la velocidad a la que se suprimen datos de B_{sys} .
Rx_n	es la velocidad a la que se suprimen datos de TB_n .
Rbx_n	es la velocidad a la que los datos de cabida útil de paquete PES se suprimen de MB_n , cuando se utiliza el método de la fuga. Sólo está definida para trenes elementales de vídeo.
$Rbx_n(j)$	es la velocidad a la que se suprimen datos de cabida útil de paquetes PES de MB_n cuando se utiliza el método del retardo VBV. Sólo está definida para trenes elementales de vídeo.
Rx_{sys}	es la velocidad a la que se suprimen datos de TB_{sys} .
R_{es}	es la velocidad de tren elemental de vídeo codificada en un encabezamiento de secuencia.

2.4.2.1 Frecuencia de reloj de sistema

La información de temporización referenciada en el T-STD es transportada por varios campos de datos definidos en la presente Especificación. Véanse 2.4.3.4 y 2.4.3.6. En los campos PCR esta información se codifica como el valor muestreado de un reloj de sistema de programa. Los campos PCR se llevan en el campo de adaptación de los paquetes de tren de transporte con un valor PID igual al PCR_PID definido en la sección de correspondencia de programas TS del programa que se decodifica.

En la práctica, los decodificadores pueden reconstruir este reloj a partir de estos valores y sus respectivos tiempos de llegada. A continuación se indican las constricciones mínimas que se aplican a la frecuencia de reloj de sistema de programa, representada por los valores de los campos PCR cuando son recibidos por un decodificador.

El valor de la frecuencia de reloj de sistema se mide en Hz y satisfará las siguientes constricciones:

$$27\,000\,000 - 810 \leq \text{frecuencia de reloj de sistema} \leq 27\,000\,000 + 810$$

$$\text{velocidad de cambio de la frecuencia de reloj de sistema en función del tiempo} \leq 75 \times 10^{-3} \text{ Hz/s}$$

NOTA – Las fuentes de datos codificados deben trabajar con una tolerancia más estricta para facilitar el funcionamiento con los equipos de grabación y reproducción de los consumidores.

Una frecuencia de reloj de sistema de programa puede ser más exacta que lo necesario. Esta mayor exactitud se puede transmitir al decodificador mediante el descriptor de reloj de sistema, descrito en 2.6.20.

Las velocidades binarias definidas en esta Especificación se miden en términos de frecuencia de reloj de sistema. Por ejemplo, una velocidad binaria de 27 000 000 bits por segundo en el T-STD indicaría que se transfiere un byte de datos cada ocho (8) ciclos del reloj de sistema.

La notación "system_clock_frequency" (frecuencia de reloj de sistema) se utiliza en varias partes en esta Especificación para denominar la frecuencia de un reloj que satisface estos requisitos. Para facilitar la notación, las ecuaciones en las cuales aparecen PCR, PTS o DTS conducen a valores de tiempo que son exactos para algún múltiplo integral de $(300 \times 2^{33} / \text{system_clock_frequency})$ segundos. Esto se debe a la codificación de información de temporización PCR como 33 bits de 1/300 de la frecuencia de reloj de sistema más 9 bits para el residuo, y la codificación como 33 bits de la frecuencia de reloj de sistema dividida por 300 para PTS y DTS.

2.4.2.2 Entrada al decodificador-objetivo de sistemas del tren de transporte

La entrada al decodificador-objetivo de sistemas del tren de transporte (T-STD) es un tren de transporte. Un tren de transporte puede contener múltiples programas con bases de tiempos independientes. Sin embargo, el T-STD decodifica solamente un programa a la vez. En el modelo T-STD, todas las indicaciones de temporización se refieren a la base de tiempos de ese programa.

Los datos procedentes del tren de transporte entran en el T-STD a una velocidad constante por piezas. El instante $t(i)$ en el que el i -ésimo byte entra en el T-STD se define decodificando los campos de referencia de reloj de programa (PCR) del tren de entrada, codificados en el campo de adaptación de paquetes del tren de transporte del programa que se ha de decodificar y contando los bytes en el tren de transporte completo entre PCR sucesivas de dicho programa. El campo PCR se codifica (véase la ecuación 2-1) en dos partes: una, en unidades del periodo del producto de 1/300 por la frecuencia de reloj de sistema, denominada base de referencia de reloj de programa (véase la ecuación 2-2), y otra parte, denominada extensión de referencia de reloj de programa (véase la ecuación 2-3), en unidades del periodo de la frecuencia de reloj de sistema. Los valores codificados en ellas vienen dados por $\text{PCR_base}(i)$, (véase la ecuación 2-2) y $\text{PCR_ext}(i)$, (véase la ecuación 2-3), respectivamente. El valor codificado en el campo PCR indica el instante $t(i)$, donde i es el byte que contiene el último bit del campo de base de referencia de reloj de programa.

Específicamente:

$$\text{PCR}(i) = \text{PCR_base}(i) \times 300 + \text{PCR_ext}(i) \quad (2-1)$$

donde:

$$\text{PCR_base}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 300) \% 2^{33} \quad (2-2)$$

$$\text{PCR_ext}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 1) \% 300 \quad (2-3)$$

Para todos los bytes, el tiempo de llegada de entrada $t(i)$ mostrado en la siguiente ecuación 2-4, se calcula a partir de $\text{PCR}(i'')$ y la velocidad de transporte a la cual llegan los datos, donde la velocidad de transporte se determina como el número de bytes en el tren de transporte entre los bytes que contienen el último bit de dos campos de base de referencia de reloj de programa sucesivos del mismo programa dividido por la diferencia entre los valores de tiempo codificados en estos mismos dos campos PCR.

$$t(i) = \frac{\text{PCR}(i'')}{\text{system_clock_frequency}} + \frac{i - i''}{\text{transport_rate}(i)} \quad (2-4)$$

donde:

- i es el índice de un byte del tren de transporte para $i'' < i < i'$;
- i'' es el índice del byte que contiene el último bit del campo de base de referencia de reloj de programa más reciente aplicable al programa que se decodifica;
- $PCR(i')$ es el tiempo codificado en los campos de base y extensión de referencia de reloj de programa, en unidades de reloj de sistema.

La velocidad de transporte viene dada por:

$$transport_rate(i) = \frac{((i - i'') \times system_clock_frequency)}{PCR(i') - PCR(i'')} \quad (2-5)$$

donde:

- i' es el índice del byte que contiene el último bit de la base de referencia de reloj de programa que sigue inmediatamente, aplicable al programa que se decodifica.

NOTA – $i'' < i \leq i'$

En el caso de una discontinuidad de la base de tiempo, mostrada por el indicador de discontinuidad en el campo de adaptación del paquete de transporte, la definición dada por la ecuación 2-4 y la ecuación 2-5 para el tiempo de llegada de los bytes a la entrada del T-STD no es aplicable entre la última PCR de la base de tiempo antigua y la primera PCR de la base de tiempo nueva. En este caso, el tiempo de llegada de estos bytes se determina por la ecuación 2-4, con la modificación de que la velocidad de transporte utilizada es la aplicable entre la última y la penúltima PCR de la base de tiempo antigua.

Se especifica una tolerancia para los valores PCR. Esta tolerancia de PCR se define como el valor máximo permitido de la inexactitud en las PCR recibidas. Esta inexactitud de PCR puede deberse a la imprecisión en los valores PCR o a la modificación de los valores PCR durante la remultiplexación. No incluye los errores en el tiempo de llegada del paquete, debidos a la fluctuación de fase en la red o a otras causas. La tolerancia de PCR es ± 500 ns.

En el modelo de T-STD, la inexactitud se reflejará como una inexactitud en la velocidad de transporte calculada con la ecuación 2-5.

Trenes de transporte con múltiples programas y velocidad variable

Los trenes de transporte pueden contener múltiples programas con bases de tiempo independientes. Se requieren distintos conjuntos de PCR, como se indica por los respectivos valores PCR_PID, uno para cada uno de estos programas independientes; por esta razón, las PCR no pueden estar coubicadas. La velocidad del tren de transporte es constante por piezas para el programa que entra en el T-STD. En consecuencia, si la velocidad del tren de transporte es variable, sólo puede variar en las PCR del programa considerado. Puesto que las PCR, y por lo tanto los puntos del tren de transporte en que la velocidad varía, no están coubicadas, la velocidad a la cual el tren de transporte entra en el T-TSD tendría que variar en función del programa que está entrando en el T-STD. Por consiguiente, no es posible establecer un plan coherente de entrega al T-STD para un tren de transporte completo cuando dicho tren de transporte contiene múltiples programas con bases de tiempo independientes y la velocidad del tren de transporte es variable. No obstante, es fácil establecer trenes de transporte de velocidad binaria constante que lleven múltiples programas de velocidad variable.

2.4.2.3 Almacenamiento en memoria tampón

Los paquetes de tren de transporte completos que contienen información de sistemas, para el programa seleccionado para decodificación, entran en la memoria tampón de transporte de sistemas, TB_{sys} , a la velocidad del tren de transporte. Comprenden paquetes de tren de transporte cuyos valores de PID son 0, 1, 2 ó 3, y todos los paquetes del tren de transporte identificados mediante la tabla de asociación de programas (véase el cuadro 2-30) que tienen el valor $program_map_PID$ (valor de PID de correspondencia de programas) para el programa seleccionado. Los datos de la tabla de información de red (NIT, *network information table*) especificados por el PID de NIT no se transfieren a TB_{sys} .

NOTA 1 – Si la tabla de información de control IPMP es grande, la velocidad de repetición de esta tabla se debería adaptar según los requisitos de la memoria tampón.

Todos los bytes que entran en la memoria tampón TB_n son suprimidos a la velocidad Rx_n especificada a continuación. Los bytes que forman parte del paquete PES o su contenido son entregados a la memoria tampón principal B_n para flujos elementales de audio y datos de sistema y a la memoria tampón de multiplexación MB_n para flujos elementales de vídeo. Otros bytes no lo son, y pueden ser utilizados para controlar el sistema. Los paquetes de flujo de transporte duplicados no son entregados a B_n ni a MB_n ni a B_{sys} .

ISO/CEI 13818-1:2007 (S)

La memoria tampón TB_n se vacía como sigue:

- cuando no hay datos en TB_n , R_x será cero;
- en los demás casos, para vídeo:

$$R_{x_n} = 1, 2 \times R_{\text{máx}} [\text{perfil}, \text{nivel}]$$

donde:

$R_{\text{máx}}[\text{perfil}, \text{nivel}]$ se especifica de acuerdo con el perfil y nivel que figuran en el cuadro 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2. Dicho cuadro 8-13 especifica el límite superior de la velocidad de cada tren elemental de vídeo dentro de un perfil y nivel específicos.

R_{x_n} es igual a $1, 2 \times R_{\text{máx}}$ para los trenes vídeo con parámetros constreñidos de ISO/CEI 11172-2, donde $R_{\text{máx}}$ indica la velocidad binaria máxima para un tren de bits de parámetros constreñidos de ISO/CEI 11172-2.

Para audio de ISO/CEI 13818-7 con ADTS:

Número de canales	R_{x_n} [bit/s]
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Canales: El número de canales de salida de audio de anchura de banda completa más el número de elementos de canal con acoplamiento conmutado independientes dentro del mismo tren de audio elemental. Por ejemplo, en el caso típico en que no hay elementos de canal con acoplamiento conmutado independientes, mono es 1 canal, estéreo es 2 canales y canal ambifónico 5.1 es 5 canales (no se cuenta el canal LFE).

Para otro audio:

$$R_{x_n} = 2 \times 10^6 \text{ bits por segundo}$$

Para datos de sistemas:

$$R_{x_n} = 1 \times 10^6 \text{ bits por segundo}$$

R_{x_n} se mide con respecto a la frecuencia de reloj de sistema.

Los paquetes de tren de transporte completos que contienen información de sistemas, para el programa seleccionado para decodificación, entran en la memoria tampón de transporte de sistemas, TB_{sys} , a la velocidad del tren de transporte. Comprenden paquetes de tren de transporte cuyos valores de PID son 0, 1, 2 y 3 (si los hubiere), y todos los paquetes del tren de transporte identificados mediante la tabla de asociación de programas (véase el cuadro 2-30) que tienen el valor `program_map_PID` (valor de PID de correspondencia de programas) para el programa seleccionado. Los datos de la tabla de información de red (NIT, network information table) especificados por el PID de NIT no se transfieren a TB_{sys} .

Los bytes se extraen de TB_{sys} a la velocidad $R_{x_{\text{sys}}}$ y se entregan a B_{sys} . Cada byte se transfiere instantáneamente.

Los paquetes de tren de datos duplicados no se entregan a B_{sys} .

Los paquetes de transporte que no entran en ninguna TB_n o TB_{sys} se descartan.

El tamaño de la memoria tampón de transporte tiene un valor fijo de 512 bytes.

Los tamaños de memoria tampón de tren elemental EBS_1 a EBS_n se definen para vídeo como iguales al tamaño de memoria tampón vbv transportado en el encabezamiento de secuencia. Véanse el resumen de parámetros constreñidos en ISO/CEI 11172-2 y el cuadro 8-14 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Los tamaños de memoria tampón de multiplexación MBS_1 a MBS_n se definen para vídeo como sigue:

Para nivel bajo y nivel principal:

$$MBS_n = BS_{mux} + BS_{oh} + VBV_{m\acute{a}x} [perfil, nivel] - vbv_buffer_size$$

donde BS_{oh} , memoria tampón para tara de paquetes PES, se define como:

$$BS_{oh} = (1/750) \text{ segundos} \times R_{m\acute{a}x} [perfil, nivel]$$

y BS_{mux} , memoria tampón múltiplex adicional, se define como:

$$BS_{mux} = 0,004 \text{ segundos} \times R_{m\acute{a}x} [perfil, nivel]$$

y donde $VBV_{m\acute{a}x}[perfil, nivel]$ se define en el cuadro 8-14 de la Rec. UIT-T H.262 | ISO/CEI 13818-2 y $R_{m\acute{a}x}[perfil, nivel]$ se define en el cuadro 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, y vbv_buffer_size se lleva en el encabezamiento de secuencia descrito en 6.2.2 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Para nivel alto 1440 y nivel alto:

$$MBS_n = BS_{mux} + BS_{oh}$$

donde BS_{oh} se define como:

$$BS_{oh} = (1/750) \text{ segundos} \times R_{m\acute{a}x} [perfil, nivel]]$$

y BS_{mux} se define como:

$$BS_{mux} = 0,004 \text{ segundos} \times R_{m\acute{a}x} [perfil, nivel]$$

y donde $R_{m\acute{a}x}[perfil, nivel]$ se define en el cuadro 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Para trenes de bits conformes a ISO/CEI 11172-2 con parámetros constreñidos:

$$MBS_n = BS_{mux} + BS_{oh} + vbv_m\acute{a}x - vbv_buffer_size$$

donde BS_{oh} se define como:

$$BS_{oh} = (1/750) \text{ segundos} \times R_{m\acute{a}x}$$

y BS_{mux} se define como:

$$BS_{mux} = 0,004 \text{ segundos} \times R_{m\acute{a}x}$$

y donde $R_{m\acute{a}x}$ y $vbv_m\acute{a}x$ se refieren a la velocidad binaria máxima y al tamaño máximo de memoria tampón vbv_buffer_size para un tren de bits con parámetros constreñidos de ISO/CEI 11172-2, respectivamente.

Una porción $BS_{mux} = 4 \text{ ms} \times R_{m\acute{a}x}[perfil, nivel]$ del MBS_n se atribuye para almacenamiento en memoria tampón, para permitir la multiplexación. El resto está disponible para BS_{oh} y puede también estar disponible para multiplexación inicial.

NOTA 2 – La ocupación de la memoria tampón por la tara de paquetes PES está directamente limitada en los trenes PES por el PES-STD, que se define en 2.5.2.4. Es posible, pero no necesario, utilizar trenes PES para construir trenes de transporte.

Tamaño de memoria tampón, BS_n

Los tamaños principales de memoria tampón BS_1 a BS_n se definen como sigue:

Audio

Para audio de ISO/CEI 13818-7 con ADTS:

Número de canales	R _{x_n} [bit/s]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Canales: El número de canales de salida de audio de anchura de banda completa más el número de elementos de canal con acoplamiento conmutado independientes dentro del mismo tren de audio elemental. Por ejemplo, en el caso típico en que no hay elementos de canal con acoplamiento conmutado independientes, mono es 1 canal, estéreo es 2 canales y canal ambifónico 5.1 es 5 canales (no se cuenta el canal LFE).

Para otro audio:

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ bytes}$$

El tamaño de la memoria tampón para decodificación de unidad de acceso BS_{dec} y de la memoria tampón para la tara de paquetes PES BS_{oh} están constreñidos por la siguiente expresión:

$$BS_{dec} + BS_{oh} \leq 2848 \text{ bytes}$$

Una porción (736 bytes) de la memoria tampón de 3584 bytes se atribuye para almacenamiento tampón, para permitir la multiplexación. El resto, 2848 bytes, se comparte para el almacenamiento tampón de unidades de acceso BS_{dec}, BS_{oh} y multiplexación adicional.

Sistemas

La memoria tampón principal B_{sys} para datos de sistema tiene un tamaño de BS_{sys} = 1536 bytes.

Vídeo

En el caso de trenes elementales de vídeo, los datos se transfieren de MB_n a EB_n utilizando uno de estos dos métodos: el método de la fuga o el método de retardo VBV.

Método de la fuga

En el método de la fuga se transfieren datos de MB_n a EB_n utilizando la velocidad de fuga R_{bx}. El método de la fuga se utiliza cuando se da uno cualquiera de los casos siguientes:

- el descriptor de STD (véase 2.6.32) para el tren elemental no está presente en el tren de transporte;
- el descriptor de STD está presente y la bandera válida de fuga tiene el valor '1';
- el descriptor de STD está presente y la bandera válida de fuga tiene el valor '0', y los campos de retardo vbv codificados en el tren de vídeo tienen el valor 0xFFFF; o
- el estado del modo truco es verdadero (véase 2.4.3.7).

Para el nivel bajo y el nivel principal:

$$Rbx_n = R_{m\acute{a}x} [perfil, nivel]$$

Para el nivel alto 1440 y el nivel alto:

$$Rbx_n = \text{Min}\{1,05 \times R_{es}, R_{m\acute{a}x}[perfil, nivel]\}$$

Para un tren de bits para parámetros constreñidos según ISO/CEI 11172-2:

$$R_{bx_n} = 1, 2 \times R_{m\acute{a}x}$$

donde $R_{m\acute{a}x}$ es la velocidad binaria máxima para un tren de bits con parámetros constreñidos conforme a ISO/CEI 11172-2.

Si hay datos de cabida útil de paquete PES en MB_n , y EB_n no está llena, la cabida útil de paquetes PES se transfiere de MB_n a EB_n a una velocidad igual a R_{bx_n} . Si EB_n está llena, no se suprimen datos en MB_n . Cuando un byte de datos se transfiere de MB_n a EB_n , todos los bytes de encabezamiento de paquete PES que se encuentran en MB_n y preceden inmediatamente a ese byte se suprimen y descartan instantáneamente. Cuando no hay datos de cabida útil de paquete PES presentes en MB_n no se suprimen datos en esta memoria tampón. Todos los datos que entran en MB_n salen de dicha memoria tampón. Todos los bytes de datos de cabida útil de paquete PES entran instantáneamente en EB_n después de salir de MB_n .

Método de retardo vbv

En el método de retardo vbv se especifica de manera precisa el instante en que cada byte de datos de vídeo codificados se transfiere de MB_n a EB_n , utilizando valores de retardo de vbv codificados en el tren elemental de vídeo. El método del retardo vbv se utiliza cuando el descriptor de STD (véase 2.6.32) para este tren elemental está presente en el tren de transporte, la bandera válida de fuga en el descriptor tiene el valor '0', y los campos de retardo vbv codificados en el tren de vídeo tienen un valor diferente de 0xFFFF. Si cualquiera de los valores del retardo vbv en una secuencia de vídeo no es igual a 0xFFFF, ninguno de los campos de retardo vbv de esa secuencia será igual a 0xFFFF (véanse ISO/CEI 11172-2 y la Rec. UIT-T H.262 | ISO/CEI 13818-2).

Cuando se utiliza el método de retardo vbv, el byte final del código de comienzo de imagen de vídeo para la imagen j se transfiere de MB_n a EB_n en el instante $td_n(j) - vbv_delay(j)$, donde $td_n(j)$ es el tiempo de decodificación de la imagen j , definido anteriormente, y $vbv_delay(j)$ es el retardo en segundos, indicado por el campo de retardo vbv de la imagen j . La transferencia de bytes entre los bytes finales de códigos de comienzo de imagen sucesivos (incluido el byte final del segundo código de comienzo) a una memoria tampón EB_n , se efectúa a una velocidad constante por pieza, $R_{bx}(j)$, que se especifica para cada imagen j . Específicamente, la velocidad $R_{bx}(j)$ de transferencia a esta memoria tampón viene dada por:

$$R_{bx}(j) = NB(j) / (vbv_delay(j) - vbv_delay(j+1) + td_n(j+1) - td_n(j)) \quad (2-6)$$

donde $NB(j)$ es el número de bytes entre los bytes finales de los códigos de comienzo de imagen (incluido el byte final del segundo código de comienzo) de las imágenes j y $j + 1$, excluidos los bytes de encabezamiento de paquete PES.

NOTA 3 – $vbv_delay(j+1)$ y $td_n(j+1)$ pueden tener valores diferentes de los que cabe esperar normalmente para una visualización periódica de vídeo si la bandera de bajo retardo en la extensión de secuencia de vídeo está puesta a '1'. Puede que no sea posible determinar los valores correctos mediante el examen del tren de bits.

La $R_{bx}(j)$ derivada de la ecuación 2-6 será menor o igual que $R_{m\acute{a}x}[\text{perfil, nivel}]$ para trenes elementales de tipo tren 0x02 (véase el cuadro 2-34), donde $R_{m\acute{a}x}[\text{perfil, nivel}]$ se define en la Rec. UIT-T H.262 | ISO/CEI 13818-2, y será menor o igual que la velocidad binaria máxima permitida para trenes elementales de vídeo con parámetros constreñidos del tipo tren 0x01, véase ISO/CEI 11172-2.

Cuando se transfiere un byte de datos de MB_n a EB_n , todos los bytes del encabezamiento de paquete PES que se encuentran en MB_n y preceden inmediatamente a ese byte son instantáneamente suprimidos y descartados. Todos los datos que entran en MB_n salen de dicha memoria tampón. Todos los bytes de datos de cabida útil de paquete PES entran instantáneamente en EB_n al salir de MB_n .

Supresión de unidades de acceso

Para cada memoria tampón de tren elemental EB_n y para cada memoria tampón principal B_n , todos los datos para la unidad de acceso que han estado en la memoria tampón durante más tiempo, $A_n(j)$, y todos los bytes de relleno que le preceden inmediatamente y que están presentes en la memoria tampón en el instante $td_n(j)$ se suprimen en el instante $td_n(j)$. El tiempo de decodificación $td_n(j)$ se especifica en los campos DTS o PTS (véase 2.4.3.6). Los tiempos de decodificación $td_n(j+1)$, $td_n(j+2)$, ... de unidades de acceso sin campos DTS o PTS codificados que siguen inmediatamente a la unidad de acceso j pueden derivarse de la información contenida en el tren elemental. Véase el anexo C de la Rec. UIT-T H.262 | ISO/CEI 13818-2, ISO/CEI 13818-3, o ISO/CEI 11172. Véase también 2.7.5. En el caso de audio, todos los encabezamientos de paquete PES que están almacenados inmediatamente antes de la unidad de acceso o que están incorporados en los datos de la unidad de acceso se suprimen instantáneamente al suprimirse la unidad de acceso. Al suprimirse la unidad de acceso, se decodifica instantáneamente en una unidad de presentación.

Datos de sistema

En el caso de datos de sistema, los datos se suprimen de la memoria tampón principal B_{sys} a una velocidad de R_{sys} siempre que haya por lo menos un byte disponible en la memoria tampón B_{sys} .

$$R_{sys} = \max(80\,000 \text{ bits/s}, \text{transport_rate}(i) \times 8 \text{ bits/byte} / 500) \quad (2-7)$$

NOTA 4 – La finalidad de aumentar R_{sys} en el caso de altas velocidades de transporte es permitir una mayor velocidad de datos para la información específica de programa.

Bajo retardo

Cuando la bandera de bajo retardo (*low_delay*) en la extensión de secuencia vídeo se pone a '1' (6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2), la memoria tampón EB_n puede estar subutilizada. En este caso, cuando la memoria tampón del tren elemental de T-STD, EB_n , es examinada en el instante especificado por $td_n(j)$, los datos completos para la unidad de acceso pueden no estar presentes en la memoria tampón EB_n . Cuando esto ocurre, la memoria tampón será reexaminada a intervalos de dos periodos de campo hasta que los datos para la unidad de acceso completa estén presentes en la memoria tampón. En este momento, toda la unidad de acceso será suprimida instantáneamente de la memoria tampón EB_n . No se producirá desbordamiento de la memoria tampón EB_n .

Cuando la bandera de bajo retardo se pone a '1', se permite continuamente y sin límite la subutilización de la memoria tampón EB_n . El decodificador T-STD suprimirá datos de unidades de acceso de la memoria tampón EB_n lo más pronto posible de acuerdo con el párrafo anterior y cualesquiera valores DTS o PTS codificados en el tren de bits. Obsérvese que el decodificador puede ser incapaz de restablecer la decodificación correcta y tiempos de visualización indicados por DTS y PTS hasta que cesa la situación de subutilización de la memoria tampón EB_n y se encuentra PTS o DTS en el tren de bits.

Modo truco

Cuando la bandera DSM modo truco (*DSM_trick_mode*) (véase 2.4.3.6) se pone a '1' en el encabezamiento de paquete PES de un paquete que contiene el comienzo de una unidad de acceso vídeo tipo B y el campo control modo truco se pone a '001' (cámara lenta) o '010' (trama congelada), o '100' (cámara lenta inversa), la unidad de acceso de imagen B no se suprime de la memoria tampón de datos vídeo EB_n hasta el último de los posibles múltiples instantes en que cualquier campo de la imagen es decodificado y presentado. La repetición de la presentación de campos e imágenes se define en 2.4.3.8 en relación con cámara lenta, cámara lenta inversa y control de identidad de campo. La unidad de acceso se suprime instantáneamente de EB_n en el tiempo indicado, que depende del valor de control de repetición.

Cuando la bandera de DSM modo truco se pone a '1' en el encabezamiento de paquete PES de un paquete que contiene el primer byte de un código de comienzo de imagen, el estado modo truco se convierte en verdadero cuando el código de comienzo de imagen en el paquete PES es suprimido de la memoria tampón EB_n . El estado modo truco sigue siendo verdadero hasta que un encabezamiento de paquete PES es recibido por el T-STD en el cual la bandera DSM modo truco se pone a '0' y el primer byte del código de comienzo de imagen después del encabezamiento de paquete PES es suprimido de la memoria tampón EB_n . Cuando el estado modo truco es verdadero, la memoria tampón EB_n puede estar subutilizada. Todas las otras constricciones de trenes normales se mantienen cuando el estado modo truco es verdadero.

2.4.2.4 Decodificación

Los trenes elementales almacenados en B_1 a B_n y EB_1 a EB_n son decodificados instantáneamente por los decodificadores D_1 a D_n y pueden ser retardados en las memorias tampón de reordenación O_1 a O_n antes de ser presentados al espectador a la salida de T-STD. Las memorias tampón de reordenación se utilizan solamente en el caso de un tren elemental de vídeo cuando algunas unidades de acceso no son transportadas en orden de presentación. Estas unidades de acceso tendrán que ser reordenadas antes de la presentación. En particular, si $P_n(k)$ es una imagen I o una imagen P transportada antes de una o más imágenes B, debe ser retardada en la memoria tampón de reordenación, O_n , del T-STD antes de ser presentada. Toda imagen almacenada previamente en O_n se presenta antes que la imagen vigente pueda ser almacenada. $P_n(k)$ se debe retardar hasta que se codifica la siguiente imagen I o imagen P. Mientras está almacenada en la memoria tampón de reordenación, las imágenes B subsiguientes son decodificadas y presentadas.

El instante en el cual una unidad de presentación $P_n(k)$ es presentada al espectador es $tp_n(k)$. Para las unidades de presentación que no requieren retardo de reordenación, $tp_n(k)$ es igual a $td_n(j)$ porque las unidades de acceso son decodificadas instantáneamente; éste es el caso, por ejemplo, para las tramas B. Para las unidades de presentación que son retardadas, $tp_n(k)$ y $td_n(j)$ difieren por el tiempo que $P_n(k)$ es retardado en la memoria tampón de reordenación, que es un múltiplo del periodo de imagen nominal. Se ha de tener cuidado de utilizar un retardo de reordenación adecuado desde el principio de los trenes elementales de vídeo para satisfacer los requisitos de todo el tren. Por ejemplo, un tren que inicialmente sólo tienen imágenes I y P pero que posteriormente incluye imágenes B debe incluir retardo de reordenación comenzando en el principio del tren.

La Rec. UIT-T H.262 | ISO/CEI 13818-2 explica la reordenación de imágenes vídeo con más detalle.

2.4.2.5 Presentación

La función de un sistema de decodificación es reconstruir unidades de presentación a partir de datos comprimidos y presentarlas en una secuencia sincronizada en los tiempos de presentación correctos. Aunque los dispositivos de presentación reales de audio y vídeo generalmente tienen retardos finitos y diferentes y pueden tener retardos adicionales impuestos por las funciones de postprocesamiento o de salida, el decodificador-objetivo de sistema modela estos retardos como cero.

En el T-STD de la figura 2-1, la visualización de una unidad de presentación vídeo (una imagen) se produce instantáneamente en su momento de presentación, $tp_n(k)$.

En el T-STD, la salida de una unidad de presentación de audio comienza en su tiempo de presentación, $tp_n(k)$, cuando el decodificador presenta instantáneamente la primera muestra. Las muestras siguientes en la unidad de presentación se presentan en consecuencia a la velocidad de muestreo del audio.

2.4.2.6 Gestión de las memorias tampón

Los trenes de transporte se construirán de tal modo que se satisfagan las condiciones definidas en esta subcláusula. En esta subcláusula se utiliza la notación definida para el decodificador-objetivo de sistema.

TB_n y TB_{sys} no desbordarán. TB_n y TB_{sys} se vaciarán por lo menos una vez cada segundo. B_n no desbordará ni será subutilizada. B_{sys} no desbordará.

EB_n no podrá ser subutilizada cuando la bandera de bajo retardo en la extensión de secuencia de vídeo esté puesta a '1' (véase 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2) o el estado modo truco sea verdadero.

Cuando está aplicado el método de la fuga para especificar transferencias, MB_n no desbordará, y se vaciará por lo menos una vez por segundo. EB_n no desbordará.

Cuando está aplicado el método del retardo vbv para especificar transferencias, MB_n no desbordará ni será subutilizada, y EB_n no desbordará.

El tiempo empleado por cualquier dato para pasar por las memorias tampón de los decodificadores-objetivos de sistemas será igual o menor que un segundo, salvo para los datos de vídeo de imagen fija y los trenes ISO/CEI 14496. Específicamente: $td_n(j) - t(i) \leq 1$ segundo para todos los valores de j y todos los bytes i en la unidad de acceso $A_n(j)$.

En el caso de datos de vídeo de imagen fija, el retardo deberá cumplir la expresión $td_n(j) - t(i) \leq 60$ segundos para todos los valores de j , y todos los bytes i en la unidad de acceso $A_n(j)$.

Para los trenes ISO/CEI 14496, el retardo es constreñido por $td_n(j) - t(i) \leq 10$ segundos para todos los valores de j , y todos los bytes i en la unidad de acceso $A_n(j)$.

Definición de desbordamiento y subutilización

Sea $F_n(t)$ la ocupación total instantánea de la memoria B_n del T-STD.

$F_n(t) = 0$ instantáneamente antes de $t = t(0)$.

No se produce desbordamiento si:

$$F_n(t) \leq BS_n$$

para todas t y n .

No se produce subutilización si:

$$0 \leq F_n(t)$$

para todas t y n .

2.4.2.7 Extensiones de T-STD para el transporte de datos ISO/CEI 14496

Para decodificar los datos ISO/CEI 14496 de un tren de transporte, se amplía el modelo de decodificador objetivo de tren de transporte (T-STD). Los parámetros de T-STD para la decodificación de trenes elementales ISO/CEI 14496 se definen en 2.11.2, mientras que en 2.11.3 se definen las extensiones y parámetros de T-STD para decodificar escenas ISO/CEI 14496 y trenes asociados.

2.4.2.8 Extensiones T-STD para el transporte de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10

Para definir la decodificación en el T-STD de trenes de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 transportados en un tren de transporte, es necesario ampliar el modelo T-STD. Esta ampliación y los parámetros T-STD para decodificar los trenes vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 se definen en 2.14.3.1.

2.4.3 Especificación de la sintaxis y semántica del tren de transporte

La siguiente sintaxis describe un tren de bytes. Los paquetes del tren de transporte tendrán una longitud de 188 bytes.

2.4.3.1 Tren de transporte

Véase el cuadro 2-1.

Cuadro 2-1 – Tren de transporte

Sintaxis	N.º de bits	Mnemónico
<pre>MPEG_transport_stream() { do { transport_packet() } while (nextbits() == sync_byte) }</pre>		

2.4.3.2 Capa de paquete del tren de transporte

Véase el cuadro 2-2.

Cuadro 2-2 – Paquete de transporte de esta Recomendación | Norma Internacional

Sintaxis	N.º de bits	Mnemónico
<pre>transport_packet(){ sync_byte transport_error_indicator payload_unit_start_indicator transport_priority PID transport_scrambling_control adaptation_field_control continuity_counter if(adaptation_field_control == '10' adaptation_field_control == '11'){ adaptation_field() } if(adaptation_field_control == '01' adaptation_field_control == '11') { for (i = 0; i < N; i++){ data_byte } } }</pre>	<p>8</p> <p>1</p> <p>1</p> <p>1</p> <p>13</p> <p>2</p> <p>2</p> <p>4</p> <p>8</p>	<p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p>

2.4.3.3 Definición semántica de campos en la capa de paquete del tren de transporte

sync_byte (byte de sincronización) – El byte de sincronización es un campo de 8 bits fijos cuyo valor es '0100 0111' (0x47). Se debe de evitar la emulación del byte de sincronización en la elección de valores para otros campos de aparecen regularmente, como PID.

transport_error_indicator (indicador de error de transporte) – El indicador de error de transporte es una bandera de un bit. Cuando se pone a '1' indica que por lo menos existe un bit erróneo que no puede corregirse en el paquete de tren de transporte asociado. Este bit puede ser puesto a '1' por entidades externas a la capa de transporte. Cuando se pone a '1', este bit no será reiniciado a '0' a menos que se hayan corregido los valores de bit erróneos.

payload_unit_start_indicator (indicador de comienzo de unidad de cabida útil) – El indicador de comienzo de unidad de cabida útil es una bandera de un bit que tiene significado normativo para los paquetes del tren de transporte que llevan paquetes PES (véase 2.4.3.6) o datos PSI (véase 2.4.4).

Cuando la cabida útil de paquete del tren de transporte contiene datos de paquetes PES, el indicador de comienzo de unidad de cabida útil tiene el siguiente significado: un '1' indica que la cabida útil de este paquete de tren de transporte comenzará con el primer byte de un paquete PES y un '0' indica que ningún paquete PES comenzará en este paquete de tren de transporte. Si el indicador de comienzo de unidad de cabida útil se pone a '1', sólo un paquete PES comienza en cualquier paquete de tren de transporte. Esto se aplica también a trenes privados de stream_type 6 (véase el cuadro 2-34).

Cuando la cabida útil del paquete de tren de transporte contiene datos PSI, el indicador de comienzo de unidad de cabida útil tiene el siguiente significado: si el paquete de tren de transporte lleva el primer byte de una sección PSI, el valor de indicador de comienzo de unidad de cabida útil será '1', lo que indica que el primer byte de la cabida útil de este paquete de tren de transporte lleva el campo de puntero. Si el paquete de tren de transporte no lleva el primer byte de una sección PSI, el valor de indicador de comienzo de unidad de cabida útil será '0', lo que indica que no hay campo de puntero en la cabida útil. Véanse 2.4.4.1 y 2.4.4.2. Esto se aplica también a trenes privados de stream_type 5 (véase el cuadro 2-34).

Para paquetes nulos, el indicador de comienzo de unidad de cabida útil se pondrá '0'.

El significado de este bit para paquetes de tren de transporte que sólo llevan datos privados no se define en esta Especificación.

transport_priority (prioridad de transporte) – La prioridad de transporte es un indicador de un bit. Cuando se pone a '1' indica que el paquete asociado es de mayor prioridad que los otros paquetes que tienen el mismo PID y que no tienen el bit puesto a '1'. El mecanismo de transporte puede utilizar esto para dar prioridad a sus datos dentro de un tren elemental. Según la aplicación, el campo de prioridad puede estar codificado con independencia del PID o dentro de un PID solamente. Este campo puede ser cambiado por codificadores o decodificadores específicos del canal.

PID – El PID es un campo de 13 bits que indica el tipo de los datos almacenados en la cabida útil del paquete. El valor de PID 0x0000 se reserva para la tabla de asociación de programas (véase el cuadro 2-30). El valor de PID 0x0001 se reserva para la tabla de acceso condicional (véase el cuadro 2-32). El valor de PID 0x0002 se reserva para la tabla de descripción del tren de transporte (véase el cuadro 2-36), el valor de PID 0x0003 se reserva para el cuadro de información de control IPMP (véase la norma ISO/CEI 13818-11) y los valores de PID 0x0004 – 0x000F están reservados. El valor de PID 0x1FFF está reservado para paquetes nulos (véase el cuadro 2-3).

Cuadro 2-3 – Tabla de PID

Valor	Descripción
0x0000	Tabla de asociación de programas
0x0001	Tabla de acceso condicional
0x0002	Tabla de descripción de tren de transporte
0x0003	IPMP Control Information Table
0x0004-0x000F	Reservado
0x0010 ... 0x1FFE	Pueden asignarse como PID de red, PID de correspondencia de programas, PID de tren elemental, o para otros fines
0x1FFF	Paquete nulo
NOTA – Los paquetes de transporte con valores de PID 0x0000, 0x0001 y 0x0010-0x1FFE están autorizados para transportar una PCR.	

transport_scrambling_control (control de aleatorización de transporte) – Este campo de 2 bits indica el modo de aleatorización de la cabida útil del paquete de tren de transporte. El encabezamiento de paquete del tren de transporte, que incluye el campo de adaptación cuando está presente, no será aleatorizado. En el caso de un paquete nulo, el valor del campo de control de aleatorización de transporte se pondrá a '00' (véase el cuadro 2-4).

Cuadro 2-4 – Valores de control de aleatorización

Valor	Descripción
00	No aleatorizado
01	Definido por el usuario
10	Definido por el usuario
11	Definido por el usuario

adaptation_field_control (control de campo de adaptación) – Este campo de 2 bits indica si este encabezamiento de paquete de tren de transporte está seguido por un campo de adaptación y/o cabida útil (véase el cuadro 2-5).

Cuadro 2-5 – Valores de control de campo de adaptación

Valor	Descripción
00	Reservado para uso futuro por ISO/CEI
01	Ningún campo de adaptación, cabida útil solamente
10	Campo de adaptación solamente, ninguna cabida útil
11	Campo de adaptación seguido por cabida útil

Los decodificadores de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 descartarán los paquetes de tren de transporte con el campo de control de campo de adaptación puesto a un valor de '00'. En el caso de un paquete nulo, el valor del control de campo de adaptación se pondrá a '01'.

continuity_counter (contador de continuidad) – El contador de continuidad es un campo de 4 bits que se incrementa con cada paquete de tren de transporte con el mismo PID. El contador de continuidad se reinicia a 0 después de su valor máximo. El contador de continuidad no se incrementará cuando el control de campo de adaptación del paquete equivale a '00' o a '10'.

En los trenes de transporte, se pueden enviar paquetes duplicados sólo como dos paquetes de tren de transporte consecutivos del mismo PID. Los paquetes duplicados tendrán el mismo valor de contador de continuidad que el paquete original y el campo de control de campo de adaptación será igual a '01' u '11'. En los paquetes duplicados, cada byte del paquete original estará duplicado, con la excepción de que en los campos de referencia de reloj de programa, si están presentes, se codificará un valor válido.

El contador de continuidad en un paquete de tren de transporte determinado es continuo cuando difiere por un valor positivo de uno del valor de contador de continuidad en el paquete de tren de transporte anterior del mismo PID, o cuando se cumple cualquiera de las condiciones de no incremento (control de campo de adaptación puesto a '00' o '10', o paquetes duplicados como se describe anteriormente). El contador de continuidad puede ser discontinuo cuando el indicador de discontinuidad se pone a '1' (véase 2.4.3.4). En el caso de un paquete nulo, el valor de contador de continuidad no está definido.

data_byte (bytes de datos) – Los bytes de datos serán bytes de datos contiguos procedentes de los paquetes PES (véase 2.4.3.6), secciones PSI (véase 2.4.4), bytes de relleno de paquetes después de secciones PSI o datos privados en estas estructuras según lo indicado por el PID. En el caso de paquetes nulos, con valor de PID 0x1FFF, se puede asignar cualquier valor a los bytes de datos. El número de bytes de datos, N, viene dado por 184 menos el número de bytes en el campo de adaptación, como se indica en 2.4.3.4.

2.4.3.4 Campo de adaptación

Véase el cuadro 2-6.

Cuadro 2-6 – Campo de adaptación de tren de transporte

Sintaxis	N.º de bits	Mnemónico
adaptation_field() {		
adaptation_field_length	8	uimsbf
if (adaptation_field_length > 0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_point_flag	1	bslbf
transport_private_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if (PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf
Reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if (OPCR_flag == '1') {		
original_program_clock_reference_base	33	uimsbf
Reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
if (splicing_point_flag == '1') {		
splice_countdown	8	tcimsbf
}		
if (transport_private_data_flag == '1') {		
transport_private_data_length	8	uimsbf
for (i = 0; i < transport_private_data_length; i++) {		
private_data_byte	8	bslbf
}		
}		
if (adaptation_field_extension_flag == '1') {		
adaptation_field_extension_length	8	uimsbf
ltw_flag	1	bslbf
piecewise_rate_flag	1	bslbf
seamless_splice_flag	1	bslbf
Reserved	5	bslbf
if (ltw_flag == '1') {		
ltw_valid_flag	1	bslbf
ltw_offset	15	uimsbf
}		
if (piecewise_rate_flag == '1') {		
Reserved	2	bslbf
piecewise_rate	22	uimsbf
}		
if (seamless_splice_flag == '1') {		
Splice_type	4	bslbf
DTS_next_AU[32..30]	3	bslbf
marker_bit	1	bslbf
DTS_next_AU[29..15]	15	bslbf
marker_bit	1	bslbf
DTS_next_AU[14..0]	15	bslbf
marker_bit	1	bslbf
}		
for (i = 0; i < N; i++) {		
Reserved	8	bslbf
}		
}		
for (i = 0; i < N; i++) {		
stuffing_byte	8	bslbf
}		
}		
}		

2.4.3.5 Definición semántica de campos en el campo de adaptación

adaptation_field_length (longitud de campo de adaptación) – La longitud del campo de adaptación es un campo de 8 bits que especifica el número de bytes en el campo de adaptación que sigue inmediatamente a la longitud de campo de adaptación. El valor 0 es para insertar un solo byte de relleno en un paquete de tren de transporte. Cuando el valor de control de campo de adaptación es '11', el valor de longitud de campo de adaptación estará en la gama 0 a 182. Cuando el valor de campo de adaptación es '10', el valor de la longitud de campo de adaptación será 183. Para los paquetes de tren de transporte que llevan paquetes PES se necesita relleno cuando los datos de paquete PES son insuficientes para llenar completamente los bytes de cabida útil del paquete de tren de transporte. El relleno se efectúa definiendo un campo de adaptación más largo que la suma de las longitudes de los elementos de datos en el mismo, de modo que los bytes de cabida útil que quedan después del campo de adaptación acomoden exactamente los datos de paquete PES disponibles. El espacio suplementario en el campo de adaptación se rellena con bytes de relleno.

Éste es el único método de relleno autorizado para paquetes de tren de transporte que llevan paquetes PES. Para los paquetes de tren de transporte que llevan PSI, en 2.4.4 se describe otro método de relleno.

discontinuity_indicator (indicador de discontinuidad) – Este es un campo de un bit que, cuando se pone a '1' indica que el estado de discontinuidad es verdadero para el paquete de tren de transporte vigente. Cuando el indicador de discontinuidad está puesto a '0' o no está presente, el estado de discontinuidad es falso. El indicador de discontinuidad se utiliza para indicar dos tipos de discontinuidades: discontinuidades de la base de tiempo de sistema y discontinuidades del contador de continuidad.

Una discontinuidad de la base de tiempo de sistema se indica mediante la utilización del indicador de discontinuidad en paquetes de tren de transporte de un PID designado como una PCR_PID (véase 2.4.4.9). Cuando el estado de discontinuidad es verdadero para un paquete de tren de transporte de un PID designado como una PCR_PID, la siguiente PCR de un paquete de tren de transporte con el mismo PID representa una muestra de un nuevo reloj de tiempo de sistema para el programa asociado. Por definición, el punto de discontinuidad de la base de tiempo de sistema se define como el instante en el cual el primer byte de un paquete que contiene una PCR de una nueva base de tiempo de sistema llega a la entrada del T-STD. El indicador de discontinuidad deberá ponerse a '1' en el paquete en que se produce la discontinuidad de la base de tiempo de sistema. El bit indicador de discontinuidad se puede también poner a '1' en paquetes de tren de transporte con la misma PCR_PID antes del paquete que contiene la nueva PCR de base de tiempo del sistema. En este caso, una vez que el indicador de continuidad se ha puesto a '1' se mantendrá fijado a '1' en todos los paquetes de tren de transporte que tengan la misma PCR_PID hasta, inclusive, el paquete de tren de transporte que contiene la primera PCR de la nueva base de tiempo de sistema. Después de la aparición de la primera discontinuidad de la base de tiempo de sistema, no menos de dos PCR para la nueva base de tiempo de sistema se recibirán antes de que pueda aparecer otra discontinuidad de la base de tiempo de sistema. Además, salvo si el estado del modo truco es verdadero, los datos procedentes de no más de dos bases de tiempo de sistema estarán presentes en el conjunto de memorias tampón T-STD para un programa en cualquier momento.

Antes de la aparición de una discontinuidad de base de tiempo de sistema, el primer byte de un paquete de tren de transporte que contiene una PTS o DTS referente a la nueva base de tiempo de sistema no deberá llegar a la entrada del T-STD. Después de la aparición de una discontinuidad de base de tiempo de sistema, el primer byte de un paquete de tren de transporte que contiene una PTS o una DTS referente a la anterior base de tiempo de sistema no deberá llegar a la entrada del T-STD.

Una discontinuidad del contador de continuidad se indica por el uso del indicador de discontinuidad en cualquier paquete de tren de transporte. Cuando el estado de discontinuidad es verdadero en cualquier paquete de tren de transporte de un PID no designado como una PCR_PID, el contador de continuidad en el paquete puede ser discontinuo con respecto al anterior paquete de tren de transporte del mismo PID. Cuando el estado de discontinuidad es verdadero en un paquete de transporte de un PID que está designado como una PCR_PID, el contador de continuidad sólo puede ser discontinuo en el paquete en que se produce una discontinuidad de la base de tiempo de sistema. Se produce un punto de discontinuidad del contador de continuidad cuando el estado de discontinuidad es verdadero en un paquete de tren de transporte y el contador de continuidad en el mismo paquete es discontinuo con respecto al anterior paquete de tren de transporte del mismo PID. Se producirá un punto de discontinuidad del contador de continuidad una vez, como máximo, desde la iniciación del estado de discontinuidad hasta la conclusión del estado de discontinuidad. Además, para todos los PID que no están designados como PCR_PID, cuando el indicador de discontinuidad está puesto a '1' en el paquete de un PID específico, el indicador de discontinuidad puede ponerse a '1' en el paquete de tren de transporte de ese mismo PID, pero no se pondrá a '1' en tres paquetes de tren de transporte consecutivos del mismo PID.

A los efectos de esta cláusula, un punto de acceso al tren elemental se define del modo siguiente:

- Vídeo conforme a ISO/CEI 11172-2 y vídeo conforme a Rec. UIT-T H.262 | ISO/CEI 13818-2 – El primer byte de una cabecera de secuencia de vídeo.
- ISO/CEI 14496-2 visual – El primer byte de una cabecera de secuencia de objeto visual.

- Vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10 – El primer byte de una unidad de acceso AVC. Los conjuntos de parámetros SPS y PPS a los cuales se hace referencia en esta unidad de acceso AVC y en todas las subsiguientes del tren de vídeo codificado se incluirán después de este punto de acceso en el tren de bytes y antes de su activación.
- Audio – El primer byte de una trama de audio.

Después de una discontinuidad del contador de continuidad en un paquete de transporte que se designa como portador de datos de tren elemental, el primer byte de datos de tren elemental en un paquete de trenes de transporte del mismo PID será el primer byte de un punto de acceso al tren elemental. En el caso de vídeo conforme a ISO/CEI 11172-2, o vídeo conforme a Rec. UIT-T H.262 | ISO/CEI 13818-2 o vídeo conforme a ISO/CEI 14496-2, el primer byte del punto de acceso al tren elemental también puede ser el primer byte de un código de fin de secuencia (`sequence_end_code`) seguido de un punto de acceso al tren elemental.

Cada paquete de tren de transporte que contiene datos de tren elemental con un PID no designado como una PCR_PID y en el cual aparece un punto de discontinuidad de contador de continuidad, y se produce una PTS o DTS, llegará a la entrada del T-STD después de haberse producido la discontinuidad de base de tiempo de sistema para el programa asociado. Cuando el estado de la discontinuidad es verdadero, si aparecen dos paquetes de tren de transporte del mismo PID que tienen el mismo valor de contador de continuidad y tienen valores de control del campo de adaptación fijados a '01' u '11', se puede descartar el segundo paquete. Un tren de transporte no se construirá de tal manera que al descartar uno de estos paquetes se pierdan datos de cabida útil de paquete PES o de datos PSI.

Después de la aparición de un indicador de discontinuidad puesto a '1' en un paquete del tren de transporte que contiene información PSI, puede aparecer una sola discontinuidad en el número de versión de secciones PSI. Cuando se produce esta discontinuidad, se enviará una versión de `TS_program_map_sections` (secciones de correspondencia de programas TS) del programa apropiado con `section_length = 13`, y el siguiente indicador vigente = 1, de modo que no se describen descriptores de programa ni trenes elementales. Esto será seguido de una versión de la sección de correspondencia de programa TS para cada programa con el número de versión incrementado en uno, y el siguiente indicador vigente = 1, con una definición de programa completa. Esto indica un cambio de versión en datos PSI.

random_access_indicator (indicador de acceso aleatorio) – El indicador de acceso aleatorio es un campo de 1 bit que indica que los paquetes de tren de transporte vigente y subsiguiente con el mismo PID contienen alguna información para ayudar al acceso aleatorio en ese punto.

Concretamente, cuando el bit está puesto a '1', el siguiente paquete PES que empezará en la cabida útil de paquetes trenes de transporte con el PID vigente, contendrá un punto de acceso al tren elemental definido en la semántica del campo indicador de discontinuidad (`discontinuity_indicator` field). Además, en el caso de vídeo, habrá una indicación de tiempo de la presentación para la primera imagen a continuación del punto de acceso al tren elemental.

En el caso del audio, estará presente una indicación de tiempo de presentación en el paquete PES que contenga el primer byte de la trama de audio. En la PCR_PID el indicador de acceso aleatorio sólo se podrá poner a '1' en el paquete de tren de transporte que contiene los campos PCR.

elementary_stream_priority_indicator (indicador de prioridad de tren elemental) – El indicador de prioridad de tren elemental es un campo de 1 bit que indica, entre paquetes con el mismo PID, la prioridad de los datos de tren elemental transportados dentro de la cabida útil de este paquete de tren de transporte. Un '1' indica que la cabida útil tiene una prioridad más alta que las cabidas útiles de otros paquetes de tren de transporte.

En el caso de vídeo conforme a ISO/CEI 11172-2 o conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 14496-2, este campo se puede poner a '1' solamente si la cabida útil contiene uno o varios bytes de un sector (rebanada) intracodificado.

En el caso de vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10, este campo se puede poner a '1' solamente si la cabida útil contiene uno o varios bytes de un sector cuyo `slice_type` sea 2, 4, 7 ó 9.

Un valor de '0' indica que la cabida útil tiene la misma prioridad que todos los otros paquetes que no tienen este bit puesto a '1'.

PCR_flag (bandera PCR) – La bandera PCR es una bandera de 1 bit. Un valor de '1' indica que el campo de adaptación contiene un campo PCR codificado en dos partes. Un valor de '0' indica que el campo de adaptación no contiene ningún campo PCR.

OPCR_flag (bandera OPCR) – La bandera OPCR es una bandera de 1 bit. Un valor de '1' indica que el campo de adaptación contiene un campo OPCR codificado en dos partes. Un valor de '0' indica que el campo de adaptación no contiene campos OPCR.

splicing_point_flag (bandera de punto de empalme) – La bandera de punto de empalme es una bandera de 1 bit que puesto a '1' indica que estará presente un campo de cuenta atrás de empalme en el campo de adaptación asociado, que especifica la aparición de un punto de empalme. Un valor de '0' indica que el campo de cuenta atrás de empalme no está presente en el campo de adaptación.

transport_private_data_flag (bandera de datos privados de transporte) – La bandera de datos privados de transporte es una bandera de 1 bit. Un valor de '1' indica que el campo de adaptación contiene uno o más bytes de datos privados. Un valor de '0' indica que el campo de adaptación no contiene ningún byte de datos privados.

adaptation_field_extension_flag (bandera de extensión de campo de adaptación) – La bandera de extensión de campo de adaptación es un campo de 1 bit que puesto a '1' indica la presencia de una extensión de campo de adaptación. Un valor de '0' indica que no está presente una extensión de campo de adaptación en el campo de adaptación.

program_clock_reference_base; program_clock_reference_extension (base de referencia de reloj de programa; extensión de referencia de reloj de programa) – La referencia de reloj de programa (PCR) es un campo de 42 bits codificado en dos partes. La primera parte, la base de referencia de reloj de programa, es un campo de 33 bits cuyo valor viene dado por PCR_base(i) en la ecuación 2-2. La segunda parte, la extensión de referencia de reloj de programa, es un campo de 9 bits cuyo valor viene dado por PCR_ext(i) en la ecuación 2-3. La PCR indica el tiempo previsto de llegada del byte que contiene el último bit de la base de referencia de reloj de programa a la entrada del decodificador-objetivo de sistema.

original_program_clock_reference_base; original_program_clock_reference_extension (base de referencia de reloj de programa original; extensión de referencia de reloj de programa original) – La referencia de programa original facultativa (OPCR, *original program clock reference*) es un campo de 42 bits codificado en dos partes. Estas dos partes, la base y la extensión, se codifican idénticamente a las dos partes correspondientes del campo PCR. La presencia de OPCR es indicada por la bandera OPCR. El campo OPCR será codificado solamente en paquetes de tren de transporte en los cuales el campo PCR está presente. Se permite OPCR en trenes de transporte monoprograma y multiprogramas.

OPCR ayuda en la reconstrucción de un tren de transporte monoprograma a partir de otro tren de transporte. Al reconstruir el tren de transporte monoprograma original, la OPCR puede ser copiada en el campo PCR. El valor PCR resultante es válido solamente si el tren de transporte monoprograma original es reconstruido enteramente con exactitud. Esto incluiría por lo menos los paquetes de datos privados y PSI que estuviesen presentes en el tren de transporte original y posiblemente requeriría otras configuraciones privadas. Significa también que la OPCR debe ser una copia idéntica de su PCR asociada en el tren de transporte monoprograma original.

La OPCR se expresa como sigue:

$$OPCR(i) = OPCR_base(i) \times 300 + OPCR_ext(i) \quad (2-8)$$

donde:

$$OPCR_base(i) = ((system_clock_frequency \times t(i)) DIV 300) \% 2^{33} \quad (2-9)$$

$$OPCR_ext(i) = ((system_clock_frequency \times t(i)) DIV 1) \% 300 \quad (2-10)$$

El campo OPCR es omitido por el decodificador. Este campo no será modificado por ningún multiplexor o decodificador.

splice_countdown (cuenta atrás de empalme) – La cuenta atrás de empalme es un campo de 8 bits que representa un valor que puede ser positivo o negativo. Un valor positivo especifica el número restante de paquetes de tren de transporte, del mismo PID, que siguen al paquete de tren de transporte asociado hasta que alcanza un punto de empalme. Se excluyen los paquetes de tren de transporte duplicados y los paquetes de tren de transporte que sólo contienen campos de adaptación. El punto de empalme está colocado inmediatamente después que el último byte del paquete de tren de transporte en el cual el campo de cuenta atrás de empalme asociado alcanza el valor cero. En el paquete de tren de transporte donde la cuenta atrás de empalme alcanza el valor cero, el último byte de datos de la cabida útil de paquete de tren de transporte será el último byte de una trama de audio codificada o una imagen codificada. En el caso de vídeo, la unidad de acceso correspondiente puede o no estar terminada por un código de fin de secuencia. Los paquetes de tren de transporte con el mismo PID que siguen pueden contener datos procedentes de un tren elemental diferente del mismo tipo.

La cabida útil del siguiente paquete de tren de transporte del mismo PID (quedan excluidos los paquetes duplicados y los que no tienen cabida útil) comenzará con el primer byte de un paquete PES. En el caso de audio, la cabida útil de

paquete PES comenzará con un punto de acceso. En el caso de vídeo, la carga útil de paquete PES comenzará con un punto de acceso, o con un código de fin de secuencia, seguido por un punto de acceso. Por consiguiente, la anterior trama de audio codificada o imagen codificada se alinea con el límite de paquete, o se rellena para que quede así alineada. A continuación del punto de empalme puede también estar presente el campo de cuenta atrás. Cuando `splice_countdown` es un número negativo de valor menos n ($-n$), indica que el paquete de tren de transporte asociado es el n -ésimo paquete que sigue al punto de empalme (excluidos los paquetes duplicados y los que no tienen cabida útil).

Para la definición del punto de acceso al tren elemental, véase la semántica del indicador de discontinuidad (`discontinuity_indicator`).

transport_private_data_length (longitud de datos privados de transporte) – La longitud de datos privados de transporte es un campo de 8 bits que especifica el número de bytes de datos privados que siguen inmediatamente al campo de longitud de datos privados de transporte. El número de bytes de datos privados no será tal que los datos privados rebasen el campo de adaptación.

private_data_byte (byte de datos privados) – El byte de datos privados es un campo de 8 bits que no será especificado por UIT-T | ISO/CEI.

adaptation_field_extension_length (longitud de extensión de campo de adaptación) – Es un campo de 8 bits que indica el número de bytes de los datos de campo de adaptación extendidos que siguen inmediatamente a este campo, incluidos bytes reservados, si los hubiere.

ltw_flag (legal_time_window_flag) (bandera ltw) (bandera de ventana de hora legal) – Este es un campo de 1 bit que puesto a '1' indica la presencia del campo `ltw_offset` (desplazamiento de ltw).

piecewise_rate_flag (bandera de velocidad por piezas) – Este es un campo de 1 bit que puesto a '1' indica la presencia de velocidad por piezas.

seamless_splice_flag (bandera de empalme liso) – Es una bandera de 1 bit que se pone a '1' para indicar que los campos `splice_type` y `DTS_next_AU` están presentes. Un valor '0' indica que ninguno de los dos campos `splice_type`, ni `DTS_next_AU` están presentes. Este campo no tendrá el valor '1' en paquetes de transporte que no tengan la bandera de punto de empalme puesta a '1'. Una vez puesta a '1' en un paquete de tren de transporte en que la cuenta inversa de empalme es positiva, se pondrá a '1' en todos los paquetes de tren de transporte subsiguientes del mismo PID que tengan la bandera de punto de empalme puesta a '1', hasta que llegue el paquete en que la cuenta inversa de empalme tome el valor cero (incluido el paquete en cuestión).

Cuando la bandera está fijada, si el tren elemental llevado en este PID no es un tren de vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2, el campo tipo de empalme (`splice_type`) se pondrá a '0000'. Si el tren elemental llevado en este PID es un tren de vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2, éste satisfará las constricciones indicadas por el valor del tipo de empalme.

ltw_valid_flag (legal_time_window_valid_flag) (bandera válida de ventana de hora legal) – Este es un campo de 1 bit que puesto a '1' indica que el valor de `ltw_offset` ha de ser válido. Un valor de '0' indica que el valor en el campo `ltw_offset` no está definido.

ltw_offset (legal_time_window_offset) (desplazamiento de ventana de hora legal) – Este es un campo de 15 bits cuyo valor es definido solamente si la bandera válida de ventana de hora legal tiene un valor de '1'. Cuando está definido, el desplazamiento de la ventana de tiempo legal se expresa en unidades de $(300/f_s)$ segundos, donde f_s es la frecuencia de reloj de sistema del programa al que pertenece este PID, y cumple lo siguiente:

$$offset = t_1(i) - t(i)$$

$$ltw_offset = offset // 1$$

donde i es el índice del primer byte de este paquete de tren de transporte, `offset` es el valor codificado en este campo, $t(i)$ es el tiempo de llegada del byte i al T-STD, y $t_1(i)$ es el límite superior de un periodo de tiempo denominado la ventana de tiempo legal que está asociada con este paquete de tren de transporte.

La ventana de tiempo legal tiene la siguiente propiedad: si la entrega de este tren de transporte al decodificador T-STD comienza en el instante $t_1(i)$, es decir, al final de su ventana de tiempo legal, y todos los otros paquetes de tren de transporte del mismo programa se entregan al final de sus ventanas de tiempo legal, entonces:

- Para vídeo – La memoria tampón MB_n para este PID en el T-STD contendrá menos de 184 bytes de datos de tren elemental en el momento que entra en ella el primer byte de la cabida útil de este paquete de tren de transporte, y no se producirán violaciones de la memoria tampón en el T-STD.

- Para audio – La memoria tampón B_n para este PID en el T-STD contendrá menos de BS_{dec} . 1 byte de datos de tren elemental en el momento en que entra en ella el primer byte de este paquete de tren de transporte, y no se producirán violaciones de memoria tampón en el T-STD.

En función de factores que incluyen el tamaño de la memoria tampón MB_n y la velocidad de transferencia de datos entre las memorias tampón MB_n y EB_n , es posible determinar otro tiempo $t_0(i)$, tal que si este paquete se entrega en cualquier lugar dentro del intervalo $[t_0(i), t_1(i)]$, no se producirán violaciones en la memoria tampón de T-STD. Este periodo de tiempo se llama la ventana de tiempo legal. El valor de t_0 no está definido en esta Recomendación | Norma Internacional.

La información de este campo está destinada a dispositivos como remultiplexadores que pueden necesitar esta información para reconstruir el estado de las memorias tampón MB_n .

piecewise_rate (velocidad por piezas) – El significado de este campo de 22 bits sólo está definido cuando tanto la bandera de ventana de tiempo legal (*ltw_flag*) como la bandera válida de ventana de tiempo legal (*ltw_valid_flag*) están puestas a '1'. Cuando está definido, es un entero positivo que especifica una velocidad binaria hipotética R que se utiliza para definir los tiempos extremos de las ventanas de tiempo legal de paquetes de tren de transporte del mismo PID que siguen a este paquete pero que no incluyen el campo de desplazamiento de ventana de tiempo legal (*legal_time_window_offset*).

Supóngase que el primer byte de este paquete de tren de transporte y de los N paquetes de tren de transporte siguientes del mismo PID tienen los índices $A_i, A_{i+1}, \dots, A_{i+N}$, respectivamente, y que los últimos paquetes no tienen un valor codificado en el campo de desplazamiento de ventana de tiempo legal. Entonces, los valores $t_1(A^{i+j})$ se determinarán por la expresión:

$$t_1(A_{i+j}) = t_1(A_i) + j \times 188 \times 8 \text{ bits / byte / } R$$

donde j toma valores de 1 a N .

Todos los paquetes entre éste y el siguiente del mismo PID, para incluir un campo de desplazamiento de ventana de tiempo legal, deberán tratarse si tuvieran el valor:

$$offset = t_1(A_i) - t(A_i)$$

correspondiente al valor $t_1(\cdot)$, calculado por la fórmula anterior, y codificado en el campo de desplazamiento de ventana de tiempo legal. $t(j)$ es el tiempo de llegada del byte j al T-STD.

El significado de este campo no está definido cuando está presente en un paquete de tren de transporte que no tiene el campo de desplazamiento de ventana de tiempo legal.

splice_type (tipo de empalme) – Es un campo de 4 bits. Desde la primera aparición de este campo, tendrá el mismo valor en todos los paquetes de tren de transporte subsiguientes del mismo PID en que esté presente, hasta que llegue el paquete en el que la cuenta atrás de empalme alcanza el valor 0 (incluido el paquete en cuestión). Si el tren elemental transportado en ese PID no es un tren vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2, este campo tendrá el valor '0000'. Si el tren elemental transportado en ese PID es un tren de vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2, este campo indica las condiciones que ha de cumplir ese tren elemental con fines de empalme. Estas condiciones se describen como una función del perfil, el nivel y el tipo de empalme en los cuadros 2-7 a 2-20.

En estos cuadros, un valor para 'splice_decoding_delay' y 'max_splice_rate' significa que el tren elemental de vídeo debe cumplir las siguientes condiciones:

- 1) El último byte de la imagen codificada que termina en el paquete de tren de transporte en el cual la cuenta atrás de empalme llega a cero permanecerá en la memoria tampón VBV del modelo VBV durante un tiempo igual a (*splice_decoding_delay* $t_{n+1} - t_n$), donde a los efectos de esta subcláusula:
 - n es el índice del final de la imagen codificada en el paquete de tren de transporte en el que *splice_countdown* llega a cero, es decir, la imagen codificada referida anteriormente.
 - t_n se define en C.3.1 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.
 - ($t_{n+1} - t_n$) se define en C.9 a C.12 de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

NOTA – t_n es el instante en el que se saca la imagen codificada n de la memoria tampón VBV y ($t_{n+1} - t_n$) el plazo durante el que se presenta la imagen n .

- 2) La memoria tampón VBV del modelo VBV no desbordará si su entrada es conmutada, en el punto de empalme, a un tren de velocidad constante igual a 'max_splice_rate' durante un tiempo igual a 'splice_decoding_delay'.

Cuadro 2-7 – Tabla 1 de parámetros de empalme
Nivel principal de perfil simple, nivel principal de perfil principal, nivel principal de perfil SNR (ambas capas), nivel alto-1440 de perfil espacial (capa básica), nivel principal de perfil alto (capas media + básica), nivel principal de perfil multivisión (capa básica) – Vídeo

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $15,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 150 ms; max_splice_rate = $12,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 225 ms; max_splice_rate = $8,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $7,2 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

Cuadro 2-8 – Tabla 2 de parámetros de empalme
Nivel bajo de perfil principal, nivel bajo de perfil SNR (ambas capas), nivel principal de perfil alto (capa básica), nivel bajo de perfil multivisión (capa básica) – Vídeo

[splice_type]	Condiciones
0000	splice_decoding_delay = 115 ms; max_splice_rate = $4,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 155 ms; max_splice_rate = $3,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 230 ms; max_splice_rate = $2,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $1,8 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

Cuadro 2-9 – Tabla 3 de parámetros de empalme
Nivel alto-1440 de perfil principal, nivel alto-1440 de perfil espacial (todas las capas), nivel alto-1440 de perfil alto (capas media + básica), nivel alto-1440 de perfil multivisión (capa básica) – Vídeo

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $60,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = $45,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $30,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $28,5 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

Cuadro 2-10 – Tabla 4 de parámetros de empalme
Nivel alto de perfil principal, nivel alto-1440 de perfil alto (todas las capas), nivel alto de perfil alto (capas media + básica), nivel alto de perfil multivisión (capa básica) – Vídeo

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $80,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = $60,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $40,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $38,0 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-11 – Tabla 5 de parámetros de empalme
Nivel bajo de perfil SNR (capa básica) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 115 ms; max_splice_rate = $3,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 175 ms; max_splice_rate = $2,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 250 ms; max_splice_rate = $1,4 \times 10^6$ bit/s
0011-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-12 – Tabla 6 de parámetros de empalme
Nivel principal de perfil SNR (capa básica) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 115 ms; max_splice_rate = $10,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 145 ms; max_splice_rate = $8,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 235 ms; max_splice_rate = $5,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $4,7 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-13 – Tabla 7 de parámetros de empalme
Nivel alto-1440 de perfil espacial (capas media + básica) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $40,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = $30,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $20,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $19,0 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-14 – Tabla 8 de parámetros de empalme
Nivel principal de perfil alto (todas las capas)
Nivel alto-1440 de perfil alto (capa básica) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $20,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = $15,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $10,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $9,5 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-15 – Tabla 9 de parámetros de empalme
Nivel alto de perfil alto (capa básica), nivel principal de perfil multivisión (ambas capas) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $25,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 165 ms; max_splice_rate = $18,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 250 ms; max_splice_rate = $12,0 \times 10^6$ bit/s
0011-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-16 – Tabla 10 de parámetros de empalme
Nivel alto de perfil alto (todas las capas),
nivel alto-1440 de perfil multivisión (ambas capas) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $100,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 160 ms; max_splice_rate = $75,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $50,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $48,0 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-17 – Tabla 11 de parámetros de empalme
Nivel principal de perfil 4:2:2 – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 45 ms; max_splice_rate = $50,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 90 ms; max_splice_rate = $50,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 180 ms; max_splice_rate = $50,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 225 ms; max_splice_rate = $40,0 \times 10^6$ bit/s
0100	splice_decoding_delay = 250 ms; max_splice_rate = $36,0 \times 10^6$ bit/s
0101-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-18 – Tabla 12 de parámetros de empalme
Nivel bajo de perfil multivisión (ambas capas) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 115 ms; max_splice_rate = $8,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 155 ms; max_splice_rate = $6,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 230 ms; max_splice_rate = $4,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $3,7 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-19 – Tabla 13 de parámetros de empalme
Nivel alto de perfil multivisión (ambas capas) – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 120 ms; max_splice_rate = $130,0 \times 10^6$ bit/s
0001	splice_decoding_delay = 150 ms; max_splice_rate = $104,0 \times 10^6$ bit/s
0010	splice_decoding_delay = 240 ms; max_splice_rate = $65,0 \times 10^6$ bit/s
0011	splice_decoding_delay = 250 ms; max_splice_rate = $62,4 \times 10^6$ bit/s
0100-1011	Reservado
1100-1111	Definido por el usuario

**Cuadro 2-20 – Tabla 14 de parámetros de empalme
Nivel alto de perfil 4:2:2 – Vídeo**

[splice_type]	Condiciones
0000	splice_decoding_delay = 45 ms; max_splice_rate = 300,0 × 10 ⁶ bit/s
0001	splice_decoding_delay = 90 ms; max_splice_rate = 300,0 × 10 ⁶ bit/s
0010-0011	Reserved
0100	splice_decoding_delay = 250 ms; max_splice_rate = 180,0 × 10 ⁶ bit/s
0101-1011	Reservado
1100-1111	Definido por el usuario

DTS_next_AU (decoding_time_stamp_next_access_unit) (siguiente unidad de acceso de indicación de tiempo de decodificación) – Es un campo de 33 bits, codificado en tres partes. En el caso de decodificación continua y periódica a través de este punto de empalme, indica el tiempo de decodificación de la primera unidad de acceso que sigue al punto de empalme. Este tiempo de decodificación se expresa en la base de tiempo que es válida en el paquete de tren de transporte en el cual la cuenta atrás de empalme alcanza el valor cero. A partir de la primera aparición de este campo, tendrá el mismo valor en todos los paquetes de tren de transporte subsiguientes del mismo PID en el que está presente, hasta el paquete en el cual la cuenta atrás de empalme llega a cero (incluido el paquete en cuestión).

stuffing_byte (octeto de relleno) – Es un valor de 8 bits fijo igual a '1111 1111' que puede ser insertado por el codificador. Es descartado por el decodificador.

2.4.3.6 Paquete PES

Véase el cuadro 2-21.

Cuadro 2-21 – Paquete PES

Sintaxis	N.º de bits	Mnemónico
PES_packet() {		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if (stream_id != program_stream_map && stream_id != padding_stream && stream_id != private_stream_2 && stream_id != ECM && stream_id != EMM && stream_id != program_stream_directory && stream_id != DSMCC_stream && stream_id != ITU-T Rec. H.222.1 type E stream) {		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
Copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsbf
if (PTS_DTS_flags == '10') {		
'0010'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
}		

Cuadro 2-21 – Paquete PES

Sintaxis	N.º de bits	Mnemónico
if (PTS_DTS_flags == '11') {		
'0011'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (ESCR_flag == '1') {		
reserved	2	bslbf
ESCR_base[32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base[29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base[14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if (ES_rate_flag == '1') {		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		
if (DSM_trick_mode_flag == '1') {		
trick_mode_control	3	uimsbf
if (trick_mode_control == fast_forward) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	5	uimsbf
}		
else if (trick_mode_control == freeze_frame) {		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if (trick_mode_control == fast_reverse) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	5	uimsbf
}		
Else		
reserved	5	bslbf
}		
if (additional_copy_info_flag == '1') {		
marker_bit	1	bslbf
additional_copy_info	7	bslbf
}		
if (PES_CRC_flag == '1') {		
previous_PES_packet_CRC	16	bslbf
}		
if (PES_extension_flag == '1') {		
PES_private_data_flag	1	bslbf
pack_header_field_flag	1	bslbf
program_packet_sequence_counter_flag	1	bslbf
P-STD_buffer_flag	1	bslbf
reserved	3	bslbf
PES_extension_flag_2	1	bslbf

Cuadro 2-21 – Paquete PES

Sintaxis	N.º de bits	Mnemónico
if (PES_private_data_flag == '1') { PES_private_data }	128	bslbf
if (pack_header_field_flag == '1') { pack_field_length pack_header() }	8	uimsbf
if (program_packet_sequence_counter_flag == '1') { marker_bit program_packet_sequence_counter marker_bit MPEG1_MPEG2_identifier original_stuff_length }	1 7 1 1 6	bslbf uimsbf bslbf bslbf uimsbf
if (P-STD_buffer_flag == '1') { '01' P-STD_buffer_scale P-STD_buffer_size }	2 1 13	bslbf bslbf uimsbf
if (PES_extension_flag_2 == '1') { marker_bit PES_extension_field_length stream_id_extension_flag If (stream_id_extension_flag == '0') { stream_id_extension for (i = 0; i < PES_extension_field_length; i++){ reserved } } }	1 7 1 7 8	bslbf uimsbf bslbf uimsbf bslbf
for (i < 0; i < N1; i++) { stuffing_byte }	8	bslbf
for (i < 0; i < N2; i++) { PES_packet_data_byte }	8	bslbf
else if (stream_id == program_stream_map stream_id == private_stream_2 stream_id == ECM stream_id == EMM stream_id == program_stream_directory stream_id == DSMCC_stream stream_id == ITU-T Rec. H.222.1 type E stream) { for (i = 0; i < PES_packet_length; i++) { PES_packet_data_byte } }	8	bslbf
else if (stream_id == padding_stream) { for (i < 0; i < PES_packet_length; i++) { padding_byte } }	8	bslbf

2.4.3.7 Definición semántica de campos en el paquete PES

packet_start_code_prefix (prefijo de código de comienzo de paquete) – Es un código de 24 bits. Junto con el identificador de tren que sigue, constituye un código de comienzo de paquete que identifica el principio de un paquete. El prefijo de código de comienzo de paquete es la cadena '0000 0000 0000 0000 0000 0001' (0x0000001).

stream_id (identificador de tren) – En trenes de programas, el identificador de tren especifica el tipo y número del tren elemental de acuerdo con las asignaciones de identificador de tren indicadas en el cuadro 2-22. En los trenes de transporte, el identificador de tren puede ponerse a cualquier valor válido que describa correctamente el tipo de tren elemental definido en el cuadro 2-22. En los trenes de transporte, el tipo de tren elemental se da en la información específica de programa como se indica en 2.4.4.

PES_packet_length (longitud de paquetes PES) – Un campo de 16 bits que especifica el número de bytes en el paquete PES que sigue al último byte del campo. Un valor de 0 indica que la longitud del paquete PES no está especificada ni limitada y está autorizada solamente en paquetes PES cuya cabida útil consiste en bytes de un tren elemental de vídeo contenido en paquetes del tren de transporte.

PES_scrambling_control (control de aleatorización) – El control de aleatorización de PES de 2 bits indica el modo de aleatorización de la cabida útil del paquete PES. Cuando la aleatorización se realiza en el nivel de PES, el encabezamiento del paquete PES, que incluye los campos facultativos cuando están presentes, no será aleatorizado (véase el cuadro 2-23).

Cuadro 2-22 – Asignaciones del identificador de tren (stream-id)

Stream_id	Nota	Codificación de tren
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		número de tren de audio x xxxx de ISO/CEI 13818-3 o ISO/CEI 11172-3 o ISO/CEI 13818-7 o ISO/CEI 14496-3
1110 xxxx		número de tren de vídeo xxxx de Rec. UIT-T H.262 ISO/CEI 13818-2, ISO/CEI 11172-2, ISO/CEI 14496-2 o Rec. UIT-T H.264 ISO/CEI 14496-10
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	Anexo A a la Rec. UIT-T H.222.0 ISO/CEI 13818-1 o ISO/CEI 13818-6_DSMCC_stream
1111 0011	2	ISO/CEI_13522_stream
1111 0100	6	Rec. UIT-T H.222.1 tipo A
1111 0101	6	Rec. UIT-T H.222.1 tipo B
1111 0110	6	Rec. UIT-T H.222.1 tipo C
1111 0111	6	Rec. UIT-T H.222.1 tipo D
1111 1000	6	Rec. UIT-T H.222.1 tipo E
1111 1001	7	ancillary_stream
1111 1010		ISO/CEI 14496-1_SL-packetized_stream
1111 1011		ISO/CEI 14496-1_FlexMux_stream
1111 1100		tren metadatos
1111 1101	8	extended_stream_id
1111 1110		tren de datos reservado
1111 1101	4	program_stream_directory

El símbolo x significa que se permite el valor '0' o el valor '1', y que ambos dan por resultado el mismo tipo de tren. El número de tren viene dado por los valores tomados por las x.

NOTA 1 – Los paquetes PES de tipo program_stream_map tienen una sintaxis única que se especifica en 2.5.4.1.

NOTA 2 – Los paquetes PES de tipo private_stream_1 e ISO/CEI_13522_stream siguen la misma sintaxis de paquete PES que los correspondientes a trenes de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2 y trenes de audio de ISO/CEI 13818-3.

NOTA 3 – Los paquetes PES de tipo private_stream_2, ECM_stream y EMM_stream son similares a private_stream_1, salvo que no se especifica sintaxis después del campo PES_packet_length.

NOTA 4 – Los paquetes PES de tipo program_stream_directory tienen una sintaxis única, que se especifica en 2.5.5.

NOTA 5 – Los paquetes PES de tipo DSM-CC_stream tienen una sintaxis única que se especifica en ISO/CEI 13818-6.

NOTA 6 – Este identificador de tren está asociado con el tipo de tren 0x09 en el cuadro 2-34.

NOTA 7 – Este identificador de tren sólo se utiliza en paquetes PES que transportan datos de un tren de programa o de un tren de sistema de ISO/CEI 11172-1, en un tren de transporte (véase 2.4.3.8).

NOTA 8 – La utilización del identificador de tren 0xFD (extended_stream_id) indica que este paquete PES emplea una sintaxis ampliada que puedan identificarse otros tipos de tren.

Cuadro 2-23 – Valores de control de aleatorización de PES

Valor	Descripción
00	No aleatorizado
01	Definido por el usuario
10	Definido por el usuario
11	Definido por el usuario

PES_priority (prioridad PES) – Es un campo de 1 bit que indica la prioridad de la cabida útil en este paquete PES. Un '1' indica una prioridad más alta de la cabida útil del paquete PES que una cabida útil del paquete PES con este campo puesto a '0'. Un multiplexor puede utilizar el bit de prioridad PES para dar prioridad a sus datos dentro de un tren elemental. Este campo no será cambiado por el mecanismo de transporte.

data_alignment_indicator (indicador de alineación de datos) – Es una bandera de 1 bit. Cuando se pone a '1' indica que la cabecera del paquete PES va seguido inmediatamente por el elemento de sintaxis de vídeo o la palabra de sincronización de audio indicados en el descriptor de alineación del tren de datos en 2.6.10, si este descriptor estuviese presente. Si está puesto a '1' y el descriptor no está presente, se requiere la alineación indicada en el tipo de alineación '01' de los cuadros 2-53, 2-54 ó 55. Cuando se pone a '0' no está definido que se produzca o no la alineación.

copyright (derechos de autor) – Es un campo de 1 bit. Cuando está puesto a '1' indica que el material de la cabida útil de paquete PES asociado está protegido por derechos de autor. Cuando está puesto a '0' no está definido si el material está o no protegido por derechos de autor. Un descriptor de derechos de autor (véase 2.6.24) está asociado con el tren elemental que contiene este paquete PES y la bandera de derechos de autor se pone a '1'; el descriptor se aplica al material contenido en este paquete PES.

original_or_copy (original o copia) – Es un campo de 1 bit. Si se pone a '1', el contenido de la cabida útil del paquete PES asociado es un original. Cuando se pone a '0' indica que dicho contenido es una copia.

PTS_DTS_flags (banderas PTS_DTS) – Es un campo de 2 bits. Si está puesto a '10', los campos PTS estarán presentes en el encabezamiento de paquete PES. Si el campo de banderas PTS_DTS está puesto a '11', tanto el campo PTS como el campo DTS estarán presentes en el encabezamiento de paquete PES. Si el campo de banderas PTS_DTS está puesto a '00', no estarán presentes en el encabezamiento de paquete PES el campo PTS, ni el campo DTS. El valor '01' está prohibido.

ESCR_flag (bandera ESCR) – Una bandera de 1 bit que cuando está puesta a '1' indica que los campos de base y extensión ESCR están presentes en el encabezamiento de paquete PES. Cuando está puesta a '0' indica que no están presentes campos ESCR.

ES_rate_flag (bandera de velocidad ES) – Una bandera de 1 bit que cuando esta puesta a '1' indica que el campo de velocidad ES está presente en el encabezamiento de paquete PES. Cuando está puesta a '0' indica que no está presente el campo de velocidad ES.

DSM_trick_mode_flag (bandera de modo truco DSM) – Una bandera de 1 bit, que cuando está puesta a '1' indica la presencia de un campo de modo truco de 8 bits. Cuando está puesta a '0' indica que este campo no está presente.

additional_copy_info_flag (bandera de información de copia adicional) – Una bandera de 1 bit, que cuando está puesta a '1' indica la presencia del campo de información de copia adicional. Cuando está puesta a '0' indica que este campo no está presente.

PES_CRC_flag (bandera PES_CRC) – Una bandera de 1 bit, que cuando está puesta a '1' indica que el campo CRC está presente en el paquete PES. Cuando está puesta a '0' indica que este campo no está presente.

PES_extension_flag (bandera de extensión PES) – Una bandera de 1 bit, que cuando está puesta a '1' indica que existe un campo de extensión en el encabezamiento de paquete PES. Cuando está puesta a '0' indica que este campo no está presente.

PES_header_data_length (longitud de datos de encabezamiento PES) – Un campo de 8 bits que especifica el número total de bytes ocupado por los campos facultativos y cualesquiera bytes de relleno contenidos en el encabezamiento de paquetes PES. La presencia de campos facultativos se indica en el byte que precede al campo de longitud de datos de encabezamiento PES.

marker_bit (bit marcador) – Un campo de 1 bit que tiene el valor '1'.

PTS (presentation_time_stamp) (indicación de tiempo de presentación) – Los tiempos de presentación se relacionarán con los tiempos de decodificación de la manera siguiente: la PTS es un número de 33 bits codificado en tres campos distintos. Indica el tiempo de presentación, $tp_n(k)$, en el decodificador-objetivo de sistema, de una unidad de

presentación k del tren elemental n . El valor de PTS se especifica en unidades del periodo de la frecuencia de reloj de sistema dividida por 300 (lo que da 90 kHz). El tiempo de presentación se deriva de la PTS de acuerdo con la ecuación 2-11 siguiente. Véase 2.7.4 sobre las constricciones impuestas a la frecuencia de codificación de las indicaciones de tiempo de presentación.

$$PTS(k) = ((system_clock_frequency \times t_{pn}(k)) DIV 300) \% 2^{33} \quad (2-11)$$

donde $t_{pn}(k)$ es el tiempo de presentación de la unidad de presentación $P_n(k)$.

En el caso de audio, si una PTS está presente en el encabezamiento de paquete PES se referirá a la primera unidad de acceso que comienza en el paquete PES. Una unidad de acceso de audio comienza en un paquete PES si el primer byte de la unidad de acceso de audio está presente en el paquete PES.

En el caso de vídeo conforme a ISO/CEI 11172-2, o ISO/CEI 14496-2, si una cabecera de paquete PES contiene un PTS, éste se referirá a la unidad de acceso que contiene el primer código de comienzo de imagen que empieza en este paquete PES. Un código de comienzo de imagen empieza en un paquete PES si éste contiene el primer byte del código de comienzo de imagen. Para imágenes I/P en secuencias distintas de `low_delay` y en caso de que no haya discontinuidad de decodificación entre unidades de acceso (AU) k y k' , el tiempo de presentación $t_{pn}(k)$ será igual al tiempo de decodificación $t_{dn}(k')$ de la siguiente imagen I o P transmitida (véase 2.7.5). Si hubiera una discontinuidad de decodificación, o el tren termina, la diferencia entre $t_{pn}(k)$ y $t_{dn}(k)$ será la misma que si el tren original hubiese continuado sin una discontinuidad y sin terminar.

NOTA 1 – Una secuencia `low_delay` (de poco retardo) es una secuencia de vídeo conforme a la ISO/CEI 14496-2 en la cual la bandera `low_delay` está puesta a '1' (véase 6.2.3 de ISO/CEI 14496-2).

En el caso de vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2, si una cabecera de paquete PES contiene un PTS, éste se referirá a la unidad de acceso que contiene el primer código de comienzo de imagen que empieza en este paquete PES. Un código de comienzo de imagen empieza en un paquete PES si éste contiene el primer byte del código de comienzo de imagen. Para cuadros codificados I/P en secuencias distintas de `low_delay` y en caso de que no haya discontinuidad de decodificación entre unidades de acceso (AU) k y k' , el tiempo de presentación $t_{pn}(k)$ será igual al tiempo de decodificación $t_{dn}(k')$ de la siguiente imagen I o P transmitida (véase 2.7.5). Si hay una discontinuidad de decodificación, o el tren termina, la diferencia entre $t_{pn}(k)$ y $t_{dn}(k)$ será la misma que si el tren original hubiese continuado sin una discontinuidad y sin terminar.

NOTA 2 – Una secuencia `low_delay` (de poco retardo) es una secuencia de vídeo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2 en la cual la bandera `low_delay` está puesta a '1' (véase 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2). Obsérvese asimismo que en caso de imágenes de campo el tiempo de presentación se refiere a la primera imagen de campo del cuadro codificado.

Para vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10, si la cabecera de paquete PES contiene una PTS, ésta se referirá a la primera unidad de acceso AVC que comienza en este paquete PES. Una unidad de acceso AVC comienza en un paquete PES si éste contiene el primer byte de una unidad de acceso AVC. Para lograr la coherencia entre el modelo STD y el modelo HRD definido en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10, el valor PTS para cada unidad de acceso AVC decodificada en el STD indicará, con la precisión de sus respectivos relojes, el mismo instante que el tiempo de salida DPB nominal en el HRD, definido éste como $t_{o,n,dpb}(n) = t_{r,n}(n) + t_c * dpb_output_delay(n)$, siendo $t_{r,n}(n)$, t_c , y $dpb_output_delay(n)$ los definidos en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10.

NOTA 3 – Es posible utilizar diferentes relojes para obtener la PTS y $t_{o,n,dpb}(n)$.

El tiempo de presentación $t_{pn}(k)$ será igual al tiempo de decodificación $t_{dn}(k)$ para:

- unidades de acceso a audio;
- unidades de acceso en secuencias vídeo de poco retardo conforme a la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 14496-2;
- imágenes B en trenes de vídeo ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 14496-2.

Si hay filtrado en audio, el modelo del sistema supone que el filtrado no introduce retardo, por lo que la muestra referida por PTS en la codificación es la misma muestra referida por PTS en la decodificación. En caso de codificación escalable, véase 2.7.6.

DTS (decoding_time_stamp) (indicación de tiempo de codificación) – La DTS es un número de 33 bits codificado en tres campos distintos. Indica el tiempo de codificación, $td_n(j)$, en el decodificador-objetivo de sistema, de una unidad de acceso j del tren elemental n . El valor de DTS se especifica en unidades del periodo de la frecuencia de reloj de sistema

ISO/CEI 13818-1:2007 (S)

dividida por 300 (lo que da 90 kHz). El tiempo de codificación se deriva de la DTS de acuerdo con la ecuación 2-12 siguiente:

$$DTS(j) = ((system_clock_frequency \times td_n(j)) DIV 300) \% 2^{33} \quad (2-12)$$

donde $td_n(j)$ es el tiempo de codificación de la unidad de acceso $A_n(j)$.

En el caso de vídeo conforme a ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2, o ISO/CEI 14496-2, si una cabecera de paquete PES contiene una DTS, ésta se referirá a la unidad de acceso que contiene el primer código de comienzo de imagen que empieza en este paquete PES. Un código de comienzo de imagen empieza en un paquete PES si éste contiene el primer byte del código de comienzo de imagen.

Para vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10, si la cabecera de paquete PES contiene una DTS, éste se referirá a la primera unidad de acceso AVC que comienza en este paquete PES. Una unidad de acceso AVC comienza en un paquete PES si éste contiene el primer byte de una unidad de acceso AVC. Para lograr la coherencia entre el modelo STD y el modelo HRD definido en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10, el valor DTS para cada unidad de acceso AVC en el STD indicará, con la precisión de sus respectivos relojes, el mismo instante que el tiempo de extracción CPB nominal $t_{r,n}(n)$ en el HRD, según se define en el anexo C de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

NOTA 4 – Es posible utilizar diferentes relojes para la obtención de DTS y $t_{r,n}(n)$.

Para el caso de la codificación escalonable, véase 2.7.6.

ESCR_base (base de ESCR); ESCR_extension (extensión de ESCR) – La referencia de reloj de tren elemental es un campo de 42 bits codificado en dos partes. La primera parte, ESCR_base, es un campo de 33 bits cuyo valor viene dado por ESCR_base(i) en la ecuación 2-14. La segunda parte, ESCR_extension, es un campo de 9 bits cuyo valor se da por ESCR_ext(i) en la ecuación 2-15. El campo ESCR indica el tiempo previsto de llegada del byte que contiene el último bit de ESCR_base a la entrada del PES-STD para trenes PES (véase 2.5.2.4).

Específicamente:

$$ESCR(i) = ESCR_base(i) \times 300 + ESCR_ext(i) \quad (2-13)$$

donde:

$$ESCR_base(i) = ((system_clock_frequency \times t(i)) DIV 300) \% 2^{33} \quad (2-14)$$

$$ESCR_ext(i) = ((system_clock_frequency \times t(i)) DIV 1) \% 300 \quad (2-15)$$

Los campos ESCR y ES_rate (véase la semántica que sigue inmediatamente) contienen información de temporización relativa a la secuencia de trenes PES. Estos campos satisfarán las constricciones definidas en 2.7.3.

ES_rate (elementary_stream_rate) (velocidad de tren elemental) – El campo de velocidad ES es un entero sin signo, con una longitud de 22 bits, que especifica la velocidad a la cual el decodificador objetivo del sistema recibe bytes del paquete PES en el caso de un tren PES. La velocidad ES es válida en el paquete PES en el cual está incluida y en los paquetes PES subsiguientes del mismo tren PES hasta que se encuentra un nuevo campo de velocidad ES. El valor de la velocidad ES se mide en unidades de 50 bytes/segundo. El valor 0 está prohibido. El valor de la velocidad ES se utiliza para definir el tiempo de llegada de bytes a la entrada de un P-STD para trenes PES definidos en 2.5.2.4. El valor codificado en el campo de velocidad ES varía de un paquete PES a otro.

trick_mode_control (control de modo truco) – Un campo de 3 bits que indica que el modo truco se aplica al tren de vídeo asociado. En los casos de otros tipos de trenes elementales, el significado de este campo y el de los cinco bits siguientes no están definidos. Para la definición del estado del modo truco, véase 2.4.2.3 referente al **modo truco**.

Cuando el estado del modo truco es falso, el número de veces, N, que una imagen aparece a la salida del proceso de decodificación en el caso de secuencias progresivas se especifica para cada imagen por los campos repeat_first_field y top_field_first en el caso de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2, y se especifica mediante el encabezamiento de secuencia en el caso de vídeo de ISO/CEI 11172-2.

En el caso de secuencias entrelazadas, cuando el estado del modo truco es falso, el número de veces, N, que una imagen aparece a la salida del proceso de decodificación de secuencias progresivas se especifica para cada imagen por los campos repeat_first_field y progressive_frame en el caso de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2.

Cuando el estado del modo truco es verdadero, el número de veces que debe visualizarse una imagen depende de N.

Cuando el valor de este campo cambia o cuando deja de funcionar el modo truco, puede aparecer cualquier combinación de las siguientes discontinuidades:

- discontinuidad en la base de tiempo;
- discontinuidad en la codificación;
- discontinuidad en el contador de continuidad.

Cuadro 2-24 – Valores de control de modo truco

Valor	Descripción
'000'	Rápido hacia adelante
'001'	Cámara lenta
'010'	Trama congelada
'011'	Rápido hacia atrás
'100'	Lento hacia atrás
'101'-'111'	Reservado

En el contexto de modo truco, la velocidad no normal de decodificación y de presentación puede provocar que sean incorrectos los valores de determinados campos definidos en los datos de tren elemental de vídeo. De manera similar, pueden ser válidas las constricciones semánticas impuestas a la estructura de rebanada. Los elementos de sintaxis vídeo a los cuales se aplica esta excepción son:

- bit_rate;
- vbv_delay;
- repeat_first_field;
- v_axis_positive;
- field_sequence;
- subcarrier;
- burst_amplitude;
- subcarrier_phase.

Un decodificador no puede depender de los valores codificados en estos campos cuando está en el modo truco.

Los decodificadores no están normativamente obligados a decodificar el campo de control del modo truco. Sin embargo, todo decodificador que decodifique el campo de control del modo truco deberá cumplir los siguientes requisitos normativos.

fast forward (rápido hacia adelante) – El valor '000' en el campo de control de modo truco, cuando está presente, indica un tren vídeo hacia adelante rápido, y define el significado de los cinco bits siguientes en el encabezamiento de paquete PES. El bit de refresco de rebanada intracodificada puede estar puesto a '1', lo que indica que puede haber macrobloques faltantes que el decodificador puede sustituir con macrobloques cúbicados de imágenes previamente decodificadas. El campo id de campo, definido en el cuadro 2-25, indica qué campos deben ser visualizados. El campo de truncamiento de frecuencia indica que puede estar incluido un conjunto de coeficientes. El significado de los valores en este campo se muestra en el cuadro 2-26.

slow motion (cámara lenta) – El valor '001' en el campo de control de modo truco. Cuando este valor está presente, indica un tren vídeo de cámara lenta y define el significado de los cinco bits siguientes en el encabezamiento de paquete PES. En el caso de secuencias progresivas, la imagen debe visualizarse $N \times \text{rep_cntrl}$ veces, donde N se define como se ha indicado antes.

En el caso de secuencias progresivas de vídeo ISO/CEI 11172-2 y Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de imagen de $N \times \text{rep_cntrl}$.

En el caso de secuencias entrelazadas Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de campo de $N \times \text{rep_cntrl}$. Si la imagen es una imagen de trama, el primer campo a visualizar es el campo superior si top_field_first es '1', y el campo inferior si top_field_first es '0' (véase la Rec. UIT-T H.262 | ISO/CEI 13818-2). Este campo se visualiza durante una duración de campo de $N \times \text{rep_cntrl} / 2$. El otro campo de la imagen se visualiza entonces por una duración de campo de $N - N \times \text{rep_cntrl} / 2$.

freeze frame (trama congelada) – El valor '010' en el campo de control de modo truco. Cuando este valor está presente, indica un tren de vídeo de trama congelada y define el significado de los cinco bits siguientes del encabezamiento de paquete PES. El campo `field_id` (identificador de campo) se define en el cuadro 2-25, que identifica el campo o campos que deben ser visualizados. El campo de identificador de campo se refiere a la primera unidad de acceso vídeo que comienza en el paquete PES que contiene el campo de identificador de campo a menos que el paquete PES contenga cero bytes de cabida útil. En este último caso, el campo de identificador de campo se refiere a la unidad de acceso vídeo previa más reciente.

fast_reverse (rápido hacia atrás) – El valor '011' en el campo de control de modo truco. Cuando este valor está presente, indica un tren de vídeo inverso rápido y define el significado de los cinco bits siguientes del encabezamiento de paquete PES. El bit de refresco de rebanada intracodificada se puede poner a '1', lo que indica que puede haber macrobloques faltantes que el decodificador puede sustituir con macrobloques coubicados de las imágenes decodificadas previamente. El campo de identificador de campo, definido en el cuadro 2-25, indica qué campo o campos deben ser visualizados. El campo de truncamiento de frecuencia indica que se puede incluir un conjunto de coeficientes restringidos. El significado de los valores de este campo se muestra en el cuadro 2-26.

slow_reverse (lento hacia atrás) – El valor '100' en el campo de control de modo truco. Cuando este valor está presente en un tren de vídeo inverso lento y define el significado de los cinco bits siguientes del encabezamiento de paquete PES. En el caso de secuencias progresivas de vídeo ISO/CEI 11172-2 y Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de imagen de $N \times \text{rep_cntrl}$, donde N tiene el valor definido más arriba.

En el caso de secuencias entrelazadas Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de campo de $N \times \text{rep_cntrl}$. Si la imagen es una imagen de trama, el primer campo a visualizar es el campo inferior si `top_field_first` es '1', y el campo superior si `top_field_first` es '0' (véase la Rec. UIT-T H.262 | ISO/CEI 13818-2). Este campo se visualiza por una duración de campo de $N \times \text{rep_cntrl} / 2$. El otro campo de la imagen se visualiza entonces por una duración de campo de $N - N \times \text{rep_cntrl} / 2$.

field_id (identificador de campo) – Un campo de 2 bits que indica qué campo o campos se han de visualizar. Se codifica de acuerdo con el cuadro 2-25.

Cuadro 2-25 – Valores de control de campo de identidad de campo

Valor	Descripción
'00'	Visualización del campo superior solamente
'01'	Visualización del campo inferior solamente
'10'	Visualización de la trama completa
'11'	Reservado

intra_slice_refresh (refresco de rebanada intracodificada) – Una bandera de un bit que cuando se pone a '1' indica que pueden faltar macrobloques entre rebanadas codificadas de datos vídeo en este paquete PES. Cuando se pone a '0' esto puede no suceder. Para más información, véase la Rec. UIT-T H.262 | ISO/CEI 13818-2. El decodificador puede sustituir macrobloques que faltan con macrobloques coubicados de imágenes decodificadas previamente.

frequency_truncation (truncamiento de frecuencia) – Un campo de 2 bits que indica que se puede haber utilizado un conjunto de coeficientes restringidos en la codificación de los datos vídeo en este paquete PES. Los valores se definen en el cuadro 2-26.

Cuadro 2-26 – Valores de selección de coeficientes

Valor	Descripción
'00'	Sólo los coeficientes DC no son cero
'01'	Sólo los tres primeros coeficientes no son cero
'10'	Sólo los seis primeros coeficientes no son cero
'11'	Todos los coeficientes pueden no ser cero

rep_cntrl (control de repetición) – Un campo de 5 bits que indica el número de veces que debe visualizarse cada campo en una imagen entrelazada, o el número de veces que debe visualizarse una imagen progresiva. El hecho de que, en las imágenes entrelazadas, el primer campo que se visualice sea el campo superior o el inferior, es una función del campo de control de modo truco y del bit `top_field_first` del encabezamiento de la secuencia de vídeo. El valor '0' está prohibido.

additional_copy_info (información de copia adicional) – Este campo de 7 bits contiene datos privados relativos a la información de derechos de autor.

previous_PES_packet_CRC (CRC de paquete PES anterior) – Es un campo de 16 bits que contiene el valor CRC que da una salida de ceros de los 16 registradores en el decodificador similar al definido en el anexo A, pero con el polinomio:

$$x^{16} + x^{12} + x^5 + 1$$

después del procesamiento de los bytes de datos del paquete PES anterior, excluido el encabezamiento PES.

NOTA 5 – Esta CRC está destinada a ser utilizada en el mantenimiento de la red de modo que aisle la fuente de errores intermitentes. No está prevista su utilización por decodificadores de trenes elementales. Se calcula solamente en los bytes de datos porque los datos del encabezamiento del paquete PES pueden ser modificados durante el transporte.

PES_private_data_flag (bandera de datos privados PES) – Una bandera de 1 bit que cuando se pone a '1' indica que el encabezamiento del paquete PES contiene datos privados. Cuando se pone a '0', indica que no hay datos privados presentes en el encabezamiento PES.

pack_header_field_flag (bandera de campo de encabezamiento de saca) – Una bandera de 1 bit que cuando se pone a '1' indica que un encabezamiento de saca de ISO/CEI 11172-1 o un encabezamiento de saca de tren de programa está almacenado en este encabezamiento de paquete PES. Si este campo está en un paquete PES que está contenido en un tren de programa, el campo se pondrá a '0'. En un tren de transporte, cuando está puesto a un valor '0', indica que no hay encabezamiento de saca en el encabezamiento PES.

program_packet_sequence_counter_flag (bandera de contador de secuencias de paquete de programa) – Una bandera de 1 bit que cuando se pone a '1' indica que están presentes los campos de contador de secuencias de paquete de programa, de identificador MPEG1_MPEG2, y de longitud de relleno original en el paquete PES. Cuando se pone a un valor '0', indica que estos campos no están presentes en el encabezamiento PES.

P-STD_buffer_flag (bandera de memoria tampón P-STD) – Una bandera de 1 bit que cuando se pone a '1' indica que P-STD_buffer_scale (escala de memoria tampón P-STD) y P-STD_buffer_size (tamaño de memoria tampón P-STD) están presentes en el encabezamiento de paquete PES. Cuando se pone a un valor '0', indica que estos campos no están presentes en el encabezamiento PES.

PES_extension_flag_2 (bandera de extensión PES 2) – Un campo de 1 bit que cuando está puesto a '1' indica la presencia del campo de longitud de campo de extensión de PES y cambios asociados. Cuando está puesto a un valor '0', indica que el campo de longitud de campo de extensión PES y cualesquiera campos asociados no están presentes.

PES_private_data (datos privados PES) – Es un campo de 16 bytes que contiene datos privados. Estos datos, combinados con los campos anterior y posterior, no emularán el prefijo de código de comienzo de paquete (0x000001).

pack_field_length (longitud de campo de saca) – Es un campo de 8 bits que indica la longitud en bytes, del campo de encabezamiento de saca.

program_packet_sequence_counter (contador de secuencias de paquete de programa) – El campo de contador de secuencias de paquete de programa es un campo de 7 bits. Es un contador facultativo que se incrementa con cada paquete PES sucesivo procedente de un tren de programa o de un tren de ISO/CEI 11172-1 o los paquetes PES asociados con una definición monoprograma en un tren de transporte, que proporciona la funcionalidad similar a un contador de continuidad (véase 2.4.3.2). Permite que una aplicación extraiga la secuencia de paquetes PES original de un tren de programas o la secuencia de paquetes original del tren de sistemas ISO/CEI 11172-1 original. El contador se reiniciará a cero después que alcanza su valor máximo. No habrá repetición de paquetes PES. En consecuencia, dos paquetes PES consecutivos en el múltiplex de programa no tendrán valores idénticos de contador de secuencias de paquete de programa.

MPEG1_MPEG2_identifier (identificador MPEG1_MPEG2) – Una bandera de 1 bit que cuando se pone a '1' indica que este paquete PES transporta información de un tren de sistema de ISO/CEI 11172-1. Cuando se pone a '0', indica que este paquete PES transporta información de un tren de programa.

original_stuff_length (longitud de relleno original) – Este campo de 6 bits especifica el número de bytes de relleno utilizados en el encabezamiento de paquete PES original de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 o en el encabezamiento de paquete original de ISO/CEI 11172-1.

P-STD_buffer_scale (escala de memoria tampón P-STD) – La escala de memoria tampón P-STD es un campo de 1 bit cuyo valor sólo está definido si este paquete PES está contenido en un tren de programa. Indica el factor de escalonamiento utilizado para interpretar el campo de tamaño de memoria tampón P-STD subsiguiente. Si el ID de tren precedente indica un tren de audio, la escala de memoria tampón P-STD tendrá el valor '0'. Si el identificador de tren precedente indica un tren de vídeo, la escala de memoria tampón P-STD tendrá el valor '1'. Para todos los otros tipos de trenes, el valor puede ser '1' o '0'.

P-STD_buffer_size (tamaño de memoria tampón P-STD) – Es un entero sin signo de 13 bits cuyo valor sólo está definido si este paquete PES está contenido en un tren de programa. Define el tamaño de la memoria tampón de entrada, BS_n , en el P-STD. Si la escala de memoria tampón P-STD tiene el valor '0', el tamaño de memoria tampón P-STD mide el tamaño de memoria tampón en unidades de 128 bytes. Si la escala de memoria tampón P-STD tiene el valor '1', el tamaño de memoria tampón P-STD mide el tamaño de la memoria tampón en unidades de 1024 bytes. De este modo:

$$\begin{aligned} &\text{si } (P-STD_buffer_scale == 0) \\ &BS_n = \overline{P-STD_buffer_size} \times 128 \end{aligned} \tag{2-16}$$

si no:

$$BS_n = P-STD_buffer_size \times 1024 \tag{2-17}$$

El valor codificado del tamaño de memoria tampón P-STD tiene efecto inmediatamente cuando el campo de tamaño de memoria tampón P-STD es recibido por el decodificador-objetivo de sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (véase 2.7.7).

El tamaño BS_n será mayor o igual que el tamaño del CPB señalado por el $CpbSize[cpb_cnt_minus1]$ y especificado en $hrd_parameters()$ de NAL en el tren de vídeo AVC. Si el tren de vídeo AVC no contiene $hrd_parameters()$ de NAL, el BS_n será mayor o igual que el tamaño del NAL CPB para el formato de tren de bytes definido en el anexo A a la Rec. UIT-T H.264 | ISO/CEI 14496-10 como $1200 \times MaxCPB$ para el nivel aplicado.

PES_extension_field_length (longitud de campo de extensión PES) – Este campo de 7 bits especifica la longitud, en bytes, de los datos que siguen a este campo en el campo de extensión PES, incluidos cualesquiera bytes reservados.

stream_id_extension_flag (bandera de extensión de indicador de tren) – Una bandera de 1 bit, que cuando está puesta a '0' indica la presencia del campo $stream_id_extension$ en el encabezamiento de paquete PES. El valor '1' para este indicador está reservado.

stream_id_extension (extensión de identificador de tren) – En los trenes de programa, el $stream_id_extension$ especifica el tipo y número del tren elemental y está definido en el cuadro 2-27. En los trenes de transporte, $stream_id_extension$ puede tener cualquier valor válido que describa correctamente el tipo de tren elemental definido en el cuadro 2-27. En los trenes de transporte, el tipo de tren elemental se especifica en la información específica de programa, conforme a 2.4.4. Obsérvese que este campo se utiliza como una extensión del $stream_id$ definido antes. Sólo se utilizará este campo cuando el valor de $stream_id$ sea 1111 1101.

Cuadro 2-27 – Atribuciones de extensión de identificador de tren (stream_id_extension)

stream_id_extension	Nota	Codificación de tren
000 0000	1	Tren de información de control IPMP
000 0001	2	Tren IPMP
000 0010 ... 011 1111		reserved_data_stream
100 0000 ... 111 1111		private_stream
NOTA 1 – Los paquetes PES que tienen $stream_id_extension$ 0b000 0000 (tren de información de control IPMP) tienen una sintaxis única especificada en ISO/CEI 13818-11 (IPMP en MPEG-2). NOTA 2 – Los paquetes PES que tienen $stream_id_extension$ 0b000 0001 (tren IPMP) tienen una sintaxis única especificada en ISO/CEI 13818-11 (IPMP en MPEG-2).		

stuffing_byte (byte de relleno) – Este es un valor de 8 bits fijo igual a '1111 1111' que puede ser insertado por el decodificador, por ejemplo, para satisfacer los requisitos del canal. Es descartado por el decodificador. No estarán presentes más de 32 bytes de relleno en un encabezamiento de paquete PES.

PES_packet_data_byte (byte de datos de paquete PES) – Los bytes de datos de paquete PES serán los bytes de datos contiguos del tren elemental indicado por el identificador del tren de paquete o PID. Cuando los datos del tren elemental se ajustan a la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 13818-3, los bytes de datos de los paquetes PES se ajustarán a los bytes de esta Recomendación | Norma Internacional. Se preservará el orden de bytes del tren elemental. El número de bytes de datos de paquete PES, N, es especificado por el campo de longitud de paquete PES. N será igual al valor indicado en la longitud de paquete PES menos el número de bytes entre el último byte del campo de longitud de paquete PES y el primer byte de datos de paquete PES.

En el caso de tren privado 1, tren privado 2, tren ECM o tren EMM, el contenido del campo de bytes de datos del paquete PES es definible por el usuario y no será especificado por UIT-T | ISO/CEI en el futuro.

padding_byte (byte de relleno) – Este es un valor de 8 bits fijo igual a '1111 1111'. Es descartado por el decodificador.

2.4.3.8 Transmisión de trenes de programa y de trenes de sistemas ISO/CEI 11172-1 en el tren de transporte

El tren de transporte tiene campos facultativos para el soporte de la transmisión de trenes de programa y de trenes de sistemas ISO/CEI 11172-1 de una manera que permita una reconstrucción simple del tren correspondiente en el decodificador.

Cuando se coloca un tren de programa en un tren de transporte, los paquetes PES del tren de programa con valores de stream_id de private_stream_1, vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 11172-2, y audio de ISO/CEI 13818-3 o ISO/CEI 11172-3, son llevados en paquetes de tren de transporte.

En el caso de estos paquetes PES, cuando se reconstruye el tren de programa en el decodificador del tren de transporte, los datos de paquetes PES se copian en el tren de programa que se está reconstruyendo.

En el caso de paquetes PES de trenes de programa con identificador de tren con valores de tabla de correspondencia de trenes de programa, tren de relleno, tren privado 2, tren ECM, EMM, DSM_CC, o directorio de trenes de programa, todos los bytes del paquete PES de tren de programa, excepto el prefijo de código de comienzo de paquete, se colocan en los campos de los bytes de datos de un nuevo paquete PES. El identificador de tren de este nuevo paquete PES tiene el valor de tren auxiliar (véase el cuadro 2-22). Este nuevo paquete es llevado entonces en paquetes de tren de transporte.

Cuando se reconstruye el tren de programa en el decodificador de tren de transporte, en el caso de paquetes PES con identificador de tren con un valor de identificador de tren auxiliar, el prefijo de código de comienzo de paquete se escribe en el tren que se está reconstruyendo, seguido por campos de bytes de datos procedentes de los paquetes PES de tren de transporte.

Los trenes de ISO/CEI 11172-1 se llevan en trenes de transporte sustituyendo primeramente los encabezamientos de paquete de ISO/CEI 11172-1 por encabezamientos de paquete PES de la Rec. UIT-T H.262 | ISO/CEI 13818-2. Los valores de campo de encabezamiento de paquete ISO/CEI 11172-1 se copian en campos equivalentes de encabezamiento de paquete PES de la Rec. UIT-T H.262 | ISO/CEI 13818-2 equivalentes.

El campo contador de secuencias de paquetes de programa se incluye en el encabezamiento de cada paquete PES que trae datos de un tren de programa, o de un tren de sistema de ISO/CEI 11172-1. Esto permite reproducir en el decodificador el orden que tenían los paquetes en el tren de programa original o en el tren de sistema de ISO/CEI 11172-1 original.

El campo pack_header() de un tren de programa o de un tren de sistema de ISO/CEI 11172-1 es llevado, en el tren de transporte, en el encabezamiento del paquete PES que sigue inmediatamente.

2.4.4 Información específica de programa

La información específica de programa (PSI) incluye datos normativos y datos privados de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que permiten la demultiplexación de programas por los decodificadores. Los programas se componen de uno o más trenes elementales, cada uno etiquetado con un PID. Los trenes elementales de programas o partes de los mismos pueden ser aleatorizados para acceso condicional. Sin embargo, la información específica de programa no será aleatorizada.

En los trenes de transporte, la información específica de programa se clasifica en seis estructuras de tablas como se muestra en el cuadro 2-28. Si bien estas estructuras pueden ser consideradas como tablas simples, serán segmentadas en secciones e insertadas en paquetes del tren de transporte, algunas con PID determinados y otras con PID seleccionables por el usuario.

Cuadro 2-28 – Información específica de programa

Nombre de estructura	Tipo de tren	Número de PID reservado	Descripción
Tabla de asociación de programas	Rec. UIT-T H.222.0 ISO/CEI 13818-1	0x00	Asocia el número de programa y el PID de la tabla de correspondencia de programas
Tabla de correspondencia de programas	Rec. UIT-T H.222.0 ISO/CEI 13818-1	Asignado en la tabla de asociación de programas (PAT)	Especifica valores de PID para componentes de uno o más programas
Tabla de información de red	Privado	Asignado en la tabla de asociación de programas (PAT)	Parámetros de red físicos tales como frecuencias resultantes de multiplexación (FDM), números de transpondedor, etc.
Tabla de acceso condicional	Rec. UIT-T H.222.0 ISO/CEI 13818-1	0x01	Asocia uno o más trenes EMM (privados) con un valor de PID único
Tabla de descripción de tren de transporte	Rec. UIT-T H.222.0 ISO/CEI 13818-1	0x02	Asocia uno o más descriptores del cuadro 2-45 a un tren de transporte entero
Tabla de información de control IPMP	Rec. UIT-T H.222.0 ISO/CEI 13818-1	0x03	Incluye la lista de herramientas, el contenedor de derechos y el contenedor de herramientas IPMP, definidos en ISO/CEI 3818-11

Las tablas de PSI definidas en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 serán segmentadas en una o más secciones que son transportadas en paquetes de transporte. Una sección es una estructura sintáctica que se utilizará para hacer corresponder todas las tablas PSI definidas en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 en paquetes del tren de transporte.

Junto con las tablas PSI definidas en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, es posible transportar tablas de datos privados. Los medios por los cuales la información privada es llevada en paquetes del tren de transporte no se definen en esta Especificación. Se puede estructurar de la misma manera utilizada para transportar tablas PSI definidas en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, de modo que la sintaxis para hacer corresponder estos datos privados es idéntica a la utilizada para la correspondencia de tablas PSI definidas en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Para este fin, se define una sección privada. Si los datos privados son llevados en paquetes del tren de transporte con el mismo valor de PID que los paquetes del tren de transporte que llevan tablas de correspondencia de programas (identificados en la tabla de asociación de programas), se utilizará la sintaxis y la semántica de sección privada. Los datos llevados en bytes de datos privados pueden estar aleatorizados. Sin embargo, ningún otro campo de la sección privada estará aleatorizado. Esta sección privada permite transmitir datos con un mínimo de estructura. Cuando no se utiliza esta estructura, la correspondencia de datos privados en paquetes del tren de transporte no es definida por la presente Recomendación | Norma Internacional.

Las secciones pueden tener longitud variable. El principio de una sección es indicado por un campo de puntero en la cabida útil del paquete del tren de transporte. La sintaxis de este campo se especifica en el cuadro 2-29.

Pueden aparecer campos de adaptación en paquetes de tren de transporte que llevan secciones PSI.

En un tren de transporte, se pueden encontrar bytes de relleno de paquetes de valor 0xFF en la cabida útil de los paquetes del tren de transporte que llevan secciones PSI y/o secciones privadas únicamente después del último byte de una sección. En este caso, todos los bytes siguientes hasta el fin del paquete del tren de transporte serán también bytes de relleno de valor 0xFF. Estos bytes se pueden descartar mediante un decodificador. En este caso, la cabida útil del siguiente paquete del tren de transporte con el mismo valor de PID comenzará con un campo de puntero de valor 0x00 que indica que la siguiente sección comienza inmediatamente después.

Cada tren de transporte contendrá uno o más paquetes de tren de transporte con valor de PID 0x0000. Estos paquetes del tren de transporte juntos contendrán una tabla de asociación de programas completa con una lista completa de todos los programas dentro del tren de transporte. La versión de la tabla transmitida más recientemente con el *current_next_indicator* (indicador siguiente vigente) puesto a un valor de '1' se aplicará siempre a los datos vigentes en el tren de transporte. Cualesquiera cambios en los programas transmitidos en el tren de transporte se describirán en una versión actualizada de la tabla de asociación de programas transmitida en paquetes del tren de transporte con valor de PID 0x0000. Todas estas secciones utilizan el valor de identificador de tabla 0x00. Sólo están permitidas las secciones con este valor de identificador de tabla dentro de paquetes del tren de transporte con valor PID de 0x0000. Para que una nueva versión de la tabla de asociación de programas (PAT, *program association table*) sea válida, todas las secciones (indicadas en el último número de sección) con un número de nueva versión y con el indicador siguiente vigente puesto a '1' tienen que salir de la memoria tampón principal B_{sys} definida en el T-STD (véase 2.4.2). La PAT se hace válida cuando el último byte de la sección que se necesita para completar la tabla sale de B_{sys} .

Cuando se aleatorizan uno o varios trenes elementales dentro de un tren de transporte, se transmitirán paquetes de tren de transporte con un valor de PID 0x0001 que contienen una tabla de acceso condicional completa e incluyen descriptores de acceso condicional asociados con los trenes aleatorizados. Los paquetes de tren de transporte transmitidos formarán juntos una versión completa de la tabla de acceso condicional. La versión de la tabla transmitida más recientemente con `current_next_indicator` puesto a un valor '1' se aplicará siempre a los datos vigentes en el tren de transporte. Todos los cambios de la aleatorización que hacen que la tabla existente sea no válida o incompleta se describirán en una versión actualizada de la tabla de acceso condicional. Estas secciones utilizarán el valor de identificador de tabla 0x01. Sólo están permitidas secciones con este valor de identificador de tabla dentro de paquetes del tren de transporte con valor de PID de 0x0001. Para que una nueva versión de la tabla de acceso condicional (CAT, *conditional access table*) sea válida, todas las secciones (indicadas en el último número de sección) con un número de nueva versión y con el indicador siguiente vigente puesto a '1' tienen que salir de la memoria tampón principal B_{sys} . La CAT se hace válida cuando el último byte de la sección que se necesita para completar la tabla sale de B_{sys} .

Cada tren de transporte contendrá uno o más paquetes de trenes de transporte con valores de PID que están etiquetados según la tabla de asociación de programas como paquetes del tren de transporte que contienen secciones de correspondencia de programas TS. Cada programa indicado en la tabla de asociación de programas se describirá en una sección única de correspondencia de programas TS. Cada programa estará totalmente definido en el propio tren de transporte. Los datos privados que tienen un campo PID elemental asociado en la sección apropiada de tabla de correspondencia de programas forman parte del programa. Otros datos privados pueden existir en el tren de transporte sin que estén indicados en la sección de tabla de correspondencia de programas. La versión transmitida más recientemente de la sección de correspondencia de programa TS con `current_next_indicator` puesto a un valor '1' se aplicará siempre a los datos vigentes dentro del tren de transporte. Todos los cambios en la definición de cualquiera de los programas transmitidos dentro del tren de transporte se describirán en una versión actualizada de la sección pertinente de la tabla de correspondencia de programas transmitida en paquetes de tren de transporte con el valor de PID identificado como el PID de correspondencia de programas para ese programa específico. Todos los paquetes del tren de transporte que transmiten una sección de correspondencia de programas TS dada tendrán el mismo valor de PID. Durante la existencia continua de un programa, incluidos todos sus eventos asociados, el PID de correspondencia de programa no cambiará. Una definición de programa no abarcará más de una sección de correspondencia de programas TS. Una nueva versión de una sección de correspondencia de programas TS se hace válida cuando el último byte de esa sección con un número de nueva versión y con el indicador siguiente vigente puesto a '1' sale de B_{sys} .

Las secciones con un valor de identificador de tabla de 0x02 contendrán información de tabla de correspondencia de programas. Estas secciones pueden ser transmitidas en paquetes de tren de transporte con diferentes valores de PID.

La tabla de información de red es facultativa y su contenido es privado. Si está presente, es transmitida en paquetes del tren de transporte que tendrán el mismo valor de PID, denominado PID de red. El valor de PID de red es definido por el usuario y, cuando está presente, figurará en la tabla de asociación de programas bajo el número de programa reservado 0x0000. Si existe la tabla de información de red, tomará la forma de una o más secciones privadas.

El número máximo de bytes en una sección de una tabla PSI definida en la Rec. UIT T H.222.0 | ISO/CEI 13818-1 es 1024 bytes. El número máximo de bytes en una sección privada es 4096 bytes.

La tabla de descripción de tren de transporte es facultativa. Si está presente, es transmitida en paquetes del tren de transporte que tienen un valor de PID de 0x0002 como se especifica en el cuadro 2-28 y será aplicable al tren de transporte entero. Las secciones de descripción de tren de transporte utilizarán un valor de identificador de tabla (`table_id`) de 0x03 como se especifica en el cuadro 2-31 y su contenido está restringido a los descriptores especificados en el cuadro 2-45. La sección de descripción de tren de transporte (`TS_description_section`) pasa a ser válida cuando existe el último byte B_{sys} de la sección requerida para completar la tabla.

No se imponen restricciones a la aparición de códigos de comienzo, bytes de sincronización u otros esquemas de bits en datos PSI, sean de la presente Recomendación | Norma Internacional o privados.

2.4.4.1 Puntero

La sintaxis de `pointer_field` (campo de puntero) se describe en el cuadro 2-29.

Cuadro 2-29 – Puntero de información específica de programa

Sintaxis	N.º de bits	Mnemónico
<code>pointer_field</code>	8	<code>uimsbf</code>

2.4.4.2 Definición semántica de campos en la sintaxis de puntero

pointer_field (campo de puntero) – Es un campo de 8 bits cuyo valor será el número de bytes que siguen inmediatamente al campo de puntero hasta el primer byte de la primera sección que está presente en la cabida útil del paquete del tren de transporte (de modo que un valor de 0x00 en el campo de puntero indica que la sección comienza inmediatamente después del campo de puntero). Cuando por lo menos una sección empieza en un paquete de tren de transporte dado, el indicador de comienzo de unidad de cabida útil (véase 2.4.3.2) se pondrá a 1 y el primer byte de la cabida útil de ese paquete de tren de transporte contendrá el puntero. Cuando ninguna sección empieza en un paquete de tren de transporte dado, el indicador de comienzo de unidad de cabida útil se pondrá a 0 y no se enviará ningún puntero en la cabida útil de ese paquete.

2.4.4.3 Tabla de asociación de programas

La tabla de asociación de programas proporciona la correspondencia entre un número de programa y el valor de PID de los paquetes del tren de transporte que llevan la definición de programa. El número de programa es la etiqueta numérica asociada con un programa.

La tabla completa está contenida en una o más secciones con la siguiente sintaxis. Se puede segmentar para ocupar múltiples secciones (véase el cuadro 2-30).

Cuadro 2-30 – Sección de asociación de programas

Sintaxis	N.º de bits	Mnemónico
<code>program_association_section() {</code>		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if (program_number == '0') {		
network_PID	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchbf
}		

2.4.4.4 Asignaciones de identificador de tabla

El campo `table_id` (identificador de tabla) identifica el contenido de una sección PSI del tren de transporte, como se muestra en el cuadro 2-31.

Cuadro 2-31 – Valores del identificador de tabla (`table_id`)

Valor	Descripción
0x00	<code>program_association_section</code>
0x01	<code>conditional_access_section (CA_section)</code>
0x02	<code>TS_program_map_section</code>
0x03	<code>TS_description_section</code>
0x04	<code>ISO_IEC_14496_scene_description_section</code>
0x05	<code>ISO_IEC_14496_object_descriptor_section</code>
0x06	<code>Metadata_section</code>
0x07	Sección de información de control IPMP (definida en ISO/CEI 13818-11)
0x08-0x3F	Reservado Rec. UIT-T H.222.0 ISO/CEI 13818-1
0x40-0xFE	Privado de usuario
0xFF	Prohibido

2.4.4.5 Definición semántica de campos en la sección de asociación de programas

`table_id` (identificador de tabla) – Este es un campo de 8 bits que se pondrá a 0x00 como se muestra en el cuadro 2-31.

`section_syntax_indicator` (indicador de sintaxis de sección) – El indicador de sintaxis de sección es un campo de 1 bit que se pondrá a '1'.

`section_length` (longitud de sección) – Este es un campo de 12 bits cuyos dos primeros bits serán '00'. Los 10 bits restantes especifican el número de bytes de la sección, que comienza inmediatamente después del campo de longitud de sección, e incluye la CRC. El valor de este campo no será superior a 1021 (0x3FD).

`transport_stream_id` (identificador de tren de transporte) – Este es un campo de 16 bits que sirve como una etiqueta para identificar este tren de transporte con respecto a cualquier otro múltiplex dentro de una red. Su valor es definido por el usuario.

`version_number` (número de versión) – Este campo de 5 bits es el número de versión de toda la tabla de asociación de programas. El número de versión se incrementará en 1, módulo 32, cada vez que cambia la definición de la tabla de asociación de programas. Cuando el `current_next_indicator` se pone a '1', el número de versión será el de la tabla de asociación de programas actualmente aplicable, y cuando se pone a '0', el número de versión será el de la siguiente tabla de asociación de programas aplicable.

`current_next_indicator` (indicador siguiente vigente) – Un indicador de 1 bit, que cuando se pone a '1' indica que la tabla de asociación de programas enviada es aplicable actualmente. Cuando el bit se pone a '0', indica que la tabla enviada no es aún aplicable y será la siguiente tabla que será válida.

`section_number` (número de sección) – Este campo de 8 bits da el número de esta sección. El número de sección de la primera sección de la tabla de asociación de programa será 0x00. Se incrementará en 1 con cada sección adicional en la tabla de asociación de programas.

`last_section_number` (número de última sección) – Este campo de 8 bits especifica el número de la última sección (es decir, la sección con el número de sección más alto) de la tabla de asociación de programas completa.

`program_number` (número de programa) – Es un campo de 16 bits que especifica el programa al cual es aplicable el PID de correspondencia de programas. Cuando se pone a 0x0000, la siguiente referencia de PID será el PID de red. Para todos los demás casos, el valor de este campo es definido por el usuario. Este campo no tomará un valor más de una vez dentro de una versión de la tabla de asociación de programas.

NOTA – Por ejemplo, se puede utilizar el número de programa como una designación para un canal de radiodifusión.

`network_PID` (PID de red) – Es un campo de 13 bits que sólo se utiliza junto con el valor del número de programa puesto a 0x0000 y especifica el PID de los paquetes del tren de transporte que contendrán la tabla de información de red. El valor del campo PID de red es definido por el usuario pero sólo tomará los valores especificados en el cuadro 2-3. La presencia del PID de red es facultativa.

program_map_PID (PID de correspondencia de programas) – Es un campo de 13 bits que especifica el PID de los paquetes del tren de transporte que contendrán la sección de correspondencia de programas aplicable al programa especificado por el número de programa. Ningún número de programa tendrá más de una asignación de PID de correspondencia de programas. El valor del PID de correspondencia de programas es definido por el usuario, pero sólo tomará los valores especificados en el cuadro 2-3.

CRC_32 – Este es un campo de 32 bits que contiene el valor CRC que da una salida cero de los registradores en el decodificador definido en el anexo A después de procesar toda la sección de asociación de programas.

2.4.4.6 Tabla de acceso condicional

La tabla de acceso condicional (CA, *conditional access*) proporciona la asociación entre uno o más sistemas CA, sus trenes de EMM y cualesquiera parámetros especiales asociados con ellos. Para una definición del campo del descriptor en el cuadro 2-32, véase 2.6.16.

El cuadro está contenido en una o más secciones. Se puede segmentar para ocupar múltiples secciones.

Cuadro 2-32 – Sección de acceso condicional

Sintaxis	N.º de bits	Mnemónico
CA_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

2.4.4.7 Definición semántica de campos en la sección de acceso condicional

table_id (identificador de tabla) – Este es un campo de 8 bits que se pondrá a 0x01 como se especifica en el cuadro 2-31.

section_syntax_indicator (indicador de sintaxis de sección) – El indicador de sintaxis de sección es un campo de 1 bit que se pondrá a '1'.

section_length (longitud de sección) – Este es un campo de 12 bits cuyos dos primeros bits serán '00'. Los 10 bits restantes especifican el número de bytes de la sección, que comienza inmediatamente después del campo de longitud de sección, e incluye la CRC. El valor de este campo no será superior a 1021 (0x3FD).

version_number (número de versión) – Este campo de 5 bits es el número de versión de toda la tabla de acceso condicional. El número de versión se incrementará en 1, módulo 32, cuando se produce un cambio en la información transportada en la tabla CA. Cuando **current_next_indicator** se pone a '1', el número de versión será el de la tabla de acceso condicional actualmente aplicable y cuando se pone a '0', el número de versión será el de la siguiente tabla de acceso condicional aplicable.

current_next_indicator (indicador siguiente vigente) – Un indicador de 1 bit, que cuando se pone a '1' indica que la tabla de acceso condicional no es aplicable actualmente. Cuando el bit se pone a '0', indica que la tabla de acceso condicional enviada no es aún aplicable y será la siguiente tabla de acceso condicional que será válida.

section_number (número de sección) – Este campo de 8 bits da el número de esta sección. El número de sección de la primera sección de la tabla de acceso condicional será 0x00. Se incrementará en 1 con cada sección adicional en la tabla de acceso condicional.

last_section_number (número de última sección) – Este campo de 8 bits especifica el número de la última sección (es decir, la sección con el número de sección más alto) de la tabla de acceso condicional.

CRC_32 – Este es un campo de 32 bits que contiene el valor CRC que da una salida de cero de los registradores en el decodificador definido en el anexo A después de procesar toda la sección de acceso condicional.

2.4.4.8 Tabla de correspondencia de programas

La tabla de correspondencia de programas proporciona la correspondencia entre los números de programa y los elementos de programa que los constituyen. Un solo caso de esta correspondencia se denomina una "definición de programa". La tabla de correspondencia de programas es el conjunto completo de todas las definiciones de programa para un tren de transporte. Esta tabla se transmitirá en paquetes cuyos valores de PID son seleccionados por el codificador. Se puede utilizar más de un valor de PID, si se desea. La tabla está contenida en una o más secciones, con la siguiente sintaxis. Se puede segmentar para ocupar múltiples secciones. En cada sección, el campo de número de sección se pondrá a cero. Las secciones se identifican por el campo de número de programa.

La definición de los campos de descriptor () figuran en 2.6 (véase el cuadro 2-33).

Cuadro 2-33 – Sección de correspondencia de programas del tren de transporte

Sintaxis	N.º de bits	Mnemónico
TS_program_map_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	3	bslbf
PCR_PID	13	uimsbf
reserved	4	bslbf
program_info_length	12	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
for (i = 0; i < N1; i++) {		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID	13	uimsbf
reserved	4	bslbf
ES_info_length	12	uimsbf
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

2.4.4.9 Definición semántica de campos en la sección de correspondencia de programas del tren de transporte

table_id (identificador de tabla) – Este es un campo de 8 bits, que en el caso de una sección de correspondencia de programas TS se pondrá siempre a 0x02 como se muestra en el cuadro 2-31.

section_syntax_indicator (indicador de sintaxis de sección) – El indicador de sintaxis de sección es un campo de 1 bit que se pondrá a '1'.

section_length (longitud de sección) – Este es un campo de 12 bits, cuyos dos primeros bits serán '00'. Los 10 bits restantes especifican el número de bytes de la sección que comienza inmediatamente después del campo de longitud de sección e incluye la CRC. El valor de este campo no será superior a 1021 (0x3FD).

program_number (número de programa) – Es un campo de 16 bits que especifica el programa al cual es aplicable el PID de correspondencia de programas. Sólo se transportará una definición de programa dentro de una sección de correspondencia de programas TS. Esto entraña que una definición de programa nunca es superior a 1016 bytes (0x3F8). Véase el anexo C informativo para las maneras de tratar los casos cuando la longitud no es suficiente. El número de programa se puede utilizar, por ejemplo, como una designación para el canal de radiodifusión. Describiendo los elementos pertenecientes a un programa se pueden concatenar juntos datos de diferentes fuentes (por ejemplo, eventos secuenciales) para formar un conjunto continuo de trenes con un número de programa. Para ejemplos de las aplicaciones, véase el anexo C.

version_number (número de versión) – Este campo de 5 bits es el número de versión de la sección de correspondencia de programas TS. El número de versión se incrementará en 1, módulo 32, cuando se produce un cambio en la información transportada dentro de la sección. El número de versión se refiere a la definición de un solo programa y, por tanto, a una sola sección. Cuando el **current_next_indicator** se pone a '1', el número de versión será el de la sección de correspondencia de programas TS actualmente aplicable y cuando se pone a '0', el número de versión será el de la siguiente sección de correspondencia de programas TS aplicable.

current_next_indicator (indicador siguiente vigente) – Un campo de 1 bit, que cuando se pone a '1' indica que la sección de correspondencia de programa enviada es actualmente aplicable. Cuando el bit se pone a '0', indica que la sección de correspondencia de programa TS enviada no es aún aplicable y será la siguiente sección de correspondencia de programas TS que será válida.

section_number (número de sección) – El valor de este campo de 8 bits será 0x00.

last_section_number (número de última sección) – El valor de este campo de 8 bits será 0x00.

PCR_PID – Este es un campo de 13 bits que indica el PID de los paquetes del tren de transporte que contendrán los campos PCR válidos para el programa especificado por el número de programa. Si ninguna PCR está asociada con una definición de programa para trenes privados, este campo tomará el valor 0x1FFF. Véase la definición semántica de PRC en 2.4.3.5 y el cuadro 2-3 para las restricciones impuestas a la elección del valor PRC-PID.

program_info_length (longitud de información de programa) – Es un campo de 12 bits, cuyos dos primeros bits serán siempre '00'. Los 10 bits restantes especifican el número de bytes de los descriptores que siguen inmediatamente al campo de longitud de información de programa.

stream_type (tipo de tren) – Es un campo de 8 bits que especifica el tipo de elemento de programa transportado en los paquetes con el PID cuyo valor es especificado por el PID elemental. Los valores del tipo de tren se especifican en el cuadro 2-34.

NOTA – Un tren auxiliar de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 está disponible para tipos de datos definidos por esta Especificación, distintos de audio, vídeo y DSM CC, tales como Directorio de trenes de programa y Correspondencia de trenes de programa.

Cuadro 2-34 – Asignaciones de tipos de tren

Valor	Descripción
0x00	Reservado UIT-T ISO/CEI
0x01	Vídeo ISO/CEI 11172
0x02	Vídeo UIT-T Rec. H.262 ISO/CEI 13818-2 o tren de vídeo de parámetros restringidos ISO/CEI 11172-2
0x03	Audio ISO/CEI 11172-3
0x04	Audio ISO/CEI 13818-3
0x05	Secciones privadas Rec. UIT-T H.222.0 ISO/CEI 13818-1
0x06	Paquetes PES Rec. UIT-T H.222.0 ISO/CEI 13818-1 que contienen datos privados
0x07	ISO/CEI 13522 MHEG
0x08	Rec. UIT-T H.222.0 ISO/CEI 13818-1 anexo A DSM-CC
0x09	Rec. UIT-T H.222.1
0x0A	ISO/CEI 13818-6 tipo A
0x0B	ISO/CEI 13818-6 tipo B
0x0C	ISO/CEI 13818-6 tipo C
0x0D	ISO/CEI 13818-6 tipo D
0x0E	Rec. UIT-T H.222.0 ISO/CEI 13818-1 auxiliar
0x0F	Audio ISO/CEI 13818-7 con sintaxis de transporte ADTS
0x10	Objeto visual ISO/CEI 14496-2
0x11	Audio ISO/CEI 14496-3 con la sintaxis de transporte LATM definida en ISO/CEI 14496-3
0x12	Tren paquetizado SL o tren FlexMux ISO/CEI 14496-1 SL-transporte en paquetes PES
0x13	Tren paquetizado SL o tren FlexMux ISO/CEI 14496-1 transporte en secciones ISO/CEI 14496
0x14	Protocolo de telecarga sincronizado ISO/CEI 13818-6
0x15	Metadatos transportados en paquetes PES
0x16	Metadatos transportados en metadata_sections
0x17	Metadatos transportados en carretes de datos ISO/CEI 13818-6
0x18	Metadatos transportados en carretes de objetos ISO/CEI 13818-6
0x19	Metadatos transportados en el protocolo de descarga sincronizada ISO/CEI 13818-6
0x1A	Tren IPMP (definido en ISO/CEI 13818-11, MPEG-2 IPMP)
0x1B	Tren de vídeo AVC según el formato de vídeo definido en la Rec. UIT-T H.264 ISO/CEI 14496-10
0x1C-0x7E	Reservado para la Rec. UIT-T H.222.0 ISO/CEI 13818-1
0x7F	Tren IPMP
0x80-0xFF	Privado de usuario

elementary_PID (PID elemental) – Es un campo de 13 bits que especifica el PID de los paquetes del tren de transporte que llevan el elemento de programa asociado.

ES_info_length (longitud de información de tren elemental) – Es un campo de 12 bits, cuyos dos primeros bits serán siempre '00'. Los 10 bits restantes especifican el número de bytes de los descriptores del elemento de programa asociado que sigue inmediatamente al campo de longitud de información ES.

CRC_32 – Este es un campo de 32 bits que contiene el valor CRC que da una salida de cero de los registradores en el decodificador, definido en el anexo B después de procesar toda la sección de correspondencia de programas del tren de transporte.

2.4.4.10 Sintaxis de la sección privada

Cuando se envían datos privados en paquetes del tren de transporte con un valor de PID designado como un PID de tabla de correspondencia de programas en la tabla de asociación de programas, se utilizará la sección privada. Esta sección privada permite transmitir datos con una estructura mínima para que un decodificador pueda analizar el tren. Las secciones se pueden utilizar de dos maneras: si el indicador de sintaxis de sección está puesto a '1', se utilizará toda la estructura común a todas las tablas; si el indicador se pone a '0', sólo los campos 'table_id' a 'private_section_length' seguirán la sintaxis y semántica de estructura común y el resto de la sección privada puede adoptar cualquier forma que el usuario determine. En el anexo C (informativo) figuran ejemplos de la utilización ampliada de esta sintaxis.

Se puede formar una tabla privada con varias secciones privadas, todas con el mismo identificador de tabla (véase el cuadro 2-35).

Cuadro 2-35 – Sección privada

Sintaxis	N.º de bits	Mnemónico
private_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
Reserved	2	bslbf
private_section_length	12	uimsbf
if (section_syntax_indicator == '0') {		
for (i = 0; i < N; i++) {		
private_data_byte	8	bslbf
}		
}		
else {		
table_id_extension	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < private_section_length-9; i++) {		
private_data_byte	8	bslbf
}		
CRC_32	32	rpchof
}		
}		

2.4.4.11 Definición semántica de campos en la sección privada

table_id (identificador de tabla) – Es un campo de 8 bits, cuyo valor identifica la tabla privada a la cual pertenece esta sección. Sólo se pueden utilizar los valores definidos en el cuadro 2-31 como "usuario privado".

section_syntax_indicator (indicador de sintaxis de sección) – Es un indicador de 1 bit que cuando está puesto a '1' indica que la sección privada sigue la sintaxis de sección genérica más allá del campo de longitud de sección privada. Cuando está puesto a '0' indica que los bytes de datos privados siguen inmediatamente al campo de longitud de sección privada.

private_indicator (indicador privado) – Esta es una bandera de 1 bit, definible por el usuario, que no será especificada por el UIT-T | ISO/CEI en el futuro.

private_section_length (longitud de sección privada) – Un campo de 12 bits que especifica el número de bytes restantes en la sección privada que siguen inmediatamente al campo de longitud de sección privada hasta el fin de la sección privada. El valor de este campo no será superior a 4093 (0xFFD).

private_data_byte (byte de datos privados) – El campo de byte de datos privados es definible por el usuario y no será especificado por el UIT-T | ISO/CEI en el futuro.

table_id_extension (extensión de identificador de tabla) – Es un campo de 16 bits y su utilización y valor son definidos por el usuario.

version_number (número de versión) – Este campo de 5 bits es el número de versión de la sección privada. El número de versión será incrementado en 1, módulo 32, cuando se produce un cambio en la información transportada en la sección privada. Cuando current_next_indicator está puesto a '0', el número de versión será el de la siguiente sección privada aplicable con los mismos identificadores de tabla y número de sección.

current_next_indicator (indicador siguiente vigente) – Un campo de 1 bit, que cuando se pone a '1' indica que la sección privada enviada es aplicable actualmente. Cuando se pone a '1', el número de versión será el de la sección privada actualmente aplicable. Cuando se pone a '0', indica que la sección privada enviada no es aún aplicable y será la siguiente sección privada con los mismos números de sección e identificador de tabla que será válida.

section_number (número de sección) – Este campo de 8 bits da el número de la sección privada. El número de sección de la primera sección en una tabla privada será 0x00. El número de sección se incrementará en 1 con cada sección adicional en esta tabla privada.

last_section_number (número de última sección) – Este campo de 8 bits especifica el número de la última sección (es decir, la sección con el número de sección más alto) de la tabla privada de la cual esta sección forma parte.

CRC_32 – Este es un campo de 32 bits que contiene el valor CRC que da una salida de cero de los registradores en el decodificador definido en el anexo A después de procesar toda la sección privada.

2.4.4.12 Sintaxis de la sección de tren de transporte

Los trenes de bits conformes a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 pueden llevar la información definida en el cuadro 2-36. Los decodificadores conformes a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 pueden decodificar la información definida en este cuadro.

El cuadro de descripción de tren de transporte se define como marco soporte de los descriptores que figuran en 2.6 para un tren de transporte entero. Los descriptores se aplicarán al tren de transporte entero. Este cuadro utiliza el valor de identificador de tabla (*table_id*) de 0x03 que se especifica en el cuadro 2-31 y se lleva en paquetes de tren de transporte cuyo valor de PID es 0x0002 especificado en el cuadro 2-3.

Cuadro 2-36 – Tabla de descripción de tren de transporte

Sintaxis	N.º de bits	Mnemónico
TS_description_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
Reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

2.4.4.13 Definición semántica de campos en la sección de tren de transporte

table_id (identificador de tablas) – Es un campo de 8 bits que se pondrá a '0x03' como se especifica en el cuadro 2-31.

section_length (longitud de sección) – Este es un campo de 12 bits cuyos dos primeros bits serán '00'. Los 10 bits restantes especifican el número de bytes de la sección, que comienza inmediatamente después del campo de longitud de sección, e incluye la CRC. El valor de este campo no será superior a 1021 (0x3FD).

version_number (número de versión) – Este campo de 5 bits es el número de versión de toda la tabla de descripción de tren de transporte. El número de versión se incrementará en 1, módulo 32, cuando cambie la definición de la tabla de descripción de tren de transporte. Cuando *current_next_indicator* se ponga a '1', el número de versión será el de la tabla de descripción de tren de transporte actualmente aplicable y cuando se ponga a '0', el número de versión será el de la siguiente tabla de descripción de tren de transporte aplicable.

current_next_indicator (indicador siguiente vigente) – Un indicador de 1 bit, que cuando se pone a '1' indica que la tabla de descripción de tren de transporte es aplicable actualmente. Cuando el bit se pone a '0', indica que la tabla enviada todavía no es aplicable y será la siguiente tabla la que sea válida.

section_number (número de sección) – Este campo de 8 bits da el número de esta sección. El número de sección de la primera sección de la tabla de descripción de tren de transporte será 0x00. Se incrementará en 1 con cada sección adicional de la tabla de descripción de tren de transporte.

last_section_number (número de última sección) – Este campo de 8 bits especifica el número de la última sección (es decir, la sección con el número de sección más alto) de la tabla de descripción de tren de transporte completa.

CRC_32 – Este es un campo de 32 bits que contiene el valor CRC que da una salida de cero de los registradores en el decodificador definido en el anexo A después de procesar toda la sección de descripción de tren de transporte.

2.5 Requisitos del tren de bits del tren de programa

2.5.1 Estructura de codificación y parámetros del tren de programa

La capa de codificación de tren de programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 permite combinar un programa de uno o más trenes elementales en un solo tren. Los datos de cada tren elemental son multiplexados junto con información que permite la presentación sincronizada de los trenes elementales dentro del programa.

Un tren de programa consiste en uno o más trenes elementales de un programa multiplexados juntos. Los trenes elementales de audio y vídeo consisten en unidades de acceso.

Los datos de trenes elementales son transportados en paquetes PES. Un paquete PES consiste en un encabezamiento de paquete PES seguido de datos de paquete. Los paquetes PES se insertan en paquetes de tren de programa.

El encabezamiento de paquete PES empieza con un código de comienzo de 32 bits que identifica también al tren (véase el cuadro 2-22) al cual pertenecen los datos del paquete. El encabezamiento de paquete PES puede contener solamente una indicación de tiempo de presentación (PTS) o una indicación de tiempo de presentación y una indicación de tiempo de codificación (DTS). El encabezamiento de paquete PES contiene también otros campos facultativos. Los datos de paquete contienen un número variable de bytes contiguos de un tren elemental.

En un tren de programa, los paquetes PES están organizados en sacas. Una saca comienza con un encabezamiento de saca y es seguida por ninguno o varios paquetes PES. El encabezamiento de saca empieza con un código de comienzo de 32 bits. El encabezamiento de saca se utiliza para almacenar información de temporización y de velocidad binaria.

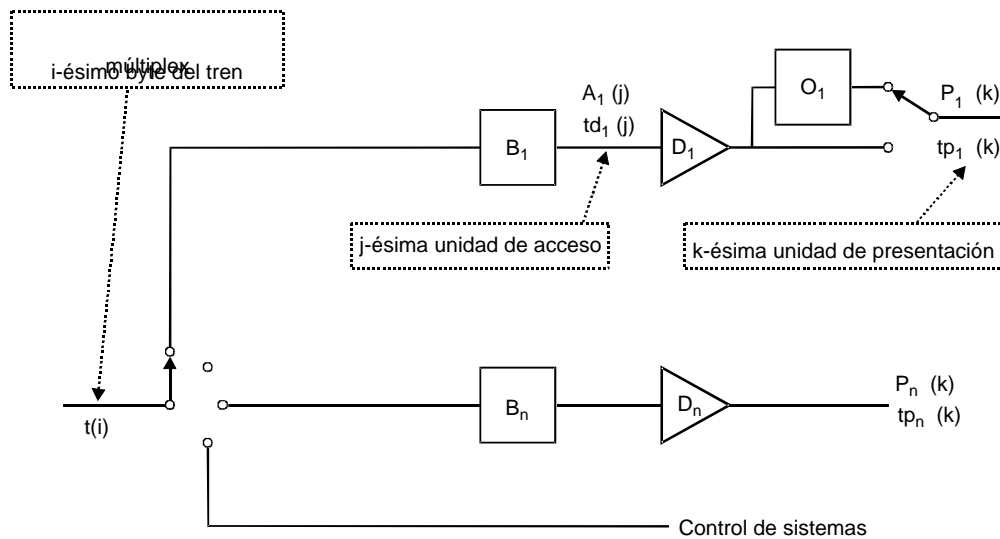
El tren de programa empieza con un encabezamiento de sistema que puede repetirse facultativamente. El encabezamiento de sistema transporta un resumen de los parámetros de sistema definidos en el tren.

Esta Recomendación | Norma Internacional no especifica los datos codificados que se pueden utilizar como parte de sistemas de acceso condicional, pero sí proporciona mecanismos para que los proveedores de servicio de programa transporten e identifiquen estos datos para el procesamiento del decodificador y para referenciar correctamente los datos aquí especificados.

2.5.2 Decodificador-objetivo de sistemas del tren de programas

La semántica del tren de programa y las constricciones impuestas a esta semántica requieren definiciones exactas de los eventos de decodificación y de los instantes en que estos eventos se producen. Las definiciones necesarias se describen en esta Especificación utilizando un decodificador ficticio conocido como el decodificador-objetivo de sistema del tren de programa (P-STD).

El P-STD es un modelo teórico utilizado para definir estos términos con precisión y modelar el proceso de decodificación durante la construcción de trenes de programa. El P-STD se define solamente con este fin. Ni la arquitectura del P-STD ni la temporización descrita excluyen la repetición ininterrumpida y sincronizada de trenes de programas procedentes de una variedad de decodificadores con diferentes arquitecturas o planes de temporización.



TISO5820-95/d07

Figura 2-2 – Notación del decodificador-objetivo de sistemas del tren de programa

La siguiente notación se utiliza para describir el decodificador-objetivo de sistemas del tren de programa y se ilustra parcialmente en la figura 2-2:

- i, i' son índices para bytes en el tren de programa. El primer byte tiene el índice 0.
- j es un índice para unidades de acceso en los trenes elementales.
- k, k', k'' son índices para unidades de presentación en los trenes elementales.
- n es un índice para los trenes elementales.
- $t(i)$ indica el tiempo en segundos en el cual el i -ésimo byte del tren de programa entra en el decodificador-objetivo de sistema. El valor $t(0)$ es una constante arbitraria.
- $SCR(i)$ es el tiempo codificado en el campo SCR medido en unidades del reloj de sistema de 27 MHz, donde i es el índice de byte del byte final del campo base de referencia de reloj de sistema.
- $A_n(j)$ es la j -ésima unidad de acceso en un tren elemental n . $A_n(j)$ está indexada según el orden de decodificación.
- $td_n(j)$ es el tiempo de decodificación, medido en segundos, en el decodificador-objetivo de sistema de la j -ésima unidad de acceso en el tren elemental n .
- $P_n(k)$ es la k -ésima unidad de presentación en un tren elemental n . $P_n(k)$ está indexada en el orden de presentación.
- $tp_n(k)$ es el tiempo de presentación, medido en segundos, en el decodificador-objetivo del sistema de la k -ésima unidad de presentación en el tren elemental n .
- t es el tiempo medido en segundos.
- $F_n(t)$ es la capacidad llena, medida en bytes, de la memoria tampón de entrada del decodificador-objetivo de sistemas para el tren elemental n en el instante t .
- B_n es la memoria tampón de entrada en el decodificador-objetivo de sistemas para el tren elemental n .
- BS_n es el tamaño de la memoria tampón de entrada del decodificador-objetivo de sistemas, medido en bytes, para el tren elemental n .
- D_n es el decodificador para el tren elemental n .
- O_n es la memoria tampón de reordenación para el tren elemental de vídeo n .

2.5.2.1 Frecuencia de reloj de sistema

La información de temporización referenciada en P-STD es transportada por varios campos de datos definidos en la presente Especificación. Los campos se definen en 2.5.3.3 y 2.4.3.6. Esta información se codifica como el valor muestreado de un reloj de sistema.

ISO/CEI 13818-1:2007 (S)

El valor de la frecuencia de reloj de sistema se mide en Hz y satisfará las siguientes constricciones:

- $27\ 000\ 000 - 810 \leq \text{system_clock_frequency} \leq 27\ 000\ 000 + 810$;
- velocidad de cambio de $\text{system_clock_frequency}$ en función del tiempo $\leq 75 \times 10^{-3}$ Hz/s.

La notación "system_clock_frequency" (frecuencia reloj de sistema) se utiliza en varias partes en esta Recomendación | Norma Internacional para referirse a la frecuencia de un reloj que satisface estos requisitos. Para facilitar la notación, las ecuaciones en las cuales SCR, PTS o DTS aparecen, conducen a valores de tiempo que son exactos a algún múltiplo integral de $(300 \times 2^{33} / \text{system_clock_frequency})$ segundos. Esto se debe a la codificación de la información de temporización SCR como 33 bits de $1/300$ de la frecuencia de reloj de sistema más 9 bits para el resto, y la codificación como 33 bits de la frecuencia de reloj de sistema dividida por 300 para PTS y DTS.

2.5.2.2 Entrada al decodificador-objetivo de sistemas del tren de transporte

Los datos del tren de programa entran en el decodificador-objetivo de sistemas. El i -ésimo byte entra en el instante $t(i)$. El tiempo en el cual este bit entra en el decodificador-objetivo de sistemas puede ser recuperado del tren de entrada decodificando los campos de referencia de reloj de sistema (SCR) de entrada y el campo de velocidad del multiplexor de programa codificados en el encabezamiento de paca. La SCR, definida en la ecuación 2-18, se codifica en dos partes: una, en unidades del periodo de $1/300 \times$ la frecuencia de reloj de sistema, denominada $\text{system_clock_reference_base}$ (véase la ecuación 2-19) y otra, denominada $\text{system_clock_reference_ext}$ (véase la ecuación 2-20) en unidades del periodo de la frecuencia de reloj de sistema. A continuación, los valores codificados en estos campos son indicados por $\text{SCR_base}(i)$ y $\text{SCR_ext}(i)$. El valor codificado en el campo SCR indica el tiempo $t(i)$, donde i se refiere al byte que contiene el último byte del campo base de referencia de reloj de sistema.

Específicamente:

$$\text{SCR}(i) = \text{SCR_base}(i) \times 300 + \text{SCR_ext}(i) \quad (2-18)$$

donde:

$$\text{SCR_base}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 300) \% 2^{33} \quad (2-19)$$

$$\text{SCR_ext}(i) = ((\text{system_clock_frequency} \times t(i)) \text{DIV } 1) \% 300 \quad (2-20)$$

El tiempo de llegada de entrada, $t(i)$, dado en la ecuación 2-21, para todos los otros bytes se construirá a partir de $\text{SCR}(i)$ y la velocidad a la cual llegan los datos, donde la velocidad de llegada dentro de cada saca es el valor representado en el campo program_mux_rate (velocidad de múltiplex de programa) en el encabezamiento de esa saca.

$$t(i) = \frac{\text{SCR}(i')}{\text{system_clock_frequency}} + \frac{i - i'}{\text{program_mux_rate} \times 50} \quad (2-21)$$

donde:

i' es el índice del byte que contiene el último bit del campo de base de referencia y de reloj de sistema en el encabezamiento de saca;

i es el índice de cualquier byte en la saca, incluido el encabezamiento de saca;

$\text{SCR}(i')$ es el tiempo codificado en los campos de base y extensión de referencia de reloj de sistema en unidades del reloj de sistema;

program_mux_rate es un campo definido en 2.5.3.3.

Después de la entrega del último byte de una saca puede haber un intervalo de tiempo durante el cual no se entreguen bytes a la entrada del P-STD.

2.5.2.3 Almacenamiento en memoria tampón

Los datos de paquete PES del tren elemental n pasan a la memoria tampón de entrada para el tren n , B_n . La transferencia del bit i de la entrada del decodificador-objetivo de sistemas a B_n es instantánea, de modo que el byte i entra en la memoria tampón para el tren n , de tamaño BS_n , en el tiempo $t(i)$.

Los bytes presentes en el encabezamiento de saca, encabezamientos de sistema mapas de tren de programa, directorios de tren de programa o encabezamientos de paquete PES del tren de programa, tales como los campos SCR, DTS, PTS y de longitud de paquete, no son entregados a ninguna de las memorias tampón pero pueden ser utilizados para controlar el sistema.

Los tamaños de memoria tampón de entrada BS_1 a BS_n son indicados por el parámetro de tamaño de memoria tampón P-STD en la sintaxis en las ecuaciones 2-16 y 2-17.

En el instante de decodificación, $td_n(j)$, todos los datos para la unidad de acceso que ha estado más tiempo en la memoria tampón, $A_n(j)$, y los bits de relleno que les precedan inmediatamente que estén presentes en la memoria tampón en el instante $td_n(j)$, se suprimen en ese momento, $td_n(j)$. El instante de decodificación $td_n(j)$ se especifica en los DTS o PTS. Los instantes de decodificación $td_n(j+1)$, $td_n(j+2)$, ... de unidades de acceso sin campos DTS o PTS codificados que sigan inmediatamente a la unidad de acceso j pueden derivarse de la información llevada en el tren de transporte. Véase el anexo C a la Rec. UIT-T H.262 | ISO/CEI 13818-2, ISO/CEI 13818-3, ISO/CEI 11172-2 o ISO/CEI 11172-3. Véase también 2.7.5. Al suprimirse la unidad de acceso en la memoria tampón, se decodifica instantáneamente para formar una unidad de presentación.

Se construirá el tren de programa y $t(i)$ se elegirá de tal modo que las memorias tampón de entrada de los tamaños BS_1 a BS_n no desborden ni queden subutilizadas en el decodificador-objetivo de sistema de programa. Esto es:

$$0 \leq F_n(t) \leq BS_n$$

para todas las t y n ,

y:

$$F_n(t) = 0$$

inmediatamente antes de $t = t(0)$.

$F_n(t)$ es el instante en que se llena la memoria tampón B_n del P-STD.

Una excepción a esta condición es que la memoria tampón B_n del P-STD puede ser subutilizada cuando la bandera de bajo retardo en el encabezamiento de secuencia de vídeo se pone a '1' (véase 2.4.2.6) o cuando el estado modo truco es verdadero (véase 2.4.3.8).

Para todos los trenes de programa, el retardo causado por el almacenamiento en la memoria tampón de la entrada del decodificador-objetivo de sistema será igual o inferior a 1 segundo, salvo en el caso de datos de vídeo de imagen fija y trenes ISO/CEI 14496. El tiempo de almacenamiento en la memoria tampón de entrada es la diferencia de tiempo entre el instante en que un byte entra en la memoria tampón de entrada y aquel en que es decodificado.

Específicamente, cuando no hay datos de vídeo de imagen fija ni tren de vídeo ISO/CEI 14496, este retardo está constreñido por:

$$td_n(j) - t(i) \leq 1 \text{ segundo}$$

cuando hay datos de vídeo de imagen fija, este retardo está constreñido por:

$$td_n(j) - t(i) \leq 60 \text{ segundos}$$

y cuando hay trenes ISO/CEI 14496, por:

$$td_n(j) - t(i) \leq 10 \text{ segundos}$$

para todos los bytes contenidos en la unidad de acceso j .

En el caso de trenes de programa, todos los bytes de cada paca entrarán en el decodificador P-STD antes de cualquier byte de la saca subsiguiente.

ISO/CEI 13818-1:2007 (S)

Cuando la bandera **low_delay (bajo retardo)** de la extensión de secuencia vídeo se pone a '1' (véase 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2), la memoria tampón VBV puede estar subutilizada. En este caso, cuando la memoria tampón del tren elemental P-STD, B_n , es examinada en el instante especificado por $td_n(j)$, los datos completos para la unidad de acceso pueden no estar presentes en la memoria tampón B_n . Cuando esto ocurre, la memoria tampón será reexaminada a intervalos de dos periodos de campo hasta que los datos para la unidad de acceso completa estén presentes en la memoria tampón. En este momento, toda la unidad de acceso será suprimida instantáneamente de la memoria tampón B_n .

La subutilización de la memoria tampón VBV puede producirse continuamente sin límites. El decodificador P-STD suprimirá datos de unidades de acceso de la memoria tampón B_n en el primer instante coherente con el párrafo anterior y cualesquiera valores de DTS o PTS codificados en el tren de bits. El decodificador puede ser incapaz de restablecer los tiempos correctos de decodificación y visualización indicados por DTS y PTS hasta que cese la situación de subutilización de la memoria tampón VBV y se encuentre una PTS o DTS en el tren de bits.

2.5.2.4 Trenes PES

Es posible construir un tren de datos como un tren contiguo de paquetes PES cada uno de los cuales contiene datos del mismo tren elemental con la misma identificación de tren. Este tren se denomina un tren PES. El modelo PES-STD para un tren PES es idéntico al del tren de programa, con la excepción de que se utiliza la referencia de reloj de tren elemental (ESCR) en lugar de la SCR y la velocidad ES en lugar de la velocidad del múltiplex de programa. El demultiplexor envía datos solamente a una memoria tampón de tren elemental.

Los tamaños de memoria tampón BS_n en el modelo PES-STD se definen como sigue:

- Para vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2:

$$BS_n = VBV_{\max}[\text{perfil, nivel}] + BS_{\text{oh}}$$

$BS_{\text{oh}} = (1/750)$ segundos $\times R_{\max}[\text{perfil, nivel}]$, donde $VBV_{\max}[\text{perfil, nivel}]$ y $R_{\max}[\text{perfil, nivel}]$ son el tamaño VBV y velocidad binaria por perfil, nivel y capa máximos definidos en los cuadros 8-14 y 8-13 de la Rec. UIT-T H.262 | ISO/CEI 13818-2. BS_{oh} está atribuida para tara de encabezamiento de paquetes PES.

- Para vídeo de ISO/CEI 11172-2:

$$BS_n = VBV_{\max} + BS_{\text{oh}}$$

$BS_{\text{oh}} = (1/750)$ segundos $\times R_{\max}$, donde R_{\max} y vbv_{\max} se refieren respectivamente a la velocidad binaria máxima y al tamaño máximo de memoria tampón vbv para un tren de bits de parámetros constreñidos de ISO/CEI 11172-2.

- Para audio de ISO/CEI 11172-3 o ISO/CEI 13818-3:

$$BS_n = 2848 \text{ bytes}$$

- Para vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10:

$$BS_n = 1200 \times \text{MaxCPB}[\text{level}] + BS_{\text{oh}}$$

Siendo $\text{MaxCPB}[\text{level}]$ el definido en el cuadro A.1 (Límites de nivel) en la Rec. UIT-T H.264 | ISO/CEI 14496-10 para cada nivel.

2.5.2.5 Decodificación y presentación

La decodificación y presentación en el decodificador-objetivo de sistemas del tren de programa son iguales a las definidas para el decodificador-objetivo de sistemas del tren de transporte en las cláusulas 2.4.2.4 y 2.4.2.5, respectivamente.

2.5.2.6 Extensiones de P-STD para el transporte de datos ISO/CEI 14496

Para decodificar los datos ISO/CEI 14496 de un tren de programa, se amplía el modelo de decodificador objetivo de tren de programa (P-STD). Para la decodificación de trenes elementales ISO-CEI 14496 en el P-STD, véase 2.11.2. La subcláusula 2.11.3 define las extensiones y parámetros del P-STD para decodificar escenas ISO/CEI 14496 y trenes asociados.

2.5.2.7 Extensiones P-STD para el transporte conforme a vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10

Para la decodificación de trenes de vídeo conformes a la Rec. UIT-T H.264 | ISO/CEI 14496-10 transportados en trenes de programa en el modelo P-STD, véase la subcláusula 2.14.3.2.

2.5.3 Especificación de la sintaxis y semántica del tren de programa

La siguiente sintaxis describe un tren de bits.

2.5.3.1 Tren de programa

Véase el cuadro 2-37.

Cuadro 2-37 – Tren de programa

Sintaxis	N.º de bits	Mnemónico
<pre>MPEG2_program_stream() { do { pack() } while (nextbits() == pack_start_code) MPEG_program_end_code }</pre>	32	bslbf

2.5.3.2 Definición semántica de campos en el tren de programa

MPEG_program_end_code (código de fin de programa MPEG) – Es la cadena de bits '0000 0000 0000 0000 0000 0001 1011 1001' (0x000001B9). Termina el tren de programa.

2.5.3.3 Capa de saca del tren de programa

Véanse los cuadros 2-38 y 2-39.

Cuadro 2-38 – Sacar de tren de programa

Sintaxis	N.º de bits	Mnemónico
<pre>pack() { pack_header() while (nextbits() == packet_start_code_prefix) { PES_packet() } }</pre>		

Cuadro 2-39 – Encabezamiento de saca de tren de programa

Sintaxis	N.º de bits	Mnemónico
pack_header() {		
pack_start_code	32	bslbf
'01'	2	bslbf
system_clock_reference_base [32..30]	3	bslbf
marker_bit	1	bslbf
system_clock_reference_base [29..15]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_base [14..0]	15	bslbf
marker_bit	1	bslbf
system_clock_reference_extension	9	uimsbf
marker_bit	1	bslbf
program_mux_rate	22	uimsbf
marker_bit	1	bslbf
marker_bit	1	bslbf
reserved	5	bslbf
pack_stuffing_length	3	uimsbf
for (i = 0; i < pack_stuffing_length; i++) {		
stuffing_byte	8	bslbf
}		
if (nextbits() == system_header_start_code) {		
system_header ()		
}		
}		

2.5.3.4 Definición semántica de campos en la saca de trenes de programa

pack_start_code (código de comienzo de saca) – Es la cadena de bits '0000 0000 0000 0000 0000 0001 1011 1010' (0x000001BA). Identifica el principio de una saca.

system_clock_reference_base (base de referencia de reloj de sistema); system_clock_reference_extension (extensión de referencia de reloj de sistema) – La referencia de reloj de sistema (SCR) es un campo de 42 bits codificado en dos partes. La primera parte, la base de referencia de reloj de sistema, es un campo de 33 bits cuyo valor viene dado por SCR_base(i) como se indica en la ecuación 2-19. La segunda parte, la extensión de referencia de reloj de sistema, es un campo de 9 bits cuyo valor viene dado por SCR_ext(i) en la ecuación 2-20. La SCR indica el tiempo de llegada previsto del byte que contiene el último bit de la base de referencia de reloj de sistema a la entrada del decodificador-objetivo de programa.

Los requisitos relativos a la frecuencia de codificación del campo SCR se indican en 2.7.1.

marker_bit (bit marcador) – Es un campo de 1 bit que tiene el valor de '1'.

program_mux_rate (velocidad de múltiplex de programa) – Es un entero de 22 bits que especifica la velocidad a la cual el P-STD recibe el tren de programa durante la saca en la cual está incluido. El valor de la velocidad de múltiplex de programa se mide en unidades de 50 bytes/segundo. El valor 0 está prohibido. El valor representado en la velocidad de múltiplex de programa se utiliza para definir el tiempo de llegada de bytes a la entrada del P-STD en 2.5.2. El valor codificado en el campo de velocidad de múltiplex de programa puede variar de una saca a otra en un tren multiplexado de programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

pack_stuffing_length (longitud de relleno de saca) – Un entero de 3 bits que especifica el número de bytes de relleno que siguen a este campo.

stuffing_byte (byte de relleno) – Es un valor de 8 bits fijo que equivale a '1111 1111' que puede ser insertado por el codificador, por ejemplo, para satisfacer las necesidades del canal. Es descartado por el decodificador. En cada encabezamiento de saca no estarán presentes más de 7 bytes de relleno.

2.5.3.5 Encabezamiento de sistema

Véase el cuadro 2-40.

Cuadro 2-40 – Encabezamiento de sistema de tren de programa

Sintaxis	N.º de bits	Mnemónico
system_header () {		
system_header_start_code	32	bslbf
header_length	16	uimsbf
marker_bit	1	bslbf
rate_bound	22	uimsbf
marker_bit	1	bslbf
audio_bound	6	uimsbf
fixed_flag	1	bslbf
CSPS_flag	1	bslbf
system_audio_lock_flag	1	bslbf
system_video_lock_flag	1	bslbf
marker_bit	1	bslbf
video_bound	5	uimsbf
packet_rate_restriction_flag	1	bslbf
reserved_bits	7	bslbf
while (nextbits () == '1') {		
stream_id	8	uimsbf
'11'	2	bslbf
P-STD_buffer_bound_scale	1	bslbf
P-STD_buffer_size_bound	13	uimsbf
}		
}		

2.5.3.6 Definición semántica de campos en el encabezamiento de sistema

system_header_start_code (código de comienzo de encabezamiento de sistemas) – Es la cadena de bits '0000 0000 0000 0000 0000 0001 1011 1011' (0x000001BB). Identifica el principio de un encabezamiento de sistema.

header_length (longitud de encabezamiento) – Este campo de 16 bits indica la longitud en bytes del encabezamiento de sistema que sigue al campo de longitud de encabezamiento. Obsérvese que futuras ampliaciones de la presente Especificación pueden ampliar el encabezamiento de sistema.

rate_bound (límite de velocidad) – Un campo de 22 bits. Es un valor entero mayor o igual que el valor máximo del campo de velocidad de multiplex de programa codificado en cualquier saca del tren de programa. Puede ser utilizado por un decodificador para evaluar si es capaz de decodificar todo el tren.

audio_bound (límite de audio) – Un campo de 6 bits. Es un entero en una gama de 0 a 32 inclusive y se fija a un valor mayor o igual que el número máximo de trenes de audio ISO/CEI 13818-3 e ISO/CEI 11172-3 en el tren de programa para el cual los procesos de decodificación están activos simultáneamente. A los efectos de esta subcláusula, el proceso de decodificación de un tren de audio de ISO/CEI 13818-3 o ISO/CEI 11172-3 está activo si la memoria tampón STD no está vacía o si una unidad de presentación está siendo presentada en el modelo P-STD.

fixed_flag (bandera fija) – La bandera fija es una bandera de 1 bit. Si está puesta a '1' indica el funcionamiento a velocidad binaria fija. Si está puesta a '0' indica el funcionamiento a velocidad binaria variable. Durante el funcionamiento a velocidad binaria fija, el valor codificado en todos los campos de referencia de reloj de sistema en el tren multiplexado Rec. UIT-T H.222.0 | ISO/CEI 13818-1 respetará la siguiente ecuación lineal:

$$SCR_base(i) = ((c1 \times i + c2) DIV 300) \% 2^{33} \quad (2-22)$$

$$SCR_ext(i) = ((c1 \times i + c2) DIV 300) \% 300 \quad (2-23)$$

ISO/CEI 13818-1:2007 (S)

donde:

- c1 es una constante de valor real válida para todas las i;
- c2 es una constante de valor real válida para todas las i;
- i es el índice en el tren multiplexado Rec. UIT-T H.222.0 | ISO/CEI 13818-1 del byte que contiene el bit final de cualquier campo de referencia de reloj de sistema del tren.

CSPS_flag (bandera CSPS) – Es un campo de 1 bit. Si está puesto a '1', el tren de programa satisface las constricciones definidas en 2.7.9.

system_audio_lock_flag (bandera de enganche de audio de sistema) – Es un campo de 1 bit que indica que hay una relación racional constante especificada entre la velocidad de muestra de audio y la frecuencia de reloj de sistema en el decodificador-objetivo de sistemas. En 2.5.2.1 se define la frecuencia de reloj de sistema y la velocidad de muestreo de audio se especifica en ISO/CEI 13818-3. La bandera de enganche de audio de sistema sólo se puede poner a '1' si, para todas las unidades de presentación en todos los trenes elementales de audio en el tren de programa, la relación de la frecuencia de reloj de sistema con la velocidad de muestreo de audio real, SCASR, es constante e igual al valor indicado en el siguiente cuadro a la velocidad de muestreo nominal indicada en el tren de audio.

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_P-STD} \quad (2-24)$$

La notación $\frac{X}{Y}$ denota división real.

Frecuencia de muestro de audio nominal (kHz)	16	32	22,05	44,1	24	48
SCASR	$\frac{27\ 000\ 000}{16\ 000}$	$\frac{27\ 000\ 000}{32\ 000}$	$\frac{27\ 000\ 000}{22\ 050}$	$\frac{27\ 000\ 000}{44\ 100}$	$\frac{27\ 000\ 000}{24\ 000}$	$\frac{27\ 000\ 000}{48\ 000}$

system_video_lock_flag (bandera de enganche de vídeo de sistema) – Es un campo de 1-bit que indica que la relación entre la base de tiempos de vídeo y la frecuencia del reloj del sistema es constante en el decodificador de sistema de que se trata. El system_video_lock_flag sólo se puede poner a '1' si, para todas las unidades de presentación en todos los trenes elementales de vídeo en el programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la relación entre la frecuencia del reloj del sistema (system_clock_frequency) y la frecuencia de la base de tiempos del vídeo actual es constante.

Para trenes de vídeo conformes a ISO/CEI 11172-2 y Rec. UIT-T H.262 | ISO/CEI 13818-2, si el system_video_lock_flag está puesto a '1', la relación entre system_clock_frequency y la velocidad de cuadro de vídeo actual, SCFR, será constante e igual al valor indicado en el siguiente cuadro a la velocidad de cuadros nominal indicada en el tren de vídeo.

Para trenes de vídeo conformes a ISO/CEI 14496-2, si el system_video_lock_flag está puesto a '1', la base de tiempo del tren de vídeo ISO/CEI 14496-2, definida por vop_time_increment_resolution, estará vinculada al STC y será exactamente igual a N veces la frecuencia de reloj del sistema (system_clock_frequency) dividida por K, siendo N y K enteros que tienen un valor fijo dentro de cada secuencia de objeto visual, con K mayor o igual a N.

Para trenes de vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10, la frecuencia de la base de tiempos AVC se define mediante el parámetro AVC time_scale (escala de tiempos). Si el system_video_lock_flag está puesto a '1' para un tren de vídeo AVC, la frecuencia de la base de tiempos AVC estará vinculada al STC y será exactamente igual a N veces la frecuencia de reloj del sistema dividida por K, siendo N y K enteros que tienen un valor fijo dentro de cada secuencia de vídeo AVC, con K mayor o igual que N.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_P-STD} \quad (2-25)$$

Velocidad de trama nominal (Hz)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

Los valores de la relación SCFR son exactos. La velocidad de trama real difiere ligeramente de la velocidad nominal cuando la velocidad nominal es 23,976, 29,97 ó 59,94 tramas por segundo.

video_bound (límite de vídeo) – El video_bound es un entero de 5-bits en la gama comprendida entre 0 y 16 y su valor es mayor o igual que el número máximo de trenes de vídeo en el tren de programa en el cual los procesos de decodificación están activos simultáneamente. A efectos de esta subcláusula, el proceso de decodificación de un tren de vídeo está activo si una de las memorias intermedias en el modelo P-STD no está vacía, o si la unidad de presentación se está presentando en el modelo P-STD.

packet_rate_restriction_flag (bandera de restricción de la velocidad de paquetes) – Es una bandera de 1 bit. Si la bandera de CSPS está puesta a '1', la bandera de restricción de la velocidad de paquetes indica la restricción que es aplicable a la velocidad de paquetes según lo especificado en 2.7.9. Si la bandera de CSPS está puesta a '0', el significado de la bandera de restricción de la velocidad de paquetes no está definido.

reserved_bits (bits reservados) – Este campo de 7 bits está reservado para uso futuro por ISO/CEI. Hasta que el UIT-T | ISO/CEI especifiquen otra cosa, tendrá el valor '1111 1111'.

stream_id (identificador de tren) – Es un campo de 8 bits que indica la codificación y número de tren elemental del tren al cual se refieren los siguientes campos P-STD_buffer_bound_scale y P-STD_buffer_size_bound.

Si el identificador de tren equivale a '1011 1000', los campos P-STD_buffer_bound_scale y P-STD_buffer_size_bound que siguen al identificador de tren se refieren a todos los trenes de audio en el tren de programa.

Si el identificador de tren equivale a '1011 1001', los campos P-STD_buffer_bound_scale que siguen al identificador de tren se refieren a todos los trenes videos en el tren de programa.

Si el identificador de tren toma cualquier otro valor, será un valor de byte mayor o igual que '1011 1100' y se interpretará que se refiere a la codificación del tren y número de tren elemental de acuerdo con el cuadro 2-22.

Cada tren elemental presente en el tren de programa tendrá sus P-STD_buffer_bound_scale y P-STD_buffer_size_bound especificado exactamente una vez por este mecanismo en cada encabezamiento de sistema.

P-STD_buffer_bound_scale (escala de límite de memoria tampón P-STD) – Es un campo de 1 bit que indica el factor de escalonamiento utilizado para interpretar el siguiente campo P-STD_buffer_size_bound. Si el indicador de tren precedente indica un tren de audio, P-STD_buffer_bound_scale tendrá el valor '0'. Si el indicador de tren precedente indica un tren de vídeo, P-STD_buffer_bound_scale tendrá el valor '1'. Para todos los otros tipos de tren, el valor de P-STD_buffer_bound_scale será '1' o '0'.

P-STD_buffer_size_bound (límite de tamaño de memoria tampón P-STD) – Es un entero sin signo de 13 bits que define un valor mayor o igual que el tamaño máximo de la memoria tampón de entrada P-STD, BS_n , en todos los paquetes para el tren n en el tren de programa. Si P-STD_buffer_bound_scale tiene el valor '0', P-STD_buffer_size_bound mide el límite de tamaño de memoria tampón en unidades de 128 bytes. Si P-STD_buffer_bound_scale tiene el valor '1', P-STD_buffer_size_bound mide el límite de tamaño de memoria tampón en unidades de 1024 bytes. Por tanto:

$$\begin{aligned} & \text{si } (P-STD_buffer_bound_scale == 0) \\ & BS_n \leq P-STD_buffer_size_bound \times 128 \end{aligned}$$

si no:

$$BS_n \leq P-STD_buffer_size_bound \times 1024$$

2.5.3.7 Capa de paquete del tren de programa

La capa de paquete del tren de programa es definida por la capa de paquete de PES, véase 2.4.3.6.

2.5.4 Correspondencia de trenes de programa

La correspondencia de trenes de programa (PSM, *program stream map*) proporciona una descripción de los trenes elementales en el tren de programa y su relación entre sí. Cuando va en un tren de transporte, esta estructura no será modificada. La correspondencia de trenes de programa está presente como un paquete PES cuando el valor de identificador de tren es 0xBC (véase el cuadro 2-22).

NOTA – Esta sintaxis difiere de la sintaxis de paquetes PES descrita en 2.4.3.6.

La definición de los campos de descriptor () figura en 2.6.

2.5.4.1 Sintaxis de la correspondencia de trenes de programa

Véase el cuadro 2-41.

Cuadro 2-41 – Correspondencia de trenes de programa

Sintaxis	N.º de bits	Mnemónico
program_stream_map() {		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimsbf
program_stream_map_length	16	uimsbf
current_next_indicator	1	bslbf
reserved	2	bslbf
program_stream_map_version	5	uimsbf
reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
elementary_stream_map_length	16	uimsbf
for (i = 0; i < N1; i++) {		
stream_type	8	uimsbf
elementary_stream_id	8	uimsbf
elementary_stream_info_length	16	uimsbf
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

2.5.4.2 Definición semántica de campos en la correspondencia de trenes de programa

packet_start_code_prefix (prefijo de código de comienzo de paquetes) – Es un código de 24 bits. Junto con el `map_stream_id` que sigue constituye un código de comienzo de paquete que identifica el principio de un paquete. El prefijo de código de comienzo de paquete es la cadena de bits '0000 0000 0000 0000 0000 0001' (0x000001 en hexadecimal).

map_stream_id (identificador de correspondencia de tren) – Es un campo de 8 bits cuyo valor será 0xBC.

program_stream_map_length (longitud de correspondencia de trenes de programas) – Es un campo de 16 bits que indica el número total de bytes en la correspondencia de trenes de programas que sigue inmediatamente a este campo. El valor máximo de este campo es 1018 bytes (0x3FA).

current_next_indicator (indicador siguiente vigente) – Es un campo de 1 bit, que cuando se pone a '1' indica que la correspondencia de trenes de programa enviada es aplicable actualmente. Cuando el bit se pone a '0' indica que la correspondencia de trenes de programa enviada no es aún aplicable y será la siguiente tabla que será válida.

program_stream_map_version (versión de correspondencia de trenes de programas) – Este campo de 5 bits es el número de versión de toda la correspondencia de trenes de programa. El número de versión se incrementará en 1, módulo 32, cada vez que cambia la definición de la correspondencia de trenes de programa. Cuando el `current_next_indicator` se pone a '1', la versión de correspondencia de tren de programa será la de la correspondencia de tren de programa actualmente aplicable, y cuando se pone a '0', la versión de correspondencia de trenes de programa será la de la siguiente correspondencia de tren de programa aplicable.

program_stream_info_length (longitud de información de tren de programa) – Es un campo de 16 bits que indica la longitud total de los descriptores que siguen inmediatamente a este campo.

marker_bit (bit marcador) – Es un campo de 1 bit que tiene el valor de '1'.

elementary_stream_map_length (longitud de correspondencia de tren elemental) – Es un campo de 16 bits que especifica la longitud total, en bytes, de toda la información de tren elemental en esta correspondencia de trenes de programa. Incluye los campos de tipo de tren, id de tren elemental y longitud de información de tren elemental.

stream_type (tipo de tren) – Es un campo de 8 bits que especifica el tipo del tren elemental de acuerdo con el cuadro 2-34. El campo de tipo de tren sólo identificará trenes elementales contenidos en paquetes PES. Un valor 0x05 está prohibido.

elementary_stream_id (identificador de tren elemental) – Es un campo de 8 bits que indica el valor del campo de identificador de tren en los encabezamientos de paquetes PES en los cuales está almacenado este tren elemental.

elementary_stream_info_length (longitud de información de tren elemental) – Es un campo de 16 bits que indica la longitud, en bytes, de los descriptores que siguen inmediatamente a este campo.

CRC_32 – Es un campo de 32 bits que contiene el valor CRC que da una salida de cero de los registradores en el decodificador definido en el anexo A después de procesar toda la correspondencia de trenes de programa.

2.5.5 Directorio del tren de programa

El directorio para un tren completo se establece con todos los datos de directorio transportados por paquetes de directorio de tren de programa identificados con `directory_stream_id` (identificador de tren de directorio). La sintaxis para los paquetes de `program_stream_directory` (directorio de tren de programa) se define en el cuadro 2-42.

NOTA 1 – Esta sintaxis difiere de la sintaxis de paquete PES descrita en 2.4.3.6.

Se pueden necesitar entradas de directorio para referenciar imágenes I en un tren vídeo definido en la Rec. UIT-T H.262 | ISO/CEI 13818-2 y en ISO/CEI 11172-2. Si una imagen I que está referenciada en una entrada al directorio está precedida por un encabezamiento de secuencia sin que intervengan encabezamientos de imagen, la entrada de directorio referenciará el primer byte de encabezamiento de la secuencia. Si una imagen I que es referenciada en una entrada de directorio está precedida por un encabezamiento del grupo de imágenes sin que intervengan encabezamientos de imágenes y sin que preceda inmediatamente al encabezamiento de secuencia, la entrada de directorio referenciará el primer byte del encabezamiento del grupo de imágenes. Cualquier otra imagen que referencie una entrada de directorio será referenciada por el primer byte del encabezamiento de imagen.

NOTA 2 – Se recomienda que las imágenes I que siguen inmediatamente a un encabezamiento de secuencia sean referenciadas en estructuras de directorio de modo que el directorio contenga una entrada en cada punto donde el decodificador puede ser reiniciado completamente.

Las entradas al directorio pueden ser obligatorias para la imagen o imágenes IDR de referencia correspondientes a un mensaje SEI de punto de recuperación en un tren de vídeo AVC. Cada una de estas entradas de directorio se referirá al primer byte en una unidad de acceso AVC.

Las referencias del directorio a trenes de audio definidos en ISO/CEI 13818-3 e ISO/CEI 11172-3 serán la palabra de sincronización de la trama de audio.

NOTA 3 – Se recomienda que la distancia entre unidades de acceso referenciadas no exceda de medio segundo.

Las unidades de acceso se referenciarán por paquetes de directorio de tren de programa en el mismo orden que aparecen en el tren de bits.

2.5.5.1 Sintaxis del paquete para el directorio del tren de programa

Véase el cuadro 2-42.

Cuadro 2-42 – Sintaxis del paquete para el directorio del tren de programa

Sintaxis	N.º de bits	Mnemónico
directory_PES_packet(){		
packet_start_code_prefix	24	bslbf
directory_stream_id	8	uimsbf
PES_packet_length	16	uimsbf
number_of_access_units	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
for (i = 0; i < number_of_access_units; i++) {		
packet_stream_id	8	uimsbf
PES_header_position_offset_sign	1	tcimsbf
PES_header_position_offset[43..30]	14	uimsbf
marker_bit	1	bslbf
PES_header_position_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
PES_header_position_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
reference_offset	16	uimsbf
marker_bit	1	bslbf
reserved	3	bslbf
PTS[32..30]	3	uimsbf
marker_bit	1	bslbf
PTS[29..15]	15	uimsbf
marker_bit	1	bslbf
PTS[14..0]	15	uimsbf
marker_bit	1	bslbf
bytes_to_read[22..8]	15	uimsbf
marker_bit	1	bslbf
bytes_to_read[7..0]	8	uimsbf
marker_bit	1	bslbf
intra_coded_indicator	1	bslbf
coding_parameters_indicator	2	bslbf
reserved	4	bslbf
}		
}		

2.5.5.2 Definición semántica de campos en el directorio del tren de programa

packet_start_code_prefix (prefijo de código de comienzo de paquete) – Es un código de 24 bits. Junto con el identificador de tren que le sigue constituye un código de comienzo de paquete que identifica el principio de un paquete. El prefijo de código de comienzo de paquete es la cadena de bits '0000 0000 0000 0000 0000 0001' (0x000001 en hexadecimal).

directory_stream_id (identificador de tren de directorio) – Es un campo de 8 bits que tendrá un valor '1111 1111' (0xFF).

PES_packet_length (longitud de paquete PES) – Es un campo de 16 bits que indica el número total de bytes en el directorio del tren de programa que sigue inmediatamente a este campo (véase el cuadro 2-22).

number_of_access_unit (número de unidades de acceso) – Este campo de 15 bits es el número de unidades de acceso que son referenciadas en este paquete PES de directorio.

prev_directory_offset (desplazamiento de información de directorio previo) – Este entero sin signo de 45 bits da el desplazamiento de la dirección en bytes del primer byte del código de comienzo del paquete del anterior paquete de directorio del tren de programa. Este desplazamiento de dirección se relaciona con el primer byte del código de comienzo del paquete que contiene este campo de previous_directory_offset. El valor '0' indica que no hay ningún paquete de directorio de tren de programa previo.

next_directory_offset (desplazamiento de directorio siguiente) – Este entero sin signo de 45 bits da el desplazamiento de la dirección en bytes del primer byte del código de comienzo de paquete del siguiente paquete de directorio de tren de programa. Este desplazamiento de dirección se relaciona con el primer byte del código de comienzo del paquete que contiene este campo next_directory_offset. El valor '0' indica que no hay un siguiente paquete de directorio de tren de programa.

packet_stream_id (identificador de tren de paquete) – Este campo de 8 bits es el identificador de tren del tren elemental que contiene la unidad de acceso referenciada por esta entrada de directorio.

PES_header_position_offset_sign (signo de desplazamiento de posición de encabezamiento PES) – Este campo de 1 bit es el signo aritmético para el desplazamiento de posición de encabezamiento PES descrito que sigue inmediatamente. Un valor de '0' indica que PES_header_position_offset es un desplazamiento positivo. Un valor de '1' indica que PES_header_position_offset es un desplazamiento negativo.

PES_header_position_offset (desplazamiento de posición de encabezamiento PES) – Este entero sin signo de 44 bits da la dirección de desplazamiento en bytes del primer byte del paquete PES que contiene la unidad de acceso referenciada. La dirección de desplazamiento se relaciona con el primer byte del código de comienzo del paquete que contiene el campo PES_header_position_offset. El valor '0' indica que no se ha referenciado ninguna unidad de acceso.

reference_offset (desplazamiento de referencia) – Este campo de 16 bits es un entero sin signo que indica la posición del primer byte de la unidad de acceso referenciada, medida en bytes, con respecto al primer byte del paquete PES que contiene el primer byte de la unidad de acceso referenciada.

PTS (presentation_time_stamp) (indicación de tiempo de presentación) – Este campo de 33 bits es la PTS de la unidad de acceso referenciada. La semántica de la codificación del campo PTS se describe en 2.4.3.6.

bytes_to_read (bytes para leer) – Este es un entero sin signo de 23 bits que da el número de bytes en el tren del programa después del byte indicado por el desplazamiento de referencia que se necesita para decodificar la unidad de acceso completamente. Este valor incluye cualesquiera bytes multiplexados en la capa de sistemas, así como los que contienen información de otros trenes.

intra_coded_indicator (indicador intracodificado) – Esta es una bandera de 1 bit que puesto a '1' indica que la unidad de acceso referenciada no está codificada con predicción. Es independiente de otros parámetros de codificación que pudieran necesitarse para decodificar la unidad de acceso. Por ejemplo, para tramas intracodificadas de vídeo, este campo se codificará como '1', mientras que para tramas 'P' y 'B' este bit se codificará como '0'. Para todos los paquetes PES que contienen datos que no son del tren de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2, este campo es indefinido (véase el cuadro 2-43).

Cuadro 2-43 – Indicador intracodificado

Valor	Significado
0	No intracodificado
1	Intracodificado

coding_parameters_indicator (indicador de parámetros de codificación) – Este campo de 2 bits se usa para indicar la ubicación de parámetros de codificación que se necesitan para decodificar las unidades de acceso referenciadas. Por ejemplo, este campo se puede utilizar para determinar la ubicación de matrices de cuantificación para tramas vídeo.

Cuadro 2-44 – Indicador de parámetros de codificación

Valor	Significado
00	Todos los parámetros de codificación están puestos a sus valores por defecto
01	Todos los parámetros de codificación están puestos en esta unidad de acceso, por lo menos uno de ellos no está puesto a un valor por defecto
10	Algunos parámetros de codificación están puestos en esta unidad de acceso
11	Ningún parámetro de codificación está codificado en esta unidad de acceso

2.6 Descriptores de programas y de elementos de programa

Los descriptores de programas y de elementos de programa son estructuras que se pueden utilizar para ampliar las definiciones de programas y de elementos de programa. Todos los descriptores tienen un formato que comienza con un valor de rótulo de 8 bits. El valor de rótulo está seguido por campos de longitud y de datos de descriptor de 8 bits.

2.6.1 Definición semántica de campos de descriptores de programas y de elementos de programa

La siguiente semántica se aplica a los descriptores definidos en 2.6.2 a 2.6.34.

descriptor_tag (rótulo de descriptor) – El rótulo de descriptor es un campo de 8 bits que identifica cada descriptor.

El cuadro 2-45 proporciona los valores de rótulo de descriptor definidos y reservados en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, y disponibles al usuario. Una 'X' en las columnas TS o PS indica la aplicabilidad del descriptor al tren de transporte o al tren de programa, respectivamente. Obsérvese que el significado de los campos en un descriptor puede depender del tren que se utiliza en el mismo. Cada caso se especifica a continuación en la semántica de descriptores.

descriptor_length (longitud de descriptor) – Es un campo de 8 bits que especifica el número de bytes del descriptor que sigue inmediatamente al campo de longitud de descriptor.

Cuadro 2-45 – Descriptores de programa y elementos de programa

descriptor_tag	TS	PS	Identificación
0	n/a	n/a	Reservado
1	n/a	n/a	Reservado
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	Video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	System_clock_descriptor
12	X	X	Multiplex_buffer_utilization_descriptor
13	X	X	Copyright_descriptor
14	X		Maximum_bitrate_descriptor
15	X	X	Private_data_indicator_descriptor
16	X	X	Smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Definido en ISO/CEI 13818-6

Cuadro 2-45 – Descriptores de programa y elementos de programa

descriptor_tag	TS	PS	Identificación
27	X	X	MPEG-4_video_descriptor
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor
31	X	X	FMC_descriptor
32	X	X	External_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor
35	X		MultiplexBuffer_descriptor
36	X	X	Content_labeling_descriptor
37	X	X	Metadata_pointer_descriptor
38	X	X	Metadata_descriptor
39	X	X	Metadata_STD_descriptor
40	X	X	Descriptor de vídeo AVC
41	X	X	IPMP_descriptor (definido en ISO/CEI 13818-11, MPEG-2 IPMP)
42	X	X	Descriptor de HRD y temporización AVC
43	X	X	MPEG-2_AAC_audio_descriptor
44	X	X	FlexMuxTiming_descriptor
45-63	n/a	n/a	Reservado Rec. UIT-T H.222.0 ISO/CEI 13818-1
64-255	n/a	n/a	Privado de usuario

2.6.2 Descriptor de tren de vídeo

El descriptor de tren de vídeo proporciona información básica que identifica los parámetros de codificación de un tren elemental de vídeo descrito en la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 11172-2 (véase el cuadro 2-46).

Cuadro 2-46 – Descriptor de tren de vídeo

Sintaxis	N.º de bits	Mnemónico
video_stream_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
multiple_frame_rate_flag	1	bslbf
frame_rate_code	4	uimsbf
MPEG_1_only_flag	1	bslbf
constrained_parameter_flag	1	bslbf
still_picture_flag	1	bslbf
if (MPEG_1_only_flag == '0'){		
profile_and_level_indication	8	uimsbf
chroma_format	2	uimsbf
frame_rate_extension_flag	1	bslbf
reserved	5	bslbf
}		
}		

2.6.3 Definición semántica de campos en el descriptor de tren de vídeo

multiple_frame_rate_flag (bandera de velocidad de trama múltiple) – Este es un campo de 1 bit que cuando está puesto a '1' indica que múltiples velocidades de trama pueden estar presentes en el tren de vídeo. Cuando está puesto a '0' solamente está presente una velocidad de trama.

frame_rate_code (código de velocidad de trama) – Este es un campo de 4 bits definido en 6.3.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, salvo que cuando la bandera de velocidad de trama múltiple está puesta a un valor de '1', la indicación de una velocidad de trama particular permite también que algunas otras velocidades de trama estén presentes en el tren vídeo, como se especifica en el cuadro 2-47.

Cuadro 2-47 – Código de velocidad de trama

Codificado como	Incluye también
23,976	
24,0	23,976
25,0	
29,97	23,976
30,0	23,976 24,0 29,97
50,0	25,0
59,94	23,976 29,97
60,0	23,976 24,0 29,97 30,0 59,94

MPEG_1_only_flag (bandera de MPEG solamente) – Es un campo de un bit que cuando se pone a '1' indica que el tren de vídeo contiene solamente datos ISO/CEI 11172-2. Cuando se pone a '0', el tren de vídeo puede contener datos de vídeo Rec. UIT-T H.262 | ISO/CEI 13818-2 y datos de vídeo ISO/CEI 11172-2 con parámetros constreñidos.

constrained_parameter_flag (bandera de parámetro constreñido) – Es un campo de un bit que cuando se pone a '1' indica que el tren de vídeo no contendrá datos de vídeo ISO/CEI 11172-2 no constreñidos. Cuando se pone a '0', el tren de vídeo puede contener parámetros constreñidos y trenes de vídeo ISO/CEI 11172-2 no constreñidos. Si MPEG_1_only_flag está puesta a '0', la bandera de parámetro constreñido debe ponerse a '1'.

still_picture_flag (bandera de imágenes fijas) – Este es un campo de un bit que puesto a '1' indica que el tren vídeo contiene solamente imágenes fijas. Si el bit está puesto a '0' el tren de vídeo puede contener datos de imágenes fijas o en movimiento.

profile_and_level_indication (indicación de perfil y nivel) – Este es un campo de 8 bits que se codifica igual que los campos `profile_and_level_indication` del tren vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2. El valor de este campo indica un perfil y un nivel más altos o al menos iguales que cualquier perfil y nivel en cualquier secuencia en el tren de vídeo asociado. A los efectos de esta subcláusula, se considera que un tren de parámetros constreñidos de ISO/CEI 11172-2 es un tren de perfil principal y nivel bajo (MP @ LL).

chroma_format (formato de cromaticidad) – Este es un campo de 2 bits que se codifica igual que los campos `chroma_format` del tren de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2. El valor de este campo será más alto o al menos igual que el valor del campo de formato de cromaticidad en cualquier secuencia vídeo el tren de vídeo asociado. A los efectos de esta subcláusula, se considera que un tren de vídeo de ISO/CEI 11172-2 tenga un campo de formato de cromaticidad con un valor '01', indicando 4:2:0.

frame_rate_extension_flag (bandera de extensión de velocidad de trama) – Esta es una bandera de 1 bit que puesta a 1 indica que cualquiera o ambos de los campos `frame_rate_extension_n` y `frame_rate_extension_d` son no cero en cualesquiera secuencias de vídeo del tren de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2. A los efectos de esta subcláusula, un tren de vídeo de ISO/CEI 11172-2 está constreñido a tener ambos campos puestos a cero.

2.6.4 Descriptor de tren de audio

El descriptor de tren de audio proporciona información básica que identifica la versión de codificación de un tren elemental de audio descrito en ISO/CEI 13818-3 o ISO/CEI 11172-3 (véase el cuadro 2-48).

Cuadro 2-48 – Descriptor de tren de audio

Sintaxis	N.º de bits	Mnemónico
audio_stream_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
free_format_flag	1	bslbf
ID	1	bslbf
Layer	2	bslbf
variable_rate_audio_indicator	1	bslbf
Reserved	3	bslbf
}		

2.6.5 Definición semántica de campos en el descriptor de tren de audio

free_format_flag (bandera de formato libre) – Este es un campo de 1 bit que puesto a '1' indica que los datos del tren de audio pueden contener una o más tramas de audio con el índice de velocidad binaria (*bitrate_index*) puesto a '0000'. Si está puesto a '0', el índice de velocidad binaria no es '0000' (véase 2.4.2.3 de ISO/CEI 13818-3) en cualquier trama de audio del tren de audio.

ID – Este es un campo de 1 bit que puesto a '1' indica que el campo ID está puesto a '1' en cada trama de audio del tren de audio (véase 2.4.2.3 de ISO/CEI 13818-3).

layer (capa) – Este es un campo de 2 bits que se codifica igual que los campos de capa del tren de audio de ISO/CEI 13818-3 o ISO/CEI 11172-3 (véase 2.4.2.3 de ISO/CEI 13818-3). La capa indicada en este campo será más alta o al menos igual que la capa más alta especificada en cualquier trama de audio del tren de audio.

variable_rate_audio_indicator (indicador de audio de velocidad variable) – Es una bandera de un bit que, cuando se pone a '0' indica que el valor codificado del campo de velocidad binaria no cambia en las tramas de audio consecutivas que han de ser presentadas sin discontinuidad.

2.6.6 Descriptor de jerarquía

El descriptor de jerarquía proporciona información para identificar los elementos de programa que contienen componentes de audio y vídeo codificados jerárquicamente y trenes privados. (Véase el cuadro 2-49.)

Cuadro 2-49 – Descriptor de jerarquía

Sintaxis	N.º de bits	Mnemónico
hierarchy_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	4	bslbf
hierarchy_type	4	uimsbf
reserved	2	bslbf
hierarchy_layer_index	6	uimsbf
reserved	2	bslbf
hierarchy_embedded_layer_index	6	uimsbf
reserved	2	bslbf
hierarchy_channel	6	uimsbf
}		

2.6.7 Definición semántica de campos en el descriptor de jerarquía

hierarchy_type (tipo de jerarquía) – La relación jerárquica entre la capa de jerarquía asociada y su capa insertada de jerarquía se define en el cuadro 2-50.

hierarchy_layer_index (índice de capa de jerarquía) – Es un campo de 6 bits que define un índice único del elemento de programa asociado en una tabla de jerarquías de capas de codificación. Los índices serán únicos dentro de una definición de programa.

hierarchy_embedded_layer_index (índice de capa insertada de jerarquía) – Es un campo de 6 bits que define el índice de tablas de jerarquía del elemento de programa al que hay que acceder antes de la decodificación del tren elemental asociado con este descriptor de jerarquía. Este campo no se define si el valor del tipo de jerarquía es 15 (capa básica).

hierarchy_channel (canal de jerarquía) – El canal de jerarquía es un campo de 6 bits que indica el número de canal previsto para el elemento de programa asociado en un conjunto ordenado de canales de transmisión. El canal de transmisión más robusto se define por el valor más bajo de este canal con respecto a la definición de la jerarquía de transmisión global.

NOTA – Un canal de jerarquía puede estar asignado a varios elementos de programa al mismo tiempo.

Cuadro 2-50 – Valores del campo de tipo de jerarquía

Valor	Descripción
0	Reservado
1	Escalonabilidad espacial
2	Escalonabilidad SNR
3	Escalonabilidad temporal
4	Partición de datos
5	Tren de bits de extensión
6	Tren privado
7	Perfil multivisión
8-14	Reservado
15	Capa básica

2.6.8 Descriptor de registro

El descriptor de registro proporciona un método para identificar de manera única e inequívoca formatos de datos privados (véase el cuadro 2-51).

Cuadro 2-51 – Descriptor de registro

Sintaxis	N.º de bits	Mnemónico
registration_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
format_identifier	32	uimsbf
for (i = 0; i < N; i++){		
additional_identification_info	8	bslbf
}		
}		

2.6.9 Definición semántica de campos en el descriptor de registro

format_identifier (identificador de formato) – Es un valor de 32 bits obtenido de una autoridad de registro designada por ISO/CEI JTC 1/SC 29.

additional_identification_info (información de identificación adicional) – El significado de los bytes de información de identificación adicional, si los hubiere, es definido por el que tiene asignado ese identificador de formato y, una vez definido, no cambiará.

2.6.10 Descriptor de alineación de tren de datos

El descriptor de alineación de tren de datos describe qué tipo de alineación está presente en el tren elemental asociado. Si el indicador de alineación de datos en el encabezamiento de paquete PES está puesto a '1' y el descriptor está presente, se requiere alineación como se especifica en este descriptor (véase el cuadro 2-52).

Cuadro 2-52 – Descriptor de alineación de tren de datos

Sintaxis	N.º de bits	Mnemónico
data_stream_alignment_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
alignment_type	8	uimsbf
}		

2.6.11 Definición semántica de campos en el descriptor de alineación de tren de datos

alignment_type (tipo de alineación) – El cuadro 2-53 describe el tipo de alineación para vídeo conforme a ISO/CEI 11172-2, la Rec. UIT-T H.262 | ISO/CEI 13818-2 o trenes visuales ISO/CEI 14496-2, cuando el indicador de alineación de datos en la cabecera del paquete PES tiene un valor de '1'. Para estos trenes de vídeo, el primer PES_packet_data_byte después de la cabecera PES será el primer byte del código de comienzo del tipo indicado en el cuadro 2-53. Al principio de una secuencia de vídeo, la alineación se producirá en el código de comienzo de la primera cabecera de secuencia.

NOTA – La especificación del tipo de alineación '01' del cuadro 2-53 no excluye la alineación desde el principio en una cabecera GOP o SEQ.

La definición de una unidad acceso se indica en 2.1.1.

Cuadro 2-53 – Valores de alineación de tren de vídeo

Tipo de alineación	Descripción
00	Reservado
01	Rebanada, o unidad de acceso de vídeo
02	Unidad de acceso de vídeo
03	Grupo de imágenes o secuencia
04	Secuencia
05-FF	Reservado

El cuadro 2-54 describe el tipo de alineación para el vídeo conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10 cuando el data_alignment_indicator en la cabecera de paquetes PES tiene un valor de '1'. En este caso el primer PES_packet_data_byte después de la cabecera PES será el primer byte de una unidad de acceso AVC o el primer byte de un sector (rebanada) AVC según lo señalado por el valor alignment_type.

Cuadro 2-54 – Valores de alineación de trenes de vídeo AVC

Tipo de alineación	Descripción
00	Reservado
01	Sector AVC o unidad de acceso AVC
02	Unidad de acceso AVC
03-FF	Reservado

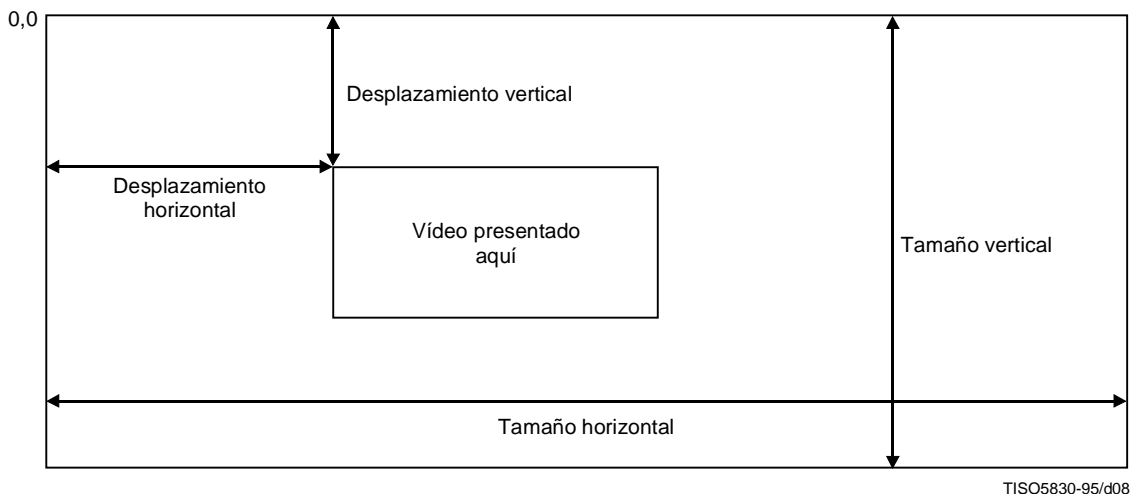
El cuadro 2-55 describe el tipo de alineación de audio cuando el indicador de alineación de datos en el encabezamiento de paquete PES tiene un valor de '1'. En este caso, el primer byte de datos de paquete PES que sigue al encabezamiento PES es el primer byte de una palabra de sincronización de audio.

Cuadro 2-55 – Valores de alineación de tren de audio

Tipo de alineación	Descripción
00	Reservado
01	Palabra de sincronización
02-FF	Reservado

2.6.12 Descriptor de cuadrícula de fondo objetivo

Es posible tener uno o más trenes de vídeo que, cuando son decodificados, no están destinados a ocupar la zona de visualización completa (por ejemplo, un monitor). La combinación de descriptor de cuadrícula de fondo objetivo y de descriptor de ventana de vídeo permite visualizar estas ventanas de vídeo en los lugares deseados. El descriptor de cuadrícula de fondo objetivo se utiliza para describir una cuadrícula de unidades de píxels proyectados en la zona de visualización. El descriptor de ventana de vídeo se utiliza para describir, para el tren asociado, el lugar en la cuadrícula en el cual debe visualizarse el píxel superior izquierdo de la ventana de visualización o del rectángulo de visualización. Esto se representa en la figura 2-3.



TISO5830-95/d08

Figura 2-3 – Zona de visualización de modelo de descriptor de cuadrícula de fondo

2.6.13 Definición semántica de campos en el descriptor de cuadrícula de fondo objetivo

horizontal_size (tamaño horizontal) – El tamaño horizontal del modelo de cuadrícula de fondo en píxels.

vertical_size (tamaño vertical) – El tamaño vertical del modelo de cuadrícula de fondo en píxels.

aspect_ratio_information (información de relación de aspecto) – Especifica la relación de aspecto de la muestra o la relación de aspecto de la visualización de la cuadrícula de fondo objetivo. La información de relación de aspecto se define en la Rec. UIT-T H.262 | ISO/CEI 13818-2 (véase el cuadro 2-56).

Cuadro 2-56 – Descriptor de la cuadrícula de fondo de destino

Sintaxis	N.º de bits	Mnemónico
target_background_grid_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
horizontal_size	14	uimsbf
vertical_size	14	uimsbf
aspect_ratio_information	4	uimsbf
}		

2.6.14 Descriptor de ventana de vídeo

El descriptor de ventana de vídeo se utiliza para describir las características de ventana del tren elemental asociado. Sus valores referencian el descriptor del modelo de cuadrícula de fondo para el mismo tren. Véase también el modelo de descriptor de cuadrícula de fondo en 2.6.12 (véase el cuadro 2-57).

Cuadro 2-57 – Descriptor de ventana de vídeo

Sintaxis	N.º de bits	Mnemónico
video_window_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
horizontal_offset	14	uimsbf
vertical_offset	14	uimsbf
window_priority	4	uimsbf
}		

2.6.15 Definición semántica de campos en el descriptor de ventana de vídeo

horizontal_offset (desplazamiento horizontal) – Su valor indica la posición del píxel superior izquierdo de la ventana de visualización de vídeo vigente o del rectángulo de visualización si se indica en la extensión de visualización de imagen en la cuadrícula de fondo objetivo para la visualización, como se define en el target_background_grid_descriptor (descriptor de cuadrícula de fondo objetivo). El píxel superior izquierdo de la ventana de vídeo será uno de los píxel de la cuadrícula de fondo objetivo (véase la figura 2-3).

vertical_offset (desplazamiento vertical) – Su valor indica la posición del píxel superior izquierdo de la ventana de visualización de vídeo vigente o del rectángulo de visualización si se indica en la extensión de visualización de imagen en la cuadrícula de fondo objetivo para la visualización, como se define en el target_background_grid_descriptor (descriptor de cuadrícula de fondo objetivo). El píxel superior izquierdo de la ventana de vídeo será uno de los píxel de la cuadrícula de fondo objetivo (véase la figura 2-3).

window_priority (prioridad de ventana) – El valor indica cómo se superponen las ventanas. Un valor de 0 es la prioridad más baja y un valor de 15 es la prioridad más alta, es decir, las ventanas con prioridad 15 son siempre visibles.

2.6.16 Descriptor de acceso condicional

El descriptor de acceso condicional se utiliza para especificar información de gestión de acceso condicional de todo el sistema, como EMM, e información específica de tren elemental, como ECM. Se puede utilizar en la sección de correspondencia de programas TS (véase 2.4.4.8) y en correspondencia de tren de programa (véase 2.5.3). Si algún tren elemental está aleatorizado, debe estar presente un descriptor CA para el programa que contiene este tren elemental. Si existe alguna información de gestión de acceso condicional para todo el sistema dentro de un tren de transporte, un descriptor CA debe estar presente en la tabla de acceso condicional.

Cuando el descriptor CA se encuentra en la sección de correspondencia de programa TS (id de tabla = 0x02), el CA_PID apunta a paquetes que contienen información de control de acceso relacionada con el programa, como los ECM. Su presencia como información de programa ampliada indica aplicabilidad a todo el programa. En el mismo caso, su presencia como información ES ampliada indica aplicabilidad al tren elemental asociado. Se prevén también datos privados.

Cuando el descriptor CA se encuentra en la sección CA (id de tabla = 0x01), el CA_PID apunta a paquetes que contienen información de gestión para todo el sistema y/o de control de acceso, como los EMM.

El contenido de los paquetes del tren transporte con información de acceso condicional es privado (véase el cuadro 2-58).

Cuadro 2-58 – Descriptor de acceso condicional

Sintaxis	N.º de bits	Mnemónico
CA_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
CA_system_ID	16	uimsbf
reserved	3	bslbf
CA_PID	13	uimsbf
for (i = 0; i < N; i++) {		
private_data_byte	8	uimsbf
}		
}		

2.6.17 Definición semántica de campos en el descriptor de acceso condicional

CA_system_ID (identificador de sistema CA) – Es un campo de 16 bits que indica el tipo de sistema CA aplicable para los trenes ECM y/o EMM asociados. La codificación es definida privadamente y no es especificada por el UIT-T | ISO/CEI.

CA_PID – Es un campo de 13 bits que indica el PID de los paquetes del tren de transporte que contendrán información ECM o EMM para los sistemas CA especificados con el ID de sistema CA asociado. El contenido (ECM o EMM) de los paquetes indicados por el CA_PID determinado a partir del contexto en el cual se encuentra el CA_PID, es decir, la sección de correspondencia de programas TS o la tabla CA en el tren de transporte o el campo de identificador de tren en el tren de programa.

La presencia de un PID 0x03 en trenes de transporte indica que existe información de tipo IPMP descrita en ISO/CEI 13818-11 que es utilizada por los componentes en el tren de transporte. En los trenes de programa, la presencia de un stream_ID_extension 0x00 indica que los componentes de este tren utilizan información de tipo IPMP descrita en ISO/CEI 13818-11. Dentro de un determinado tren Rec. UIT-T H.222.0 | ISO/CEI 13818-1, los componentes podrían utilizar tanto la IPMP, descrita en ISO/CEI 13818-11, como la información CA definida en ISO/CEI 13818-1:2006. La compatibilidad entre estas dos opciones se describe en ISO/CEI 13818-11.

2.6.18 Descriptor de idioma ISO 639

El descriptor de idioma se utiliza para especificar el idioma del tren elemental asociado (véase el cuadro 2-59).

Cuadro 2-59 – Descriptor de idioma ISO 639

Sintaxis	N.º de bits	Mnemónico
ISO_639_language_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i = 0; i < N; i++) {		
ISO_639_language_code	24	bslbf
audio_type	8	bslbf
}		
}		

2.6.19 Definición semántica de campos en el descriptor de idioma ISO 639

ISO_639_language_code (código de idioma ISO 639) – Identifica el idioma o idiomas utilizados por el elemento de programa asociado. El código de idioma ISO 639 contiene un código de tres caracteres especificado por la Parte 2 de ISO 639. Cada carácter está codificado en 8 bits de acuerdo con ISO 8859-1 e insertado en orden en el campo de 24 bits. En el caso de trenes de audio multilingües, la secuencia de campos de código de idioma ISO 639 reflejará el contenido del tren de audio.

audio_type (tipo de audio) – Es un campo de 8 bits que especifica el tipo de tren definido por el cuadro 2-60.

Cuadro 2-60 – Valores de tipo de audio

Valor	Descripción
0x00	No definido
0x01	En blanco
0x02	Degradación auditiva
0x03	Comentario para casos de degradación visual
0x04-0x7F	Privado del usuario
0x80-0xFF	Reservado

clean effects (en blanco) – Este campo indica que el tren elemental referenciado no tiene idioma.

hearing impaired (degradación auditiva) – Este campo indica que el tren elemental referenciado está preparado para personas con degradación auditiva.

visual_impaired_commentary (comentario para degradación visual) – Este campo indica que el elemento de programa referenciado está preparado para espectadores con degradaciones visuales.

2.6.20 Descriptor de reloj de sistema

Este descriptor transporta información sobre el reloj de sistema que se utilizó para generar las indicaciones de tiempos.

Si se utilizó una referencia de reloj externa, el `external_clock_reference_indicator` (indicador de referencia de reloj externo) se puede poner a '1'. El decodificador puede utilizar facultativamente la misma referencia externa, si está disponible.

Si la exactitud del reloj de sistema es superior a la exactitud de 30 ppm requerida, la exactitud del reloj puede ser comunicada codificándola en los campos `clock_accuracy` (exactitud de reloj). La exactitud de frecuencia de reloj es:

$$\text{clock_accuracy_integer} \times 10^{\text{clock_accuracy_exponent}} \text{ ppm} \quad (2-26)$$

Si `clock_accuracy_integer` se pone a 0, la exactitud de reloj de sistema es 30 ppm. Cuando el indicador de referencia de reloj externo se pone a '1', la exactitud del reloj la proporciona el reloj de referencia externa (véase el cuadro 2-61).

Cuadro 2-61 – Descriptor de reloj de sistema

Sintaxis	N.º de bits	Mnemónico
<code>system_clock_descriptor() {</code>		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
external_clock_reference_indicator	1	bslbf
reserved	1	bslbf
clock_accuracy_integer	6	uimsbf
clock_accuracy_exponent	3	uimsbf
reserved	5	bslbf
<code>}</code>		

2.6.21 Definición semántica de campos en el descriptor de reloj de sistema

external_clock_reference_indicator (indicador de referencia de reloj externa) – Es un indicador de 1 bit, que puesto a '1' indica que el reloj de sistema ha sido derivado de una referencia de frecuencia externa que puede estar disponible en el decodificador.

clock_accuracy_integer (entero de precisión de reloj) – Es un entero de 6 bits. Junto con `clock_accuracy_exponent`, da la exactitud de frecuencia fraccionaria del reloj de sistema en partes por millón.

clock_accuracy_exponent (exponente de exactitud de reloj) – Es un entero de 3 bits. Junto con `clock_accuracy_integer`, da la exactitud de frecuencia fraccionaria del reloj de sistema en partes por millón.

2.6.22 Descriptor de utilización de memoria tampón múltiplex

El `multiplex_buffer_utilization_descriptor` (descriptor de utilización de memoria tampón múltiplex) establece límites para la ocupación de la memoria tampón múltiplex STD. Esta información está destinada a sistemas como los remultiplexores que pueden utilizar esta información para apoyar una determinada estrategia de remultiplexación (véase el cuadro 2-62).

Cuadro 2-62 – Descriptor de utilización de memoria tampón múltiplex

Sintaxis	N.º de bits	Mnemónico
<code>multiplex_buffer_utilization_descriptor() {</code>		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
bound_valid_flag	1	bslbf
LTW_offset_lower_bound	15	uimsbf
reserved	1	bslbf
LTW_offset_upper_bound	15	uimsbf
<code>}</code>		

2.6.23 Definición semántica de campos en el descriptor de utilización de memoria tampón múltiplex

bound_valid_flag (bandera de límites válidos) – Un valor '1' indica que los campos de límite inferior del desplazamiento de la ventana de tiempo legal y de límite superior de desplazamiento de la ventana de tiempo legal son válidos.

LTW_offset_lower_bound (límite inferior del desplazamiento de la ventana de tiempo legal) – Este campo de 15 bits está definido solamente si la bandera de límites válidos tiene el valor '1'. Cuando está definido, este campo indica las unidades de periodos de reloj (27 MHz/300) definidas para el desplazamiento de la ventana de tiempo legal (desplazamiento de la LTW) (véase 2.4.3.4). El límite inferior del desplazamiento de la LTW representa el valor más bajo que tendría cualquier campo de desplazamiento de la LTW si dicho campo estuviese codificado en cada paquete del tren o de los trenes a que hace referencia este descriptor. Cuando el descriptor de utilización de memoria tampón múltiplex está presente, los campos reales de desplazamiento de la LTW pueden estar o no codificados en el tren de bits. Este límite es válido hasta la siguiente aparición de este descriptor.

LTW_offset_upper_bound (límite superior del desplazamiento de la ventana de tiempo legal) – Este campo de 15 bits está definido solamente si la bandera de límites válidos tiene el valor '1'. Cuando está definido, este campo indica las unidades de periodos de reloj (27 MHz/300) definidas para el desplazamiento de la ventana de tiempo legal (desplazamiento de la LTW) (véase 2.4.3.4). El límite superior del desplazamiento de la LTW representa el valor más alto que tendría cualquier campo de desplazamiento de la LTW si dicho campo estuviese codificado en cada paquete del tren o de los trenes a que hace referencia este descriptor. Cuando el descriptor de utilización de memoria tampón múltiplex está presente, los campos reales de desplazamiento de la LTW pueden estar o no codificados en el tren de bits. Este límite es válido hasta la siguiente aparición de este descriptor.

2.6.24 Descriptor de derechos de autor

El descriptor de derechos de autor proporciona un método para permitir la identificación de obras audiovisuales. Este descriptor de derechos de autor se aplica a programas o trenes elementales dentro del programa (véase el cuadro 2-63).

Cuadro 2-63 – Descriptor de derechos de autor

Sintaxis	N.º de bits	Mnemónico
copyright_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
copyright_identifier	32	uimsbf
for (i = 0; i < N; i++){		
additional_copyright_info	8	bslbf
}		
}		

2.6.25 Definición semántica de campos en el descriptor de derechos de autor

copyright_identifier (identificador de derechos de autor) – Este campo es un valor de 32 bits obtenido de una autoridad de registro.

additional_copyright_info (información de derechos de autor adicional) – El significado de los bytes de información de derechos de autor adicional, si los hubiere, es definido por aquél al que se ha asignado ese identificador de derechos de autor y, una vez definido, no cambiará.

2.6.26 Descriptor de velocidad binaria máxima

Véase el cuadro 2-64.

Cuadro 2-64 – Descriptor de velocidad binaria máxima

Sintaxis	N.º de bits	Mnemónico
maximum_bitrate_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	2	bslbf
maximum_bitrate	22	uimsbf
}		

2.6.27 Definición semántica de campos en el descriptor de velocidad binaria máxima

maximum_bitrate (velocidad binaria máxima) – La velocidad binaria máxima se codifica como un entero positivo de 22 bits en el campo. El valor indica un límite superior de la velocidad binaria máxima, incluida la tara de transporte, que se encontrará en este elemento de programa, o programa. La velocidad binaria máxima se expresa en unidades de 50 bytes/segundo. El descriptor de velocidad binaria máxima está incluido en la tabla de correspondencia de programas. Su presencia como información de programa ampliado indica la aplicabilidad a todo el programa. Su presencia como información ES indica aplicabilidad al elemento de programa asociado.

2.6.28 Descriptor de indicador de datos privados

Véase el cuadro 2-65.

Cuadro 2-65 – Descriptor de indicador de datos privados

Sintaxis	N.º de bits	Mnemónico
private_data_indicator_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
private_data_indicator	32	uimsbf
}		

2.6.29 Definición semántica de campos en el descriptor de indicador de datos privados

private_data_indicator (indicador de datos privados) – El valor de indicador de datos privados es privado y no será definido por el UIT-T | ISO/CEI.

2.6.30 Descriptor de memoria tampón alisadora

Este descriptor es facultativo y da información sobre el tamaño de una memoria tampón alisadora, SB_n , asociada con este descriptor, y sobre la correspondiente velocidad a que salen los datos de esa memoria tampón, para el elemento o los elementos de programa a que hace referencia el descriptor.

En el caso de trenes de transporte, los bytes de paquetes de tren de transporte del elemento o elementos de programa asociados, presentes en el tren de transporte, se introducen en una memoria tampón SB_n del tamaño dado por sb_size , en el instante definido por la ecuación 2-4.

En el caso de trenes de programa, los bytes de todos los paquetes PES de los trenes elementales asociados se introducen en una memoria tampón SB_n del tamaño dado por sb_size , en el instante definido por la ecuación 2-21.

Cuando hay datos presentes en esta memoria, los bytes se extraen de ella a una velocidad definida por sb_leak_rate . La memoria tampón SB_n no desbordará nunca. Durante la existencia continua de un programa, los valores de los elementos del descriptor de memoria tampón alisadora de los diferentes elementos del programa no cambiarán.

El significado del descriptor de memoria tampón alisadora sólo está definido cuando está incluido en la tabla de correspondencia de programas (PMT) o en la tabla de correspondencia de trenes de programa.

Si, en el caso de un tren de transporte, está presente en la información de tren elemental de la tabla de correspondencia de programas, todos los paquetes de tren de transporte del PID de ese elemento de programa entran en la memoria tampón alisadora.

Si, en el caso de un tren de transporte, está presente en la información de programa, los siguientes paquetes de tren de transporte entrarán en la memoria alisadora:

- todos los paquetes de tren de transporte de todos los PID listados como PID elementales en la información de programa ampliada;
- también todos los paquetes de tren de transporte del PID que es igual PMT_PID de esta sección; y
- todos los paquetes de tren de transporte del programa PCR_PID .

Todos los bytes que entran en la memoria tampón asociada, salen también de ella.

En un momento dado cualquiera tiene que haber, como máximo, un descriptor referente a cualquier elemento de programa individual y, como máximo, un descriptor referente al programa en su totalidad.

Cuadro 2-66 – Descriptor de memoria tampón alisadora

Sintaxis	N.º de bits	Mnemónico
smoothing_buffer_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	2	bslbf
sb_leak_rate	22	uimsbf
reserved	2	bslbf
sb_size	22	uimsbf
}		

2.6.31 Definición semántica de los campos del descriptor de memoria tampón alisadora

sb_leak_rate (velocidad de salida (o de fuga) de la memoria tampón alisadora) – Este campo de 22 bits se codifica como un entero positivo. Su contenido indica el valor de la velocidad de salida de la memoria tampón SB_n para el tren elemental asociado u otros datos, en unidades de 400 bit/s.

sb_size (tamaño de la memoria tampón alisadora) – Este campo de 22 bits se codifica como un entero positivo. Su contenido indica el valor del tamaño de la memoria tampón alisadora de multiplexación SB_n para el tren elemental asociado u otros datos, en unidades de 1 byte (véase el cuadro 2-66).

2.6.32 Descriptor de STD

Este descriptor es opcional y se aplica únicamente al modelo T-STD y a trenes elementales de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2, y se utiliza como se indica en 2.4.2. Este descriptor no se aplica a trenes de programa (véase el cuadro 2-67).

Cuadro 2-67 – Descriptor de STD

Sintaxis	N.º de bits	Mnemónico
STD_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	7	bslbf
leak_valid_flag	1	bslbf
}		

2.6.33 Definición semántica de los campos del descriptor de STD

leak_valid_flag (bandera de velocidad de salida válida) – La bandera de velocidad de salida válida es una bandera de 1 bit. Cuando está puesta a '1', la transferencia de datos de la memoria tampón MB_n a la memoria tampón EB_n en el decodificador T-STD se efectúa por el método de la fuga indicado en 2.4.2.3. Si esta bandera tiene el valor '0' y los campos `vbv_delay` presentes en el tren de vídeo asociado no tienen el valor 0xFFFF, la transferencia de datos de la memoria tampón MB_n a la memoria tampón EB_n se efectúa por el método del retardo `vbv` indicado en 2.4.2.3.

2.6.34 Descriptor de IBP

Este descriptor facultativo da información sobre algunas características de los tipos de secuencia trama en el tren de vídeo conforme a ISO/CEI 11172-2, Rec. UIT-T H.262 | ISO/CEI 13818-2, o ISO/CEI 14496-2 (véase el cuadro 2-68).

Cuadro 2-68 – Descriptor de IBP

Sintaxis	N.º de bits	Mnemónico
ibp_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
closed_gop_flag	1	uimsbf
identical_gop_flag	1	uimsbf
max_gop-length	14	uimsbf
}		

2.6.35 Definición semántica de los campos del descriptor IBP

closed_gop_flag (bandera de grupo de imágenes cerrado) – Esta es una bandera de 1 bit que, cuando se pone a '1', indica que se codifica un encabezamiento de grupo de imágenes antes de cada trama I y que closed_gop_flag se pone a '1' en todos los encabezamientos de grupo de imágenes en la secuencia de vídeo.

identical_gop_flag (bandera de grupo de imágenes idéntico) – Esta es una bandera de 1 bit que, cuando se pone a '1', indica que el número de tramas P y de tramas B entre tramas I, y los tipos de codificación de imagen y tipos de secuencia de imagen entre imágenes I son los mismos en toda la secuencia, excepto quizás las imágenes hasta la segunda imagen I.

max_gop_length (bandera de grupo de imágenes máximo) – Es un entero sin signo de 14 bits que indica el número máximo de imágenes codificadas entre dos cualesquiera imágenes I consecutivas en la secuencia. El valor cero está prohibido.

2.6.36 Descriptor de vídeo MPEG-4

Para trenes individuales ISO/CEI 14496-2 transportados directamente en paquetes PES, definidos en 2.11.2, el descriptor de vídeo MPEG-4 proporciona información básica para identificar los parámetros de codificación de estos trenes elementales visuales. El descriptor de vídeo MPEG-4 no se aplica a los trenes ISO/CEI 14496-2 encapsulados en paquetes SL y en paquetes FlexMux, definidos en 2.11.3.

Cuadro 2-69 – Descriptor de vídeo MPEG-4

Sintaxis	N.º de bits	Mnemónico
MPEG-4_video_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MPEG-4_audio_profile_and_level	8	uimsbf
}		

2.6.37 Definición semántica de campos en el descriptor de vídeo MPEG-4

MPEG-4_video_profile_and_level (perfil y nivel de vídeo MPEG-4) – Este campo de 8 bits identificará el perfil y nivel del tren vídeo ISO/CEI 14496-2. Este campo será codificado con el mismo valor que el campo profile_and_level_indication en el encabezamiento de secuencia de objeto visual del tren ISO/CEI 14496-2 asociado.

2.6.38 Descriptor de audio MPEG-4

Para cada tren individual ISO/CEI 14496-3 directamente transportado en paquetes PES, definidos en 2.11.2, el descriptor de audio MPEG-4 proporciona información básica para identificar los parámetros de codificación de estos trenes elementales de audio. El descriptor de audio MPEG-4 no se aplica a los trenes ISO/CEI 14496-3 encapsulados en paquetes SL y en paquetes FlexMux, definidos en 2.11.3.

Cuadro 2-70 – Descriptor de audio MPEG-4

Sintaxis	N.º de bits	Mnemónico
MPEG-4_audio_descriptor () { descriptor_tag descriptor_length MPEG-4_audio_profile_and_level }	8 8 8	uimsbf uimsbf uimsbf

2.6.39 Definición semántica de campos en el descriptor de audio MPEG-4

MPEG-4_audio_profile_and_level (perfil y nivel de audio MPEG-4) – Este campo de 8 bits identificará el perfil y el nivel del tren de audio ISO/CEI 14496-3 correspondiente al siguiente cuadro 2-71.

Cuadro 2-71 – Valores de asignación de perfil y nivel de audio MPEG-4

Valor	Descripción
0x00-0x0F	Reservado
0x10	Perfil principal, nivel 1
0x11	Perfil principal, nivel 2
0x12	Perfil principal, nivel 3
0x13	Perfil principal, nivel 4
0x14-0x17	Reservado
0x18	Perfil escalable, nivel 1
0x19	Perfil escalable, nivel 2
0x1A	Perfil escalable, nivel 3
0x1B	Perfil escalable, nivel 4
0x1C-0x1F	Reservado
0x20	Perfil vocal, nivel 1
0x21	Perfil vocal, nivel 2
0x22-0x27	Reservado
0x28	Perfil de síntesis, nivel 1
0x29	Perfil de síntesis, nivel 2
0x2A	Perfil de síntesis, nivel 3
0x2B-0x2F	Reservado
0x30	Perfil de audio de alta calidad, nivel 1
0x31	Perfil de audio de alta calidad, nivel 2
0x32	Perfil de audio de alta calidad, nivel 3
0x33	Perfil de audio de alta calidad, nivel 4
0x34	Perfil de audio de alta calidad, nivel 5
0x35	Perfil de audio de alta calidad, nivel 6
0x36	Perfil de audio de alta calidad, nivel 7
0x37	Perfil de audio de alta calidad, nivel 8
0x38	Perfil de audio de bajo retardo, nivel 1
0x39	Perfil de audio de bajo retardo, nivel 2
0x3A	Perfil de audio de bajo retardo, nivel 3
0x3B	Perfil de audio de bajo retardo, nivel 4
0x3C	Perfil de audio de bajo retardo, nivel 5
0x3D	Perfil de audio de bajo retardo, nivel 6
0x3E	Perfil de audio de bajo retardo, nivel 7
0x3F	Perfil de audio de bajo retardo, nivel 8

Cuadro 2-71 – Valores de asignación de perfil y nivel de audio MPEG-4

Valor	Descripción
0x40	Perfil de audio natural, nivel 1
0x41	Perfil de audio natural, nivel 2
0x42	Perfil de audio natural, nivel 3
0x43	Perfil de audio natural, nivel 4
0x44-0x47	Reservado
0x48	Perfil de interfuncionamiento de redes audio móviles, nivel 1
0x49	Perfil de interfuncionamiento de redes audio móviles, nivel 2
0x4A	Perfil de interfuncionamiento de redes audio móviles, nivel 3
0x4B	Perfil de interfuncionamiento de redes audio móviles, nivel 4
0x4C	Perfil de interfuncionamiento de redes audio móviles, nivel 5
0x4D	Perfil de interfuncionamiento de redes audio móviles, nivel 6
0x4E-0x4F	Reservado
0x50	Perfil AAC, nivel 1
0x51	Perfil AAC, nivel 2
0x52	Perfil AAC, nivel 4
0x53	Perfil AAC, nivel 5
0x54-0x57	Reservado
0x58	Perfil AAC de alta eficiencia, nivel 2
0x59	Perfil AAC de alta eficiencia, nivel 3
0x5A	Perfil AAC de alta eficiencia, nivel 4
0x5B	Perfil AAC de alta eficiencia, nivel 5
0x5C-0xFF	Reservado

2.6.40 Descriptor IOD

El descriptor IOD encapsula la estructura InitialObjectDescriptor. Un descriptor de objeto inicial permite acceder a un conjunto de trenes ISO/CEI 14496 identificando los valores ES_ID de trenes de descripción de escena y descriptor de objeto ISO/CEI 14496-1. Los trenes de descripción de escena y de descripción de objeto contienen más información sobre los trenes ISO/CEI 14496 que forman parte de la escena. Para una descripción del procedimiento de acceso de contenido, véase el anexo R. El InitialObjectDescriptor se especifica en 8.6.3 de ISO/CEI 14496-1.

Dentro de un tren de transporte el descriptor IOD será transportado en el bucle de descriptor inmediatamente después del campo program_info_length en la tabla de correspondencia de programas. Si hay presente una correspondencia de trenes de programa en el tren de programa, el descriptor IOD será transportado en el bucle de descriptor inmediatamente después del campo program_stream_info_length en la correspondencia de trenes de programa. Puede haber más de un descriptor IOD asociado a un programa.

NOTA – Esta Especificación no indica cómo la etiqueta IOD puede ser utilizada por información de servicio de nivel más alto para seleccionar únicamente una de las presentaciones de ISO/CEI 14496 identificadas por múltiples descriptores IOD.

Cuadro 2-72 – Descriptor IOD

Sintaxis	N.º de bits	Mnemónico
IOD_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Scope_of_IOD_label	8	uimsbf
IOD_label	8	uimsbf
InitialObjectDescriptor ()	8	uimsbf
}		

2.6.41 Definición semántica de campos en el descriptor IOD

Scope_of_IOD_label (alcance de la etiqueta IOD) – Este campo de 8 bits especifica el alcance del campo de etiqueta IOD. Un valor de 0x10 indica que la etiqueta IOD es única dentro del tren de programa o dentro del programa específico de un tren de transporte en el que se transporta el descriptor IOD. Un valor de 0x11 indica que la etiqueta IOD es única dentro del tren de transporte en el que se transporta el descriptor IOD. Todos los demás valores del campo alcance de la etiqueta IOD están reservados.

IOD_label (etiqueta IOD) – Este campo de 8 bits especifica la etiqueta del descriptor IOD.

InitialObjectDescriptor () (descriptor de objeto inicial) – Esta estructura se define en 8.6.3.1 de ISO/CEI 14496-1.

2.6.42 Descriptor SL

El descriptor SL se utilizará cuando sólo un tren paquetizado SL ISO/CEI 14496-1 está encapsulado en paquetes PES. El descriptor SL asocia el ES_ID de este tren SL paquetizado a un PID elemental en el caso de un tren de transporte o a un elementary_stream_id en el caso de un tren de programa. Dentro de un tren de programa, el descriptor SL será transportado para el correspondiente tren elemental en el bucle de descriptor inmediatamente después del campo ES_info_length en la tabla de correspondencia de programas. Si hay presente una correspondencia de trenes de programa en un tren de programa, el descriptor SL será transportado en el bucle de descriptor inmediatamente después del campo elementary_stream_info_length dentro del correspondencia de trenes de programa.

NOTA – Los trenes SL paquetizados se pueden utilizar en un tren de programa. Sin embargo, sólo existe un identificador de tren para los trenes paquetizados ISO/CEI 14496-1 SL. Para asociar múltiples trenes como éstos dentro de un tren de programa a una escena ISO/CEI 14496-1, FlexMux tiene que ser utilizado y señalizado adecuadamente por un descriptor FMC. Esta limitación no existe en un tren de transporte, donde el descriptor SL proporciona una correspondencia inequívoca entre un valor ES_ID de ISO/CEI 14496-1 y un valor de PID elemental de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Cuadro 2-73 – Descriptor SL

Sintaxis	N.º de bits	Mnemónico
SL_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ES_ID	16	uimsbf
}		

2.6.43 Definición semántica de campos en el descriptor SL

ES_ID – Este campo de 16 bits especificará el identificador de un tren paquetizado SL ISO/CEI 14496-1.

2.6.44 Descriptor FMC

El descriptor FMC indica que la herramienta FlexMux de ISO/CEI 14496-1 ha sido utilizada para multiplexar trenes paquetizados SL en un tren FlexMux antes de ser encapsulado en paquetes PES o secciones ISO/IEC 14496. El descriptor FMC asocia canales FlexMux a valores de ES_ID de los trenes paquetizados SL en el tren FlexMux.

Se requiere un descriptor FMC para cada elemento de programa referenciado por un valor de PID elemental (elementary_PID_value) en un tren de transporte y para cada identificador de tren elemental (elementary_stream_id) en un tren de programa que transporta un tren FlexMux. Dentro de un tren de transporte, el descriptor FMC será transportado por el correspondiente tren elemental en el bucle de descriptor inmediatamente después del campo ES_info_length en la tabla de correspondencia de programas si hay presente una correspondencia de trenes de programa en el bucle de descriptor inmediatamente después del campo elementary_stream_info_length en la correspondencia de trenes de programa.

Para cada tren paquetizado SL en un tren FlexMux, el canal FlexMux será identificado por una sola entrada en el descriptor FMC.

Cuadro 2-74 – Descriptor FMC

Sintaxis	N.º de bits	Mnemónico
FMC_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<descriptor_length; i += 3) {		
ES_ID	16	uimsbf
FlexMuxChannel	8	uimsbf
}		
}		

2.6.45 Definición semántica de campos en el descriptor FMC

ES_ID – Este campo de 16 bits especifica el identificador de un tren paquetizado SL ISO/CEI 14496-1.

FlexMuxChannel () (canal FlexMux) – Este campo de 8 bits especifica el número del canal FlexMux utilizado para este tren paquetizado SL.

2.6.46 Descriptor ES_ID externo

El descriptor ES_ID externo asigna un ES_ID, definido ISO/CEI 14496-1, a un elemento de programa al que no se ha asignado ningún valor ES_ID por otros medios. Este ES_ID permite hacer referencia a un componente no definido en ISO/CEI 14496 en la descripción de escena o, por ejemplo, asociar el componente no definido en ISO/CEI 14496 con un tren IPMP.

Dentro de un tren de transporte, la asignación de un ES_ID se efectuará transportando un descriptor ES_ID externo para el tren elemental correspondiente en el bucle de descriptor inmediatamente después del campo ES_info_length en la tabla de correspondencia de programas. Si hay presente una correspondencia de trenes de programa, el descriptor External_ES_ID será transportado en el bucle de descriptor inmediatamente después del campo elementary_stream_info_length en la correspondencia de trenes de programa.

Cuadro 2-75 – Descriptor External_ES_ID

Sintaxis	N.º de bits	Mnemónico
External_ES_ID_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
External_ES_ID	16	uimsbf
}		

2.6.47 Definición semántica de campos en el descriptor ES ID externo

External_ES_ID (ES_ID externo) – Este campo de 16 bits asigna un identificador ES_ID, definido en ISO/CEI 14496-1, a un componente de un programa.

2.6.48 Descriptor Muxcode

El descriptor Muxcode transporta estructuras MuxCodeTableEntry definidas en 11.2.4.3 de ISO/CEI 14496-1. MuxCodeTableEntries configura el modo MuxCode de FlexMux.

Uno o más descriptores Muxcode pueden estar asociados a cada elementary_PID o elementary_stream_id, respectivamente, que transportan un tren FlexMux ISO/CEI 14496-1 que utiliza el código Muxcode. Dentro de un tren de transporte, el descriptor Muxcode será transportado para el correspondiente tren elemental en el bucle de descriptor inmediatamente después del campo ES_info_length en la tabla de correspondencia de programas. Si hay presente una correspondencia de trenes de programa en un tren de programa, el descriptor Muxcode será transportado en el bucle de descriptor inmediatamente después del campo elementary_stream_info_length en la correspondencia de trenes de programa.

Se puede actualizar MuxCodeTableEntries con nuevas versiones. En caso de tales actualizaciones, el número de versión de cada tabla de correspondencia de programas o la program_stream_map_version de cada correspondencia de trenes de programa, respectivamente, que transportan el descriptor Muxcode en su bucle de descriptor, se incrementará en 1 módulo 32.

Cuadro 2-76 – Descriptor Muxcode

Sintaxis	N.º de bits	Mnemónico
<pre>Muxcode_descriptor () { descriptor_tag descriptor_length for (i = 0; i < N; i++) { MuxCodeTableEntry () } }</pre>	<p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p>

2.6.49 Definición semántica de campos en el descriptor Muxcode

MuxCodeTableEntry () (entrada de tabla Muxcode) – Esta estructura se define en 11.2.4.3 de ISO/CEI 14496-1.

2.6.50 Descriptor FmxBufferSize

El descriptor FmxBufferSize transporta el tamaño de la memoria tampón FlexMux (FB) para cada tren paquetizado SL multiplexado en un tren FlexMux.

Se asociará un descriptor FmxBufferSize a cada elementary_PID o elementary_stream_id, respectivamente, que transportan un tren FlexMux ISO/CEI 14496-1. Dentro de un tren de transporte, el descriptor FmxBufferSize será transportado por el correspondiente tren elemental en el bucle de descriptor inmediatamente después del campo ES_info_length en la tabla de correspondencia de programas. Si hay presente una correspondencia de trenes de programa en un tren de programa, el descriptor FmxBufferSize será transportado en el bucle de descriptor inmediatamente después del campo elementary_stream_info_length dentro de la correspondencia de trenes de programa.

Cuadro 2-77 – Descriptor FmxBufferSize

Sintaxis	N.º de bits	Mnemónico
<pre>FmxBufferSize_descriptor () { descriptor_tag descriptor_length DefaultFlexMuxBufferDescriptor() for (i=0; i<descriptor_length; i += 4) { FlexMuxBufferDescriptor() } }</pre>	<p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p>

2.6.51 Definición semántica de campos en el descriptor FmxBufferSize

FlexMuxBufferDescriptor() (descriptor de memoria tampón FlexMux) – Este descriptor especifica el tamaño de la memoria tampón FlexMux para un tren paquetizado SL transportado dentro del tren FlexMux stream. Se define en 11.2 de ISO/CEI 14496-1.

DefaultFlexMuxBufferDescriptor() (descriptor de memoria tampón FlexMux por defecto) – Este descriptor especifica el tamaño por defecto de la memoria tampón FlexMux para este tren FlexMux. Se define en 11.2 de ISO/CEI 14496-1.

2.6.52 Descriptor de memoria tampón múltiplex

El descriptor de memoria tampón múltiplex transporta el tamaño de la memoria tampón múltiplex MB_n , así como la velocidad de fuga Rx_n a la cual los datos son transferidos de la memoria tampón de transporte TB_n a la memoria tampón MB_n para un PID elemento de programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1 referenciado por un valor de PID elemental en la tabla de correspondencia de programas.

Se asociará un descriptor de memoria tampón múltiplex a cada PID elemental que contiene un tren FlexMux ISO/CEI 14496 o un tren paquetizado SL, incluidos los que contienen secciones ISO/CEI 14496. Para la definición de estas memorias tampón y velocidades en el modelo T-STD de decodificación del contenido ISO/CEI 14496, véase 2.11.3.9.

El descriptor de memoria tampón múltiplex será transportado en el bucle de descriptor inmediatamente después del campo `ES_info_length` en la tabla de correspondencia de programas.

Cuadro 2-78 – Descriptor MultiplexBuffer

Sintaxis	N.º de bits	Mnemónico
MultiplexBuffer_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MB_buffer_size	24	uimsbf
TB_leak_rate	24	uimsbf
}		

2.6.53 Definición semántica de campos en el descriptor de memoria tampón múltiplex

MB_buffer_size (tamaño de memoria MB) – Este campo de 24 bits especificará en octetos el tamaño de la memoria tampón MB_n del tren elemental n que está asociado con este descriptor.

TB_leak_rate (velocidad de fuga de TB) – Este campo de 24 bits especificará en unidades de 400 bits por segundo la velocidad a la cual los datos son transferidos de la memoria tampón de transporte TB_n a la memoria múltiplex MB_n para el tren elemental n que está asociado con este descriptor.

2.6.54 Descriptor FlexMuxTiming

Véase el cuadro 2-79.

Cuadro 2-79 – Descriptor FlexMuxTiming

Sintaxis	N.º de bits	Mnemónico
FlexMuxTiming_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
FCR_ES_ID	16	uimsbf
FCRResolution	32	uimsbf
FCRLength	8	uimsbf
FmxRateLength	8	uimsbf
}		

2.6.55 Definición semántica de campos en el descriptor FlexMuxTiming

FCR_ES_ID – Es el `ES_ID` asociado a este tren de referencia de reloj.

FCRResolution (Resolución de FCR) – Es la resolución de la base de tiempo de objeto, en ciclos por segundo.

FCRLength (Longitud de FCR) – Es la longitud del campo `fmxClockReference` medido en paquetes FlexMux con `index = 238`. Una longitud igual a cero indicará que no hay paquetes FlexMux con `index = 238` en este tren FlexMux. `FCRLength` tomará valores entre cero y 64.

FmxRateLength (Longitud de FmxRate) – Es la longitud del campo `fmxRate` en paquetes FlexMux con `index = 238`. `FmxRateLength` tomará valores entre 1 y 32.

2.6.56 Descriptor de etiquetado del contenido

El descriptor de etiquetado del contenido asigna una etiqueta al contenido y los metadatos pueden utilizarla para establecer una referencia al contenido asociado. Esta etiqueta (`content_reference_id_record`), es específica al formato de aplicación de los metadatos. El descriptor de etiquetado del contenido está asociado con un segmento de contenido. A los fines del texto de este punto, un segmento de contenido es una porción de tiempo de un programa, un tren elemental (audio o vídeo) o cualquier combinación de programas o trenes elementales. El descriptor puede estar incluido en la tabla de correspondencia de programas (PTM, *programan map table*), en el bucle de descriptor del programa o del tren elemental, pero también se puede incluir en tablas no definidas en la presente especificación, por ejemplo, tablas que describen segmentos de programas o trenes elementales. De otra parte, el descriptor de etiquetado del contenido contiene la información sobre la base de tiempo del contenido que se utiliza y sobre la diferencia entre la base de tiempo del contenido y la base de tiempo de los metadatos. Cuando se utiliza el concepto de tiempo de lectura normal (NPT, *normal play time*) del DSM-CC, según se especifica en ISO/CEI 13818-6, como base de tiempo del contenido, se incluye el ID de la base de tiempo NPT. Este descriptor permite el transporte de datos privados. Véase el cuadro 2-80.

Cuadro 2-80 – Descriptor de etiquetado del contenido

Sintaxis	N.º de bits	Mnemónico
<code>Content_labeling_descriptor () {</code>		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
metadata_application_format	16	uimsbf
if (metadata_application_format== 0xFFFF){		
metadata_application_format_identifier	32	uimsbf
}		
content_reference_id_record_flag	1	bslbf
content_time_base_indicator	4	uimsbf
reserved	3	bslbf
if (content_reference_id_record_flag == '1'){		
content_reference_id_record_length	8	uimsbf
for (i=0; i<content_reference_id_record_length;i++){		
content_reference_id_byte	8	bslbf
}		
}		
if (content_time_base_indicator== 1 2){		
reserved	7	bslbf
content_time_base_value	33	uimsbf
reserved	7	bslbf
metadata_time_base_value	33	uimsbf
}		
if (content_time_base_indicator== 2){		
reserved	1	bslbf
contentId	7	uimsbf
}		
if (content_time_base_indicator==3 4 5 6 7){		
time_base_association_data_length	8	uimsbf
for (i=0; i<time_base_association_data_length;i++){		
reserved	8	bslbf
}		
}		
for (i=0; i<N;i++){		
private_data_byte	8	bslbf
}		
}		

2.6.57 Definición semántica de los campos en el descriptor de etiquetado de contenido

metadata_application_format (formato de aplicación de metadatos) – El `metadata_application_format` es un campo de 16 bits codificado como se define en el cuadro 2-81, que especifica cuál es la aplicación encargada de definir la utilización, la sintaxis y la semántica del registro `content_reference_id` y de cualquier otro campo de este descriptor definido en un contexto privado. Véase asimismo 2.12.1. El valor 0xFFFF indica que el formato va determinado por el valor del campo `metadata_application_format_identifier`.

**Cuadro 2-81 – Campo metadata_application_format
(formato de aplicación de metadatos)**

Valor	Descripción
0x0000 – 0x000F	Reservado
0x0010	Codificado ISO 15706 (ISAN) en su formato binario (véanse notas 1 y 3)
0x0011	Codificado ISO 15706-2 (V-ISAN) en su formato binario (véanse notas 2 y 3)
0x0012 – 0x00FF	Reservado
0x0100 – 0xFFFFE	Definido por el usuario
0xFFFF	Definido por el campo metadata_application_format_identifier
NOTA 1 – Para ISAN, el campo content_reference_id_byte se pone a codificación binaria y a content_reference_id_record_length se le asigna el valor 0x08. NOTA 2 – Para V-ISAN, el campo content_reference_id_byte se pone a codificación binaria y a content_reference_id_record_length se le asigna el valor 0x0C. NOTA 3 – Para lograr el interfuncionamiento entre aplicaciones de metadatos que utilizan los valores de metadata_application_format de 0x0010 y 0x0011, se recomienda poner a '1' content_reference_id_flag y poner a '00' content_time_base_indicator.	

metadata_application_format_identifier (identificador del formato de aplicación de metadatos) – La codificación de este campo de 32 bits es plenamente equivalente a la codificación del campo format_identifier en el registration_descriptor, según se define en 2.6.8.

NOTA - La autoridad de registro asignada para el campo format_identifier es la SMPTE.

content_reference_id_record_flag (bandera de registro del identificador de la referencia del contenido) – La bandera content_reference_id_record de 1 bit señala la presencia de un content_reference_id_record en este descriptor.

content_time_base_indicator (indicador de base de tiempo del contenido) – El campo de 4 bits content_time_base_indicator especifica la base de tiempo del contenido utilizada. Si el descriptor está asociado con un programa, la base de tiempo del contenido se aplica a todos los trenes que forman parte de dicho programa. El valor de 1 indica que se utiliza el reloj de sistema (STC, *system time clock*), mientras que el valor 2 indica la utilización del tiempo de lectura normal NPT que se define en ISO/CEI 13818-6. Los valores comprendidos entre 8 y 15 indican que se utilizan bases de tiempo del contenido privadas. Cuando se codifica con el valor 0, no se define ninguna base de tiempo del contenido en este descriptor. No se define en esta especificación la correspondencia al contenido de las referencias temporales en los metadatos cuando no se especifica la base de tiempo del contenido para un programa o tren.

**Cuadro 2-82 – Valores del indicador de base de tiempo del contenido
(content_time_base_indicator)**

Valor	Descripción
0	No se define la base de tiempo del contenido en este descriptor
1	Utilización del STC
2	Utilización del NPT
3-7	Reservado
8-15	Utilización de una base de tiempo del contenido privada

content_reference_id_record_length (longitud del registro del identificador de la referencia del contenido) – El campo de 8 bits content_reference_id_record_length especifica el número de content_reference_id_bytes inmediatamente posteriores a este campo. Este campo no puede codificarse con el valor 0.

content_reference_id_byte (byte de identificación de la referencia del contenido) – El content_reference_id_byte es parte de una cadena de uno o más bytes contiguos que asignan una o más identificaciones de referencia (etiquetas) al contenido asociado a este descriptor. El formato de esta cadena de bytes está definido por el valor codificado del campo metadata_application_format.

content_time_base_value (valor de base de tiempo del contenido) – El campo de 33 bits content_time_base_value especifica el valor de la base de tiempo del contenido que se indica en el campo content_time_base_indicator, en unidades de 90 kHz.

metadata_time_base_value (valor de base de tiempo de los metadatos) – El campo de 33 bits `metadata_time_base_value` se codifica en unidades de 90 kHz. Este campo se codifica con el valor de la base de tiempo de los metadatos en el instante en que la base de tiempo que se indica en `content_time_base_indicator` alcanza el valor codificado del campo `content_time_base_value`. Obsérvese que se puede utilizar cualquier notación para la base de tiempo de los metadatos, pero este valor debe codificarse en unidades de 90 kHz. Por ejemplo, si se utiliza un código temporal de la SMPTE, el número de horas, minutos, segundos y tramas se expresa en el correspondiente número de unidades de 90 kHz.

contentId (identificador del contenido) – El campo de 7 bits `contentId` especifica el valor del campo `content_Id` en el descriptor de referencia NPT para la base de tiempo NPT aplicada.

time_base_association_data_length (longitud de los datos de asociación de la base de tiempo) – El campo de 8 bits `time_base_association_data_length` especifica el número de bytes reservados inmediatamente posteriores a este campo. Los bytes reservados pueden utilizarse para transportar datos de asociación de la base de tiempo, para las bases de tiempo definidas en el futuro.

private_data_byte (byte de datos privados) – El campo de 8 bits `private_data_byte` representa datos en formato de dominio privado. Estos bytes pueden utilizarse para aportar información adicional, según convenga. La utilización de estos bytes queda definida por el formato de aplicación de metadatos.

2.6.58 Descriptor de puntero de metadatos

El descriptor de puntero de metadatos se refiere a un único servicio de metadatos y lo asocia con contenido audiovisual en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Los metadatos están asociados con el contenido dentro del contexto de este descriptor. Este contexto queda definido por la situación del descriptor. En un tren de transporte, el descriptor puede estar situado en la PMT, en el bucle de descriptor del programa o de un tren elemental, pero también en tablas no definidas en la presente especificación, como las tablas que describen los paquetes de programas de servicios de radiodifusión. Los metadatos se pueden colocar en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, pero estos mismos metadatos también se pueden colocar en otros sistemas, por ejemplo Internet.

El descriptor puede indicar dónde están situados los metadatos no transportados en un tren de la Rec. H.222.0 | ISO/CEI 13818-1, y la codificación de esta información es específica al formato de aplicación de metadatos. Este descriptor permite el transporte de datos privados.

Para los metadatos transportados en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, el descriptor especifica las herramientas utilizadas para este transporte. Si los metadatos son transportados en paquetes de un tren elemental paquetizado (PES), en secciones de metadatos o en secciones de telecarga sincronizadas ISO/CEI 13818-6, el campo `metadata_service_id` identifica el servicio de metadatos en el tren de metadatos a que se hace referencia. Si para transportar los metadatos se utiliza un carrusel ISO/CEI 13818-6, los datos privados pueden informar sobre el servicio de metadatos, por ejemplo el valor aplicado de `module_id` para el transporte de metadatos en el carrusel de datos y el nombre de archivo de los metadatos cuando se utiliza un carrusel de objetos.

Los receptores deben saber que desde el mismo programa o tren audiovisual (definido por el contexto del descriptor) se pueden señalar múltiples servicios de metadatos. Se utilizará un descriptor de puntero de metadatos individual para señalar cada servicio de metadatos utilizado por el programa o tren audiovisual. Del mismo modo, puede señalarse el mismo servicio de metadatos desde distintos programas o trenes audiovisuales, utilizando diferentes descriptores de puntero de metadatos para cada asociación.

**Cuadro 2-83 – Descriptor de puntero de metadatos
(metadata pointer descriptor)**

Sintaxis	N.º de bits	Mnemónico
Metadata_pointer_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
metadata_application_format	16	uimsbf
if (metadata_application_format == 0xFFFF){		
metadata_application_format_identifier	32	uimsbf
}		
metadata_format	8	uimsbf
if (metadata_format == 0xFF){		
metadata_format_identifier	32	uimsbf
}		
metadata_service_id	8	uimsbf
metadata_locator_record_flag	1	bslbf
MPEG_carriage_flags	2	uimsbf
reserved	5	bslbf
if (metadata_locator_record_flag == '1'){		
metadata_locator_record_length	8	uimsbf
for (i=0; i< metadata_locator_record_length; i++){		
metadata_locator_record_byte	8	bslbf
}		
}		
if (MPEG_carriage_flags == 0 1 2){		
program_number	16	uimsbf
}		
if (MPEG_carriage_flags == 1){		
transport_stream_location	16	uimsbf
transport_stream_id	16	uimsbf
}		
for (i=0; i<N; i++){		
private_data_byte	8	bslbf
}		
}		

2.6.59 Definición semántica de campos en el descriptor de puntero de metadatos

metadata_application_format (formato de aplicación de metadatos) – El campo de 16 bits metadata_application_format especifica cuál es la aplicación encargada de definir la utilización, la sintaxis y la semántica del registro metadata_locator_record y de cualquier otro campo del descriptor definido en un contexto privado. La codificación de este campo se define en el cuadro 2-81 de la subcláusula 2.6.57.

metadata_application_format_identifier (identificador del formato de aplicación de metadatos) – La codificación de este campo se define en 2.6.57.

metadata_format (formato de metadatos) – El campo de 8 bits metadata_format indica el formato y la codificación de los metadatos. La codificación de este campo se especifica en el cuadro 2-84.

Cuadro 2-84 – Valores del formato de metadatos

Valor	Descripción
0x00-0x0F	Reservado
0x10	TeM ISO/CEI 15938-1
0x11	BiM ISO/CEI 15938-1
0x12-0x3E	Reservado
0x3F	Definido por el formato de aplicación de metadatos
0x40-0xFE	Uso privado
0xFF	Definido por el campo metadata_format_identifier

Los valores 0x10 y 0x11 identifican los datos definidos en ISO/CEI 15938-1. El valor 0x3F indica que el formato está definido por el valor del campo metadata_application_format. Los valores comprendidos entre 0x40 y 0xFE, inclusive, permiten la señalización del uso de formatos privados. El valor 0xFF indica que el formato está identificado en el campo metadata_format_identifier.

ISO/CEI 13818-1:2007 (S)

metadata_format_identifier (identificador del formato de metadatos) – La codificación de este campo de 32 bits es plenamente equivalente a la codificación del campo `format_identifier` del `registration_descriptor`, según se define en 2.6.8.

NOTA – La SMPTE es la Autoridad de Registro asignada para el campo `format_identifier`.

metadata_service_id (identificador del servicio de metadatos) – Este campo de 8 bits identifica el servicio de metadatos. Se utiliza para la extracción del servicio de metadatos de un tren de metadatos.

metadata_locator_record_flag (bandera de registro del localizador de los metadatos) – El campo de 1 bits `metadata_locator_record_flag` puesto a '1' indica que los metadatos asociados están disponibles en un lugar que no es un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y que se especifica en `metadata_locator_record`.

MPEG_carriage_flags (banderas de transporte MPEG) – El campo de 2 bits `MPEG_carriage_flags` especifica si el tren de metadatos que contiene el servicio de metadatos asociado se transporta en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y, en tal caso, si los metadatos asociados se transportan en un tren de transporte o un tren de programa. La codificación de este campo se define en el cuadro 2-85.

Cuadro 2-85 – Banderas de transporte MPEG (MPEG_carriage_flags)

Valor	Descripción
0	Transportado en el mismo tren de transporte en que se incluye el descriptor de puntero de metadatos.
1	Transportado en un tren de transporte distinto de aquél donde se encuentra el descriptor de puntero de metadatos.
2	Transportado en un tren de programa. Puede o no ser el mismo tren de programas en que se incluye el descriptor de puntero de metadatos.
3	Ninguno de los casos anteriores.

metadata_locator_record_length (longitud de registro de localizador de metadatos) – El campo de 8 bits `metadata_locator_record_length` especifica el número de `metadata_locator_record_bytes` inmediatamente posteriores. Este campo no puede codificarse con el valor 0.

metadata_locator_record_byte (byte de registro de localizador de metadatos) – El `metadata_locator_record_byte` es parte de una cadena de uno o más bytes contiguos que forman el registro de localizador de metadatos. Este registro especifica uno o más lugares que no son el tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. El formato del registro de localizador de metadatos queda definido por la aplicación de metadatos especificada en el campo `metadata_application_format`. Este registro puede contener, por ejemplo, localizadores uniformes de recursos (URL) de Internet que especifican dónde pueden encontrarse los metadatos (podría ser otra situación, además de su aparición en el tren de transporte). Si se codifica el `MPEG_carriage_flags` con los valores 0, 1 ó 2 y hay un registro de localización de metadatos, ello quiere decir que los mismos metadatos se encuentran en distintos lugares.

program_number (número de programa) – El campo de 16 bits `program_number` identifica el número de programa MPEG-2 en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que transporta los metadatos asociados. Si el campo `MPEG_carriage_flags` tiene el valor 0, significa que el tren de transporte es el tren actual, y si el campo `MPEG_carriage_flags` tiene el valor 1, se trata del tren de transporte identificado por los campos `transport_stream_location` y `transport_stream_id`.

transport_stream_location (situación del tren de transporte) – El campo de 16 bits `transport_stream_location` es definido en dominio privado. Por ejemplo, este campo puede ser utilizado por las aplicaciones para indicar el `original_network_id` definido por el ETSI.

transport_stream_id (identificador del tren de transporte) – El campo de 16 bits `transport_stream_id` identifica el tren de transporte en que van los metadatos asociados.

private_data_byte (byte de datos privados) – El `private_data_byte` es un campo de 8 bits. El `private_data_bytes` representa datos en formato de dominio privado. Estos bytes pueden utilizarse para ofrecer información adicional, según convenga.

2.6.60 Descriptor de metadatos

El descriptor de metadatos especifica los parámetros de un servicio de metadatos transportado en un tren de transporte o un tren de programa MPEG-2. En un tren de transporte MPEG-2, el descriptor se incluye en la PMT, en el bucle de descriptor de un tren elemental que transporta el servicio de metadatos. El descriptor especifica el formato de los metadatos asociados y el valor del campo `metadata_service_id` identifica el servicio de metadatos al que se aplica el descriptor de metadatos. Según sea necesario, el descriptor puede llevar información que identifique el servicio de metadatos en un conjunto de metadatos transmitidos en un carrusel DSM-CC. También pueden transportarse de modo optativo datos privados propios del formato de aplicación de metadatos.

El descriptor de metadatos indica asimismo si es necesario configurar el decodificador, y puede transportar los bytes de configuración del decodificador, pero esto sólo es práctico cuando hay pocos bytes. Si la información de configuración del decodificador es demasiado extensa para ser transportada por el descriptor, ésta deberá incluirse en un servicio de metadatos, sea en el mismo servicio de metadatos o en otro servicio de metadatos dentro del mismo programa. Gracias al descriptor de metadatos se puede identificar el servicio de metadatos que contiene la configuración del decodificador. Si se utiliza un carrusel DSM-CC para transportar la configuración del decodificador, también se puede indicar cómo extraer esta información del carrusel.

**Cuadro 2-86 – Descriptor de metadatos
(metadata descriptor)**

Sintaxis	N.º de bits	Mnemónico
Metadata_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
metadata_application_format	16	uimsbf
if (metadata_application_format == 0xFFFF) {		
metadata_application_format_identifier	32	uimsbf
}		
metadata_format	8	uimsbf
if (metadata_format == 0xFF) {		
metadata_format_identifier	32	uimsbf
}		
metadata_service_id	8	uimsbf
decoder_config_flags	3	bslbf
DSM-CC_flag	1	bslbf
reserved	4	bslbf
if (DSM-CC_flag == '1') {		
service_identification_length	8	uimsbf
for(i=0; i<service_identification_length; i++) {		
service_identification_record_byte	8	bslbf
}		
}		
if (decoder_config_flags == '001') {		
decoder_config_length	8	uimsbf
for(i=0; i<decoder_config_length; i++) {		
decoder_config_byte	8	bslbf
}		
}		
if (decoder_config_flags == '011') {		
dec_config_identification_record_length	8	uimsbf
for(i=0; i<dec_config_id_record_length; i++) {		
dec_config_identification_record_byte	8	bslbf
}		
}		
if (decoder_config_flags == '100') {		
decoder_config_metadata_service_id	8	uimsbf
}		
if (decoder_config_flags == '101' '110') {		
reserved_data_length	8	uimsbf
for(i=0; i<reserved_data_length; i++) {		
reserved	8	bslbf
}		
}		
for (i=0; i<N; i++) {		
private_data_byte	8	bslbf
}		
}		

2.6.61 Definición semántica de campos en el descriptor de metadatos

metadata_application_format (formato de aplicación de metadatos) – El campo de 16 bits `metadata_application_format` especifica cuál es la aplicación encargada de definir la utilización, la sintaxis y la semántica de `service_identification_record` y de cualesquiera bytes del descriptor definidos en un contexto privado. La codificación de este campo se define en el cuadro 2-81.

metadata_application_format_identifier (identificador del formato de aplicación de metadatos) – La codificación de este campo se define en 2.6.57.

metadata_format (formato de metadatos): La codificación de este campo se define en 2.6.59.

metadata_format_identifier (identificador del formato de metadatos) – La codificación de este campo se define en 2.6.59.

metadata_service_id (identificador del servicio de metadatos) – Este campo de 8 bits identifica el servicio de metadatos al que se aplica el descriptor de metadatos.

decoder_config_flags (banderas de configuración del decodificador) – El campo de 3 bits decoder_config_flags indica si se transporta información de configuración del decodificador y de qué manera.

Cuadro 2-87 – Banderas de configuración del decodificador (decoder_config_flags)

Valor	Descripción
000	No es necesario configurar el decodificador.
001	La configuración del decodificador se transporta en este descriptor en el campo decoder_config_byte.
010	La configuración del decodificador se transporta en el mismo servicio de metadatos al que se aplica este descriptor de metadatos.
011	La codificación del decodificador se transporta en un carrusel DSM-CC. Este valor se utilizará únicamente si el servicio de metadatos al que se aplica este descriptor utiliza el mismo tipo de carrusel DSM-CC.
100	La configuración del decodificador se transporta en otro servicio de metadatos dentro del mismo programa, que se indica en el campo decoder_config_metadata_service_id en este descriptor de metadatos.
101, 110	Reservado.
111	Definido en dominio privado.

DSM-CC_flag (bandera del DSM-CC) – Se trata de una bandera de 1 bit codificada con el valor 1 si el tren al que asocia este descriptor se transporta en un carrusel de datos o un carrusel de objetos ISO/CEI 13818-6.

NOTA 1 – El valor de tipo de tren asignado a este tren de metadatos indica si se utiliza un carrusel de objetos o un carrusel de datos.

service_identification_length (longitud de identificación del servicio) – Este campo especifica el número de service_identification_record_bytes inmediatamente posteriores.

service_identification_record_byte (byte de registro de identificación del servicio) – Este byte es parte de una cadena de uno o más bytes contiguos que especifican el service_identification_record. Este registro contiene datos sobre la extracción del servicio de metadatos de un carrusel DSM-CC. El formato del registro de localizador de metadatos queda definido por la aplicación indicada en el formato de aplicación de metadatos. Cuando se utiliza un carrusel de objetos DSM-CC, el registro podría ser el identificador de objeto unívoco (el IOP:IOR() del DSM-CC ISO/CEI 13818-6, subcláusulas 11.3.1 y 5.7.2.3) para el servicio de metadatos, por ejemplo. Del mismo modo, en caso de utilizarse un carrusel de datos DSM-CC, el registro puede incluir el transaction_id y el module_id del servicio de metadatos, por ejemplo.

decoder_config_length (longitud de configuración del decodificador) – Este campo especifica el número de decoder_config_bytes inmediatamente posteriores.

decoder_config_byte (byte de configuración del codificador) – Estos bytes comprenden la información de configuración del decodificador. Esta secuencia de bytes incluye la información de configuración que necesita el receptor para decodificar este servicio. Normalmente se transporta en el descriptor de metadatos únicamente cuando la información de configuración es muy pequeña.

decoder_config_DSM-CC_id (identificador de la configuración del decodificador en DSM-CC) – Se trata del identificador de telecarga de la información de configuración del decodificador, si esta información se transmite en un carrusel de datos DSM-CC, o del identificador del objeto información de configuración del decodificador si se transporta en un carrusel de objetos DSM-CC.

NOTA 2 – El valor de tipo de tren asignado a este tren de metadatos indica si se utiliza un carrusel de objetos o un carrusel de datos.

dec_config_identification_record_length (longitud de registro de identificación de la configuración del decodificador) – Este campo especifica el número de dec_config_identification_record_bytes inmediatamente posteriores.

dec_config_identification_record_byte (byte del registro de identificación de la configuración del decodificador) – Este byte forma parte de una cadena de uno o más bytes contiguos que especifican el dec_config_identification_record. Este registro explica la manera de extraer la configuración del decodificador necesaria del carrusel DSM-CC. El formato del registro de localizador de metadatos queda definido por el formato de aplicación de metadatos. Cuando se utiliza un carrusel de objetos DSM-CC, el registro podría ser el identificador de

objeto unívoco (el IOP:IOR() del DSM-CC ISO/CEI 13818-6, subcláusulas 11.3.1 y 5.7.2.3) para la configuración del decodificador, por ejemplo. Del mismo modo, en caso de utilizarse un carrusel de datos DSM-CC, el registro puede indicar el `transaction_id` y el `module_id` de la configuración del decodificador.

decoder_config_metadata_service_id (identificador del servicio de metadatos configuración del decodificador) – Valor del `metadata_service_id` asignado al servicio de metadatos que contiene la configuración del decodificador. El servicio de metadatos indicado por el `decoder_config_metadata_service_id` y el servicio de metadatos que utiliza la configuración del decodificador deben estar en el mismo programa. Por consiguiente, en un tren de transporte, los descriptores de metadatos para estos dos servicios de metadatos deben estar en la misma PMT. El descriptor de metadatos que corresponde al servicio de metadatos indicado por `decoder_config_metadata_service_id` deberá tener un campo `decoder_config_flag` con un valor de '001', '010' o '011'.

reserved_data_length (longitud de datos reservados) – Este campo especifica el número de bytes reservados inmediatamente posteriores.

private_data_byte (byte de datos privados) – El `private_data_byte` es un campo de 8 bits. El `private_data_bytes` representa datos en formato de dominio privado. Estos bytes pueden utilizarse para ofrecer información adicional, según convenga.

2.6.62 Descriptor del STD de metadatos

Este descriptor define los parámetros del modelo de decodificador de sistemas objetivo (STD) (parámetros definidos en 2.12.10) para el procesamiento del tren de metadatos al que se asocia el descriptor.

**Cuadro 2-88 – Descriptor del STD de metadatos
(metadata STD descriptor)**

Sintaxis	N.º de bits	Mnemónico
Metadata_STD_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	2	bslbf
metadata_input_leak_rate	22	uimsbf
reserved	2	bslbf
metadata_buffer_size	22	uimsbf
reserved	2	bslbf
metadata_output_leak_rate	22	uimsbf
}		

2.6.63 Definición semántica de campos en el descriptor del STD de metadatos

metadata_input_leak_rate (velocidad de entrada de metadatos) – El campo de 22 bits `metadata_input_leak_rate` especifica la velocidad de paso o fuga del tren de metadatos asociado en el modelo T-STD, de la memoria tampón TB_n a la memoria tampón B_n . La velocidad de paso se especifica en unidades de 400 bits/s. Si los metadatos se transportan en un tren de programas, no se especifica la codificación del campo `metadata_input_leak_rate`, puesto que la velocidad de paso a B_n es igual a la velocidad del tren de programa.

metadata_buffer_size (tamaño de la memoria tampón de metadatos) – El campo de 22 bits `metadata_buffer_size` especifica el tamaño de la memoria tampón B_n en el modelo STD para el tren de metadatos asociado. El tamaño de B_n se especifica en unidades de 1024 bytes.

metadata_output_leak_rate (velocidad de salida de metadatos) – El campo de 22 bits `metadata_output_leak_rate` especifica la velocidad de paso o fuga del servicio de metadatos asociado en el modelo STD, de la memoria tampón B_n al decodificador. La velocidad de paso se especifica en unidades de 400 bits/s. Para los trenes de metadatos transportados de manera sincrónica (tipo de tren 0x15 ó 0x19), las unidades de acceso de metadatos se eliminan instantáneamente de B_n bajo el control de las indicaciones de tiempo del PTS y entonces no se especifica la codificación del campo `metadata_output_leak_rate`.

2.6.64 Descriptor de vídeo AVC

Para trenes de vídeo conformes a la Rec. UIT-T H.264 | ISO/CEI 14496-10, el descriptor de vídeo AVC proporciona información básica para especificar los parámetros de codificación en el tren de vídeo AVC correspondiente, como por ejemplo los parámetros perfil y nivel incluidos en el SPS de un tren de vídeo AVC.

El descriptor de vídeo AVC también indica la presencia de imágenes fijas AVC y de imágenes de 24 horas AVC en el tren de vídeo AVC. Si no se incluye este descriptor en el PMT de un tren de vídeo AVC en un tren de transporte o en el PSM, si lo hubiere, para un tren de vídeo AVC en un tren de programa, ese tren de vídeo AVC no contendrá imágenes fijas AVC ni imágenes de 24 horas AVC. (Véase el cuadro 2-89.)

Cuadro 2-89 – Descriptor de vídeo AVC

Sintaxis	N.º de bits	Mnemónico
AVC_video_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_idc	8	uimsbf
constraint_set0_flag	1	bslbf
constraint_set1_flag	1	bslbf
constraint_set2_flag	1	bslbf
AVC_compatible_flags	5	bslbf
level_idc	8	uimsbf
AVC_still_present	1	bslbf
AVC_24_hour_picture_flag	1	bslbf
reserved	6	bslbf
}		

2.6.65 Definición de semántica de los campos en un descriptor de vídeo AVC

profile_idc, constraint_set0_flag, constraint_set1_flag, constraint_set2_flag, AVC_compatible_flags y level_idc – Estos campos, salvo el AVC_compatible_flags se codificarán de acuerdo con la semántica de los mismos definida en la Rec. UIT-T H.264 | ISO/CEI 14496-10. La semántica de AVC_compatible_flags es exactamente la misma a la semántica de los campos definidos para los 5 bits entre la bandera constraint_set2 y el campo level_idc en el conjunto de parámetros secuencia, según se define en la Rec. UIT-T H.264 | ISO/CEI 14496-10. Todo el tren de vídeo AVC correspondiente al descriptor AVC será conforme con el perfil, nivel y restricciones indicados por estos campos.

NOTA – En una o varias secuencias del tren de vídeo AVC el nivel puede ser inferior al nivel indicado en el descriptor de vídeo AVC, pero puede suceder que un perfil sea además un subconjunto del perfil indicado en el descriptor de vídeo AVC. Sin embargo, para todo el tren de vídeo AVC sólo se utilizarán herramientas que estén incluidas en el perfil indicado en el descriptor de vídeo AVC, si lo hubiere. Por ejemplo, si se indica el perfil principal, puede utilizarse el perfil básico en algunas secuencias, pero utilizando únicamente las herramientas que tiene el perfil principal. Si los conjuntos de parámetros secuencia en un tren de vídeo AVC indican perfiles diferentes, y no se indican otras restricciones, puede ser necesario examinar el tren para determinar con qué perfil, si lo hubiere, es conforme todo el tren. Si se hace corresponder al tren de vídeo AVC un descriptor de vídeo AVC que no es conforme con un solo perfil, el tren de vídeo AVC se deberá subdividir en dos o más subtrenes de manera que los descriptores de vídeo AVC puedan indicar un perfil sencillo para cada subtren.

AVC_still_present – Este campo de 1 bit indica, cuando está puesto a '1', que el tren de vídeo AVC puede incluir imágenes fijas AVC. Cuando está puesto a '0', el tren de vídeo AVC correspondiente no contendrá imágenes fijas AVC.

AVC_24_hour_picture_flag – Esta bandera de 1 bit indica, cuando está puesta a '1' que el tren de vídeo AVC correspondiente puede contener imágenes de 24 horas AVC. Para la definición de imagen de 24 horas AVC véase 2.1.2. Si esa bandera está puesta a '0', el tren de vídeo AVC correspondiente no contendrá imágenes de 24 horas AVC.

2.6.66 Temporización AVC y descriptor HRD

La temporización AVC y el descriptor HRD proporcionan la temporización y los parámetros HRD del correspondiente tren de vídeo AVC. Para cada tren de vídeo AVC transportado en un tren conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la temporización AVC y el descriptor HRD estarán incluidos en el PMT o en el PSM, si el tren de programa contuviera el PSM, a no ser que el tren de vídeo AVC transporte parámetros VUI con el timing_info_present_flag puesto a '1':

- para cada imagen IDR; y
- para cada imagen correspondiente a un mensaje SEI del punto de recuperación.

La ausencia de la temporización AVC y del descriptor HRD en el PMT de un tren de vídeo AVC, indica la utilización de un método de fuga en el T-STD según se define en 2.14.3.1 para la transferencia de MB_n a EB_n, pero dicha utilización también puede indicarse mediante el hrd_management_valid_flag puesto a '0' en la temporización AVC y en el descriptor HRD. Si la velocidad de transferencia en la memoria intermedia EB_n puede determinarse a partir de los parámetros HRD contenidos en un tren de vídeo AVC, y si esta velocidad de transferencia se utiliza en el T-STD para la

transferencia entre MB_n y EB_n , la temporización AVC y el descriptor HRD con el `hrd_management_valid_flag` puesto a '1' se incluirá en el PMT para ese tren de vídeo AVC. (Véase el cuadro 2-90.)

Cuadro 2-90 – Temporización AVC y descriptor HRD

Sintaxis	N.º de bits	Mnemónico
AVC timing and HRD descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
hrd_management_valid_flag	1	bslbf
reserved	6	bslbf
picture_and_timing_info_present	1	bslbf
if (picture_and_timing_info_present) {		
90kHz_flag	1	bslbf
reserved	7	bslbf
if (90kHz_flag == '0') {		
N	32	uimsbf
K	32	uimsbf
}		
num_units_in_tick	32	uimsbf
}		
fixed_frame_rate_flag	1	bslbf
temporal_poc_flag	1	bslbf
picture_to_display_conversion_flag	1	bslbf
reserved	5	bslbf
}		

2.6.67 Definición de semántica de los campos en la temporización AVC y en el descriptor HRD

hrd_management_valid_flag – Este campo de 1 bit sólo se define en trenes de transporte.

Cuando el temporizador AVC y el descriptor HRD corresponden a un tren de vídeo AVC transportado en un tren de transporte, se aplica lo siguiente. Si el `hrd_management_valid_flag` está puesto a '1', los mensajes SEI del periodo de almacenamiento intermedio y SEI de temporización de imágenes, definidos en el anexo C de la Rec. UIT-T H.264 | ISO/CEI 14496-10, estarán contenidos en el tren de vídeo AVC correspondiente. Estos mensajes SEI de periodo de almacenamiento transportarán los valores `initial_cpb_removal_delay` e `initial_cpb_removal_delay_offset` codificados para el NAL HRD. Si el `hrd_management_valid_flag` está puesto a '0', la transferencia de cada byte de MB_n a EB_n en el T-STD se realizará de conformidad con el plan de entrega para ese byte en el CPB en el NAL HRD, de acuerdo con lo indicado en los valores `initial_cpb_removal_delay` e `initial_cpb_removal_delay_offset` codificados para `SchedSelIdx = cpb_cnt_minus1`. Cuando el `hrd_management_valid_flag` está puesto a '0', se utilizará el método de fuga definido en 2.14.3.1 para la transferencia de MB_n a EB_n en el T-STD.

Cuando la temporización AVC y el descriptor HRD correspondan a un tren de vídeo AVC transportado en el tren de programa, el significado del `hrd_management_valid_flag` no está definido.

picture_and_timing_info_present – Este campo de 1 bit, cuando está puesto a '1', indica que el `90kHz_flag` y los parámetros de precisión correspondientes al reloj de sistema de 90 kHz están contenidos en este descriptor.

90kHz_flag, **N**, **K** – El `90kHz_flag`, indica, cuando está puesto a '1', que la frecuencia de la base de tiempos AVC es de 90 kHz. Para un tren de vídeo AVC la frecuencia de la base de tiempos AVC se define mediante el parámetro AVC `time_scale` en los parámetros VUI, según se determina en el anexo E a la Rec. UIT-T H.264 | ISO/CEI 14496-10. La relación entre la escala de tiempos (`time_scale`) AVC y el STC se definirá mediante los parámetros **N** y **K** en este descriptor, del modo siguiente.

$$time_scale = \frac{(N \times system_clock_frequency)}{K}$$

siendo `time_scale` la frecuencia exacta de la base de tiempos AVC, con **K** mayor o igual que **N**.

ISO/CEI 13818-1:2007 (S)

Si el 90kHz_flag está puesto a '1', N es igual a 1 y K es igual a 300. Si el 90kHz_flag está puesto a '0', los valores de N y K se indican en los valores codificados de los campos N y K.

NOTA 1 – Esto permite la correspondencia del tiempo expresado en unidades de time_scale con las unidades de 90 kHz, necesaria para el cálculo de las indicaciones de tiempo PTS y DTS, por ejemplo en decodificadores con unidades de acceso a AVC para los cuales la cabecera PES no codifica PTS o DTS.

num_units_in_tick – Se codifica exactamente del mismo modo que el campo num_units_in_tick en los parámetros VUI del anexo E a la Rec. UIT-T H.264 | ISO/CEI 14496-10. La información que proporciona este campo se aplica a todo el tren de vídeo AVC al que se asocia la temporización AVC y el descriptor HRD.

fixed_frame_rate_flag – Se codifica exactamente del mismo modo que el fixed_frame_rate_flag en los parámetros VUI en el anexo E a la Rec. UIT-T H.264 | ISO/CEI 14496-10. Cuando esta bandera está puesta a '1' indica que la velocidad de cuadros codificados es constante dentro del tren de vídeo AVC correspondiente. Cuando la bandera está puesta a '0' significa que este descriptor no proporciona información alguna sobre la velocidad de cuadro del tren de vídeo AVC correspondiente.

temporal_poc_flag – Cuando el temporal_poc_flag está puesto a '1' y el fixed_frame_rate_flag está puesto a '1', el tren de vídeo AVC correspondiente transportará la información del contador de orden de imágenes (POC, *picture order count*) (PicOrderCnt) para lo cual las imágenes se cuentan en unidades de $\Delta t_{fi,dpb}(n)$, siendo $\Delta t_{fi,dpb}(n)$ lo especificado en la ecuación E-10 a la Rec. UIT-T H.264 | ISO/CEI 14496-10. Cuando el temporal_poc_flag está puesto a '0', no se transporta información alguna sobre la posible relación entre la información POC en el tren de vídeo AVC y el tiempo.

NOTA 2 – Esto reduce la sobretara necesaria para indicar la temporización de cada unidad de acceso. Es posible calcular una PTS y una DTS eficaces para las unidades de acceso para las cuales no se transporta ninguna PTS/DTS explícitamente. La repetición del campo presentado más recientemente de la paridad adecuada (o cuadro) sucede cuando la diferencia entre las PTS de la imagen actual y la siguiente es mayor que $2 \times \Delta t_{fi,dpb}$ (o mayor que $\Delta t_{fi,dpb}$ cuando frame_mbs_only_flag es igual a 1).

picture_to_display_conversion_flag – Este campo de 1 bit indica, cuando está puesto a '1' que el tren de vídeo AVC puede transportar información de visualización en imágenes codificadas mediante el campo pic_struct en mensajes SEI picture_timing (véase el anexo D a la Rec. UIT-T H.264 | ISO/CEI 14496-10) y/o proporcionando la información del contador del orden de imágenes (POC) (PicOrderCnt) cuando las imágenes se cuentan en unidades de $\Delta t_{fi,dpb}(n)$ (véase también la semántica de temporal_poc_flag) de modo que la información de temporización para unidades de acceso AVC sucesivas se pueda obtener de la imagen anterior en el orden de presentación o de decodificación.

Cuando el picture_to_display_conversion_mode_flag está puesto a '0', los mensajes SEI de temporización de imágenes en el tren de vídeo AVC, si los hubiere, no contendrán en el campo pic_struct, y por consiguiente el pic_struct_present_flag se pondrá a '0' en los parámetros VUI en el tren de vídeo AVC.

2.6.68 Descriptor de audio AAC MPEG-2

Para cada tren ISO/CEI 13818-7 transportado directamente en paquetes PES, el descriptor de audio AAC MPEG-2 definido en el cuadro 2-91 proporciona la información básica para identificar los parámetros de codificación de esos trenes elementales de audio.

**Cuadro 2-91 – Descriptor de audio AAC MPEG-2
(MPEG-2 AAC_audio_descriptor)**

Sintaxis	N.º de bits	Mnemónico
MPEG-2_AAC_audio_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MPEG-2_AAC_profile	8	uimsbf
MPEG-2_AAC_channel_configuration	8	uimsbf
MPEG-2_AAC_additional_information	8	uimsbf
}		

2.6.69 Semántica de los campos del descriptor de audio AAC MPEG-2

MPEG-2_AAC_profile – Campo de 8 bits que indica el perfil AAC con arreglo al índice del cuadro 31 de ISO/CEI 13818-7:2006.

MPEG-2_AAC_channel_configuration – Campo de 8 bits que indica el número y la configuración de los canales audio que presenta el decodificador AAC al oyente para un determinado programa. Los valores en la gama de 1 a 6 indican el número y la configuración de canales audio correspondientes al "número de índice del tren de bits por

defecto" del cuadro 42 de ISO/CEI 13818-7:2006. El resto de los valores indican que el número y configuración de canales audio no están definidos.

MPEG-2_AAC_additional_information – Campo de 8 bits que indica si los datos de ampliación del ancho de banda definidos en ISO/CEI-13818-7:2006 están incorporados en el tren de bits AAC de conformidad con el cuadro 2-92.

Cuadro 2-92 – Valores del campo MPEG-2_AAC_additional_information

Valor	Descripción
0x00	Datos AAC de conformidad con ISO/CEI 13818-7:2006
0x01	Datos AAC que contienen datos de ampliación del ancho de banda con arreglo a ISO/CEI 13818-7:2006
0x02-0xFF	Reservado

2.7 Restricciones impuestas a la semántica del tren multiplexado

2.7.1 Frecuencia de codificación de la referencia de reloj de sistema

El tren de programa se construirá de modo que el intervalo de tiempo entre los bytes que contienen el último bit de los campos de base de referencia de reloj de sistema en pacas sucesivas será inferior o igual a 0,7 s. De este modo:

$$|t(i) - t(i')| \leq 0,7 \text{ s}$$

para todas las i e i' , donde i e i' son los bytes que contienen el último bit de los campos de base de referencia de reloj de sistemas consecutivos.

2.7.2 Frecuencia de codificación de la referencia de reloj de programa

El tren de transporte se construirá de modo que el intervalo de tiempo entre el byte que contiene el último bit de los campos de base de referencia de reloj de programa en apariciones sucesivas de las PRC en paquetes del tren de transporte del PCR_PID para cada programa sea inferior o igual a 0,1 s. De este modo:

$$|t(i) - t(i')| \leq 0,1 \text{ s}$$

para todas las i e i' , donde i e i' son los índices de los bytes que contienen el último bit de campos de base de referencia de reloj de programa consecutivos en paquetes del tren de transporte del PCR_PID para cada programa.

Debe haber por lo menos dos (2) PCR, del PCR_PID en un tren de transporte, entre discontinuidades PCR consecutivas (véase 2.4.3.4), para facilitar el enganche de fase y la extrapolación de los tiempos de entrega de bytes.

2.7.3 Frecuencia de codificación de la referencia de reloj de tren elemental

El tren de programa y el tren de transporte se construirán de modo que si el campo de referencia de reloj de tren elemental se codifica en cualquier paquete PES que contiene datos de un tren elemental dado, el intervalo de tiempo en el STD de PES entre los bytes que contienen el último bit de los campos de base de ESCR sea inferior o igual a 0,7 s. En trenes PES, la codificación de ESCR se requiere con el mismo intervalo. De este modo:

$$|t(i) - t(i')| \leq 0,7 \text{ s}$$

para todas las i e i' , donde i e i' son los índices de los bytes que contienen en los últimos bits de campos de base de ESCR consecutivos.

NOTA – La codificación de campos de referencia de reloj de tren elemental es facultativa, no hay que codificarlos, pero si se codifican, se aplica esta restricción.

2.7.4 Frecuencia de codificación de indicación de tiempo de presentación

El tren de programa y el tren de transporte se construirán de modo que la diferencia máxima entre las indicaciones de tiempo de presentación codificadas referentes a cada tren de vídeo o audio elemental sea 0,7 s. De este modo:

$$|tp_n(k) - tp_n(k'')| \leq 0,7 \text{ s}$$

ISO/CEI 13818-1:2007 (S)

para n , k , y k'' conformes:

- $P_n(k)$ y $P_n(k'')$ son unidades de presentación para las cuales se codifican indicaciones de tiempo de presentación;
- k y k'' se eligen de modo que no haya unidades de presentación, $P_n(k')$ con una indicación de tiempo de presentación codificada y con $k < k' < k''$; y
- no existe ninguna discontinuidad de decodificación en el tren elemental n entre $P_n(k)$ y $P_n(k'')$.

La restricción de 0,7 s no se aplica en el caso de:

- imágenes fijas según se define en 2.1;
- imágenes fijas AVC;
- unidades de acceso a AVC con una velocidad de cuadro muy baja, cuando el tiempo de presentación de las unidades de acceso subsiguientes difiere en más de 0,7 s. En este caso particular, los parámetros VUI `num_units_in_tick` y `time_scale` figurarán en el tren de vídeo AVC o en la temporización AVC y el descriptor HRD correspondientes al tren de vídeo AVC.

NOTA – El tiempo de presentación de una unidad de acceso AVC es equivalente al tiempo de salida DPB $t_{o,dpb}(n)$ definido en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10.

2.7.5 Codificación condicional de indicaciones de tiempo

Para cada tren elemental de un tren de programa o de un tren de transporte, la indicación de tiempo de presentación (PTS) se codificará para la primera unidad de acceso.

Existe una discontinuidad de decodificación en el comienzo de una unidad de acceso $A_n(j)$ en un tren elemental si el tiempo de decodificación $td_n(j)$ de esa unidad de acceso es mayor que el valor más grande admisible dada la tolerancia especificada en la frecuencia de reloj de sistema. Para vídeo, excepto cuando el estado del modo truco es verdadero o cuando la bandera de bajo retardo es '1', esto se permite solamente en el comienzo de una secuencia de vídeo. Si existe una discontinuidad de decodificación en un tren elemental de audio o vídeo, en el tren de transporte o en el tren de programa, se codificará PTS con referencia a la primera unidad de acceso después de cada discontinuidad de decodificación, excepto cuando el estado del modo truco es verdadero.

Cuando `low_delay` es '1', una PTS se codificará para la primera unidad de acceso después de la subutilización de una memoria tampón EB_n o B_n .

Una PTS sólo puede estar presente en un encabezamiento de paquete PES de tren elemental de vídeo o de audio Rec. UIT-T H.222.0 | ISO/CEI 13818-1 si el primer byte de un código de comienzo de imagen o el primer byte de una unidad de acceso están contenidos en el paquete PES.

Aparecerá una indicación de tiempo de decodificación (DTS) en un encabezamiento de paquete PES solamente si se cumplen las dos condiciones siguientes:

- una PTS está presente en el encabezamiento de paquete PES;
- el tiempo de decodificación difiere del tiempo de presentación.

Para cada imagen de 24 horas AVC no se codificará ningún valor explícito de PTS y DTS en la cabecera PES. Para tales unidades de acceso AVC, los decodificadores deducirán el tiempo de presentación a partir de los parámetros contenidos en el tren de vídeo AVC. Por consiguiente, cada tren de vídeo AVC que contiene una o más imágenes de 24 horas AVC:

- transportará mensajes SEI de temporización de imágenes con valores codificados de `cpb_removal_delay` y `dpb_output_delay`; o
- transportará parámetros VUI con `fixed_frame_rate_flag` puesto a '1' y transportará información del contador de orden de imágenes (POC) (`PicOrderCnt`) siendo imágenes que se cuentan en unidades de $\Delta t_{fi,dpb}(n)$, donde $\Delta t_{fi,dpb}(n)$ se especifica en la ecuación E-10 de la Rec. UIT-T H.264 | ISO/CEI 14496-10.

NOTA 1 – Los requisitos indicados en el segundo punto se satisfacen cuando la temporización AVC y el descriptor HRD corresponden con el tren de vídeo AVC con el `fixed_frame_rate_flag` puesto a '1' y el `temporal_poc_flag` puesto a '1'.

Lo que se indica a continuación se aplica a las unidades de acceso AVC en trenes de vídeo AVC transportados en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Para cada unidad de acceso AVC que no representa una imagen de 24 horas AVC se proporcionará una cabecera PES con una PTS codificada y, si procede, un valor DTS, a no ser que todas las condiciones expresadas de conformidad con uno de los siguientes puntos sea cierta:

- La secuencia de vídeo AVC contiene los siguientes mensajes SEI, según lo indicado por los parámetros VUI:

- a) mensajes SEI de temporización de imagen que proporcionan los parámetros `cpb_removal_delay` y `dpb_output_delay`, y
- b) mensajes SEI de periodo de almacenamiento intermedio que proporcionan los parámetros `initial_cpb_removal_delay` e `initial_cpb_removal_delay_offset`.

NOTA 2 – Cuando la secuencia de vídeo AVC contiene mensajes SEI de temporización de imágenes, estos mensajes están presentes para cada unidad de acceso AVC, según lo exige la Rec. UIT-T H.264 | ISO/CEI 14496-10. Cuando la secuencia de vídeo AVC contiene mensajes SEI de periodo de almacenamiento intermedio, estos mensajes estarán presentes en cada unidad de acceso IDR y para cada unidad de acceso que corresponda a un mensaje SEI de punto de recuperación, según lo exige la Rec. UIT-T H.264 | ISO/CEI 14496-10.

- Un temporizador AVC y un descriptor HRD corresponden al tren de vídeo AVC y en este descriptor `fixed_frame_rate_flag` está puesto a '1' y `temporal_poc_flag` está puesto a '1'.
- Un temporizador AVC y un descriptor HRD corresponden a un tren de vídeo AVC y en este descriptor `fixed_frame_rate_flag` está puesto a '1', `picture_to_display_conversion_flag` está puesto a '1', `temporal_poc_flag` está puesto a '0' y la secuencia de vídeo AVC contiene mensajes SEI de temporización de imágenes con el campo `pic_struct`.
NOTA 3 – En este caso en particular el campo `pic_struct` se utiliza para determinar los valores PTS subsiguientes.
- Un temporizador AVC y un descriptor HRD corresponden a un tren de vídeo AVC y en este descriptor `fixed_frame_rate_flag` está puesto a '1' y `temporal_poc_flag` está puesto a '0' y `picture_to_display_conversion_flag` está puesto a '0'.

NOTA 4 – En este caso la información POC en el tren de vídeo AVC se utiliza para determinar los valores PTS subsiguientes.

2.7.6 Constricciones de temporización para la codificación escalonable

Si una secuencia audio se codifica utilizando un tren de bits de extensión, por ejemplo el especificado en ISO/CEI 13818-3, las correspondientes unidades de decodificación/presentación en las dos capas tienen valores PTS idénticos.

Si una secuencia de vídeo se codifica como una mejora SNR de otra secuencia, como se especifica en 7.8 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, el conjunto de tiempos de presentación para ambas secuencias deberá ser el mismo.

Si una secuencia de vídeo se codifica como dos particiones, como se especifica en 7.10 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, el conjunto de tiempos de presentación para ambas particiones deberá ser el mismo.

Si una secuencia de vídeo se codifica como una mejora escalonable espacial, como se especifica en 7.7 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, se aplicará lo siguiente:

- Si ambas secuencias tienen la misma velocidad de trama, el conjunto de tiempos de presentación para ambas secuencias deberá ser el mismo.
NOTA – Esto no entraña que el tipo de codificación de imagen sea el mismo en ambas capas.
- Si las secuencias tienen diferentes velocidades de trama, el conjunto de tiempos de presentación será tal que el mayor número posible de tiempos de presentaciones sean comunes a ambas secuencias.
- La imagen a partir de la cual se hace la predicción espacial será una de las siguientes:
 - la imagen de capa más baja coincidente o decodificada más recientemente;
 - la imagen de capa más baja coincidente o decodificada más recientemente, que sea una imagen I o P;
 - la segunda imagen de capa más baja decodificada más recientemente que sea una imagen I o P, siempre que la capa más baja no tenga la bandera de bajo retardo puesta a '1'.

Si una secuencia de vídeo se codifica como una mejora temporalmente escalonable de otra secuencia, como se especifica en 7.9 de la Rec. UIT-T H.262 | ISO/CEI 13818-2, las imágenes de capa más baja siguientes pueden utilizarse como la referencia. Los tiempos son relativos a los tiempos de presentación:

- la imagen de capa más baja coincidente o presentada más recientemente;
- la siguiente imagen de capa más baja que habrá de presentarse.

2.7.7 Frecuencia de codificación de tamaño de memoria tampón P-STD en encabezamientos de paquete PES

En un tren de programa, los campos `P-STD_buffer_scale` y `P-STD_buffer_size` aparecerán en el primer paquete PES de cada tren y de nuevo siempre que el valor cambia. Pueden aparecer también en cualquier otro paquete PES.

2.7.8 Codificación de encabezamiento de sistema en el tren de programa

En un tren de programa, el encabezamiento de sistema puede estar presente en cualquier saca que sigue inmediatamente al encabezamiento de saca. El encabezamiento de sistema estará presente en la primera saca de un tren de programa. Los valores codificados en todos los encabezamientos de sistema del tren de programa serán idénticos.

2.7.9 Tren de programa de parámetros de sistema constreñidos

Un tren de programa es un "tren de parámetros de sistema constreñidos" (CSPS, *constrained system parameters stream*), si se conforma con los límites especificados en esta subcláusula. Los trenes de programa no están restringidos a los límites especificados por el CSPS. Un CSPS puede ser identificado por medio de la bandera CSPS definida en el encabezamiento de sistema en 2.5.3.5. El CSPS es un subconjunto de todos los trenes de programas posibles.

Velocidad de paquete

En el CSPS, la velocidad máxima a la cual los paquetes llegarán a la entrada del P-STD es 300 paquetes por segundo si el valor codificado en el campo `rate_bound` (véase 2.5.3.6) es menor o igual a 4 500 000 bit/s si la bandera de restricción de velocidad de paquetes está puesta a '1', e inferior o igual a 2 000 000 bit/s si la bandera de restricción de velocidad de paquetes está puesta a '0'. Para velocidades binarias más altas, la velocidad de paquetes CSPS está limitada por una relación lineal al valor codificado en el campo `rate_bound`.

Específicamente, para todas las sacas p en el tren de programa cuando la bandera de restricción de velocidad de paquetes está puesta a '1' (véase 2.5.3.5):

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[1, \frac{R_{\text{máx}}}{4,5 \times 10^6} \right] \quad (2-27)$$

y cuando la bandera de restricción de velocidad de paquetes está puesta a '0':

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[1, \frac{R_{\text{máx}}}{2,5 \times 10^6} \right] \quad (2-28)$$

donde

$$R_{\text{máx}} = 8 \times 50 \times \text{rate_bound} \quad \text{bit/s} \quad (2-29)$$

NP es el número de prefijos de código de comienzo de paquete y de códigos de comienzo de encabezamiento entre códigos de comienzo de saca adyacente o entre el último código de comienzo de saca y el código de fin de programa MPEG según se define en el cuadro 2-37 y la semántica en 2.5.3.2;

t(i) es el tiempo, medido en segundos, codificado en la referencia de reloj de sistema de la saca p ;

t(i') es el tiempo, medido en segundos, codificado en la referencia de reloj de sistema para la saca $p + 1$, que sigue inmediatamente a la saca p o en el caso de la saca final en el tren de programa, el tiempo de llegada del byte que contiene el último bit del código de fin de programa MPEG.

Tamaño de memoria tampón del decodificador

En el caso de un CSPS, el tamaño máximo de cada memoria tampón de entrada en el decodificador-objetivo del sistema está limitado. Se aplican diferentes límites para trenes elementales de vídeo y trenes elementales de audio.

En el caso de un tren elemental de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 11172-2 en un CSPS, se aplica lo siguiente:

BS_n tiene un tamaño que es igual a la suma del tamaño del verificador de almacenamiento en memoria tampón de vídeo (VBV) especificado en la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 11172-2, respectivamente, y una cantidad adicional de almacenamiento BS_{add} . BS_{add} y se especifica como:

$$BS_{\text{add}} \leq \text{MAX} [6 \times 1024, R_{\text{vmáx}} \times 0,001] \text{ bytes}$$

siendo $R_{\text{vmáx}}$ la velocidad binaria de vídeo máxima del tren elemental de vídeo de la Rec. UIT-T H.262 | ISO/CEI 13818-2 o ISO/CEI 11172-2.

En el caso de un tren elemental de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 en un CSPS, se aplica lo siguiente:

BS_n tiene un tamaño igual a la suma de cpb_size y una cantidad adicional de almacenamiento BS_{add} . BS_{add} que se especifica como:

$$BS_{add} \leq MAX [6 \times 1024, R_{vm\acute{a}x} \times 0,001] \text{ bytes}$$

siendo $R_{vm\acute{a}x}$ la velocidad binaria de vídeo máxima del tren de vídeo AVC y

cpb_size el tamaño de $CpbSize[cpt_cnt_minus1]$ de la CPB para el formato de tren de bytes indicado en $hrd_parameters()$ de NAL en el tren de vídeo AVC. Si el tren de vídeo AVC no contiene $hrd_parameters()$ de NAL, el cpb_size tendrá el tamaño definido por $1200 \times MaxCPB$ en el anexo A a la Rec. UIT-T H.264 | ISO/CEI 14496-10 para el nivel aplicado.

En el caso de un tren elemental de audio en un CSPS, se aplica lo siguiente:

$$BS_n \leq 4096 \text{ bytes}$$

En el caso de un tren elemental de audio de ISO/CEI 13818-7 con ADTS en un CSPS, se aplica lo siguiente para soportar 8 canales:

$$BS_n \leq 8976 \text{ bytes}$$

2.7.10 Tren de transporte

Eganche de velocidad de muestra en trenes de transporte

En el tren de transporte habrá una relación racional constante especificada entre la velocidad de muestreo de audio y la frecuencia de reloj de sistema en el decodificador-objetivo de sistema y, de manera similar, una relación racional especificada entre la velocidad de trama de vídeo y la frecuencia de reloj de sistema. En 2.4.2 se define la frecuencia de reloj de sistema. La velocidad de trama de vídeo es la especificada en la Rec. UIT-T H.262 | ISO/CEI 13818-2 o en ISO 11172-2. La velocidad de muestreo de audio es la especificada en ISO/CEI 13818-3 o ISO/CEI 11172-3. Para todas las unidades de presentación en trenes elementales de audio en el tren de transporte, la relación entre la referencia de reloj de sistema y la velocidad de muestreo de audio real, SCASR, es constante e igual al valor indicado en la siguiente tabla a la velocidad de muestreo nominal indicada en el tren de audio:

$$SCASR = \frac{system_clock_frequency}{audio_sample_rate_in_the_T-STD} \tag{2-30}$$

La notación $\frac{X}{Y}$ denota división real.

Frecuencia de muestreo de audio nominal (kHz)	16	32	22,05	44,1	24	48
SCASR	$\frac{27\ 000\ 000}{16\ 000}$	$\frac{27\ 000\ 000}{32\ 000}$	$\frac{27\ 000\ 000}{22\ 050}$	$\frac{27\ 000\ 000}{44\ 100}$	$\frac{27\ 000\ 000}{24\ 000}$	$\frac{27\ 000\ 000}{48\ 000}$

Para todas las unidades de presentación en cada tren de vídeo de ISO/CEI 11172-2 y de la Rec. UIT-T H.262 | ISO/CEI 13818-2 del tren de transporte, la relación entre la frecuencia de reloj del sistema y la velocidad de cuadro de vídeo real, SCFR, es constante e igual al valor indicado en el siguiente cuadro a la velocidad de cuadros nominal indicada en el tren de vídeo.

$$SCFR = \frac{system_clock_frequency}{frame_rate_in_the_T-STD} \tag{2-31}$$

Velocidad de trama nominal (Hz)	23,976	24	25	29,97	30	50	59,94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

ISO/CEI 13818-1:2007 (S)

Los valores de SCFR son exactos. La velocidad de trama real difiere ligeramente de la velocidad nominal cuando la velocidad nominal es 23,976, 29,97 ó 59,94 tramas por segundo.

Para los trenes de vídeo de ISO/CEI 14496-2 que se conducen por el tren de transporte, la base de tiempos del tren de vídeo de ISO/CEI 14496-2, definida mediante `vop_time_increment_resolution`, estará vinculado al STC y será exactamente igual a N veces la frecuencia del reloj del sistema (`system_clock_frequency`) dividida por K, siendo N y K enteros que tienen un valor fijo en cada secuencia de objeto visual, con K mayor o igual que N.

Para los trenes de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 la base de tiempos del tren de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 estará vinculada a la frecuencia de reloj del sistema. La frecuencia de la base de tiempos AVC se define mediante el parámetro `time_scale`, y esta frecuencia será exactamente igual a N veces la frecuencia del reloj del sistema dividida por K, siendo N y K enteros que tienen un valor fijo en cada secuencia de vídeo AVC con K mayor o igual que N. Por ejemplo, si la escala de tiempos (`time_scale`) es 90 000, la frecuencia de la base de tiempos AVC es exactamente igual a la frecuencia del reloj del sistema dividido por 300.

2.8 Compatibilidad con ISO/CEI 11172

El tren de programa de la presente Recomendación | Norma Internacional se define para que sea compatible hacia adelante con ISO/CEI 11172-1. Los decodificadores del tren de programa definidos en la presente Recomendación | Norma Internacional soportarán también la decodificación de ISO/CEI 11172-1.

2.9 Registro de identificadores de derechos de autor

2.9.1 Generalidades

Las partes 1, 2 y 3 de ISO/CEI 13818 proporcionan el soporte para la gestión de la protección de los derechos de autor de los trabajos audiovisuales. En la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 la identificación de trabajos audiovisuales se efectúa por medio de un descriptor de derechos de autor, mientras que en la Rec. UIT-T H.262 | ISO/CEI 13818-2 y en ISO/CEI 13818-3 se emplean campos que permiten identificar a los titulares de los derechos de autor mediante campos de sintaxis en la sintaxis de trenes elementales. En esta Recomendación | Norma Internacional se presenta el método para obtener y registrar los identificadores de derechos de autor de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

En la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se especifica un identificador de derechos de autor de 32 bits único que es un identificador de código de tipo de trabajo (como ISBN, ISSN, ISRC, etc.) incluido en el descriptor de derechos de autor. El identificador de derechos de autor permite la identificación de un gran número de autoridades de registro de derechos de autor. Cada Autoridad de Registro de derechos de autor puede especificar una sintaxis y una semántica para identificar los trabajos audiovisuales u otros trabajos protegidos por derechos de autor dentro de esa organización particular de registro de derechos de autor mediante el empleo adecuado del campo información de derechos de autor adicional (`additional_copyright_info`) de longitud variable que contiene el número de derechos de autor.

En la subcláusula siguiente y en los anexos L, M y N se indican de manera resumida los beneficios y las responsabilidades de todas las partes por lo que respecta al registro del identificador de derechos de autor.

2.9.2 Implementación de una autoridad de registro (RA)

El JTC 1 de la ISO/CEI instará a que se presenten candidaturas para que una organización internacional asuma la función de autoridad de registro del **copyright_identifier (identificador de derechos de autor)** definido en 2.6.24. La organización elegida funcionará como la autoridad de registro. La así llamada autoridad de registro desempeñará sus tareas de conformidad con el anexo H de las directrices del JTC 1. El identificador de derechos de autor registrado (`registered_copyright_identifier`) se denominará en lo sucesivo identificador registrado (RID).

Una vez elegida la autoridad de registro, el JTC 1 promoverá la creación de un grupo de gestión de registros (RMG) encargado de examinar las apelaciones presentadas por las organizaciones cuya petición de un RID, para emplearlo según la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, hubiera sido rechazada por la autoridad de registro.

Los anexos L, M y N contienen información sobre los procedimientos que se han de seguir para registrar un identificador de derechos de autor único.

2.10 Registro de formato de datos privados

Este texto proporciona el descriptor de registro de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 para que los usuarios de la presente Especificación puedan llevar datos sin ambigüedades cuando su formato no es reconocido por esta Especificación. Esta disposición permite que la presente Especificación admita todo tipo de datos, al tiempo que suministra un método para identificar de manera inequívoca las características de los datos privados subyacentes.

2.10.1 Generalidades

En la siguiente subcláusula y en los anexos O y P se indican de manera resumida los beneficios y las responsabilidades de todas las partes por lo que respecta al registro del formato de datos privados.

2.10.2 Implementación de una autoridad de registro (RA, *registration authority*)

El JTC 1/SC 29 de la ISO/CEI instará a los organismos miembros de la ISO o a los Comités Nacionales de la CEI a que presenten candidaturas para ejercer la función de autoridad de registro del **identificador de formato** (**format_identifier**) definido en 2.6.8 y 2.6.9. La organización elegida funcionará como la autoridad de registro. La así llamada autoridad de registro desempeñará sus tareas de conformidad con el anexo H de las directrices del JTC 1. El identificador de formato de datos privados registrado (registered private data format_identifier) se llamará en adelante identificador registrado (RID, *registered identifier*).

Una vez elegida la autoridad de registro, el JTC 1 promoverá la creación de un grupo de gestión de registros (RMG, *registration management group*) encargado de examinar las apelaciones presentadas por las organizaciones cuya petición de un RID, para emplearlo según esta Especificación, hubiera sido rechazada por la autoridad de registro.

Los anexos O y P contienen información sobre los procedimientos que se han de seguir para registrar un identificador de formato único.

2.11 Transporte de datos ISO/CEI 14496

2.11.1 Introducción

Un tren conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 puede transportar trenes elementales individuales de ISO/CEI 14496-2 y 14496-3 así como escenas audiovisuales de ISO/CEI 14496-1 con sus correspondientes trenes. Normalmente, los trenes de ISO/CEI 14496 serán elementos de un programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, definidos mediante el PMT en un tren de transporte y el PSM en un tren de programa.

Para el transporte de datos de ISO/CEI 14496 en trenes de transporte y trenes de programa se hace la distinción entre los trenes elementales individuales y escenas audiovisuales de ISO/CEI 14496-1 con sus correspondientes trenes. Para el transporte de trenes elementales individuales de ISO/CEI 14496-2 y 14496-3 sólo se utilizan herramientas del sistema de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, según se define en 2.11.2. Para el transporte de escenas audiovisuales de ISO/CEI 14496-1 y sus correspondientes trenes elementales de ISO/CEI 14496 contenidos en trenes SL_packetized de ISO/CEI 14496-1 o trenes FlexMux, se utilizan herramientas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y de ISO/CEI 14496-1 según se define en 2.11.3.

El transporte de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 por trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se especifica en 2.14.

2.11.2 Transporte de trenes elementales ISO/CEI 14496-2 y 14496-3 individuales en paquetes PES

2.11.2.1 Introducción

Los trenes elementales ISO/CEI 14496-2 y 14496-3 individuales pueden ser transportados en paquetes PES como PES_packet_data_bytes. Para el paquetizado PES no se aplican restricciones específicas de alineación de datos. Para la sincronización, los PTS y, cuando proceda, los DTS se codifican en el encabezamiento de paquete PES que transporta datos de trenes elementales ISO/CEI 14496; para la codificación de PTS y DTS se aplican las mismas restricciones que para los trenes elementales ISO/CEI 13818. Para una visión general de cómo transportar trenes ISO/CEI 14496 individuales dentro de un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, véase el cuadro 2-93.

Cuadro 2-93 – Transporte de trenes ISO/CEI 14496 individuales en trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

Objeto visual de ISO/CEI 14496-2	Transporte en paquetes PES	Stream_type = 0x10	Stream_id = '1110 xxxx'
Audio de ISO/CEI 14496-3	Transporte en paquetes PES	Stream_type = 0x11	Stream_id = '110x xxxx'

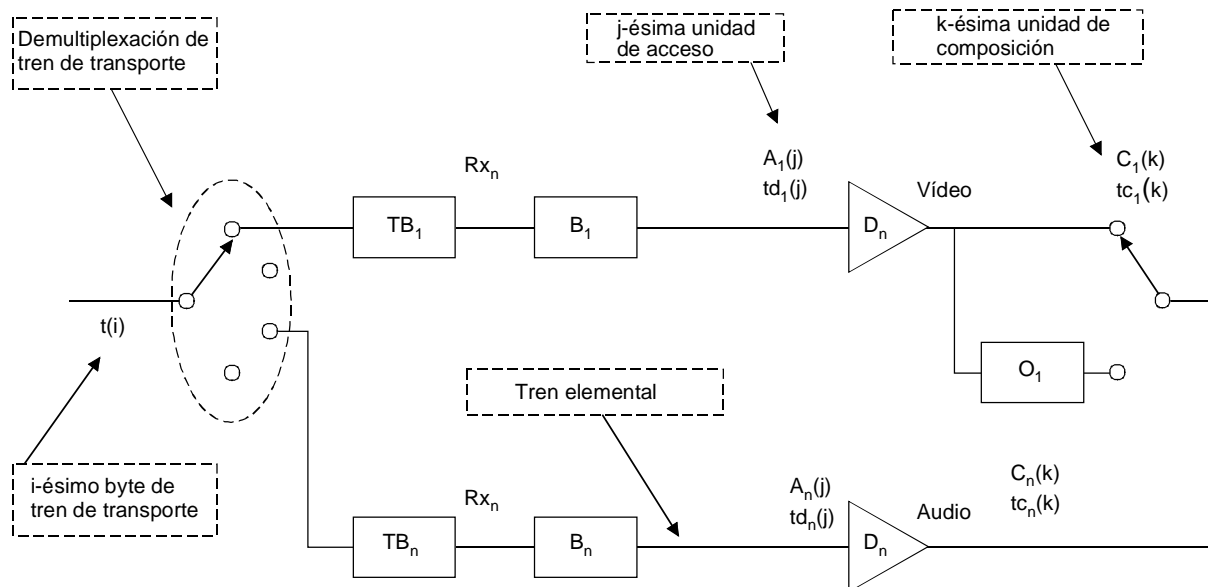
Si un PTS o un DTS está presente en un encabezamiento de paquete PES, hará referencia al objeto visual que sigue, sea el primer código de comienzo VOP o el primer código de comienzo de objeto de textura fija que comienza en el paquete PES. Cada tren vídeo ISO/CEI 14496-2 transportado por trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 contendrá la información requerida para decodificar el tren vídeo ISO/CEI 14496-2; en consecuencia, el tren contendrá encabezamientos de secuencias de objetos visuales, encabezamientos de objetos visuales y encabezamientos de capas de objetos vídeo.

En el caso de un tren elemental ISO/CEI 14496-3, antes del paquetizado PES el tren de datos elemental será encapsulado primero en la sintaxis de transporte LATM/LOAS AudioSyncStream(), definida en ISO/CEI 14496-3. Si está presente un PTS en el encabezamiento de paquete PES, hará referencia a la primera trama de audio que sigue a la primera palabra de sincronización que comienza en la cabida útil del paquete PES.

El transporte de trenes elementales de ISO/CEI 14496-2 e ISO/CEI 14496-3 individuales en paquetes PES se identificará mediante valores apropiados del identificador de tren (stream_id) y tipo de tren (stream_type), que indican el uso de vídeo de ISO/CEI 14496-2 o audio de 14496-3. Además, ese transporte será señalado por el descriptor de vídeo MPEG-4 o el descriptor de audio MPEG-4, respectivamente. Estos descriptores serán transportados en el bucle de descriptor de la entrada del respectivo tren elemental en el cuadro de correspondencia de programas en el caso de un tren de transporte o en la correspondencia de trenes de programa, si los hubiere, en el caso de un tren de programa. La Rec. UIT-T H.222.0 | ISO/CEI 13818-1 no especifica la presentación de trenes elementales de ISO/CEI 14496-2 e ISO/CEI 14496-3 en el contexto de un programa.

2.11.2.2 Extensiones de STD para trenes elementales ISO/CEI 14496 individuales

El modelo T-STD incluye una memoria tampón de transporte TB_n y una memoria tampón múltiplex B_n antes de la codificación de cada tren elemental ISO/CEI 14496 n individuales. Obsérvese que en el T-STD, la memoria tampón múltiplex única B_n está aplicada para vídeo ISO/CEI 14496-2, como se indica en la figura 2-4, en lugar del modelo con dos memorias tampón MB_n y EB_n utilizado para vídeo ISO/CEI 13818-2 en T-STD. Para las memorias tampón TB_n y B_n y la velocidad R_{x_n} entre TB_n y B_n se aplican las siguientes restricciones.



T1609310-00/d09

Figura 2-4 – Extensiones del modelo T-STD para trenes elementales ISO/CEI 14496 individuales

En el caso de transporte de un tren ISO/CEI 14496-2:

Tamaño BS_n de la memoria tampón B_n:

$$BS_n = BS_{mux} + BS_{oh} + VB_{V_{m\acute{a}x}}[\text{perfil}, \text{nivel}]$$

donde:

BS_{oh}, el almacenamiento de tara de paquete se define como:

$$BS_{oh} = (1/750) \text{ segundos} \times \text{m\acute{a}x} \{R_{m\acute{a}x}[\text{perfil}, \text{nivel}], 2\ 000\ 000 \text{ bit/s}\}$$

y

BS_{mux}, el almacenamiento múltiplex adicional se define como:

$$BS_{mux} = 0,004 \text{ segundos} \times \text{m\acute{a}x} \{R_{m\acute{a}x}[\text{perfil}, \text{nivel}], 2\ 000\ 000 \text{ bit/s}\}$$

Velocidad R_{x_n} :

$$R_{x_n} = 1,2 \times R_{\text{m}\acute{a}\text{x}}[\text{perfil, nivel}]$$

donde:

$VBV_{\text{m}\acute{a}\text{x}}[\text{perfil, nivel}]$ y $R_{\text{m}\acute{a}\text{x}}[\text{perfil, nivel}]$ se definen en ISO/CEI 14496-2 para cada perfil y nivel. Para perfiles y niveles para los cuales no se especifica un valor $VBV_{\text{m}\acute{a}\text{x}}$, el tamaño de B_n y la velocidad R_{x_n} son definidas por el usuario.

En el caso de transporte de un tren ISO/CEI 14496-3:

Tamaño BS_n de memoria tampón B_n para audio ISO/CEI 14496-3 AAC,

$$\text{en los demás casos } BS_n = BS_{\text{mux}} + BS_{\text{dec}} + BS_{\text{oh}} = 3584 \text{ bytes}$$

En este caso, el tamaño de la memoria tampón de decodificación de unidades de acceso de BS_{dec} y la memoria tampón de taras de paquete PES BS_{oh} están constreñidos por:

$$BS_{\text{dec}} + BS_{\text{oh}} \leq 2848 \text{ bytes}$$

Una porción (736 bytes) de la memoria tampón de 3584 bytes se atribuye para almacenamiento tampón para permitir la multiplexación. El resto, 2848 bytes, se comparte para almacenamiento tampón de unidades de acceso BS_{dec} , BS_{oh} y multiplexación adicional.

Velocidad R_{x_n} para audio ISO/CEI 14496-3 AAC idéntica a la definida para audio ISO/CEI 13818-7 ADTS en 2.4.2.3,

$$\text{en los demás casos } R_{x_n} = 2\,000\,000 \text{ bit/s}$$

El modelo P-STD incluye una memoria tampón múltiplex B_n antes de la decodificación de cada tren elemental n ISO/CEI 14496 individual. El tamaño BS_n de la memoria B_n en el P-STD es definido por el campo P-STD_buffer_size en el encabezamiento PES.

2.11.3 Transporte de escenas audiovisuales ISO/CEI 14496 y trenes ISO/CEI asociados

2.11.3.1 Introducción

Esta subcláusula describe el encapsulado y la señalización cuando una escena audiovisual representada por datos ISO/CEI 14496 es transportada en un tren de programa o en un tren de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. El contenido ISO/CEI 14496 consiste en el descriptor de objeto inicial y en un número variable de trenes, tales como trenes de descriptor de objeto, trenes de descripción de escenas (que transportan unidades de instrucción BIFS o unidades de acceso de animación BIFS), trenes IPMP, trenes OCI y trenes audiovisuales. Cada tren ISO/CEI 14496 estará contenido en un tren paquetizado SL y facultativamente puede ser multiplexado en un tren FlexMux, ambos definidos en ISO/CEI 14496-1. Para el transporte de un tren de programa o tren de transporte UIT-T H.222.0 | ISO/CEI 13818-1, estos trenes paquetizados SL y los trenes FlexMux contendrán los campos referencia de reloj de objeto (OCR, *object clock reference*) y referencia de reloj FlexMux (FCR, *FlexMux clock reference*) especificados en 2.11.3.4 y 2.11.3.5, respectivamente. Los trenes paquetizados SL o los trenes FlexMux son encapsulados después en paquetes PES o en secciones ISO_CEI_14496 antes de la paquetización de trenes de transporte o la multiplexación de trenes de programa. Las secciones ISO_CEI_14496 se construyen en el formato largo de las secciones H.222.0 | ISO/CEI 13818-1.

2.11.3.2 Asignación de valores de ES_ID

Una escena ISO/CEI-14496-1 transportada por un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 puede asociar un número de trenes ISO/CEI 14496 y otros trenes ISO/CEI 13818 mediante el uso del parámetro ES_ID. La escena y los trenes asociados pueden ser transportados por el mismo tren Rec. UIT-T H.222.0 | ISO/CEI 13818-1, pero una escena puede hacer también referencia a trenes transportados por otro medio, por ejemplo, por una red IP. En la presente Especificación no se define la manera de identificar estos otros medios.

En ISO/CEI 14496-1 se definen las reglas de alcance de nombre para identificadores. Estas reglas permiten utilizar el mismo valor de ES_ID para dos trenes diferentes dentro del contenido ISO/CEI 14496. Cuando se transportan una o múltiples escenas ISO/CEI 14496-1 en un programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1, no se producirán valores duplicados de ES_ID dentro del programa, de manera que cada tren paquetizado SL ISO/CEI 14496 o canal FlexMux 14496-1 tiene un valor ES_ID único en el programa.

2.11.3.3 Temporización de escenas ISO/CEI 14496 y trenes asociados

Cuando es transportado por un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la base de tiempo de objeto de cada tren ISO/CEI 14496 estará enganchada al STC de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, es decir:

$$\text{Si } X(t) = f_{\text{stc}}(t) / f_{\text{object}}(t)$$

entonces el valor de $X(t)$ será constante en cualquier instante t .

donde:

$f_{\text{stc}}(t)$ indica la frecuencia prevista del STC en el instante t , a saber, 27 000 000 Hz

$f_{\text{object}}(t)$ indica la frecuencia de la base de tiempo de objeto en el instante t .

La base de tiempo de objeto de los trenes ISO/CEI 14496 transportados por un tren Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se transmite como sigue:

- La base de tiempo de objeto de un tren paquetizado SL transportado en paquetes PES sin el uso del FlexMux será transmitido por OCR codificadas en el encabezamiento de paquete SL de ese tren. Véase 2.11.3.4.
- La base de tiempo de objeto de los trenes paquetizados SL transportados en paquetes PES dentro de un tren FlexMux será transmitida por FCRs en ese tren FlexMux. Véase 2.11.3.5. Por consiguiente, todos los trenes ISO/CEI 14496 contenidos dentro del mismo tren FlexMux comparten la misma base de tiempo de objeto.
- La base de tiempo de objeto de un tren paquetizado SL transportado en secciones será transmitido por otro tren ISO/CEI 14496 dentro del tren de transporte o tren de programa indicado por el campo OCR_ES_ID en el descriptor ES para ese tren.

Se aplicarán las siguientes constricciones para codificar las OCR y FCR en los trenes paquetizados SL y en los trenes FlexMux transportados por un tren Rec. UIT-T H.222.0 | ISO/CEI 13818-1:

- Las OCR y las FCR de cada tren paquetizado SL y cada tren FlexMux asociado a la misma escena tendrán la misma resolución.
- La resolución de las OCR y FCR para una escena, f_{cr} , tendrá un valor menor o igual que 90 000 Hz.
- La relación $(f_{\text{stc}}(t)/300)/f_{\text{cr}}$, será un valor mayor o igual que uno. Por consiguiente, la resolución de los elementos de sintaxis OCR y FCR sólo pueden tomar valores tales como 90 000 Hz, 45 000 Hz, 30 000 Hz, 22 500 Hz, 18 000 Hz, etc.

Dentro de las constricciones anteriores y la restricción ISO/CEI 14496-1 de que la resolución f_{cr} presentará un número entero de ciclos por segundos, f_{cr} puede seleccionarse como apropiada para la escena.

Las indicaciones de tiempo ISO/CEI 14496 codificadas en el encabezamiento de paquete SL se referirán a instantes de la base de tiempo de objeto del tren transportado en el paquete SL. La resolución de cada una de estas indicaciones de tiempo será menor en un factor 2^k que la resolución de las OCR o FCR asociadas al tren, siendo k un entero positivo mayor o igual que cero. Para obtener el mismo ajuste automático, la longitud de los campos de indicación de tiempo `TimeStampLength` será k bits menor que la longitud del campo OCR o FCR, `OCRLength` y `FCRLength`, respectivamente. Por tanto, para cada tren se aplicarán las siguientes condiciones para la codificación de las indicaciones de tiempo:

- `TimeStampResolution` = $(\text{OCRResolution o FCRResolution, respectivamente})/2^k$, siendo k un entero positivo mayor o igual que cero. ISO/CEI 14496-1 requiere que `TimeStampResolution` represente un número entero de ciclos por segundo.
- `TimeStampLength` = `OCRLength o FCRLength respectivamente` – k .

La relación entre un valor del STC y el correspondiente valor de la base de tiempo de objeto de un tren se establece asociando campos PTS en encabezamientos de paquete PES con la OCR o FCR en encabezamientos de paquete SL y paquetes de tren FlexMux respectivamente, especificados en 2.11.3.6 y 2.11.3.7.

2.11.3.4 Temporización de entrega de trenes SL paquetizados

Para transportar el contenido ISO/CEI 14496 en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, se utilizan trenes paquetizados SL ISO/CEI 14496-1. En cada tren paquetizado SL transportado en un paquete PES sin el uso de FlexMux, el campo `objectClockReference` se codificará como sigue:

- 1) En el primer encabezamiento SL de un tren paquetizado SL habrá presente un campo `objectClockReference` (OCR).

- 2) El tren paquetizado SL se construirá de modo que el intervalo de tiempo entre los bytes que contienen el último bit de los campos OCR sucesivos sea menor o igual que 0,7 segundos. De este modo:

$$|t(i'') - t(i')| \leq 0,7 \text{ s}$$

para todos los i' e i'' , donde i' e i'' son los índices de los bytes que contienen el último bit de los campos OCR consecutivos en el tren FlexMux.

Si se codifica una referencia de reloj de objeto en un encabezamiento de paquete SL, se codificará también el campo `instantBitrate`.

2.11.3.5 Temporización de entrega de trenes FlexMux

Junto con los trenes paquetizados SL se puede utilizar también la herramienta FlexMux de ISO/CEI 14496-1 para transportar el contenido ISO/CEI 14496 en los trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. La cabida útil de los paquetes FlexMux consistirá en los paquetes SL especificados en ISO/CEI 14496-1. En cada tren FlexMux transportado en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, el campo `fmxClockReference` se codificará como sigue:

- 1) En el primer paquete FlexMux de un tren FlexMux habrá presente un campo `fmxClockReference` (FCR).
- 2) El tren FlexMux se construirá de modo que el intervalo de tiempo entre los bytes que contienen el último bit de los campos FCR sucesivos sea menor o igual que 0,7 s. De este modo:

$$|t(i'') - t(i')| \leq 0,7 \text{ s}$$

para todos los i' e i'' , donde i' e i'' son los índices de los bytes que contienen el último bit de los campos FCR consecutivos en el tren FlexMux.

- 3) Todas las indicaciones de tiempo ISO/CEI 14496 contenidas en los trenes paquetizados SL transportados con un tren FlexMux se referirán a instantes de la base de tiempo de objeto transmitida por los campos FCR en el tren FlexMux. Los trenes paquetizados SL transportados en paquetes FlexMux no necesitan transportar campos OCR. Si hay presentes campos OCR, pueden ser ignorados.

2.11.3.6 Transporte de trenes paquetizados SL en paquetes PES

Un tren paquetizado SL ISO/CEI 14496-1 puede corresponder con un tren PES. Sólo un paquete SL de un tren paquetizado SL constituirá la cabida útil de un paquete PES. Los paquetes PES que transportan un tren paquetizado SL se identificarán por `stream_id = 0xFA` en el encabezamiento de paquete PES.

Cuando un campo OCR está codificado en el encabezamiento de paquete SL, la indicación de tiempo de presentación (PTS) estará codificada en el encabezamiento del paquete PES que transporta este encabezamiento de paquete SL. Esta PTS se codificará con el valor de 33 bits de la parte a 90 kHz del STC que corresponde al valor de la base de tiempo de objeto en el instante de tiempo indicado por la OCR.

El `ES_ID` asociado con el tren paquetizado SL será señalado por el descriptor SL especificado en 2.6.46.

2.11.3.7 Transporte de paquetes FlexMux en trenes PES

Los paquetes PES con una cabida útil formada por paquetes FlexMux serán identificados por `stream_id = 0xFB` en el encabezamiento de paquete PES. Un número entero de paquetes FlexMux constituirá la cabida útil de un paquete PES, es decir, la cabida útil de un paquete PES que transporta un tren FlexMux comenzará con un encabezamiento de paquete FlexMux y terminará con el último byte de un paquete FlexMux.

Si un campo `fmxClockReference` (FCR) se codifica en uno de los paquetes FlexMux contenidos en un paquete PES, la PTS se codificará en el encabezamiento del paquete PES que contiene este paquete FlexMux. Esta PTS se codificará con el valor de 33 bits de la parte de 90 kHz del STC que corresponde al valor de la base de tiempo de objeto del tren FlexMux en el instante de tiempo indicado por la FCR. Cuando múltiples paquetes FlexMux con un campo FCR codificado están contenidos en un paquete PES, la PTS corresponderá al tiempo indicado por la FCR en el primer paquete FlexMux encontrado en la cabida útil del paquete PES.

Los `ES_ID` asociados a cada tren paquetizado SL transportado en el tren FlexMux serán señalizados por un descriptor FMC como se especifica en 2.6.44.

2.11.3.8 Transporte de paquetes SL y paquetes FlexMux en secciones

Para el transporte de contenido ISO/CEI 14496 en secciones, se definen las secciones `ISO_CEI_14496`. Sólo los trenes de descriptor de objeto paquetizados SL y los trenes de descripción de escena utilizarán secciones `ISO_CEI_14496`.

ISO/CEI 13818-1:2007 (S)

Una sección ISO_CEI_14496 contendrá un paquete SL entero de un tren paquetizado SL o un número entero de paquetes FlexMux, cada uno de los cuales transporta un paquete SL del mismo tren elemental ISO/CEI 14496-1.

El cuadro 2-94 muestra la sintaxis de las secciones ISO_CEI_14496 para transportar trenes elementales ISO/CEI 14496-1, calificados por el identificador de tabla como descriptor de objeto o datos de tren de descripción de escena. Los datos del tren de descriptor de objeto consisten en una tabla de descriptores de objetos que comprende varios descriptores de objeto. La tabla de descriptores de objetos puede transmitirse en múltiples secciones ISO_CEI_14496. Los datos de descripción de escena consisten en una tabla de descripción de escena que puede comprender varias instrucciones BIFS. Esta tabla puede transmitirse en múltiples secciones ISO_CEI_14496. No es necesario recibir una tabla completa para procesar su cabida útil. Sin embargo, los datos de cabida útil de las secciones serán procesados en el orden correcto, indicado por el valor del campo section_number en los bytes de encabezamiento de sección ISO_CEI_14496.

Cuadro 2-94 – Sintaxis de sección para el transporte de tren ISO/CEI 14496

Sintaxis	N.º de bits	Mnemónico
ISO_IEC_14496_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
ISO_IEC_14496_section_length	12	uimsbf
table_id_extension	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
if (PMT_has_SL_descriptor(current_PID)) {		
SL_Packet()		
}		
else if (PMT_has_FMC_descriptor(current_PID)) {		
for (i=1; i<N; i++)		
FlexMuxPacket()		
}		
else {		
for (i=1; i<N; i++)		
reserved	8	bslbf
}		
CRC_32	32	rpchof
}		

table_id (id de tabla) – Este campo de 8 bits se pondrá a '0x04' o '0x05' en caso de una ISO_CEI_14496_section. Un valor de '0x04' indica una sección ISO_CEI_14496_scene_description que transporta un tren de descripción de escena ISO/CEI 14496-1. Un valor de '0x05' indica una sección ISO_CEI_14496_object_descriptor transporta un tren de descriptor de objeto ISO/CEI 14496-1.

section_syntax_indicator (indicador de sintaxis de sección) – Este campo de 1 bit se pondrá a '1'.

private_indicator (indicador privado) – Este campo de 1 bit no se define en esta Especificación.

ISO_CEI_14496_section_length (longitud de sección ISO_CEI 14496) – Este campo de 12 bits especificará el número de bytes restantes en la sección inmediatamente después del campo ISO_CEI_14496_section_length hasta el final de la sección ISO_CEI_14496. El valor de este campo no excederá de 4093 (0xFFD).

table_id_extension (id de extensión de tabla) – Este campo de 16 bits no se define en esta Especificación; su uso y valor son definidos por el usuario.

version_number (número de versión) – Este campo de 5 bits representará el número de versión de la tabla de descriptores de objeto o de la tabla de descripción de escena, respectivamente. El número de versión será incrementado en 1 módulo 32 con cada nueva versión de la tabla. El control de versión se deja a discreción de la aplicación.

current_next_indicator (indicador siguiente vigente) – Este campo de 1 bit se pondrá a 1.

section_number (número de sección) – Este campo de 8 bits representará el número de la sección ISO_CEI_14496. Este número de la primera sección ISO_CEI_14496 de la tabla de descriptores de objeto o de la tabla de descripción de escena tendrá un valor igual a 0x00. El valor de número de sección será incrementado en 1 con cada sección adicional en la tabla.

last_section_number (número de última sección) – Este campo de 8 bits especificará el número de la última sección de la tabla de descriptores de objeto o de la tabla de descripción de escena de la cual esta sección forma parte.

PMT_has_SL_descriptor(current_PID) [PTM tiene descriptor SL (PID vigente)] – Una pseudofunción que será verdadera si el bucle de descriptor en la tabla de correspondencia de programas para el elemento de programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que transporta esta sección ISO_CEI_14496 contiene un descriptor SL.

SL_Packet() (paquete SL) – Un paquete de capa de sincronización especificado en 10.2.2 de ISO/CEI 14496-1.

PMT_has_FMC_descriptor(current_PID) [PTM tiene descriptor FMC (PID vigente)] – Una pseudofunción que será verdadera si el bucle de descriptor en la tabla de correspondencia de programas para el elemento de programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que transporta esta sección ISO_CEI_14496 contiene un descriptor FMC.

FlexMuxPacket() (paquete FlexMux) – Un paquete FlexMux especificado en 11.2.4 de ISO/CEI 14496-1.

CRC_32 – Este campo de 32 bits contendrá el valor CRC que da una salida de cero de los registros en el decodificador definido en el anexo A después del procesamiento de toda la sección ISO_CEI_14496.

2.11.3.9 Extensiones de T-STD

2.11.3.9.1 Modelo T-STD para el contenido ISO/CEI 14496

La figura 2-5 muestra el decodificador-objetivo de sistema de transporte (T-STD) para la entrega de elementos de programa ISO/CEI 14496 encapsulados en trenes de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

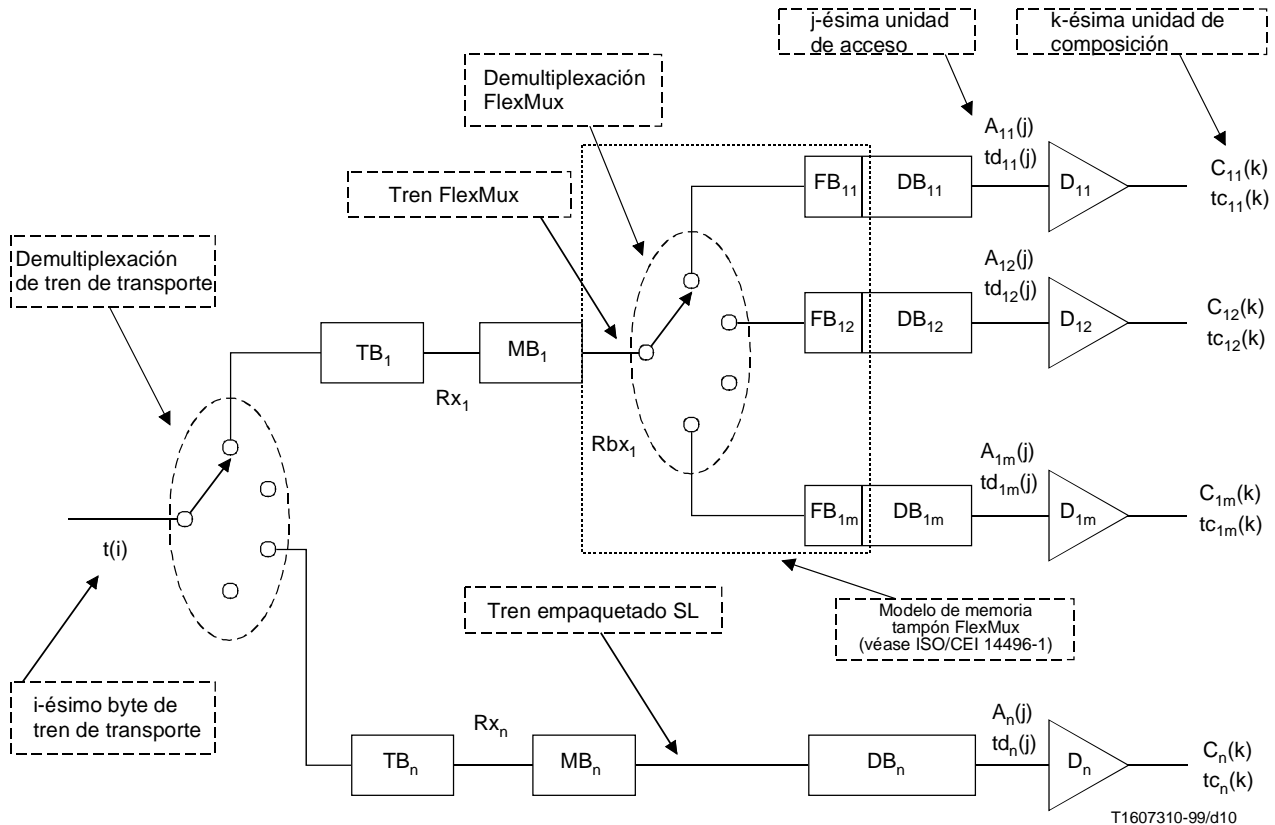


Figura 2-5 – Modelo T-STD para el contenido ISO/CEI 14496

En la figura 2-5 y su descripción se utiliza la siguiente notación:

TB_n	es la memoria tampón de transporte.
MB_n	es la memoria tampón múltiplex para el tren FlexMux o para el tren paquetizado SL n.
FB_{np}	es la memoria tampón FlexMux para el tren elemental en el canal FlexMux p del tren FlexMux n.
DB_{np}	es el decodificador para el tren elemental en el canal FlexMux p del tren FlexMux n.
DB_n	es la memoria tampón del decodificador para el tren elemental n.
D_{np}	es la memoria tampón del decodificador para el tren elemental en el canal FlexMux p del tren FlexMux n.
D_n	es el decodificador para el tren elemental n.
Rx_n	es la velocidad a la cual los datos son suprimidos de TB_n .
Rbx_n	es la velocidad a la cual los datos son suprimidos de MB_n .
$A_{np}(j)$	es la j-ésima unidad de acceso en el tren elemental en el canal FlexMux p del tren FlexMux n. $A_{np}(j)$ está indexada según el orden de decodificación.
$A_n(j)$	es la j-ésima unidad de acceso en el tren elemental n. $A_n(j)$ está indexada según el orden de decodificación.
$Td_{np}(j)$	es el tiempo de decodificación, medido en segundos, en el decodificador objetivo del sistema, de la j-ésima unidad de acceso en el tren elemental en el canal FlexMux p del tren FlexMux n.
$Td_n(j)$	es el tiempo de decodificación, medido en segundos, en el decodificador objetivo del sistema, de la j-ésima unidad de acceso en el tren elemental n.
$C_{np}(k)$	es la k-ésima unidad de composición en el tren elemental en el canal FlexMux p del tren FlexMux n. $C_{np}(k)$ resulta de decodificar $A_{np}(j)$. $C_{np}(k)$ está indexada según el orden de composición.
$C_n(k)$	es la k-ésima unidad de composición en el tren elemental n. $C_n(k)$ resulta de la decodificación de $A_n(j)$. $C_n(k)$ está indexada según el orden de composición.
$tc_{np}(k)$	es el tiempo de composición, medido en segundos, en el decodificador objetivo del sistema, de la k-ésima unidad de composición en el tren elemental en el canal FlexMux p del tren FlexMux n.
$tc_n(k)$	es el tiempo de composición, medido en segundos, en el decodificador objetivo del sistema, de la k-ésima unidad de composición en el tren elemental n.
$t(i)$	indica el tiempo en segundos en el cual el i-ésimo byte del tren de transporte entra en el decodificador objetivo del sistema.

2.11.3.9.2 Procesamiento de trenes FlexMux

Los paquetes completos del tren de transporte que contienen datos del tren FlexMux n son pasados a la memoria tampón de transporte para el tren FlexMux n, TB_n . El tamaño de TB_n se fija en 512 bytes. Todos los bytes que entran en TB_n son suprimidos de TB_n a la velocidad Rx_n , especificada por el campo TB_leak_rate en el descriptor de memoria tampón múltiplex asociado con el tren FlexMux n. Cuando no hay datos en la memoria tampón TB_n , la velocidad Rx_n es igual a cero. Los paquetes de tren de transporte duplicados no son entregados a MB_n .

En el caso de transporte en paquetes PES, el encabezamiento del paquete PES y los bytes de datos de cabida útil son entregados a la memoria tampón MB_n ; todos los otros bytes que salen de TB_n no entran en MB_n y pueden ser usados para controlar el sistema. En el caso de transporte en secciones ISO_CEI_14496, los bytes de encabezamiento de las secciones, de cabida útil y de datos CRC-32 son entregados a la memoria tampón MB_n ; todos los demás bytes no entran en MB_n y pueden ser utilizados para controlar el sistema. En ambos casos, el tamaño de MB_n es especificado por el campo MB_buffer_size en el descriptor de memoria tampón múltiplex.

Los bytes de paquetes del tren FlexMux en la memoria tampón MB_n son todos entregados a su memoria tampón asociada a la velocidad especificada por el campo $fmXRate$ codificado en el tren FlexMux y de acuerdo con el modelo de memoria tampón FlexMux definido en 11.2.9 de ISO/CEI 14496-1. Sólo los bytes de datos de cabida útil de paquetes FlexMux en el canal FlexMux p del tren FlexMux n entran en la memoria tampón FB_{np} . Los bytes de encabezamiento de paquetes FlexMux en el canal FlexMux p del tren FlexMux n se descartan y pueden utilizarse para controlar el sistema. La velocidad especificada por el campo $fmXRate$ será aplicable para todos los paquetes FlexMux en el tren que sigue inmediatamente al paquete de canal de referencia de reloj FlexMux hasta que se encuentre el siguiente paquete de canal de referencia de reloj FlexMux. Cuando no hay presentes datos del tren FlexMux en MB_n , no se suprimen datos de MB_n . Los bytes del encabezamiento de paquetes PES o del encabezamiento de sección ISO_CEI_14496 que preceden inmediatamente a un encabezamiento FlexMux son suprimidos instantáneamente y descartados y pueden ser utilizados para controlar el sistema. Los bytes de los campos CRC-32 de la sección ISO_CEI_14496 que siguen inmediatamente al último paquete del tren FlexMux en la cabida útil de la sección son suprimidos instantáneamente y descartados y pueden ser utilizados para verificar la integridad de los datos. Los bytes

del canal de referencia de reloj FlexMux son suprimidos instantáneamente y descartados y pueden ser utilizados para enganchar la base de tiempo de objeto ISO/CEI 14496 con el STC. Cuando no hay un paquete PES ni bytes de datos de cabida útil de sección, respectivamente presentes en MB_n , no se suprimen datos de MB_n . Todos los datos que entran en MB_n salen de esta memoria. Todos los bytes de cabida útil de paquetes PES del tren n entran instantáneamente en el demultiplexor FlexMux al salir de MB_n .

2.11.3.9.3 Definición de la memoria tampón FlexMux, FB_{np}

Para cada canal p de un tren FlexMux n , el tamaño de la memoria tampón FB_{np} se define utilizando el descriptor de tamaño de memoria tampón Fmx . Los bytes de cabida útil de paquetes FlexMux se transfieren de la memoria tampón FB_{np} a la memoria tampón del decodificador DB_{np} de acuerdo con el modelo de memoria tampón FlexMux definido en 11.2.9 de ISO/CEI 14496-1. Sólo los bytes de cabida útil de paquetes SL en el canal FlexMux p del tren FlexMux n entran en la memoria tampón DB_{np} . Los bytes de encabezamiento de paquetes SL en el canal FlexMux p del tren FlexMux n se descartan y pueden utilizarse para controlar el sistema.

2.11.3.9.4 Procesamiento de trenes SL paquetizados

Los paquetes completos del tren de transporte que contienen datos del tren paquetizado SL n son pasados a la memoria tampón de transporte para el tren paquetizado SL, TB_n . Todos los bytes que entran en TB_n son suprimidos a la velocidad R_{x_n} , especificada por el campo TB_leak_rate en el descriptor de memoria tampón múltiplex. Cuando no hay datos en la memoria tampón TB_n , la velocidad R_{x_n} es igual a cero. Los paquetes de tren de transporte duplicados no son entregados a MB_n .

En el caso de transporte en paquetes PES, los bytes de encabezamiento y de datos de cabida útil de paquete PES son entregados a la memoria tampón MB_n ; todos los bytes que salen de TB_n no entran en MB_n , y pueden ser utilizados para controlar el sistema. En el caso de transporte en secciones ISO_CEI_14496, los bytes de encabezamiento, de cabida útil y de datos CRC-32 son entregados a la memoria tampón MB_n ; todos los otros bytes no entran en MB_n y pueden ser utilizados para controlar el sistema. En ambos casos, el tamaño de MB_n es especificado por el campo MB_buffer_size en el descriptor de memoria tampón múltiplex.

Los bytes de tren paquetizados SL en la memoria MB_n son entregados todos a la memoria tampón de decodificador DB_n a la velocidad especificada por el campo $instantBitRate$ codificado en el tren paquetizado SL y de acuerdo con el modelo de decodificador del sistema definido en 7.4 de ISO/CEI 14496-1. La velocidad especificada por el campo será aplicable a todos los bytes de datos en el tren paquetizado SL hasta que se encuentre el siguiente campo $instantBitRate$. Si no hay bytes de trenes paquetizados SL en MB_n , no se suprimen bytes de MB_n . Los bytes del encabezamiento de paquete PES o del encabezamiento de sección ISO_CEI_14496 que preceden inmediatamente al encabezamiento de paquete SL son suprimidos instantáneamente y descartados y pueden ser utilizados para controlar el sistema. Los bytes de los campos CRC-32 de la sección ISO_CEI_14496 que siguen inmediatamente al byte de cabida útil del último paquete SL son suprimidos instantáneamente y descartados y pueden ser utilizados para verificar la integridad de los datos. Cuando no hay bytes de datos de cabida útil de paquetes PES o de sección, respectivamente presentes en MB_n , no se suprimen datos de MB_n . Todos los datos que entran en MB_n salen de esta memoria. Todos los bytes de cabida útil de paquetes PES del tren n entran instantáneamente en la memoria tampón DB_n al dejar MB_n , con la excepción de los encabezamientos de paquetes SL. Los bytes de los encabezamientos de paquetes SL no entran en DB_n y pueden ser utilizados para controlar el sistema. El tamaño de la memoria tampón de decodificador DB_n viene dado por $bufferSizeDb$ del descriptor de configuración del decodificador definido en ISO/CEI 14496-1.

2.11.3.9.5 Gestión de memorias tampón

Los trenes de transporte se construirán de modo que se satisfagan las condiciones definidas en esta subcláusula.

TB_n no desbordará y estará vacía por lo menos una vez cada segundo. No se producirá desbordamiento de MB_n . No se producirá desbordamiento de FB_{np} . No se producirá subutilización ni desbordamiento de DB_{np} ni de DB_n . Se produce subutilización de DB_{np} cuando uno o más bytes de una unidad de acceso no están presentes en DB_{np} en el tiempo de decodificación asociado con esta unidad de acceso. Se produce subutilización de DB_n cuando uno o más bytes de una unidad de acceso no están presentes en DB_n en el tiempo de decodificación asociado con esta unidad de acceso.

2.11.3.10 Transporte dentro de un tren de transporte

2.11.3.10.1 Visión general

Un tren de transporte puede contener uno o más programas, cada uno descrito por una tabla de correspondencia de programas. Se puede transportar el contenido ISO/CEI 14496, además de los tipos de trenes ya definidos para este programa. Los elementos del contenido ISO/CEI 14496 pueden ser transportados en uno o más elementos de programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1 referenciados por un valor de PID único dentro de un tren de transporte. Como un caso especial, es posible que un programa dentro de un tren de transporte consista solamente en elementos de programa ISO/CEI 14496. El contenido ISO/CEI 14496 asociado a un programa y transportado en el tren de transporte

será referenciado en la tabla de correspondencia de programas de ese programa. Se utilizará un descriptor de objeto inicial para definir una escena ISO/CEI 14496-1; el uso de este descriptor se especifica en 2.11.3.10.2.

El transporte del contenido ISO/CEI 14496 en un PID es señalizado por un valor de tipo de tren de 0x12 ó 0x13 en la tabla de correspondencia de programas en asociación con ese valor PID. Un valor de 0x12 indica transporte en paquetes PES. El campo stream_id en el encabezamiento de paquete PES señala si los paquetes PES contienen un solo paquete SL o varios paquetes FlexMux. Un valor de tipo de tren de 0x13 en la tabla de correspondencia de programas indica que el elemento de programa transporta un tren de descriptor de objeto o un tren de instrucción BIFS contenido en secciones. En este caso, el identificador de tabla en el encabezamiento de sección indica si se transporta un tren de descriptor de objeto en las secciones o un tren de instrucción BIFS. Véase también el cuadro 2-95. La sección contiene un paquete SL o varios paquetes FlexMux, según lo indicado por la presencia de un descriptor SL o un descriptor FMC respectivamente, en el bucle de descriptor de la tabla de correspondencia de programas para el elemento de programa Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que transporta las secciones. Cuando se transporta el contenido ISO/CEI 14496, el descriptor SL y el descriptor FMC especificarán el ES_ID para cada tren ISO/CEI 14496 encapsulado. Cuando cambia la asignación de valores ES_ID, la tabla de correspondencia de programas será actualizada y se incrementará el número de versión de PTM en 1 módulo 32. En el anexo R figura un ejemplo de procedimiento de acceso de contenido para los componentes de programa ISO/CEI 14496.

Cuadro 2-95 – Opciones definidas por ISO/CEI para el transporte de una escena ISO/CEI 14496 y trenes asociados por la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

Trenes de descriptor de objeto de ISO/CEI 14496-1	Encapsulado en paquetes SL	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Transporte en secciones ISO_CEI_14496	Stream_type = 0x13	Table_id = 0x05
	Encapsulado en paquetes SL seguidos por multiplex en paquetes FlexMux	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Transporte en secciones ISO_CEI_14496	Stream_type = 0x13	Table_id = 0x05
Trenes de descripción de escenas de ISO/CEI 14496-1	Encapsulado en paquetes SL	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1010'
		Transporte en secciones ISO_IEC_14496	Stream_type = 0x13	Table_id = 0x04
	Encapsulado en paquetes SL seguidos por multiplex en paquetes FlexMux	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1011'
		Transporte en secciones ISO_CEI_14496	Stream_type = 0x13	Table_id = 0x04
Todos los demás trenes de ISO/CEI 14496	Encapsulado en paquetes SL	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1010'
	Encapsulado en paquetes SL seguidos por multiplex en paquetes FlexMux	Transporte en paquetes PES	Stream_type = 0x12	Stream_id = '1111 1011'

2.11.3.10.2 Descriptor de objeto inicial

En el caso de transporte de una escena ISO/CEI 14496-1, el descriptor de objeto inicial de ISO/CEI 14496-1 sirve como el punto de acceso inicial a todos los trenes asociados. El descriptor de objeto inicial será transportado en el descriptor IOD situado en el bucle de descriptor que sigue inmediatamente al campo program_info_length en la tabla de correspondencia de programas a la cual está asociada esta escena. Contiene el descriptor ES que identifica trenes de descripción de escena y de descriptor de objeto que forman parte del programa. Puede contener también descriptores ES que identifican uno o más trenes IPMP u OCI asociados. La identificación de los trenes se hace por medio de los ES_ID especificados en la cláusula 8 de ISO/CEI 14496-1.

2.11.3.11 Modelo P-STD para el contenido ISO/CEI 14496

La figura 2-6 muestra el modelo STD cuando se transportan sistemas de datos ISO/CEI 14496 en un tren de programa.

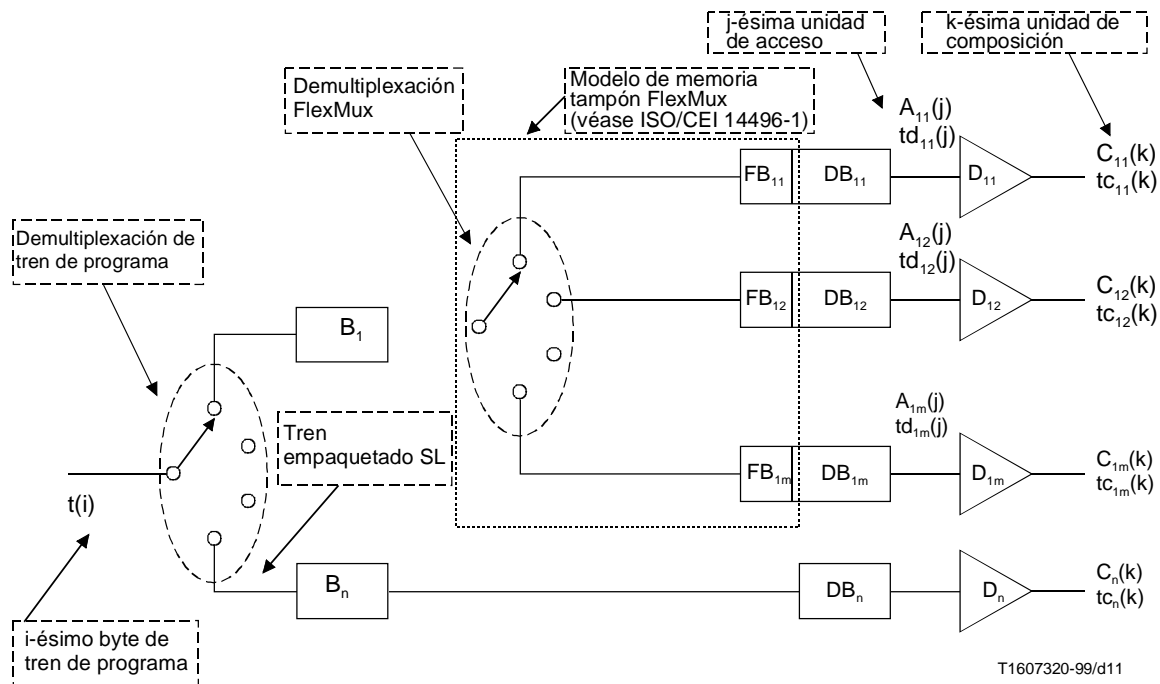


Figura 2-6 – Modelo P-STD para el tren de sistemas ISO/CEI 14496

En la figura 2-6 y su descripción se utiliza la siguiente notación:

- B_n es la memoria tampón de entrada para el tren FlexMux n o para el tren paquetizado SL n .
- FB_{np} es la memoria tampón FlexMux para el tren elemental en el canal FlexMux p del tren FlexMux n .
- DB_{np} es la memoria tampón del decodificador para el tren elemental en el canal FlexMux p del tren FlexMux n .
- DB_n es la memoria tampón del decodificador para el tren elemental n .
- D_{np} es el decodificador para el tren elemental en el canal FlexMux p del tren FlexMux n .
- D_n es el decodificador para el tren elemental n .
- $A_{np}(j)$ es la j -ésima unidad de acceso en el tren elemental en el canal FlexMux p del tren FlexMux n . $A_{np}(j)$ está indexada según el orden de decodificación.
- $A_n(j)$ es la j -ésima unidad de acceso en el tren elemental n . $A_n(j)$ está indexada según el orden de decodificación.
- $Td_{np}(j)$ es el tiempo de decodificación, medido en segundos, en el decodificador objetivo del sistema, de la j -ésima unidad de acceso en el tren elemental en el canal FlexMux p del tren FlexMux n .
- $Td_n(j)$ es el tiempo de decodificación, medido en segundos, en el decodificador objetivo del sistema, de la j -ésima unidad de acceso en el tren elemental n .
- $C_{np}(k)$ es la k -ésima unidad de composición en el tren elemental en el canal FlexMux p del tren FlexMux n . $C_{np}(k)$ resulta de decodificar $A_{np}(j)$. $C_{np}(k)$ está indexada según el orden de composición.
- $C_n(k)$ es la k -ésima unidad de composición en el tren elemental n . $C_n(k)$ resulta de la decodificación de $A_n(j)$. $C_n(k)$ está indexada según el orden de composición.
- $tc_{np}(k)$ es el tiempo de composición, medido en segundos, en el decodificador objetivo del sistema, de la k -ésima unidad de composición en el tren elemental en el canal FlexMux p del tren FlexMux n .
- $tc_n(k)$ es el tiempo de composición, medido en segundos, en el decodificador objetivo del sistema, de la k -ésima unidad de composición en el tren elemental n .
- $t(i)$ indica el tiempo en segundos en el cual el i -ésimo byte del tren de transporte entra en el decodificador objetivo del sistema.

2.11.3.11.1 Procesamiento de trenes FlexMux

A la entrada de STD, cada byte en la cabida útil de paquetes PES que transportan un tren paquetizado SL n o un tren FlexMux n es transferido instantáneamente a la memoria tampón B_n . El i -ésimo byte entra en B_n en el instante $t(i)$. Los bytes de encabezamiento de paquete PES no entran en la memoria tampón B_n y pueden utilizarse para controlar el sistema. El tamaño de B_n es especificado por el campo P-STD_buffer_size en el encabezamiento del paquete PES que transporta el tren n .

Los bytes de paquetes de tren FlexMux en la memoria tampón B_n son entregados todos a su memoria tampón FlexMux asociada a la velocidad especificada por el campo fmxRate codificado en el tren FlexMux y de acuerdo con el modelo de memoria tampón FlexMux definido en 11.2.9 de ISO/CEI 14496-1. Sólo los bytes de datos de cabida útil de paquetes FlexMux en el canal FlexMux p del tren FlexMux n entran en la memoria tampón FB_{np} . Los bytes de encabezamiento de paquetes FlexMux en el canal FlexMux p del tren FlexMux n se descartan y pueden utilizarse para controlar el sistema. La velocidad especificada por el campo fmxRate será aplicable para todos los paquetes FlexMux en el tren hasta que se encuentre el siguiente paquete de canal de referencia de reloj FlexMux. Los bytes del canal de referencia de reloj FlexMux son suprimidos instantáneamente y descartados y pueden ser utilizados para enganchar la base de tiempo de objeto ISO/CEI 14496 con el STC. Cuando no hay datos de cabida útil de paquetes PES presentes en B_n , no se suprimen datos de B_n . Todos los datos que entran en B_n salen de esta memoria. Todos los bytes de cabida útil de paquetes PES del tren n entran instantáneamente en el demultiplexor FlexMux al salir de B_n .

2.11.3.11.2 Definición de la memoria tampón FlexMux, FB_{np}

Para cada canal p de un tren FlexMux n , el tamaño de la memoria tampón FlexMux FB_{np} se define utilizando el descriptor FmxBufferSize si hay presente una correspondencia de trenes de programa en el tren de programa. Los bytes de cabida útil de paquetes FlexMux se transfieren de la memoria tampón FB_{np} a la memoria tampón del decodificador DB_{np} de acuerdo con el modelo de memoria tampón FlexMux definido en 11.2.9 de ISO/CEI 14496-1. Sólo los bytes de cabida útil de paquetes SL en el canal FlexMux p del tren FlexMux n entran en la memoria tampón DB_{np} . Los bytes de encabezamiento de paquetes SL en el canal FlexMux p del tren FlexMux n se descartan y pueden utilizarse para controlar el sistema.

2.11.3.11.3 Procesamiento de trenes paquetizados SL

A la entrada del STD cada byte en la cabida útil de los paquetes PES que transportan un tren paquetizado SL n es transferido instantáneamente a la memoria tampón B_n . El i -ésimo byte entra en B_n en el instante $t(i)$. Los bytes de encabezamiento de paquete PES no entran en la memoria tampón B_n y pueden utilizarse para controlar el sistema. El tamaño de B_n es especificado por el campo de tamaño de memoria tampón P-STD en el encabezamiento del paquete PES que transporta el tren n . Los bytes de trenes paquetizados SL en la memoria tampón B_n son entregados a la memoria tampón del decodificador DB_n a la velocidad especificada por el campo instantBitRate codificado en el tren paquetizado SL y de acuerdo con el modelo de codificador del sistema definido en 7.4 de ISO/CEI 14496-1. La velocidad especificada por el campo instantBitRate será aplicable a todos los bytes de datos en el tren paquetizado SL hasta que se encuentre el siguiente campo instantBitRate. Cuando no hay datos de cabida útil de paquetes PES presentes en B_n , no se suprimen datos de B_n . Todos los datos que entran en B_n salen de esta memoria. Todos los bytes del tren n entran instantáneamente en la memoria tampón DB_n al salir de B_n , con la excepción de los encabezamientos de paquetes SL. Los bytes de los encabezamientos de paquetes SL no entran en DB_n y pueden ser utilizados para controlar el sistema. El tamaño de la memoria tampón de decodificador DB_n viene dado por bufferSizeDb del descriptor de configuración de decodificador definido en ISO/CEI 14496-1.

2.11.3.11.4 Gestión de memorias tampón

Los trenes de programa se construirán de modo que no se produzca desbordamiento de B_n . No se producirá desbordamiento de FB_{np} . No se producirá subutilización ni desbordamiento de DB_{np} ni de DB_n . Se produce subutilización de DB_{np} cuando uno o más bytes de una unidad de acceso no están presentes en DB_{np} en el tiempo de decodificación asociado con esta unidad de acceso. Se produce subutilización de DB_n cuando uno o más bytes de una unidad de acceso no están presentes en DB_n en el tiempo de decodificación asociado con esta unidad de acceso.

2.11.3.12 Transporte dentro de un tren de programa

2.11.3.12.1 Visión general

Un tren de programa sólo contiene un programa. Se puede transportar datos ISO/CEI 14496, además de los tipos de trenes ya definidos para este programa. Como un caso especial, es posible también que un tren de programa sólo transporte datos ISO/CEI 14496. Si hay presente una correspondencia de trenes de programa, se hará referencia como sigue al contenido ISO/CEI 14496 transportado en un tren de programa. El transporte de escenas ISO/CEI 14496-1 y los trenes ISO/CEI 14496 asociados en paquetes SL y FlexMux se indica mediante el identificador de tren apropiado y el descriptor de objeto inicial; el uso de este descriptor se especifica en 2.11.3.12.2. Para cada tren ISO/CEI 14496 transportado, el descriptor SL y el descriptor FMC especificarán el ES_ID. Cuando cambia la asignación de los

valores ES_ID, la correspondencia de tren de programa, si está presente, será actualizada y la versión de correspondencia de tren de programa será incrementada en 1 módulo 32. Obsérvese que en una correspondencia de programas, el contenido ISO/CEI 14496 puede también ser referenciado por medios privados.

Para un ejemplo de un procedimiento de acceso de contenido para los componentes de programa ISO/CEI 14496 dentro de un tren de programa, véase el anexo R.

2.11.3.12.2 Descriptor de objeto inicial

En el caso de transporte de una escena, el descriptor de objeto inicial de ISO/CEI 14496-1 sirve como el punto de acceso inicial a todos los trenes asociados. Si hay presente una correspondencia de trenes de programa, el descriptor de objeto inicial será transportado en el descriptor IOD situado en el bucle de descriptor que sigue inmediatamente al campo `program_stream_info_length`. Contiene descriptores ES que identifican trenes de descripción de escena y de descriptor de objeto que forman parte del programa. Puede contener también descriptores ES que identifican uno o más trenes IPMP u OCI asociados. La identificación de los trenes se hace por medio de los ES_ID especificados en la cláusula 8 de ISO/CEI 14496-1. En un tren de programa, el descriptor de objeto inicial puede también ser transportado por medios privados.

2.12 Transporte de metadatos

2.12.1 Introducción

Un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 puede transportar metadatos. El formato de los metadatos puede venir definido por la ISO o cualquier otra autoridad. En este punto se definen el transporte de los metadatos y los mecanismos de transporte, así como la señalización relativa a los metadatos, el modelo de temporización de los metadatos aplicado y las extensiones del modelo STD para la decodificación de los metadatos.

Un servicio de metadatos se define como un conjunto coherente de metadatos con el mismo formato que se entrega a un receptor con un fin específico. Los servicios de metadatos están contenidos en trenes de metadatos, cada uno de los cuales transporta uno o más servicios de metadatos. La presente especificación supone que se utiliza el concepto de unidades de acceso de metadatos dentro de un servicio de metadatos. La definición de unidad de acceso de metadatos es específica a cada formato de metadatos, pero se supone que cada servicio de metadatos representa una concatenación (o serie) de unidades de acceso de metadatos.

Cuando se transporta un servicio de metadatos por un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, se asigna un identificador de servicio de metadatos individual a cada uno de esos servicios. Un identificador de servicio de metadatos hace referencia de manera unívoca a un servicio de metadatos de entre todos los servicios de metadatos disponibles en el mismo tren de transporte o tren de programa, y no *únicamente* dentro del tren de metadatos. El identificador de servicio de metadatos se utiliza para extraer el servicio de metadatos y toda la información necesaria para decodificarlo.

Para decodificar los metadatos puede ser necesario disponer de datos de configuración del decodificador. Si la decodificación de un servicio de metadatos transportado en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 requiere datos de configuración del decodificador, estos datos de configuración del decodificador de los metadatos deberán ser transportados dentro del mismo programa del mismo tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

En la subcláusula 2.12.2 se aborda la temporización de los metadatos, mientras que en 2.12.3 se describen someramente las herramientas definidas para el transporte de metadatos en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. La utilización de las herramientas de transporte disponibles se especifica en 2.12.4 a 2.12.8, y en 2.12.9 se expone la señalización relacionada con los metadatos. Por último, en 2.12.10 se especifica el modelo STD para la decodificación de metadatos.

Puesto que pueden transportarse muchos tipos de metadatos, es esencial señalar tanto el formato y la codificación precisos de los metadatos, como el significado semántico de los metadatos. El formato y la codificación van indicados en "formato de metadatos", mientras que el significado semántico se indica en "formato de aplicación de metadatos". En otras palabras, el formato de metadatos indica cómo deben decodificarse los metadatos, mientras que el formato de aplicación de metadatos indica la manera de utilizar los metadatos y principalmente qué aplicación los utiliza. Esta diferenciación es importante, puesto que distingue la codificación o representación de los metadatos y su significado, permitiendo así que la aplicación sea independiente del medio de transporte de los metadatos.

2.12.2 Modelo temporal de los metadatos

Los metadatos pueden consistir en códigos de tiempo asociados al contenido, por ejemplo, para indicar el inicio de un segmento del contenido. Las indicaciones temporales incluidas en los metadatos corresponden a una determinada línea temporal del contenido que es propia del formato de metadatos y/o del formato de aplicación de metadatos considerados. Por ejemplo, podría ser que un formato (de aplicación) de metadatos utilice el UTC, mientras que otro formato de aplicación de metadatos utilizará los códigos temporales de la SMPTE. Para permitir el transporte del

contenido en cualquier momento y por cualquier medio, se supone que la línea temporal del contenido definida en los metadatos es independiente del transporte, aunque esto no es obligatorio.

Para transportar contenidos y los metadatos asociados por trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, es necesario mantener referencias temporales exactas entre los metadatos y los contenidos. Esto también es necesario si los metadatos se transportan por otros medios. Para ello, en la presente especificación se supone que se utiliza el modelo de línea temporal de la figura 2-7.

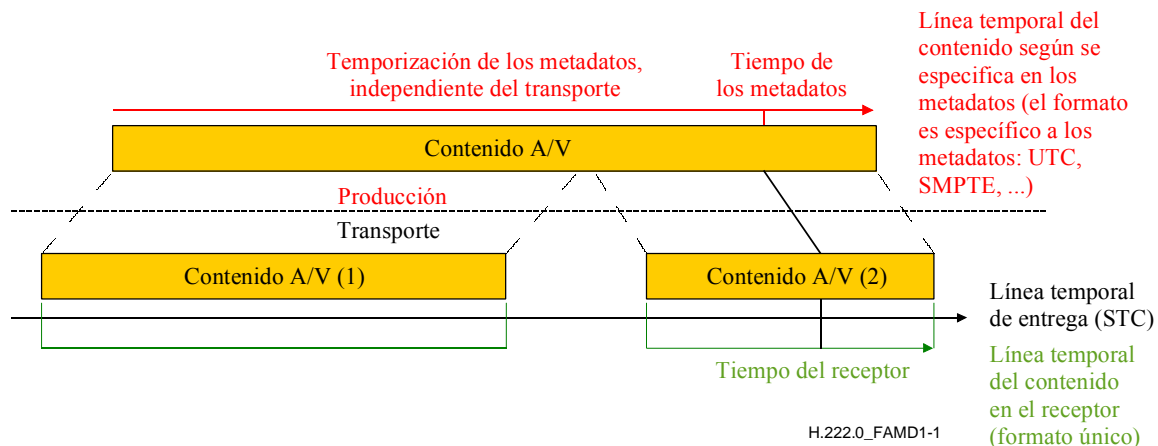


Figura 2-7 – Modelo de temporización para la entrega de contenidos y metadatos

Los metadatos se asocian al contenido audiovisual, por regla general independientemente del transporte, en la producción o en cualquier otra etapa anterior al transporte. Siempre que es necesario, se incorpora información temporal en los metadatos para indicar, por ejemplo, segmentos específicos dentro del contenido: se utiliza una línea temporal del contenido definida en los metadatos. Por ejemplo, se pueden utilizar los códigos temporales UTC o SMPTE. El formato de la línea temporal es independiente de cualquier código temporal que pueda o no estar incorporado directamente en el tren audiovisual. Por ejemplo, la línea temporal definida en los metadatos puede utilizar los códigos UTC, mientras que en el tren de vídeo se incorporan códigos temporales de la SMPTE.

Deben cumplirse los siguientes requisitos en cada uno de los trenes de metadatos:

- no debe haber discontinuidades temporales en la línea temporal del contenido definida en los metadatos;
- la línea temporal del contenido definida en los metadatos estará ajustada al reloj de muestreo del contenido;
- cada una de las referencias temporales del tren de metadatos debe referirse a la misma línea temporal del contenido definida en los metadatos.

En el momento del transporte, se asocia al contenido una temporización específica para el transporte, que se denomina línea temporal de entrega. Si se transporta en trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la línea temporal de entrega viene dada por el reloj de tiempo del sistema (STC). El contenido puede entregarse como una porción contigua de información, pero también es posible interrumpir la entrega del contenido, por ejemplo, en caso de interrupciones de un programa por noticias. Se admite pues que haya discontinuidades de la línea temporal en éste y en otros casos.

Cuando en los metadatos se utilizan referencias temporales, el decodificador de sistemas objetivo (STD) debe asociar sin ambigüedad estas referencias con los valores temporales del contenido recibido. Para ello es necesario definir una línea temporal del contenido en el receptor. El STC puede utilizarse como línea temporal del contenido en el receptor, pero no garantiza necesariamente una asociación temporal sin ambigüedades porque puede sufrir discontinuidades. Por consiguiente, también puede utilizarse como línea temporal del contenido en el receptor el NPT (tiempo de lectura normal) del DSM-CC ISO/CEI 13818-6. En cualquier modo de reproducción, es decir normal, inversa, cámara lenta, rápido hacia adelante, rápido hacia atrás e imagen fija, el NPT garantiza una asociación temporal sin ambigüedades, independiente de las discontinuidades del STC y de las inserciones de otro contenido. Obsérvese que es necesario transmitir un nuevo NPT_reference_descriptor a cada nuevo ciclo del STC.

Para mantener las referencias temporales exactas entre los metadatos y el contenido, es necesario conocer la relación entre el tiempo en los metadatos (MT), definido en la línea temporal del contenido en los metadatos, y el correspondiente tiempo en el receptor (RT) de la línea temporal del contenido en el receptor. Esto se consigue estableciendo la diferencia temporal (en unidades de 90 kHz) entre la línea temporal del contenido en los metadatos y la línea temporal del contenido en el receptor. El descriptor de etiquetado del contenido especifica esta diferencia, que

determina el valor de la base de tiempo de los metadatos en el momento en que la base de tiempo del contenido en el receptor alcanza un valor especificado. Véase la figura 2-7.

La temporización en los sistemas de metadatos puede referirse a imágenes específicas o a tramas de audio, por ejemplo, utilizando los códigos temporales del SMPTE. La diferencia entre la línea temporal del contenido en los metadatos y la línea temporal del contenido en el receptor se expresa en unidades de 90 kHz y, por consiguiente, la referencia temporal de los metadatos se traducirá en valores de 90 kHz en los receptores. Para evitar cualquier inexactitud, los receptores utilizarán el valor más próximo para toda referencia a una imagen o trama de audio. Por ejemplo, las referencias temporales de los metadatos traducidas a unidades de 90 kHz se harán corresponder a la imagen o la trama cuyo valor PTS sea más cercano al valor traducido.

Si se utiliza el NPT, durante la reproducción en cualquier modo y en cualquier momento, siempre habrá la misma diferencia entre la base de tiempo de los metadatos y la base de tiempo del NPT. Siempre y cuando no haya discontinuidades del STC ni inserciones de otro contenido, la diferencia entre la base de tiempo de los metadatos y la base de tiempo del STC también será constante, pero únicamente en el modo de reproducción normal. También es necesario que la diferencia sea constante en el caso de líneas temporales de dominio privado, pero posiblemente haya restricciones que no se definen en esta especificación.

Si los metadatos se transportan de forma sincrónica en paquetes PES o mediante el protocolo de telecarga sincronizado DSM-CC, se asignan indicaciones de tiempo de presentación (PTS) a los metadatos. Estos PTS podrán, por ejemplo, indicar el momento en que los metadatos son válidos. Esto implica un conocimiento *a priori* de la manera en que se asocian los metadatos a la temporización de la entrega. Ahora bien, en los metadatos transportados sincrónicamente también puede haber referencias temporales que hay que hacer corresponder entre la línea temporal del contenido en los metadatos y la línea temporal del contenido en el receptor, teniendo en cuenta la diferencia especificada entre ambas líneas temporales. Véase asimismo la figura 2-8.

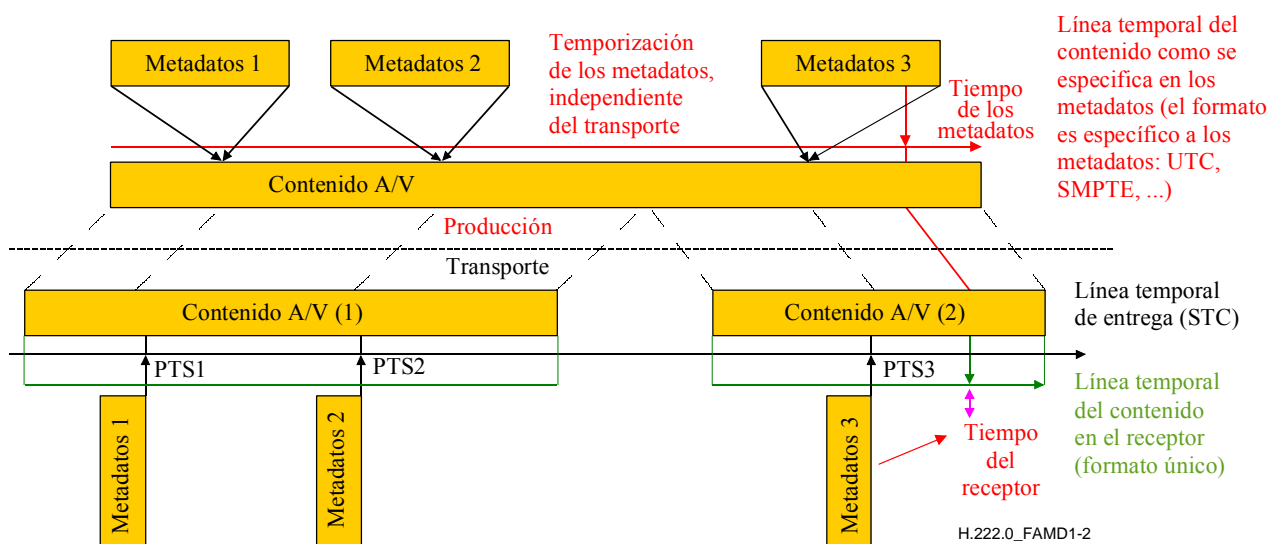


Figura 2-8 – Entrega de los metadatos en paquetes PES

2.12.3 Opciones para el transporte de metadatos

Para tener en cuenta las muy diversas características de los metadatos, se definen una serie de herramientas para el transporte de metadatos en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

En esta Especificación se definen dos herramientas para la entrega sincrónica de los metadatos:

- transporte en paquetes PES;
- utilización del protocolo de telecarga sincronizado DSM-CC.

Además, se definen tres herramientas para la entrega asincrónica de los metadatos:

- transporte en secciones de metadatos;
- utilización de carruseles de datos DSM-CC;
- utilización de carruseles de objetos DSM-CC.

Cabe notar que algunas de las opciones de transporte asíncrono soportan carruseles y estructuras de archivo. La elección de la herramienta de transporte depende de los requisitos que se aplican a la entrega de los metadatos y de los requisitos de las herramientas, según se describe en los siguientes párrafos.

ISO/CEI 13818-1:2007 (S)

Los metadatos pueden asimismo transportarse por medios privados como los paquetes PES con un valor de identificador de tren 0xBD o 0xBF (`private_stream_id_1` o `private_stream_id_2`) o en secciones de dominio privado. Esta especificación no detalla cómo se utilizan los medios privados para el transporte de metadatos, pero prevé la señalización de dichos metadatos utilizando los descriptores que se definen en 2.6.56 a 2.6.63.

La referenciación básica de los servicios de metadatos es la misma para todas las herramientas, es decir, se hace utilizando un identificador de servicio de metadatos. No obstante, hay diferencias entre las distintas herramientas. Cuando se utilizan paquetes PES, secciones de metadatos o secciones de telecarga sincronizadas DSM-CC, los datos de cada servicio de metadatos se señalan explícitamente dentro del tren de metadatos mediante el campo `metadata_service_id`. En cambio, cuando se utilizan carruseles DSM-CC, la forma de señalización se deja a discreción de las aplicaciones de metadatos. Cabe notar que en la presente especificación se prevé el transporte de un servicio de metadatos en un carrusel DSM-CC, pero no se limita el número de servicios de metadatos que pueden transportarse en un solo carrusel DSM-CC.

Los datos de configuración del decodificador en los metadatos se señalan explícitamente cuando esta información se incluye en el descriptor de metadatos en paquetes PES con un `stream_type` 0x15 y un `stream_id` 0xFC, en secciones de metadatos o en secciones de telecargas sincronizadas DSM-CC. Cuando los datos de configuración del decodificador en los metadatos se transportan en un carrusel DSM-CC, es necesario señalar estos datos, pero la presente especificación no define la forma de hacerlo. Dicha señalización se deja a discreción de las aplicaciones.

2.12.4 Utilización de paquetes PES para transportar metadatos

Los paquetes PES son un mecanismo de transporte sincrónico de metadatos. Gracias al PTS en la cabecera de los paquetes PES, las unidades de acceso de metadatos se asocian a un determinado instante del STC, sin necesidad de que haya referencias temporales en los metadatos. Esto implica un conocimiento *a priori* de la manera en que se asocian los metadatos a la temporización de entrega. Se asignan valores `stream_id` y `stream_type` específicos para marcar los paquetes PES que transportan metadatos. Véase 2.12.9.

Si se utilizan paquetes PES con `stream_type` 0x15 y `stream_id` 0xFC para transportar los metadatos, se utilizará un contenedor de unidad de acceso de metadatos como herramienta para alinear los paquetes PES y las unidades de acceso de metadatos, utilizando `metadata_AU_cells`. Esto permite una indicación de acceso aleatorio cuyo significado depende del formato de los metadatos, y un contador de secuencia de células para detectar la pérdida de `metadata_AU_cells`. Cada unidad de acceso de metadatos se transporta y, si procede, se fragmenta en una o más `metadata_AU_cells`. En cada paquete PES que transporta metadatos, el primer `PES_packet_data_byte` será el primer byte de una `metadata_AU_cell`. La indicación de tiempo PTS en la cabecera del paquete PES vale para todas las unidades de acceso de metadatos transportados en el mismo paquete PES. La indicación de tiempo PTS señala el momento en que las unidades de acceso de metadatos se decodifican instantáneamente y se eliminan de la memoria tampón B_n en el STD. Obsérvese que la relación entre una unidad de acceso de metadatos decodificada y el contenido audiovisual queda fuera del ámbito de la presente especificación.

Un paquete PES puede contener una única `metadata_AU_cell`. Esto resulta útil si la unidad de acceso de metadatos no cabe en un único paquete PES, en cuyo caso la fragmentación de la unidad de acceso de metadatos la realiza la `metadata_AU_cell`.

Cuando los metadatos se transportan en paquetes PES en un tren de programa al que se aplica una correspondencia de trenes de programa, esta correspondencia del tren de programa especificará qué paquetes PES contienen los metadatos asociados.

2.12.4.1 Contenedor de unidad de acceso de metadatos

Se utilizará un contenedor de unidad de acceso de metadatos al transportar unidades de acceso de metadatos en paquetes PES con un `stream_type` 0x15 y un `stream_id` 0xFC, o en secciones de telecarga sincronizadas DSM-CC de `stream_type` 0x19. El contenedor define una estructura consistente en una serie concatenada de `Metadata_AU_cells`. Codificando el tamaño de los metadatos contenidos en cada `metadata_AU_cell`, los receptores pueden hacer un análisis sintáctico independiente de los metadatos. El analizador sintáctico puede extraer los metadatos y presentarlos a un decodificador de metadatos sin conocimiento *a priori* de los detalles de los metadatos. Las `Metadata_AU_cell` deben alinearse con el transporte, es decir, el primer byte de la carga útil del paquete PES o de la sección de telecarga sincronizada DSM-CC deberá ser el primer byte de la `Metadata_AU_cell`.

Si una unidad de acceso de metadatos no cabe enteramente en una `metadata_AU_cell`, la unidad de acceso de metadatos deberá fragmentarse en múltiples `metadata_AU_cells`; el campo `fragmentation_indication` de cada una de dichas `metadata_AU_cell` indicará cuál es la `metadata_AU_cell` que contiene un fragmento.

La indicación de tiempo PTS en la cabecera o de la sección de telecarga sincronizada vale para todas las `Metadata_AU_cell` transportadas en el mismo paquete PES o en la misma sección de telecarga sincronizada, respectivamente.

**Cuadro 2-96 – Contenedor de unidad de acceso de metadatos
(Metadata Access Unit Wrapper)**

Sintaxis	N.º de bits	Mnemónico
<pre>Metadata_AU_wrapper () { for (i=0; i<N;i++){ Metadata_AU_cell () } }</pre>		

**Cuadro 2-97 – Célula de unidad de acceso de metadatos
(Metadata AU cell)**

Sintaxis	N.º de bits	Mnemónico
<pre>Metadata_AU_cell () { metadata_service_id sequence_number cell_fragment_indication decoder_config_flag random_access_indicator reserved AU_cell_data_length for (i=0; I<AU_cell_data_length;i++){ AU_cell_data_byte } }</pre>	<p>8</p> <p>8</p> <p>2</p> <p>1</p> <p>1</p> <p>4</p> <p>16</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p>

metadata_service_id (identificador de servicio de metadatos) – Es un campo de 8 bits que identifica el servicio de metadatos asociado con la unidad de acceso de metadatos transportada en esta célula de unidad de acceso de metadatos.

sequence_number (número de secuencia) – Este campo de 8 bits especifica el número de secuencia de la metadata_AU_cell. Este número se incrementa en uno por cada metadata_AU_cell sucesiva que constituye el metadata_AU_wrapper, independientemente del valor codificado del metadata_service_id.

cell_fragment_indication (indicación de fragmento de célula) – Este campo de 2 bits contiene información sobre la unidad de acceso de metadatos transportada en esta metadata_AU_cell, de acuerdo con el cuadro 2-98.

**Cuadro 2-98 – Indicación de fragmentación de la célula
(Cell fragment indication)**

Valor	Descripción
11	Una única célula transporta la unidad de acceso de metadatos completa.
10	La primera de una serie de células que transportan datos de una unidad de acceso de metadatos.
01	La última de una serie de células que transportan datos de una unidad de acceso de metadatos.
00	Una célula de una serie de células que transportan datos de una unidad de acceso de metadatos, que no es ni la primera ni la última.

random_access_indicator (indicador de acceso aleatorio) – Este campo de 1 bit, cuando está codificado con el valor 1, indica que los metadatos transportados en esta metadata_AU_cell representan un punto de entrada al servicio de metadatos que se puede decodificar sin disponer de información de las metadata_AU_cell anteriores. El significado del punto de acceso aleatorio queda definido por el formato de los metadatos

decoder_config_flag (bandera de configuración del decodificador) – Este campo de 1 bit señala la presencia de información de configuración del decodificador en la unidad de acceso de metadatos transportada. Cabe notar que ello no impide la presencia de metadatos en la unidad de acceso próxima a los datos de configuración del decodificador.

AU_cell_data_length (longitud de los datos de la célula unidad de acceso) – Este campo de 16 bits especifica el número de AU_cell_data_bytes inmediatamente posteriores.

AU_cell_data_byte (byte de datos de la célula unidad de acceso) – Este campo de 8 bits contiene bytes contiguos de una unidad de acceso de metadatos.

2.12.5 Utilización del protocolo de telecarga sincronizado DSM-CC para transportar metadatos

Para un transporte sincronizado, además de utilizar los paquetes PES, puede utilizarse el protocolo de telecarga sincronizado DSM-CC. Si se utilizan secciones de telecarga sincronizadas DSM-CC para transportar los metadatos, se utilizará el contenedor de unidad de acceso de metadatos que se define en 2.12.4.1 para englobar las unidades de acceso de metadatos. Esto permite una indicación de acceso aleatorio cuyo significado depende del formato de los metadatos, y un contador de secuencia de células para detectar la pérdida de metadata_AU_cells. En cada sección de telecarga sincronizada DSM-CC que transporta metadatos, el primer byte de la carga útil deberá ser el primer byte de una Metadata_AU_cell. La indicación de tiempo PTS en la cabecera vale para todas las unidades de acceso de metadatos transportadas en la misma sección de telecarga sincronizada DSM-CC. El PTS señala el momento en que las unidades de acceso de metadatos se decodifican instantáneamente y se eliminan de la memoria tampón B_n en el STD. Obsérvese que la relación entre las unidades de acceso de metadatos decodificadas y el contenido audiovisual queda fuera del ámbito de la presente especificación. Se asigna un valor stream_type específico (como se detalla en el cuadro 2-34) para señalar el transporte de metadatos en secciones de telecarga sincronizadas DSM-CC.

2.12.6 Utilización de secciones de metadatos para transportar metadatos

El método de secciones de metadatos se puede utilizar para transportar de forma asíncrona las unidades de metadatos sin recurrir a un mecanismo de entrega carrusel. La sintaxis y la semántica de las secciones de metadatos se definen en esta subcláusula. Cada sección de metadatos podrá llevar una unidad de acceso de metadatos completa o una parte de una unidad de acceso de metadatos, según se defina en el campo section_fragment_indication.

Para su transporte en secciones de metadatos, las unidades de acceso de metadatos se estructuran en una o más tablas de metadatos. Cada tabla de metadatos contiene una o más unidades de acceso de metadatos completas de uno o más servicios de metadatos. Teóricamente, el mecanismo de transporte de las tablas de metadatos es comparable al mecanismo de transporte de las tablas de correspondencia de programas y las tablas de asociación de programas. Cada tabla de metadatos puede estar formada por múltiples secciones de metadatos y contener metadatos de varios servicios.

Se asignan valores stream_type y table_id específicos a las secciones de metadatos. Asimismo, los datos de configuración del decodificador de metadatos pueden transportarse en las secciones, señaladas por un valor de descripción de metadatos, asignado por el descriptor de configuración del decodificador de metadatos.

Cuadro 2-99 – Sintaxis de la sección para el transporte de metadatos

Sintaxis	N.º de bits	Mnemónico
Metadata_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
random_access_indicator	1	bslbf
decoder_config_flag	1	bslbf
metadata_section_length	12	uimsbf
metadata_service_id	8	uimsbf
reserved	8	bslbf
section_fragment_indication	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i=1; i<N; i++){		
metadata_byte	8	bslbf
}		bslbf
CRC_32	32	rpchof
}		

table_id (identificador de tabla) – El table_id es un campo de 8 bits que deberá fijarse a "0x06" para cada sección de metadatos.

section_syntax_indicator (indicador de sintaxis de sección) – Este campo de 1 bit debe tener el valor 1.

private_indicator (indicador privado) – Este campo de 1 bit no queda abarcado por la presente especificación.

random_access_indicator (indicador de acceso aleatorio) – Este campo de 1 bit, cuando está codificado con el valor 1, indica que los metadatos transportados en esta sección de metadatos representan un punto de acceso al servicio de metadatos cuya decodificación es posible sin disponer de información de las secciones de metadatos anteriores. El significado del punto de acceso aleatorio queda definido por el formato de los metadatos.

decoder_config_flag (bandera de configuración del decodificador) – Este campo de un 1 bit, cuando va codificado con el valor 1, indica que la información de configuración del decodificador está incluida en la unidad de acceso de metadatos transportada en esta sección de metadatos.

metadata_section_length (longitud de la sección de metadatos) – Este campo de 12 bits deberá especificar el número de bytes restantes en la sección, inmediatamente posteriores al campo metadata_section_length, incluida la verificación por redundancia cíclica (CRC). El valor de este campo no deberá exceder de 4093 (0xFFD).

metadata_service_id (identificador del servicio de metadatos) – Este campo de 8 bit identifica el servicio de metadatos asociado a la unidad de acceso de metadatos transportada en esta sección de metadatos. Cada tabla de metadatos puede contener metadatos procedentes de múltiples servicios de metadatos.

section_fragment_indication (indicación de fragmentación de la sección) – Este campo de 2 bits contiene información sobre la fragmentación de la unidad de acceso de metadatos transportada en esta sección de metadatos, de acuerdo con el cuadro 2-100.

**Cuadro 2-100 – Indicación de fragmentación de la sección
(section fragment indication)**

Valor	Descripción
11	Una única sección de metadatos que transporta una unidad de acceso de metadatos completa.
10	La primera de una serie de secciones de metadatos que transportan datos de una unidad de acceso de metadatos.
01	La última de una serie de secciones de metadatos que transportan datos de una unidad de acceso de metadatos.
00	Una sección de metadatos de una serie de secciones de metadatos que transportan datos de una unidad de acceso de metadatos, que no es ni la primera ni la última.

version_number (número de versión) – Este campo de 5 bits es el número de versión de toda la tabla de metadatos. El número de versión se incrementará en 1 módulo 32 cada vez que cambie la información contenida en la tabla de metadatos. Cuando el current_next_indicator se pone a 1, el version_number será el número de la tabla de metadatos actualmente aplicable, y cuando se pone a 0, el version_number será el número de la siguiente tabla de metadatos aplicable.

current_next_indicator (indicador actual siguiente) – Este campo de 1 bit indica que la tabla de metadatos enviada es aplicable actualmente (si está puesto a 1), o que la tabla de metadatos enviada no es aún aplicable y será la siguiente tabla válida (si está puesto a 0).

section_number (número de sección) – Este campo de 8 bits da el número de la sección de metadatos. El section_number de la primera sección de la tabla de metadatos será 0x00. El section_number se incrementará en 1 para cada sección adicional en la tabla de metadatos.

last_section_number (número de la última sección) – Este campo de 8 bit especifica el número de la última sección (la sección que tiene el número más alto) de la tabla de metadatos completa de la que es parte esta sección.

metadata_byte (byte de metadatos) – Campo de 8 bits que contiene uno de los bytes contiguos de una unidad de acceso de metadatos.

CRC_32 – Campo de 32 bits que contiene el valor CRC que da un resultado cero de los registros en el decodificador definido en el anexo A después de procesar toda la sección de metadatos.

2.12.7 Utilización en un carrusel de datos DSM-CC para transportar metadatos

Las herramientas DSM-CC definidas en ISO/CEI 13818-6 para los carruseles de datos se pueden utilizar si es necesario recurrir a un mecanismo de entrega de carrusel, sin necesidad de expresar la organización jerárquica de la estructura de metadatos en el mecanismo de transporte. La información sobre el carrusel que contiene los metadatos se incluyen en el descriptor de metadatos definido en 2.6.60 y 2.6.62. Se asigna un valor `stream_type` específico para señalar el transporte de metadatos en un carrusel de datos DSM-CC. Obsérvese que la señalización de los servicios de metadatos dentro de un carrusel de datos DSM-CC es necesaria, pero no queda definida por la presente especificación.

2.12.8 Utilización de un carrusel de objetos DSM-CC para transportar metadatos

Si es necesario recurrir a un mecanismo de entrega carrusel con capacidad para expresar la organización jerárquica de la estructura de metadatos en el transporte, pueden utilizarse las herramientas y estructuras de ficheros DSM-CC definidas en ISO/CEI 13818-6 para los carruseles de objetos de usuario a usuario. Estas estructuras de archivo ofrecen herramientas para estructurar los metadatos de la forma más apropiada para un análisis sintáctico eficiente y para expresar la organización jerárquica de los metadatos. La información necesaria para identificar el carrusel que contiene los metadatos se incluye en el descriptor de metadatos que se define en 2.6.60 y 2.6.61. Puede ser el IOP:IOR() definido en 11.3.1 y 5.7.2.3 para el DSM-CC de ISO/CEI 13818-6. Se asigna un valor `stream_type` específico para señalar el transporte de metadatos en un carrusel de objetos DSM-CC. Obsérvese que la señalización del servicio de metadatos dentro de un carrusel de objetos DSM-CC es necesaria, pero no queda definida por la presente especificación.

2.12.9 Señalización relacionada con los metadatos

La señalización relacionada con los metadatos abarca cuatro esferas distintas:

- señalización de servicios y trenes de metadatos;
- señalización de contenido para su utilización por un sistema de metadatos;
- asociación de los metadatos al contenido; y
- señalización de los datos de configuración del decodificador.

2.12.9.1 Señalización de los servicios y trenes de metadatos

El transporte de metadatos se identifica por un valor `stream_type` comprendido entre 0x15 y 0x19, inclusive, que especifica cuál de los cinco métodos descritos en 2.12.4 a 2.12.8 se utiliza para el transporte de metadatos y, si procede, por un valor `stream_id` de 0xFC que indica un tren de metadatos.

Para identificar inequívocamente un servicio de metadatos se asigna un valor `metadata_service_id` a cada uno de estos servicios en el transporte. El valor asignado debe ser único dentro del tren de transporte o el tren de programa que transporta el servicio de metadatos. Si los metadatos se transportan en paquetes PES con un `stream_id` 0xFC, o en secciones de metadatos, o en secciones de telecarga sincronizadas ISO/CEI 13818-6, el valor `metadata_service_id` asignado se señala explícitamente en la cabecera de la célula `metadata_AU_cell` o en la sección de metadatos. Si se utiliza un carrusel ISO/CEI 13818-6 para transportar los metadatos, la señalización de los servicios de metadatos se deja a discreción de la aplicación. El descriptor de metadatos especifica el formato de los metadatos y ofrece información sobre los datos de configuración del decodificador; transporta información sobre el servicio de metadatos al que está asociado, y de esta forma se establece su vinculación con un servicio.

2.12.9.2 Señalización del contenido para su utilización por un sistema de metadatos

En las subcláusulas 2.6.56 y 2.6.57 se define un descriptor de etiquetado del contenido que puede utilizarse para asignar una referencia específica al formato de aplicación de metadatos (`content_reference_id_record`) a un contenido audiovisual o de cualquier otro tipo transportado en un tren de transporte o un tren de programa MPEG-2. El `content_reference_id_record` puede ser utilizado por el sistema de metadatos como etiqueta para referirse a dicho contenido. El contenido puede representar, por ejemplo, un programa o un tren, o segmentos de cualquiera de ellos. El descriptor de etiquetado del contenido también indica la base de tiempo del contenido, que se utiliza para la referenciación temporal de los metadatos, incluida la diferencia constante entre la base de tiempo de los metadatos y la base de tiempo del contenido. El descriptor permite transportar datos privados. El campo `metadata_application_format` puede imponer restricciones para `content_reference_record`, por ejemplo limitar el periodo de tiempo durante el cual es válido.

2.12.9.3 Asociación de los metadatos al contenido

En 2.6.58 y 2.6.59 se define el descriptor de puntero de metadatos, que asocia un único servicio de metadatos a contenidos audiovisuales o de otro tipo en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Los metadatos se asocian al contenido dentro del contexto definido por la ubicación del descriptor. En un tren de transporte, el descriptor se puede colocar en la PMT, en el bucle de descriptor de un tren de programa o un tren elemental, pero también se

puede colocar en tablas no definidas en la presente especificación, como las tablas que describen los paquetes de programas de servicios de radiodifusión.

El descriptor de puntero de metadatos señala, desde el contexto del contenido, el servicio de metadatos asociado al contenido. El descriptor incluye el valor del `metadata_service_id` asignado al servicio de metadatos asociado, así como uno o más lugares donde se encuentran los metadatos asociados. Por ejemplo, los metadatos se pueden colocar en el mismo tren de transporte que el contenido, o en otro tren de transporte, pero también en medios distintos a los trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, como Internet.

2.12.9.4 Señalización de los datos de configuración del decodificador

Para decodificar los metadatos puede ser necesario disponer de datos de configuración del decodificador de metadatos. De ser así, los datos de configuración del decodificador se incluirán en uno de los servicios de metadatos en el mismo programa, en el mismo tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que el servicio de metadatos. Si se necesitan datos de configuración del decodificador para decodificar un servicio de metadatos, el descriptor de metadatos puede transportar dichos datos pero también puede obtenerlos por consulta de los datos de configuración del decodificador del mismo servicio de metadatos o de otro distinto. En un tren de transporte, estos otros servicios pueden encontrarse buscando en la PMT un `metadata_descriptor` con el `metadata_service_id` que se especifica en el campo `decoder_config_metadata_service_id` (y con el mismo `metadata_format` y el mismo `metadata_application` format).

2.12.9.5 Resumen de la señalización de metadatos

En la figura 2-9 se ofrece un ejemplo de señalización de metadatos en que un único programa transporta el contenido o esencia, ("programa de contenido"), y los metadatos se transportan en un programa distinto ("programa de metadatos"). En este ejemplo, el programa de metadatos y el programa de contenido están en el mismo tren de transporte.

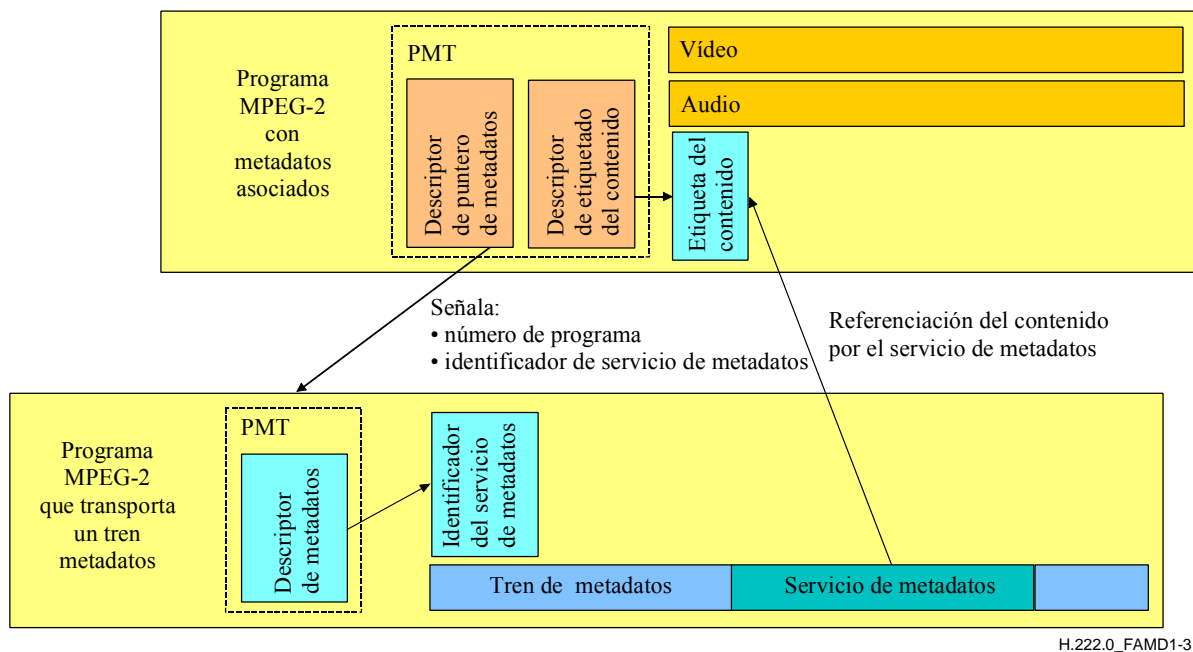


Figura 2-9 – Referenciación y señalización de los metadatos

En el programa de contenido hay dos descriptores relacionados con los metadatos: el descriptor `content_labeling` y el descriptor `metadata_pointer`. El descriptor `content_labeling` asocia con el contenido una etiqueta (en el diagrama la "etiqueta del contenido") codificada en el descriptor en el campo `content_reference_id`. La etiqueta puede ser utilizada por el servicio de metadatos para referirse a la esencia, en su totalidad o en parte, o con un segmento definido en el tiempo. Por ejemplo, el descriptor `content_labeling` puede crear una etiqueta "Noticias del 1/1/02", y los metadatos pueden referirse a un contenido específico dentro del conjunto "Noticias del 1/1/02", por ejemplo, indicando el instante específico de ese segmento.

El descriptor de puntero de metadatos indica dónde se encuentra el servicio de metadatos para el contenido especificado. En este ejemplo, los metadatos se transportan en un programa distinto, pero sería igualmente válido transportar los metadatos en el mismo programa que el contenido, o por medios que quedan fuera del ámbito de la presente especificación, por ejemplo, un URL. Este descriptor también da el valor del identificador de servicio de metadatos asignado al servicio de metadatos. Esto es necesario dado que el tren de metadatos puede transportar

múltiples servicios de metadatos para muchos programas distintos y es necesario que cada uno de ellos pueda identificar unívocamente su servicio de metadatos.

En el programa de metadatos, el descriptor de metadatos señala a qué servicio de metadatos dentro del tren de metadatos se aplica. En caso de utilizarse, el descriptor de metadatos indica dónde puede encontrarse la información de configuración del decodificador.

Cuando el receptor que decodifica el programa de contenido identifica un descriptor de puntero de metadatos en la PMT, el receptor extrae el descriptor de metadatos del programa de metadatos. En caso de ser necesario, en primer lugar se extraen los datos de configuración y se configura en consecuencia el decodificador, tras lo cual se puede empezar a decodificar el servicio de metadatos.

2.12.10 Modelo STD para los metadatos

El modelo de decodificador de sistemas adjetivo (STD) especifica las restricciones normativas de los trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que transportan metadatos. Para decodificar metadatos en el STD se aplican los modelos T-STD y P-STD normales con memoria tampón B_n , velocidad de entrada R_{x_n} de los metadatos en B_n y velocidad de salida R_{metadata} del B_n al D_{metadata} (decodificador de metadatos). Véase la figura 2-10.

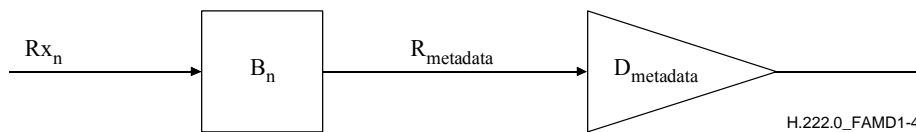


Figura 2-10 – Decodificación de metadatos en el STD

Los metadatos entran en la memoria tampón B_n a la velocidad R_{x_n} . En el modelo P-STD, la velocidad R_{x_n} es equivalente a la velocidad del tren de programa. En el modelo T-STD, la velocidad R_{x_n} es la velocidad de salida del TB_n y es equivalente a la velocidad definida por el campo `metadata_input_leak_rate` en el descriptor STD de metadatos. El tamaño BS_n de la memoria tampón B_n es equivalente al tamaño definido en el campo `metadata_buffer_size` en el descriptor STD de metadatos. En caso de entrega sincrónica, la decodificación de metadatos es instantánea y está controlada por las indicaciones de tiempo PTS. En el momento de la decodificación, es decir, cuando el reloj de sistema (STC) coincide con la indicación de tiempo PTS, los metadatos asociados se eliminan instantáneamente de la memoria B_n . En caso de entrega asíncrona, los metadatos se eliminan de la memoria B_n a una velocidad R_{metadata} equivalente a la velocidad definida en el campo `metadata_output_leak_rate` en el descriptor STD de metadatos. La memoria tampón B_n no debe desbordarse.

Obsérvese que el modelo STD define restricciones para la entrega de los metadatos, pero no especifica restricción alguna para la temporización utilizada en los metadatos.

2.13 Transporte de datos ISO 15938

2.13.1 Introducción

El transporte de metadatos en trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, como se describe en 2.12, permite transportar datos ISO 15938 codificando adecuadamente el campo `metadata_format`. En este punto, se define una instancia específica para el transporte de los datos ISO 15938. El transporte de datos ISO 15938 deberá cumplir todos los requisitos definidos en 2.12, además de los requisitos definidos en esta subcláusula.

2.13.2 Datos de configuración del decodificador ISO 15938

Para decodificar los datos ISO 15938 es necesario disponer de datos de configuración del decodificador. Por consiguiente, si un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 transporta datos ISO 15938, el descriptor de metadatos deberá señalar que se transportan los datos de configuración del decodificador asociados en el mismo tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, codificando el valor del `decoder_config_flags` en 001, 010, 011 ó 100.

2.14 Transporte de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10

2.14.1 Introducción

En esta especificación se define el transporte de tren elemental de la Rec. UIT-T H.264 | ISO/CEI 14496-10 en sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, para trenes de programa y de transporte. Normalmente, un tren de la Rec. UIT-T H.264 | ISO/CEI 14496-10 será un elemento de un programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, según se define mediante el PMT en un tren de transporte y el PSM en un tren de programa. La gestión del transporte y

de las memorias intermedias de trenes de vídeo AVC se define utilizando los parámetros existentes de esta Recomendación | Norma Internacional, como PTS y DTS, así como la información contenida en el tren de vídeo AVC.

El transporte de trenes de vídeo AVC en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 define la correspondencia precisa entre los parámetros STD y HRD que pueden aparecer en un tren de vídeo AVC. Se definen requisitos de la presencia de parámetros HRD en el tren de vídeo AVC para garantizar que se pueda verificar si se cumple cada requisito STD para cada tren de vídeo AVC transportado en un tren de transporte o un tren de programa.

NOTA 1 – Aunque en la información de temporización contenida en el tren de vídeo AVC quizá no utilice un reloj de 90 kHz, las indicaciones de tiempo PTS y DTS se tienen que expresar en unidades de 90 kHz.

Cuando se transporta un tren de la Rec. UIT-T H.264 | ISO/CEI 14496-10 en un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, los datos codificados de la primera estarán contenidos en paquetes PES y cumplirán con el formato de tren de bytes definido en el anexo B a la UIT-T H.264 | ISO/CEI 14496-10, con las siguientes restricciones:

- Cada unidad de acceso AVC contendrá una unidad NAL delimitadora de unidad de acceso.

NOTA 2 – En la Rec. UIT-T H.264 | ISO/CEI 14496-10 se exige que una unidad NAL delimitadora de unidad de acceso, si la hubiere, sea la primera unidad NAL dentro de una unidad de acceso AVC. Estas unidades de acceso simplifican la detección de las fronteras entre imágenes; gracias a estas unidades no es necesario procesar el contenido de las cabeceras de sector y son particularmente útiles en los perfiles básicos y extendidos donde el orden de los sectores puede ser arbitrario.
- Cada unidad NAL del tren de bytes que transporta un delimitador de unidad de acceso contendrá un solo elemento sintáctico `zero_byte`.

NOTA 3 – La sintaxis y semántica de las unidades NAL del tren de bytes se definen en el anexo B a la Rec. UIT-T H.264 | ISO/CEI 14496-10.
- Todos los conjuntos de parámetros secuencia e imagen (SPS y PPS) necesarios para decodificar el tren de vídeo AVC estarán contenidos en ese tren de vídeo AVC.

NOTA 4 – La Rec. UIT-T H.264 | ISO/CEI 14496-10 también permite la entrega de parámetros SPS y PPS por mecanismos externos. En esta Especificación no se proporciona soporte para ese tipo de entrega, y por lo tanto es necesario que los parámetros SPS y PPS se transporten dentro del tren de vídeo AVC.
- Cada secuencia de vídeo AVC que contiene `hrd_parameters()` con `low_delay_hrd_flag` puesto a '1', transportará parámetros VUI en los cuales `timing_info_present_flag` estará puesto a '1'.

NOTA 5 – Si `low_delay_hrd_flag` está puesto a '1', en el modelo STD estará permitida la infrautilización de la capacidad de la memoria intermedia; véanse las cláusulas 2.14.3 y 2.14.4. Al poner a '1' `timing_info_present_flag` se garantiza que el tren de vídeo AVC contiene información suficiente para determinar el tiempo de salida DPB y el tiempo de supresión CPB de unidades de acceso AVC, en caso de que se produzca una infrautilización de la capacidad.

Para proporcionar información específica de visualización, por ejemplo la relación de aspecto (`aspect_ratio`), se recomienda firmemente que cada tren de vídeo AVC transporte parámetros VUI con información suficiente para garantizar que el tren de vídeo AVC decodificado pueda visualizarse correctamente en los receptores.

2.14.2 Transporte en paquetes PES

El vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10 se transporta en paquetes PES como `PES_packet_data_bytes`, utilizando uno de los 16 valores de identificador de tren (`stream_id`) asignados al vídeo, y señalando el tren de vídeo de la misma Recomendación mediante el valor del tipo de tren (`stream-type`) asignado en el PMT o el PSM (véase el cuadro 2-34). El nivel más alto que puede producirse en un tren de vídeo AVC, así como el perfil con el que el tren es completamente conforme, deben indicarse utilizando el descriptor de vídeo AVC. Si este descriptor está asociado a un tren de vídeo AVC, se transportará en el bucle del descriptor para la entrada del tren elemental respectivo en el cuadro de correspondencia de programa en caso de un tren de transporte o en el mapa de tren de programa, si lo hubiere, en caso del tren de programa. En esta Recomendación | Norma Internacional no se especifica la presentación de trenes de la Rec. UIT-T H.264 | ISO/CEI 14496-10 en el contexto de un programa.

En la paquetización PES no se aplican restricciones específicas de alineamiento de datos. Para la sincronización y gestión de STD, se codifican las PTS y, si procede, las DTS, en la cabecera del paquete PES que transporta los datos del tren elemental de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10. Para la codificación de las PTS y DTS se aplican las restricciones y la semántica definidos en 2.4.3.7 y 2.7.

2.14.3 Extensiones STD

2.14.3.1 Extensiones T-STD

El modelo T-STD incluye una memoria intermedia de transporte TB_n y una memoria intermedia de multiplexación MB_n antes de la memoria intermedia EB_n para decodificar cada tren elemental n de vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10. Véase la figura 2-11.

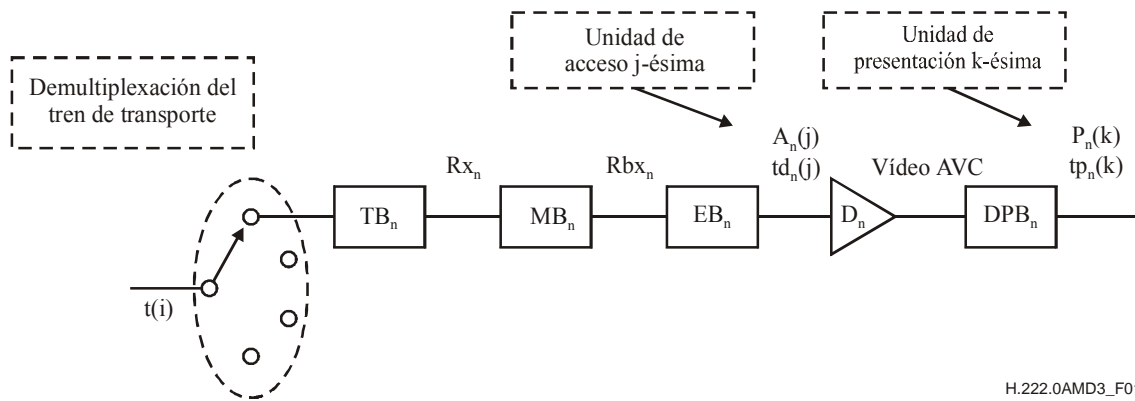


Figura 2-11 – Extensiones del modelo T-STD para vídeo de la Rec. UIT-T H.264 | ISO/CEI 14496-10

H.222.0AMD3_F01

Gestión de memoria intermedia DPB_n

El transporte de un tren de vídeo AVC conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 no tiene efecto alguno sobre el tamaño de la memoria intermedia DPB_n. Para decodificar un tren de vídeo AVC en el STD el tamaño de la DPB_n se define en la Rec. UIT-T H.264 | ISO/CEI 14496-10. La memoria intermedia DPB debe gestionarse como se indica en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10 (cláusulas C.2 y C.4). Una unidad de acceso AVC decodificada que entra instantáneamente en la DPB_n después de la decodificación de unidad de acceso AVC, es decir, en el instante de tiempo de supresión de la CPB de la unidad de acceso AVC. La unidad de acceso AVC decodificada se presenta en el instante de salida de la DPB. Si el tren de vídeo AVC no proporciona información suficiente para determinar el tiempo de supresión de la CPB y el instante de salida de la DPB de las unidades de acceso AVC, estos instantes de tiempo se determinarán en el modelo STD a partir de las indicaciones de tiempo PTS y DTS del modo siguiente:

- 1) El tiempo de supresión de la CPB de la unidad de acceso n de AVC es el instante de tiempo indicado por DTS(n), siendo DTS(n) el valor DTS en la unidad de acceso n de AVC.
- 2) El tiempo de salida de la DPB de la unidad de acceso n de AVC es el instante de tiempo indicado por PTS(n), siendo PTS(n) el valor de PTS de la unidad de acceso n de AVC.

NOTA 1 – Las secuencias vídeo AVC en las cuales low_delay_hrd_flag en hrd parámetros() está puesto a 1 transportan información suficiente para determinar el tiempo de salida de la DPB y el tiempo de supresión de la CPB de cada unidad de acceso AVC. Por consiguiente para unidades de acceso AVC en las que puede producirse una infrutilización de la capacidad STD, el tiempo de supresión de la CPB y el tiempo de salida de la DPB se definen mediante los parámetros HRD, y no mediante las indicaciones de tiempo DTS y PTS.

Gestión de las memorias intermedias TB_n, MB_n y EB_n

La entrada en la memoria TB_n y su tamaño TBS_n se especifican en 2.4.2.3. Para las memorias MB_n y EB_n, y para la velocidad Rx_n entre TB_n y MB_n y la velocidad Rbx_n entre MB_n y EB_n se aplican las siguientes restricciones para el transporte de trenes de la Rec. UIT-T H.264 | ISO/CEI 14496-10:

Tamaño EBS_n de la memoria EB_n:

$$EBS_n = cpb_size$$

Siendo cpb_size el tamaño CpbSize[cpb_cnt_minus1] de la CPB del formato del tren de byte indicado en los hrd_parameters() de NAL transportados en parámetros VUI en el tren de vídeo AVC. Si el tren de vídeo AVC no contiene hrd_parameters() de NAL, el cpb_size tendrá el tamaño definido por 1200 × MaxCPB en el anexo A a la Rec. UIT-T H.264 | ISO/CEI 14496-10 para el nivel del tren de vídeo AVC.

Tamaño MBS_n de la memoria MB_n:

$$MBS_n = BS_{mux} + BS_{oh} + 1200 \times MaxCPB[level] - cpb_size$$

Siendo BS_{oh}, el volumen almacenado en la memoria intermedia de la tara del paquete, definido como:

$$BS_{oh} = (1/750) \text{ segundos} \times \text{máx} \{ 1200 \times MaxBR[level], 2\,000\,000 \text{ bit/segundo} \}$$

y BS_{mux} , el volumen almacenado en la memoria intermedia del múltiplex adicional, definido mediante:

$$BS_{\text{mux}} = 0,004 \text{ segundos} \times \text{máx} \{ 1200 \times \text{MaxBR}[\text{level}], 2\,000\,000 \text{ bit/segundo} \}$$

Siendo $\text{MaxCPB}[\text{level}]$ y $\text{MaxBR}[\text{level}]$ los definidos para el formato del tren de bytes en el cuadro A.1 (Límites de nivel) en la Rec. UIT-T H.264 | ISO/CEI 14496-10 para el nivel del tren de vídeo AVC; y

cpb_size es el tamaño $\text{CpbSize}[\text{cpb_cnt_minus1}]$ de la CPB del formato del tren de bytes indicado en los $\text{hrd_parameters}()$ de NAL transportados en parámetros VUI en el tren de vídeo AVC. Si el tren de vídeo AVC no contiene $\text{hrd_parameters}()$ de NAL, el cpb_size será igual a $1200 \times \text{MaxCPB}$ definido en el anexo A a la Rec. UIT-T H.264 | ISO/CEI 14496-10 para el nivel del tren de vídeo AVC.

Velocidad R_{x_n} :

Cuando TB_n no contiene datos, R_{x_n} es igual a cero.

En caso contrario, $R_{x_n} = \text{bit_rate}$

Siendo bit_rate la velocidad binaria $\text{BitRate}[\text{cpb_cnt_minus1}]$ del flujo de datos en la CPB para el formato de tren de bytes indicado en los $\text{hrd_parameters}()$ de NAL transportados en los parámetros VUI en el tren de vídeo AVC. Si el tren de vídeo AVC no contiene $\text{hrd_parameters}()$ de NAL el bit_rate será igual a la velocidad binaria $1200 \times \text{MaxBR}[\text{level}]$ definido en el anexo A a la Rec. UIT-T H.264 | ISO/CEI 14496-10 para el nivel del tren de vídeo AVC.

Transferencia entre MB_n y EB_n

Si hubiera un $\text{AVC_timing_and_HRD_descriptor}$ con $\text{hrd_management_valid_flag}$ puesto a '1', la transferencia de datos de MB_n a EB_n será conforme con el plan definido de HRD para la llegada de datos a la CPB como se define en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10.

De lo contrario, se utilizará el método de fuga para la transferencia de datos de la MB_n a EB_n del modo siguiente:

Velocidad R_{bx_n} :

$$R_{bx_n} = 1200 \times \text{MaxBR}[\text{level}]$$

Siendo $\text{MaxBR}[\text{level}]$ el formato del tren de bytes definido en el cuadro A.1 (Límites de nivel) en la Rec. UIT-T H.264 | ISO/CEI 14496-10 para cada nivel.

Si MB_n contiene datos de cabida útil del paquete PES, y la memoria EB_n no está llena, la cabida útil del paquete PES se transfiere de MB_n a EB_n a una velocidad igual a R_{bx_n} . Si EB_n está llena, no se extraen los datos de MB_n . Cuando un byte de datos se transfiere de la MB_n a la EB_n , los bytes de cabecera del paquete PES que están en la MB_n y que preceden a ese byte se extraen instantáneamente y se descartan. Si la MB_n no contiene datos de cabida útil de paquete PES no se extraen los datos de la MB_n . Todos los datos que entran a la MB_n salen luego. Todos los bytes de datos de cabida útil de paquete EB_n entran instantáneamente a la EB_n después de salir de la MB_n .

Extracción de unidades de acceso AVC contenidos en la EB_n

Cada unidad de acceso AVC $A_n(j)$ que esté contenida en la EB_n se extrae inmediatamente en un instante $td_n(j)$. El tiempo de decodificación $td_n(j)$ se especifica mediante la DTS o mediante el tiempo de extracción de la CPB, según se obtiene a partir de la información contenida en el tren de vídeo AVS.

Retardo STD

El retardo total de datos de la Rec. UIT-T H.264 | ISO/CEI 14496-10 distintos de los datos de imágenes fijas AVC a través de las memorias intermedias de los decodificadores objetivo del sistema TB_n , MB_n y EB_n deberá cumplir que $td_n(j) - t(i) \leq 10$ segundos para toda j , y todos los bytes i en la unidad de acceso AVC $A_n(j)$.

El retardo de los datos de imágenes fijas AVC a través de las memorias intermedias de los decodificadores objetivo del sistema TB_n , MB_n y EB_n deberán cumplir que $td_n(j) - t(i) \leq 60$ segundos para toda j , y todos los bytes i en la unidad de acceso AVC $A_n(j)$.

Condiciones de gestión de la memoria intermedia

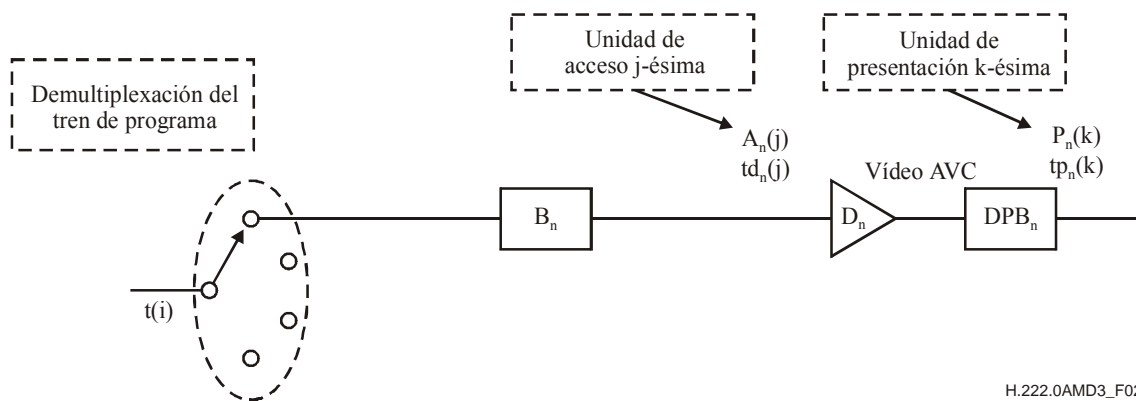
Los trenes de transporte deben construirse de modo que se satisfagan las siguientes condiciones de gestión de la memoria intermedia:

- No deberá producirse desbordamiento en la TB_n , la cual se vaciará al menos una vez cada segundo.
- No se producirá desbordamiento en MB_n , EB_n y DPB_n .
- No se producirá infrautilización de la capacidad en la EB_n , excepto cuando estén incluidos parámetros VUI en la secuencia vídeo AVC con `low_delay_hrd_flag` puesto a '1'. La infrautilización de la capacidad de la EB_n se produce para una unidad de acceso AVC $A_n(j)$ cuando la EB_n no contenga uno o más bytes de la $A_n(j)$ en el instante de decodificación $td_n(j)$.

NOTA 2 – Un tren de vídeo AVC puede transportar información para determinar la conformidad del tren de vídeo AVC con el HRD, según se especifica en el anexo C a la Rec. UIT-T H.264 | ISO/CEI 14496-10. La presencia de esta información puede indicarse en un tren de transporte utilizando un temporizador AVC y un descriptor HRD con el `hrd_management_valid_flag` puesto a '1'. Independientemente de la presencia de esta información, la conformidad de un tren de vídeo AVC con el T-STD garantiza que los requisitos de gestión de memorias intermedias HRD para CPB_n se satisfacen cuando cada byte del tren de vídeo AVC se entrega y se extrae de la CPB_n en el HRD justo en el mismo instante en el que el byte se entrega y se extrae de la EB_n en el T-STD.

2.14.3.2 Extensiones P-STD

El modelo P-STD para la decodificación de un tren elemental conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10 incluye una memoria intermedia múltiplex B_n y un decodificador D_n seguido de una memoria intermedia DPB_n , (véase la figura 2-12). Para cada tren de vídeo AVC n , el tamaño BS_n de la memoria intermedia B_n en el P-STD se define mediante el campo `P-STD_buffer_size` en la cabecera del paquete PES.



H.222.0AMD3_F02

Figura 2-12 – Extensiones del modelo P-STD según la Rec. H.264 | ISO/CEI 14496-10

Gestión de memoria intermedia DPB_n

La memoria intermedia DPB_n se gestionará exactamente del mismo modo que el T-STD; véase 2.14.3.1.

Gestión de la memoria intermedia B_n

Los datos de la unidad de acceso AVC entran en la memoria intermedia B_n como se indica en 2.5.2.2. La unidad de acceso AVC $A_n(j)$ se decodifica y se extrae instantáneamente de B_n en un instante $td_n(j)$. El tiempo de decodificación $td_n(j)$ se especifica mediante el DTS o mediante el tiempo de extracción de la CPB, que se obtienen de la información contenida en el tren de vídeo AVC. Tras la decodificación, la unidad de acceso AVC entra instantáneamente en la DPB_n o sale sin entrar en la misma, de acuerdo con las normas específicas de la Rec. UIT-T. H.264 | ISO/CEI 14496-10.

Retardo STD

El retardo total de datos conforme a la Rec. UIT-T H.264 | ISO/CEI 14496-10 distintos de los datos de imágenes fijas AVC a través de la memoria intermedia de los decodificadores objetivo del sistema B_n deberá cumplir que $td_n(j) - t(i) \leq 10$ segundos para toda j , y todos los bytes i en la unidad de acceso AVC $A_n(j)$.

El retardo de datos de imágenes fijas AVC a través de la memoria intermedia de los decodificadores objetivo del sistema B_n deberá cumplir que $td_n(j) - t(i) \leq 60$ segundos para toda j y para todos los bytes i en la unidad de acceso AVC $A_n(j)$.

Condiciones de gestión de la memoria intermedia

Los trenes de programa deberán construirse de manera que se cumplan las siguientes condiciones para la gestión de la memoria:

- No se producirá un desbordamiento de B_n .
- No se producirá una infrutilización de la capacidad de B_n , salvo cuando los parámetros VUI estén contenidos en la secuencia de vídeo AVC con `low_delay_hrd_flag` puesto a '1' o cuando el estado `trick_mode` sea verdadero. La infrutilización de la capacidad B_n se produce para una unidad de acceso AVC $An(j)$ cuando uno o varios bytes de $An(j)$ no están contenidos en B_n en el instante de decodificación $tdn(j)$.

Anexo A

Modelo de decodificador CRC

(Este anexo es parte integrante de esta Recomendación | Norma Internacional)

A.0 Modelo de decodificador CRC

En la figura A.1 se especifica el modelo de decodificador CRC de 32 bits.

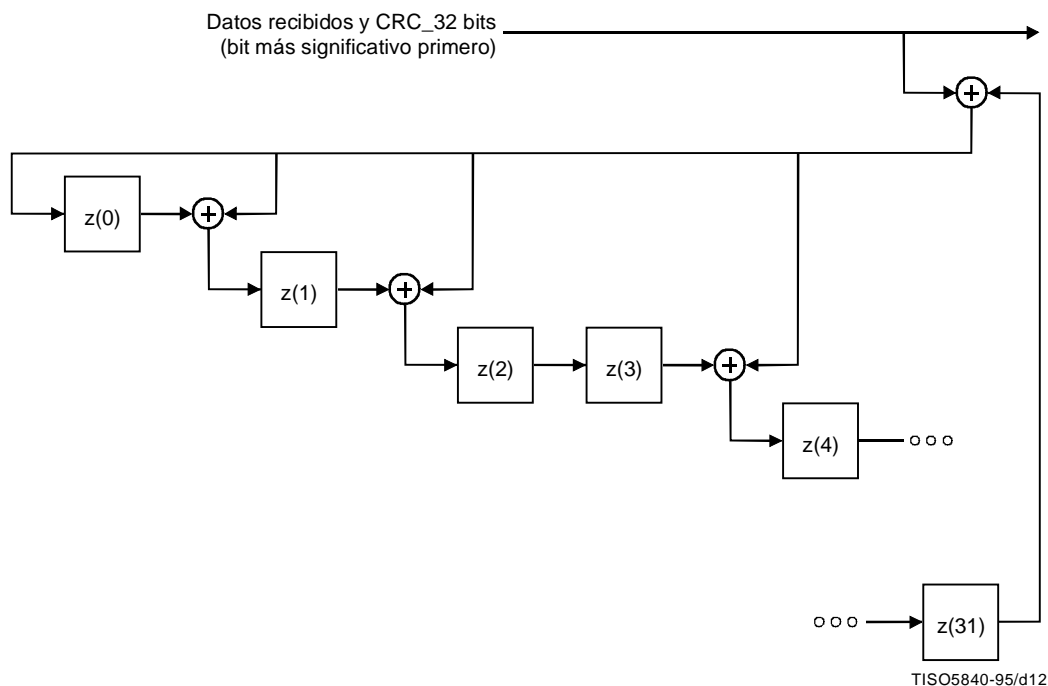


Figura A.1 – Modelo de decodificador CRC de 32 bits

El decodificador CRC de 32 bits funciona a nivel de bits y consiste en 14 sumadores '+' y 32 elementos de retardo $z(i)$. La entrada del decodificador CRC se suma a la salida de $z(31)$ y el resultado se proporciona a la entrada $z(0)$ y a una de las entradas de cada sumador restante. La otra entrada de cada sumador restante es la salida de $z(i)$, mientras que la salida de cada sumador restante se conecta a la entrada de $z(i + 1)$, con $i = 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22$ y 25 . Véase la figura A.1.

Ésta es la CRC calculada con el polinomio:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (\text{A-1})$$

En la entrada del decodificador CRC se reciben bytes. Cada byte es desplazado en el decodificador CRC un bit a la vez, con el bit más significativo primero. Por ejemplo, si la entrada es el byte 0x01, los siete '0' entran primero en el decodificador CRC seguidos por el '1'. Antes que la CRC procese los datos de una sección, la salida de cada elemento de retardo $z(i)$ se fija a su valor inicial '1'. Después de esta inicialización, cada byte de la sección es proporcionado a la entrada del decodificador CRC, incluidos los cuatro bytes CRC_32. Después de desplazar el último bit del último byte CRC_32 en el decodificador, es decir, en $z(0)$ después de la adición con la salida de $z(31)$, se lee la salida de todos los elementos de retardo $z(i)$. Cuando no hay errores, cada una de las salidas de $z(i)$ será cero. En el codificador CRC, el campo CRC_32 se codifica con un valor tal que esto esté asegurado.

Anexo B

Instrucciones y control del medio de almacenamiento digital (DSM-CC)

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

B.0 Introducción

El protocolo de instrucciones y control del medio de almacenamiento digital (DSM-CC) es un protocolo de aplicación específico destinado a proporcionar las funciones de control y operaciones básicas específicas para gestionar un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 en medios de almacenamiento digital. DSM-CC es un protocolo de bajo nivel por encima de las capas de red y por debajo de las capas de aplicación.

El DSM-CC será transparente en el sentido siguiente:

- es independiente del DSM utilizado;
- es independiente de si el DSM está en una ubicación local o distante;
- es independiente del protocolo de red con el cual interconecta el DSM-CC; y
- es independiente de los distintos sistemas operativos en los cuales funciona el DSM.

B.0.1 Finalidad

Muchas aplicaciones de instrucciones y control DSM de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 requieren acceso a un tren de bits de dicha Recomendación | Norma Internacional almacenado en una variedad de medios de almacenamiento digital en una ubicación local o distante. Diferentes DSM tienen sus propias instrucciones de control específicas, por lo que el usuario tendría que conocer diferentes conjuntos de instrucciones de control DSM específicas, para acceder a los trenes de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 desde diferentes DSM. Esto plantea muchas dificultades para el diseño de la interfaz de un sistema de aplicación de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 o de ISO/CEI 11172-1. Con miras a superar estas dificultades, se requiere un conjunto de instrucciones de control DSM comunes, que sean independientes del DSM específico utilizado. Este anexo es exclusivamente informativo. En ISO/CEI 13818-6 se define la extensión DSM-CC con un alcance más amplio.

B.0.2 Aplicaciones futuras

Además de las aplicaciones inmediatas soportadas por el actual protocolo de instrucciones y control DSM, entre las futuras aplicaciones basadas en extensiones del DSM-CC cabe incluir las siguientes:

Vídeo a la carta

Se proporcionan programas vídeo solicitados por un cliente a través de distintos canales de comunicación. El cliente podrá seleccionar un programa vídeo de una lista de programas disponibles en un servidor vídeo. Estas aplicaciones podrán ser utilizadas por hoteles, televisión por cable, instituciones educativas, hospitales, etc.

Servicios vídeo interactivos

En estas aplicaciones, el usuario mediante interacción frecuente controla la manipulación de vídeo y audio almacenados. Estos servicios pueden incluir juegos basados en vídeo, viajes vídeo controlados por el usuario, compra electrónica, etc.

Redes vídeo

Es posible que diversas aplicaciones deseen intercambiar datos de audio y vídeo almacenados a través de algún tipo de red de computador. Los usuarios podrán encaminar la información audiovisual a través de la red vídeo a sus terminales. Las aplicaciones de publicidad electrónica y multimedios son ejemplos de esta clase de aplicación.

B.0.3 Ventajas

Al especificar instrucciones de control DSM independientemente del DSM, los usuarios finales pueden realizar la decodificación de trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 sin tener que comprender plenamente el funcionamiento detallado del DSM específico utilizado.

Las instrucciones de control DSM son códigos para dar a los usuarios finales la garantía de que los trenes de bits de UIT-T H.222.0 | ISO/CEI 13818-1 se pueden visualizar y almacenar con la misma semántica, con independencia del DSM y de la interfaz de usuario. Son instrucciones fundamentales para el control del funcionamiento DSM.

B.0.4 Funciones básicas

B.0.4.1 Selección de trenes

El DSM-CC proporciona los medios de seleccionar un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 para realizar las operaciones siguientes. Estas operaciones comprenden la creación de un nuevo tren de bits. Los parámetros de esta función son:

- índice del tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 (la correspondencia entre este índice y el nombre significativo de una aplicación está fuera del ámbito del actual protocolo DSM-CC);
- modo (extracción/almacenamiento).

B.0.4.2 Extracción

El DSM-CC proporciona los medios para:

- reproducir un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 identificado;
- reproducirlo en un tiempo de presentación dado;
- fijar la velocidad de reproducción (normal o rápida);
- fijar la duración de la reproducción (hasta un tiempo de presentación especificado, el fin del tren de bits reproducido hacia adelante o el principio en reproducción inversa o la emisión de una instrucción de parada);
- fijar el sentido (hacia adelante o hacia atrás);
- hacer una pausa;
- reanudar;
- cambiar el punto de acceso en el tren de bits;
- parar.

B.0.4.3 Almacenamiento

El DSM-CC proporciona los medios para:

- almacenar un tren de bits válido durante un intervalo especificado;
- detener el almacenamiento.

El DSM-CC proporciona un subconjunto útil pero limitado de funcionalidades que pueden ser requeridas en aplicaciones de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 basadas en DSM. Cabe esperar que se añadan capacidades adicionales importantes en extensiones futuras.

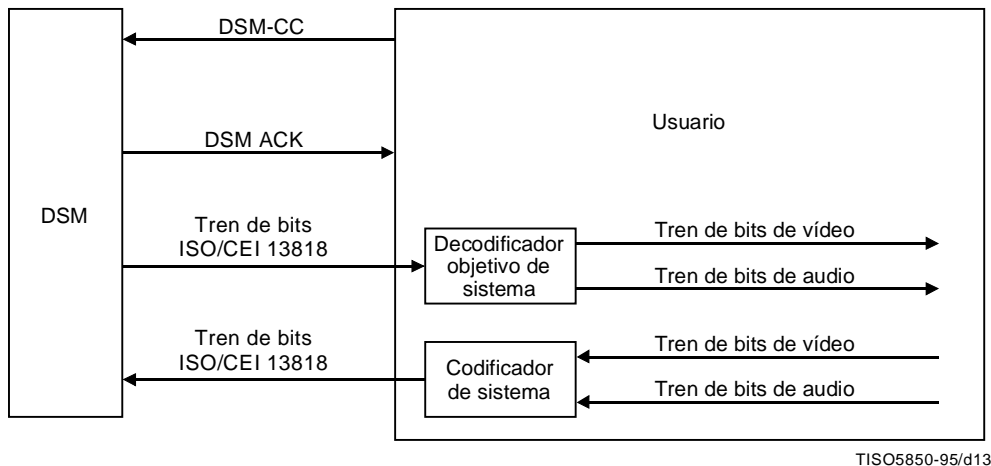
B.1 Elementos generales

B.1.1 Alcance

El alcance de este trabajo consiste en la elaboración de una Recomendación | Norma Internacional para especificar un conjunto útil de instrucciones para el control de medios de almacenamiento digital en los cuales se almacenan trenes de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Las instrucciones pueden ejecutar el telecontrol de un medio de almacenamiento digital en general independientemente del DSM específico y aplicarlo a cualquier tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 almacenado en DSM.

B.1.2 Visión de conjunto y aplicación del protocolo DSM-CC

La sintaxis y la semántica actuales del DSM-CC abarcan la aplicación de un usuario al DSM. El sistema del usuario es capaz de extraer un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y es también capaz (facultativamente) de generar un tren de bits de dicha Recomendación | Norma Internacional. El canal de control por el cual se envían las instrucciones y acuses de recibo DSM se muestra en la figura B.1 como un canal fuera de banda. Esto se puede realizar también insertando las instrucciones y acuses de recibo DSM-CC en los trenes de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 si no se dispone de un canal fuera de banda.



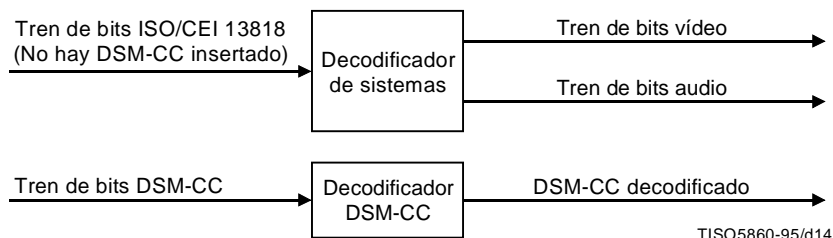
TISO5850-95/d13

Figura B.1 – Configuración de una aplicación DSM-CC

B.1.3 Transmisión de instrucciones y acuses de recibo DSM-CC

El protocolo DSM-CC se codifica en un tren de bits DSM-CC de acuerdo con la sintaxis y semántica definidas en B.2.2 a B.2.9. El tren de bits DSM-CC puede ser transmitido como un tren de bits autónomo y en un tren de bits de sistema de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Cuando el tren de bits DSM-CC se transmite en modo autónomo, su relación con el tren de bits de sistema y el proceso de decodificación se ilustran en la figura B.2. En este caso, el tren de bits DSM-CC no está insertado en el tren de bits de sistema. Este modo de transmisión se puede utilizar en la aplicación cuando el DSM está conectado directamente con el decodificador de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Se puede utilizar también en aplicaciones donde el tren de bits DSM-CC podrá ser controlado y transmitido por otros tipos de multiplexores de red.



TISO5860-95/d14

Figura B.2 – Tren de bits DSM-CC decodificado como un tren de bits autónomo

Para algunas aplicaciones, es conveniente transmitir el DSM-CC en un tren de bits de sistema de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, de modo que algunas características de este tren de bits de sistema de dicha Recomendación | Norma Internacional se puedan aplicar también al tren de bits DSM-CC. En este caso, el tren de bits DSM-CC es insertado en el tren de bits de sistema por el multiplexor de sistemas.

El tren de bits DSM-CC es codificado por el codificador de sistema según el siguiente proceso. Primero, el tren de bits DSM-CC es paquetizado en un tren de elementos paquetizados (PES), de acuerdo con la sintaxis descrita en 2.4.3.6. Después, el paquete PES es multiplexado en un tren de programa (PS) o en un tren de transporte (TS) de acuerdo con el requisito del medio de transmisión. Los procedimientos de decodificación son la inversa de los procedimientos de codificación y se ilustran en el diagrama de bloques del decodificador de sistemas de la figura B.3.

En la figura B.3, la salida de decodificador de sistemas es un tren de bits de vídeo, un tren de bits de audio y/o un tren de bits DSM-CC. El tren de bits DSM-CC es identificado por el identificador de tren, valor '1111 0010' definido en el cuadro 2-22 de la de id de tren. Una vez identificado el tren de bits DSM-CC, sigue las reglas especificadas por el T-STD o el P-STD.

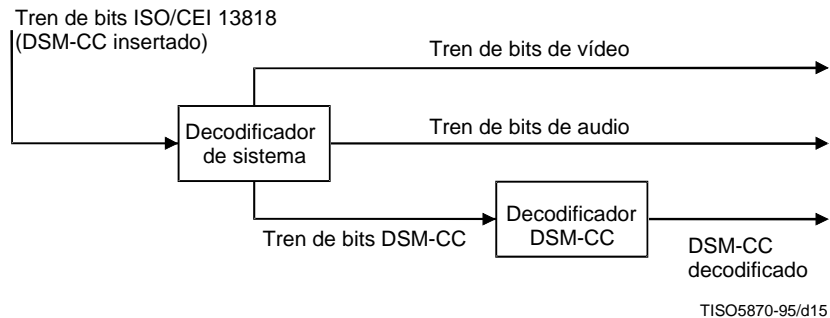


Figura B.3 – Tren de bits DSM-CC decodificado como parte del tren de bits de sistema

B.2 Elementos técnicos

B.2.1 Definiciones

A los efectos de la presente Recomendación | Norma Internacional, son aplicables las definiciones siguientes.

B.2.1.1 DSM-CC: Las instrucciones y control de medios de almacenamiento digital especificadas por la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 para el control de medios de almacenamiento digital en ubicaciones locales o distantes que contienen un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.2 DSM ACK: El acuse de recibo del receptor de la instrucción DSM-CC al iniciador de la instrucción.

B.2.1.3 tren de bits MPEG: Un tren de sistemas de ISO/CEI 11172-1 un tren de programa o un tren de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.4 servidor DSM-CC: Un sistema, local o distante, utilizado para almacenar y/o extraer un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.5 punto de acceso aleatorio: Un punto en el tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 con la propiedad de que por lo menos para un tren elemental dentro del tren de bits, la siguiente unidad de acceso, 'N', completamente contenida en el tren de bits puede ser decodificada sin referencia a unidades de acceso previas, y para cada tren elemental en el tren de bits todas las unidades con tiempos de presentación iguales o ulteriores están completamente contenidas siguientemente en el tren de bits y pueden ser completamente decodificadas por un decodificador-objetivo de sistema sin acceso a información previa al punto de acceso aleatorio. El tren de bits almacenado en el DSM puede tener ciertos puntos de acceso aleatorio; la salida del DSM puede incluir puntos de acceso aleatorio adicionales construidos por la propia manipulación por el DSM del material almacenado (por ejemplo, almacenamiento de matrices de cuantificación de modo que se pueda generar un encabezamiento de secuencias siempre que sea necesario). Un punto de acceso aleatorio tiene una PTS asociada, a saber, la PTS real o implícita de la unidad de acceso 'N'.

B.2.1.6 valor PTS operacional vigente: La PTS vigente o implícita asociada con el último punto de acceso aleatorio que precede a la unidad de acceso proporcionada por el DSM a partir del tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 vigente seleccionado. Si ninguna unidad de acceso ha sido proporcionada por este tren de bits, el DSM no es capaz de proporcionar acceso aleatorio en el tren de bits vigente y el valor PTS operacional vigente es el primer punto de acceso aleatorio en el tren de bits de dicha Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

B.2.1.7 tren de bits DSM-CC: Una secuencia de bits que cumple la sintaxis de la B.2.2.

B.2.2 Especificación de la sintaxis DSM-CC

- Cada instrucción de control DSM empezará por un código de comienzo, indicado en el cuadro B.1.
- Cada instrucción de control tendrá una longitud de paquete para especificar el número de byte en un paquete DSM-CC.

- Cuando un tren de bits DSM-CC se transmite como un paquete PES definido en 2.4.3.6 los campos hasta el campo de longitud de paquete son idénticos a los especificados en 2.4.3.6. En otras palabras, si el paquete DSM-CC se encapsula en un paquete PES, el código de comienzo de paquete PES es el único código de comienzo al principio del paquete.
- La instrucción o acuse de recibo de control real seguirá al último byte del campo de longitud de paquete.
- Será proporcionado un tren de acuse de recibo por el receptor del tren de bits de control DSM después que ha comenzado o se ha ejecutado la operación solicitada, según la instrucción específica recibida.
- El DSM es responsable en todo momento de proporcionar un tren de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 normativo. Esto puede incluir la manipulación de bits en modo truco que se define en 2.4.3.6.

Cuadro B.1 – Sintaxis DSM-CC

Sintaxis	N.º de bits	Mnemónico
DSM_CC() {		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
packet_length	16	uimsbf
command_id	8	uimsbf
If (command_id == '01') {		
control()		
} else if (command_id == '02') {		
ack()		
}		
}		

B.2.3 Semántica de campos en la especificación de sintaxis DSM-CC

packet_start_code_prefix (prefijo de código de comienzo de paquete) – Es un código de 24 bits. Junto con el identificador de tren que le sigue constituye un código de comienzo de paquete DSM-CC que identifica el principio de un tren de bits de paquete DSM-CC. El prefijo de código de comienzo de paquete es la cadena de bits '0000 0000 0000 0000 0000 0001' (0x000001).

stream_id (identificador de tren) – Es un campo de 8 bits que especifica el tipo de tren de bits y tendrá un valor '1111 0010' en el caso de tren de bits DSM-CC. Véase el cuadro 2-23.

packet_length (longitud de paquete) – Es un campo de 16 bits que especifica el número de byte en el paquete DSM-CC que sigue inmediatamente al último byte del campo.

command_id (identificador de instrucción) – Es un entero sin signo de 8 bits. Identifica si el tren de bits es un tren de instrucción de control o de acuse de recibo. Los valores se definen en el cuadro B.2.

Cuadro B.2 – Valores asignados de identificador de instrucción

Valor	Command_id
0x00	Prohibido
0x01	Control
0x02	Acuse de recibo
0x03-0xFF	Reservado

B.2.4 Capa de control

Constricciones de la fijación de banderas en el control DSM-CC:

- Como máximo, una de las banderas para seleccionar, reproducir y almacenar se pondrán a '1' para cada instrucción de control DSM. Si ninguno de estos bits está fijado, se pasará por alto esta instrucción.

- Como máximo, se fijará una de las siguientes banderas `pause_mode`, `resume_mode`, `stop_mode`, `play_flag` y `jump_flag` para cada instrucción de extracción. Si ninguno de estos bits está fijado, se pasará por alto esta instrucción.
- Como máximo, se seleccionará una `record_flag` y `stop_mode` para cada instrucción de almacenamiento. Si ninguno de estos bits está fijado, se pasará por alto esta instrucción.

Véase el cuadro B.3.

Cuadro B.3 – Control DSM-CC

Sintaxis	N.º de bits	Mnemónico
<code>control() {</code>		
select_flag	1	bslbf
retrieval_flag	1	bslbf
storage_flag	1	bslbf
reserved	12	bslbf
marker_bit	1	bslbf
If (select_flag == '1') {		
bitstream_id [31..17]	15	bslbf
marker_bit	1	bslbf
bitstream_id [16..2]	15	bslbf
marker_bit	1	bslbf
bitstream_id [1..0]	2	bslbf
select_mode	5	bslbf
marker_bit	1	bslbf
}		
if (retrieve_flag == '1') {		
jump_flag	1	bslbf
play_flag	1	bslbf
pause_mode	1	bslbf
resume_mode	1	bslbf
stop_mode	1	bslbf
reserved	10	bslbf
marker_bit	1	bslbf
if (jump_flag == '1') {		
reserved	7	bslbf
direction_indicator	1	bslbf
time_code()		
}		
if (play_flag == '1'){		
speed_mode	1	bslbf
direction_indicator	1	bslbf
reserved	6	bslbf
time_code()		
}		
}		
if (storage_flag == '1') {		
reserved	6	bslbf
record_flag	1	bslbf
stop_mode	1	bslbf
if (record_flag == '1') {		
time_code()		
}		
}		
}		

B.2.5 Semántica de campos en la capa de control

marker_bit (bit marcador) – Es un marcador de un bit que se pone siempre a '1' para evitar emulación del código de comienzo.

reserved_bits (bits reservados) – Es un campo de 12 bits reservado para uso futuro por la presente Recomendación | Norma Internacional para instrucciones de control de DSM. Hasta que UIT-T | ISO/CEI especifiquen otra cosa, tendrá el valor '0000 0000 0000'.

select_flag (bandera de selección) – Esta es una bandera de un bit que cuando se pone a '1' especifica una operación de selección de tren de bits. Cuando se pone a '0', no se producirá ninguna operación de selección de tren de bits.

retrieval_flag (bandera de extracción) – Esta es una bandera de un bit que puesta a '1' especifica que se producirá una acción de extracción específica (reproducción). La operación comienza con el valor PTS operacional vigente.

storage_flag (bandera de almacenamiento) – Esta es una bandera de un bit que cuando está puesta a '1' especifica que se ha de ejecutar una operación de almacenamiento.

bistream_ID (identificador de tren de bits) – Este campo de 32 bits se codifica en tres partes que se combinan para formar un entero sin signo que especifica el tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que se ha de seleccionar. Es responsabilidad del servidor DSM hacer corresponder los nombres del tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 almacenado en su DSM de manera única con una serie de números que podrán ser representados por el identificador de tren de bits.

select_mode (selección de modo) – Es un entero sin signo de 5 bits para especificar qué modo de funcionamiento del tren de bits se solicita. En el cuadro B.4 se especifican los modos definidos.

Cuadro B.4 – Valores asignados de selección de modo

Código	Modo
0x00	Prohibido
0x01	Almacenamiento
0x02	Extracción
0x03-0x1F	Reservado

jump_flag (bandera de salto) – Es una bandera de un bit que puesta a '1' especifica el salto del puntero de reproducción a una nueva unidad de acceso. La nueva PTS es especificada por un código de tiempo relativo con respecto al valor PTS operacional vigente. Esta función sólo es válida cuando el tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 está en el modo "parada".

play_flag (bandera de reproducción) – Esta es una bandera de un bit que puesta a '1' especifica la reproducción de un tren de bits durante un determinado periodo de tiempo. La velocidad, sentido y duración de la reproducción son parámetros adicionales en el tren de bits. La reproducción comienza a partir del valor PTS operacional vigente.

pause_mode (modo pausa) – Este es un código de un bit que especifica que se haga una pausa en la acción de reproducción y se mantenga el puntero de reproducción en el valor PTS operacional vigente.

resume_mode (modo reanudación) – Es un código de un bit que especifica que se continúe la acción de reproducción a partir del valor PTS operacional vigente. Reanudación sólo tiene significado si el tren de bit vigente está en el estado "pausa" y el tren de bits se fijará al estado de reproducción hacia adelante a velocidad normal.

stop_mode (modo parada) – Este es un código de un bit que especifica la parada de la transmisión de un tren de bits.

direction_indicator (indicador de sentido) – Este es un código de un bit que indica el sentido de reproducción. Si este bit se pone a '1', es hacia adelante. En los demás casos es hacia atrás.

speed_mode (modo de velocidad) – Este es un código de un bit para especificar la escala de velocidad. Si se pone a '1' especifica que la velocidad es normal. Si se pone a '0', especifica que la velocidad es rápida (es decir rápida hacia adelante o rápida hacia atrás).

record_flag (bandera de grabación) – Esta es una bandera de un bit que especifica la petición de grabar el tren de bits de un usuario final a un DSM durante un tiempo especificado o hasta la recepción de una instrucción de parada, lo que llegue primero.

B.2.6 Capa de acuse de recibo

Constricciones de la fijación de banderas en el control DSM-CC

Sólo uno de los bits de acuse de recibo especificados a continuación se pueden poner a '1' para cada tren de bits de acuse de recibo DSM (véase el cuadro B.5).

Cuadro B.5 – Acuse de recibo DSM-CC

Sintaxis	N.º de bits	Mnemónico
ack() {		
select_ack	1	bslbf
retrieval_ack	1	bslbf
storage_ack	1	bslbf
error_ack	1	bslbf
reserved	10	bslbf
marker_bit	1	bslbf
cmd_status	1	bslbf
If (cmd_status == '1' && (retrieval_ack == '1' storage_ack == '1')) {		
time_code()		
}		
}		

B.2.7 Semántica de campos en la capa de acuse de recibo

select_ack (acuse de recibo de selección) – Este es un campo de 1 bit, que puesto a '1' especifica que la instrucción de acuse de recibo es para acusar recibo de una instrucción de selección.

retrieval_ack (acuse de recibo de extracción) – Este es un campo de 1 bit que puesto a '1' especifica que la instrucción de acuse de recibo es para acusar recibo de una instrucción de extracción.

storage_ack (acuse de recibo de almacenamiento) – Este es un campo de 1 bit que puesto a '1' especifica que la instrucción de acuse de recibo es para acusar recibo de una instrucción de almacenamiento.

error_ack (acuse de recibo de error) – Este es un campo de 1 bit que puesto a '1' indica un error DSM. Los errores definidos normalmente son EOF [end of file (fin de fichero) en reproducción hacia adelante o comienzo de fichero en reproducción inversa] en un tren que está siendo extraído y disco completo (Disk Full) en un tren que está siendo almacenado. Si este bit está fijado a '1', cmd_status no está definido. En cualquiera de los dos casos, se selecciona el tren de bits vigente.

cmd_status (estado de instrucción) – Esta es una bandera de 1 bit que puesta a '1' especifica que la instrucción es aceptada. Si está puesta a '0', la instrucción es rechazada. Según la instrucción recibida, se aplica la siguiente semántica:

- Si select_ack está fijada y cmd_status está puesto a '1', especifica que se selecciona el tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y que el servidor está preparado para proporcionar el modo de funcionamiento seleccionado. El valor PTS operacional vigente está fijado al primer punto de acceso aleatorio del tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 nuevamente seleccionado. Si cmd_status está puesto a '0', la operación ha fracasado y no se ha seleccionado ningún tren de bits.
- Si retrieval_ack está fijado y cmd_status está puesto a '1', especifica que se ha iniciado la operación de extracción para todas las instrucciones de extracción. La posición del puntero PTS operacional vigente es informada por el time_code (código de tiempo) siguiente.
- Para la instrucción play_flag con infinite_time_flag != '1', se enviará un segundo acuse de recibo, que acusará recibo de que la operación de reproducción ha terminado alcanzando la duración definida por la instrucción play_flag.
- Si cmd_status está puesta a '0' en un acuse de recibo de extracción, la operación ha fracasado, y los posibles motivos pueden ser un ID de tren de bits no válido, un salto que rebasa el fin de un fichero, o una función no soportada, como reproducción inversa a velocidad normal.
- Si storage_ack está fijada, especifica que está comenzando la operación de almacenamiento para la instrucción record_flag, o ha sido completada por la instrucción stop_mode. La PTS de la última unidad de acceso completa almacenada es informada por el código de tiempo siguiente.

- Si la operación de grabación ha terminado alcanzando la duración definida por la instrucción `storage_flag`, se enviará otro acuse de recibo y se informará el valor PTS operacional vigente después de la grabación.
- Si `cmd_status` está puesta a '0' en un acuse de recibo de almacenamiento, la operación ha fracasado y los posibles motivos son un ID de tren de bits no válido, o la incapacidad del DSM de almacenar datos.

B.2.8 Código de tiempo

Constricciones del código de tiempo

- Una operación hacia adelante de duración especificada dada por un código de tiempo termina después que la PTS real implícita de una unidad de acceso es observada de modo que PTS menos el valor PTS operacional vigente al principio de la operación módulo 2^{33} rebasa la duración.
- Una operación hacia atrás de duración especificada dada por un código de tiempo termina después que la PTS real o implícita de una unidad de acceso es observada de modo que el valor PTS operacional vigente al comienzo de la operación menos ese módulo 2^{33} de PTS rebasa la duración.
- Para todas las instrucciones en la capa de control, el código de tiempo se especifica como una duración relativa con respecto al valor PTS operacional vigente.
- Para todas las instrucciones en la capa de acuse de recibo, el código de tiempo es especificado por el valor PTS operacional vigente.

Véase el cuadro B.6.

Cuadro B.6 – Código de tiempo

Sintaxis	N.º de bits	Mnemónico
<code>time_code() {</code>		
Reserved	7	bslbf
infinite_time_flag	1	bslbf
if (infinite_time_flag == '0') {		
reserved	4	bslbf
PTS [32..30]	3	bslbf
marker	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
}		

B.2.9 Semántica de campos en el código de tiempo

infinite_time_flag (bandera de tiempo infinito) – Esta es una bandera de un bit que cuando está puesta a '1' especifica un periodo de tiempo infinito. Esta bandera se pone a '1' en las aplicaciones cuando no se puede definir por anticipado un periodo de tiempo para una operación específica.

PTS[32..0] – La indicación de tiempo de la unidad de acceso del tren de bits. Según la función, puede ser un valor absoluto o un retardo de tiempo relativo en ciclos del reloj de sistema de 90 kHz.

Anexo C

Información específica de programa

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

C.0 Explicación de la información específica de programa en trenes de transporte

La subcláusula 2.4.4 contiene la sintaxis, semántica y texto normativos concernientes a la información específica de programa. En todos los casos hay que cumplir las constricciones de 2.4.4. Este anexo proporciona información explicativa sobre cómo utilizar las funciones PSI y examina ejemplos de cómo se puede utilizar en la práctica.

C.1 Introducción

La presente Recomendación | Norma Internacional proporciona un método para describir el contenido de paquetes del tren de transporte a los efectos de la demultiplexación y presentación de programas. La especificación de codificación incluye esta función a través de la información específica de programa (PSI). En este anexo se examina la utilización de PSI.

Puede considerarse que la PSI pertenece a seis tablas:

- 1) tabla de asociación de programas (PAT, *program association table*);
- 2) tabla de correspondencia de programas TS (PMT, *program map table*);
- 3) tabla de información de red (NIT, *network information table*);
- 4) tabla de acceso condicional (CAT, *conditional access table*);
- 5) tabla de descripción de tren de transporte (TSDT, *transport stream description table*); y
- 6) tabla de información de control IPMP (ICIT, *IPMP control information table*).

El contenido de las tablas PAT, PMT, CAT y TSDT se especifica en esta Recomendación | Norma Internacional. La tabla ICIT se define en ISO/CEI 13818-11 (IPMP en MPEG-2).

NIT es una tabla privada, pero el valor PID de los paquetes del tren de transporte que la llevan se especifica en la PAT. Las tablas NIT e ICIT deben seguir la estructura de sección definida en esta Recomendación | Norma Internacional.

C.2 Mecanismo funcional

Las tablas enumeradas anteriormente son teóricas en cuanto que nunca necesitan ser regeneradas en una forma especificada dentro de un decodificador. Si bien puede considerarse estas estructuras como simples tablas, pueden ser divididas antes de enviarlas en paquetes del tren de transporte. La sintaxis soporta esta operación permitiendo que las tablas sean divididas en secciones y proporcionando un método de correspondencia normativo con las cabidas útiles de paquetes del tren de transporte. Se proporciona también un método para transportar datos privados en un formato similar. Esto es útil porque se puede utilizar el mismo procesamiento básico en el decodificador para los datos PSI y para los datos privados, lo que ayuda a reducir los costes. Para orientación sobre la colocación óptima de PSI en el tren de transporte, véase el anexo D.

Cada sección está identificada únicamente por la combinación de los siguientes elementos:

i) **table_id (identificador de tabla)**

El `table_id` de 8 bits identifica a qué tabla pertenece la sección.

- Las secciones con `table_id` 0x00 pertenecen a la tabla de asociación de programas.
- Las secciones con `table_id` 0x01 pertenecen a la tabla de acceso condicional.
- Las secciones con `table_id` 0x02 pertenecen a la tabla de correspondencia de programas TS.
- Las secciones con `table_id` 0x03 pertenecen a la `TS_description_section`.
- Las secciones con `table_id` 0x04 pertenecen a la `ISO_CEI_14496_scene_description_section`.
- Las secciones con `table_id` 0x05 pertenecen a la `ISO_CEI_14496_object_descriptor_section`.
- Las secciones con `table_id` 0x06 pertenecen a la `metadata_section`.

- Las secciones con `table_id` 0x07 pertenecen a la `IPMP_Control_Information_section`.

El usuario puede asignar otros valores de `table_id` para fines privados.

Es posible establecer filtros que observan el campo `table_id` para identificar si una nueva sección pertenece o no a una tabla de interés.

ii) **`table_id_extension` (extensión de identificador de tabla)**

Este campo de 16 bits existe en la versión larga de una sección. En la tabla de asociación de programas se utiliza para identificar el `transport_stream_id` (identificador de tren de transporte) del tren, efectivamente una etiqueta definida por el usuario que permite distinguir un tren de transporte de otro dentro de una red o a través de redes. En la tabla de acceso condicional, este campo actualmente no tiene significado por lo que se ha marcado "reservado" lo que indica que será codificado como 0xFFFF, pero ese significado puede ser definido por UIT-T | ISO/CEI en una siguiente revisión de la presente Recomendación | Norma Internacional. En una sección de correspondencia de programas TS, el campo contiene el `program_number` (número de programa) y por tanto identifica el programa al cual se refieren los datos en la sección. La `table_id_extension` se puede utilizar también como un punto de filtro en algunos casos.

iii) **`section_number` (número de sección)**

El campo `section_number` permite que las secciones de una tabla determinada sean reensambladas en su orden original por el decodificador. La presente Recomendación | Norma Internacional no impone la obligación de que las secciones se transmitan en orden numérico, pero se recomienda, a menos que se desee transmitir algunas secciones de la tabla más frecuentemente que otras, por ejemplo, debido a consideraciones relativas al acceso aleatorio.

iv) **`version_number` (número de versión)**

Cuando cambian las características del tren de transporte descrito en PSI (por ejemplo, adición de programas suplementarios, diferente composición de trenes elementales para un programa dado), los nuevos datos PSI tienen que ser enviados con información actualizada pues la versión transmitida más recientemente de las secciones marcadas como "vigentes" debe ser siempre válida. Los decodificadores tienen que poder identificar si la sección recibida más recientemente es idéntica a la sección que ya han procesado/almacenado (en cuyo caso la sección puede ser descartada) o si es diferente, y por tanto puede significar un cambio de configuración. Esto se logra enviando una sección con los mismos `table_id`, `table_id_extension` y `section_number` que la sección anterior que contiene los datos pertinentes, pero con el siguiente valor de `version_number`.

v) **`current_next_indicator` (siguiente indicador vigente)**

Es importante saber en qué punto del tren de bits es válida la PSI. Por consiguiente, cada sección puede numerarse como válida "ahora" (vigente) o como válida en el futuro inmediato (siguiente). Esto permite la transmisión de una configuración futura antes del cambio, lo que da al decodificador la oportunidad de preparar dicho cambio. Sin embargo, no hay obligación de transmitir la siguiente versión de una sección por anticipado, pero si se transmite, será la siguiente versión correcta de esa sección.

C.3 Correspondencia de secciones con paquetes del tren de transporte

Las secciones corresponden directamente con paquetes del tren de transporte, es decir, sin una correspondencia previa con paquetes PES. Las secciones no tienen que comenzar al principio de los paquetes del tren de transporte (aunque pueden), porque el comienzo de la primera sección en la cabida útil de un paquete del tren de transporte está apuntado por el `pointer_field` (campo de puntero). La presencia del `pointer_field` es señalizada por `payload_unit_start_indicator` (indicador de comienzo de unidad de cabida útil) que se pone a un valor de '1' en paquetes PSI. (En paquetes que no son PSI, el indicador señala que un paquete PES comienza en el paquete de tren de transporte.) El `pointer_field` señala el comienzo de la primera sección en el paquete del tren de transporte. Nunca hay más de un `pointer_field` en un paquete de tren de transporte, pues el comienzo de cualquier otra sección se puede identificar contando la longitud de la primera sección y cualesquiera subsiguientes, puesto que la sintaxis no permite saltos entre secciones dentro de un paquete del tren de transporte.

Es importante señalar que dentro de paquetes del tren de transporte de cualquier valor de PID, una sección debe ser terminada antes que la otra pueda comenzar, o si no, no es posible identificar a qué encabezamiento de sección pertenecen los datos. Si una sección termina antes del fin de un paquete del tren de transporte, pero no es conveniente abrir otra sección, se proporciona un mecanismo de relleno para llenar el espacio. El relleno se efectúa llenando cada byte restante del paquete con el valor 0xFF. En consecuencia, el valor `table_id` 0xFF está prohibido porque se confundiría con el relleno. Una vez que ha aparecido un byte 0xFF al final de una sección, el resto del paquete del tren de transporte debe ser rellenado con bytes 0xFF, lo que permite al decodificador descartar el resto del paquete del tren de transporte. El relleno se puede efectuar también utilizando el mecanismo normal de campo de adaptación.

C.4 Velocidades de repetición y acceso aleatorio

En los sistemas en los que se ha de considerar el acceso aleatorio, se recomienda retransmitir las secciones PSI varias veces, aun cuando no se produzcan cambios de la configuración, pues en el caso general, el decodificador necesita los datos PSI para identificar el contenido del tren de transporte con miras a poder comenzar la decodificación. La presente Recomendación | Norma Internacional no impone requisitos de velocidad de repetición o aparición de secciones PSI. No obstante, evidentemente la repetición de secciones ayuda con frecuencia a las aplicaciones de acceso aleatorio, a la vez que aumenta la velocidad binaria utilizada por datos PSI. Si las correspondencias de programa son estáticas o casi estáticas, pueden ser almacenadas en el decodificador para permitir el acceso más rápido a los datos que si hay que esperar que sean retransmitidos. El fabricante del decodificador puede establecer un compromiso entre el volumen de almacenamiento requerido y la repercusión deseada en el tiempo de adquisición de canal.

C.5 ¿Qué es un programa?

El concepto de un programa tiene una definición precisa dentro de la presente Recomendación | Norma Internacional [véase 2.1.60 programa (sistema)]. En el caso de un tren de transporte, la base de tiempo se define por la PCR. Esta crea efectivamente un canal virtual dentro del tren de transporte.

Obsérvese que ésta no es la misma definición comúnmente utilizada en radiodifusión, donde un "programa" es un conjunto de trenes elementales no sólo con una base de tiempos común, sino también con un tiempo de comienzo y fin común. Una serie de "programas de radiodifusores" (denominados en este anexo como eventos) pueden ser transmitidos secuencialmente en un tren de transporte utilizando el mismo número de programa para crear un canal de televisión "convencional de radiodifusión" (a veces denominado un servicio).

Las descripciones de eventos podrán ser transmitidas en secciones privadas.

Un programa es denotado por un número de programa, que tiene significado solamente dentro de un tren de transporte. El número de programa es un entero sin signo de 16 bits y que permite por tanto que existan 65535 programas únicos dentro de un tren de transporte (el número de programa 0 se reserva para identificación de la NIT). Cuando el decodificador dispone de varios trenes de transporte (por ejemplo, en una red en cable), con miras a demultiplexar satisfactoriamente un programa, el decodificador debe conocer el `transport_stream_id` (identificador del tren de transporte) (para hallar el múltiplex adecuado) y el `program_number` (número de programa) del servicio (para hallar el programa correcto dentro del múltiplex).

La correspondencia del tren de transporte se puede realizar mediante la tabla de información de red (NIT, *network information table*). Obsérvese que la tabla de información de red puede estar almacenada en la memoria no volátil del decodificador para reducir el tiempo de adquisición del canal. En este caso, sólo tiene que ser transmitida con frecuencia suficiente para soportar puntualmente las operaciones de establecimiento de inicialización del decodificador. El contenido de la NIT es privado, pero adoptará al menos la estructura de sección mínima.

C.6 Atribución de número de programa

Puede no ser conveniente en todos los casos agrupar juntos todos los trenes elementales que comparten una referencia de reloj común como un programa. Es concebible tener un tren de transporte multiservicios con sólo un conjunto de PCR, común a todos. En general, aunque un radiodifusor puede preferir dividir lógicamente el tren de transporte en varios programas, donde el `PCR_PDI` (ubicación de la referencia de reloj) es siempre el mismo. Este método de dividir los trenes elementales en programas pseudoIndependientes puede tener varios usos. A continuación se dan dos ejemplos:

i) *Transmisiones multilingües en mercados separados*

Un tren de vídeo puede estar acompañado por varios trenes de audio en diferentes idiomas. Es conveniente incluir un ejemplo del descriptor de idioma `ISO_639` asociado con cada tren de audio para poder seleccionar el programa y el audio correctos. Es razonable disponer de varias definiciones de programa con diferentes números de programa, donde todos los programas referencian el mismo tren de vídeo y `PCR_PID`, pero tienen diferentes `PID` de audio. No obstante, es también razonable y posible numerar el tren de vídeo y todos los trenes de audio como un programa, cuando no se rebasa el límite de tamaño de sección de 1024 bytes.

ii) *Definiciones de programas muy grandes*

Hay un límite máximo de longitud de sección de 1024 bytes (incluido el encabezamiento de sección y `CRC_32`). Esto significa que ninguna definición de programa puede rebasar esta longitud. Para la gran mayoría de los casos, incluso cuando cada tren elemental tiene varios descriptores, este tamaño es adecuado. Sin embargo, cabe considerar los casos en sistemas de velocidad binaria muy alta, en los que se podría rebasar este límite. Entonces es posible en general identificar métodos de dividir las referencias de los trenes, de modo que no tengan que enumerarse todos juntos. Algunos trenes elementales podrán ser referenciados en más de un programa, y algunos solamente en uno o en el otro, pero no en ambos.

C.7 Utilización de PSI en un sistema típico

Un sistema de comunicaciones, especialmente en aplicaciones de radiodifusión, puede consistir en muchos trenes de transporte. Cada una de las cuatro estructuras de datos PSI puede aparecer en cada uno de los trenes de transporte en un sistema. Debe haber siempre una versión completa de la tabla de asociación de programas que numere todos los programas dentro del tren de transporte y una tabla de correspondencia de programas TS completa que contenga definiciones de programas completos para todos los programas dentro del tren de transporte. Si algunos trenes están aleatorizados, debe haber también una tabla de acceso condicional que presente las listas de los trenes de mensaje de gestión de títulos (EMM, *entitlement management message*) pertinentes. La presencia de una NIT es completamente facultativa.

Las tablas PSI corresponden con paquetes del tren de transporte mediante la estructura de sección descrita anteriormente. Cada sección tiene un campo `table_id` en su encabezamiento, que permite mezclar secciones de tablas PSI y datos privados de secciones privadas en paquetes del tren de transporte del mismo valor de PID o incluso en el mismo paquete del tren de transporte. Obsérvese sin embargo, que dentro de paquetes del mismo PID se debe transmitir una sección completa antes de que pueda comenzar la siguiente sección. Esto sólo es posible para paquetes que según la etiqueta contienen sección de tabla de correspondencia y de programas TS o paquetes NIT, porque las secciones privadas no pueden corresponder con paquetes PAT o CAT.

Se requiere que todas las secciones PAT correspondan con paquetes del tren de transporte con PID = 0x0000 y todas las secciones CA correspondan con paquetes con PID = 0x0001. Las secciones PMT pueden corresponder con paquetes de valor de PID seleccionados por el usuario, enumerado como el PMT_PID para cada programa en la tabla de asociación de programas. De manera similar, el PID para la NIT que lleva paquetes de tren de transporte es seleccionado por el usuario pero debe ser señalado por la entrada "program_number == 0x00" en la PAT, si existe la NIT.

El contenido de cualesquiera trenes de parámetros CA es enteramente privado, pero EMM y ECM deben enviarse también en paquetes del tren de transporte para cumplir esta Recomendación | Norma Internacional.

Las tablas de datos privados se pueden enviar utilizando la sintaxis de sección privada. Estas tablas se podrán emplear, por ejemplo, en un entorno de radiodifusión para describir un servicio, un evento futuro, horarios de radiodifusión e información conexas.

C.8 Relaciones de estructuras PSI

La figura C.1 muestra un ejemplo de la relación entre las cuatro estructuras PSI y el tren de transporte. Son posibles otros ejemplos, pero la figura muestra las conexiones primarias.

En las siguientes subcláusulas se describe cada tabla PSI.

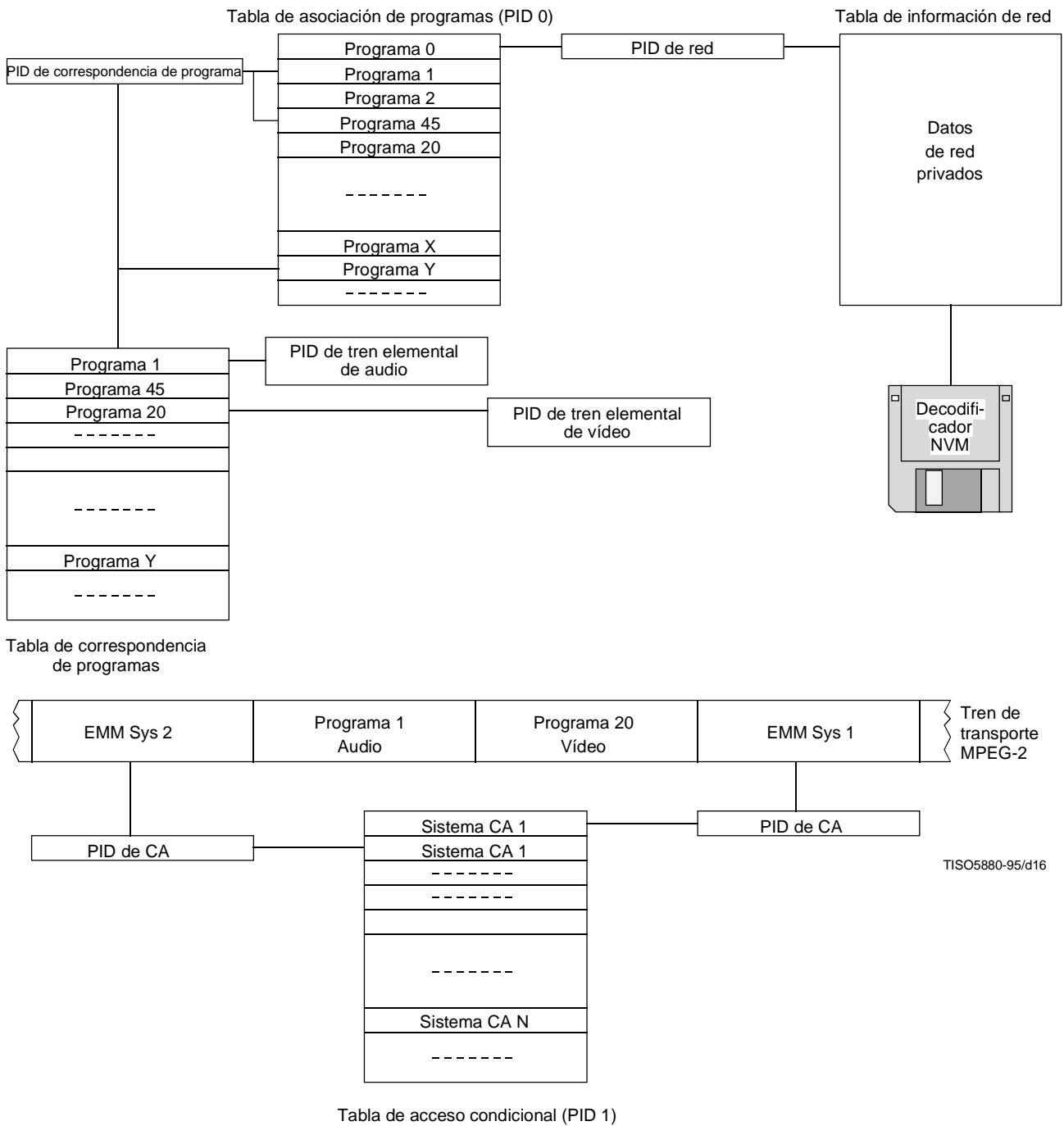


Figura C.1 – Relaciones de correspondencia de programa y red

C.8.1 Tabla de asociación de programas

Cada tren de programa debe contener una tabla de asociación de programas válida completa. La tabla de asociación de programas da la correspondencia entre un número de programa y el PID de los paquetes del tren de transporte que lleva la definición de ese programa (el PMT_PID). La PAT puede ser dividida en hasta 255 secciones antes de establecer la correspondencia con los paquetes del tren de transporte. Cada sección transporta una parte de la PAT global. Esta partición puede ser conveniente para minimizar la pérdida de datos en condiciones de error. Es decir, la pérdida de paquetes o los bits erróneos se pueden localizar en secciones más pequeñas de la PAT permitiendo así que otras secciones se reciban y decodifiquen correctamente. Si toda la información de PAT se pone en una sección, un error que origine un bit cambiado en el ID en el identificador de tabla, por ejemplo, causaría la pérdida de toda la PAT. No obstante, esto se permite mientras la sección no rebase el límite de longitud máximo de 1024 bytes.

ISO/CEI 13818-1:2007 (S)

El programa 0 (cero), está reservado y se utiliza para especificar el PID de red. Es un puntero de los paquetes del tren de transporte que llevan la tabla de información de red.

La tabla de asociación de programas se transmite siempre sin criptación.

C.8.2 Tabla de correspondencia de programas

La tabla de correspondencia de programas proporciona la correspondencia entre un número de programa y los elementos de programa que lo forman. Esta tabla está presente en paquetes del tren de transporte que tienen uno o más valores de PID seleccionados privadamente. Estos paquetes del tren de transporte pueden contener otras estructuras privadas definidas por el campo de identificador de tabla. Es posible tener secciones PMT de TS referentes a diferentes programas transmitidos en paquetes del tren de transporte que tienen un valor PID común.

La presente Recomendación | Norma Internacional requiere un mínimo de identificación de programa: número de programa, PCR PID, tipos de tren y PID de elementos de programa. Se puede transportar información adicional para programas o trenes elementales mediante la utilización de construcción de descriptores. Véase C.8.6.

Los datos privados se pueden enviar también en paquetes del tren de transporte que indiquen que transportan secciones de la tabla de correspondencia de programas TS. Esto se efectúa mediante la utilización de la sección privada. En una sección privada, la aplicación decide si el número de versión y el indicador de siguiente vigente representan los valores de estos campos para una sección o si son aplicables a muchas secciones como parte de una tabla privada más grande.

NOTA 1 – Los paquetes del tren de transporte que contienen la tabla de correspondencia de programa se transmiten sin criptación.

NOTA 2 – Es posible transmitir información sobre eventos en descriptores privados transportados dentro de `TS_program_map_section()`.

C.8.3 Tabla de acceso condicional

La tabla de acceso condicional (CA) da la asociación entre uno o más sistemas CA, sus trenes EMM y cualesquiera parámetros especiales asociados con ellos.

NOTA – El contenido (privado) de los paquetes del tren de transporte que contienen parámetros EMM y CA, si están presentes, estará en general, criptados (aleatorizado).

C.8.4 Tabla de información de red

El contenido de la tabla de información de red es privado y no se especifica en esta Recomendación | Norma Internacional. En general, contendrá la correspondencia de servicios seleccionados por el usuario con identidades de tren de transporte, frecuencias de canal, número de transpondedor de satélite, características de modulación, etc.

C.8.5 Sección privada

La sección privada puede aparecer en dos formas básicas, la versión más corta (donde sólo están presentes los campos hasta `section_length inclusive`) o la versión larga (donde todos los campos hasta `last_section_number inclusive` están presentes y después de los bytes de datos privados está presente el campo `CRC_32`).

Las secciones privadas pueden aparecer en PID que están etiquetados como `PMT_PID` o en paquetes del tren de transporte con otros valores de PID que contienen exclusivamente secciones privadas, incluido el PID asignado a la NIT. Si los paquetes de tren de transporte del PID que transporta las secciones privadas son identificados como un PID que transporta secciones privadas (valor de asignación de tipo de tren `0x05`), sólo pueden aparecer secciones privadas en paquetes del tren de transporte de ese valor de PID. Las secciones pueden ser de tipo corto o largo.

C.8.6 Descriptores

Hay varios descriptores normativos definidos en esta Recomendación | Norma Internacional. Es posible también definir muchos más descriptores privados. Todos los descriptores tienen un formato común: {rótulo, longitud, datos}. Cualquiera descriptores definidos privadamente deben observar este formato. La porción de datos de estos descriptores privados se define privadamente.

Un descriptor (el descriptor CA) se utiliza para indicar el lugar (valor de PID de paquetes de transporte) de datos ECM asociados con elementos de programa cuando se encuentra en una sección TS PMT. Cuando se encuentra en una sección CA, se refiere a los EMM.

Para ampliar el número de descriptores privados disponibles, se podrá utilizar el siguiente mecanismo: se podrá definir privadamente un `descriptor_tag` (rótulo de descriptor) privado definido para ser construido como un descriptor compuesto. Esto entraña la definición privada de otro subdescriptor como el primer campo de los bytes de datos privados del descriptor privado. La estructura descrita se indica en los cuadros C.1 y C.2.

Cuadro C.1 – Descriptor compuesto

Sintaxis	N.º de bits	Mnemónico
<pre> Composite_descriptor(){ descriptor_tag(privately defined) descriptor_length for (i = 0; i < N; i++){ sub_descriptor() } } </pre>	<p style="text-align: center;">8</p> <p style="text-align: center;">8</p>	<p style="text-align: center;">uimsbf</p> <p style="text-align: center;">uimsbf</p>

Cuadro C.2 – Subdescriptor

Sintaxis	N.º de bits	Mnemónico
<pre> sub_descriptor() { sub_descriptor_tag sub_descriptor_length for (i = 0; i < N; i++) { private_data_byte } } </pre>	<p style="text-align: center;">8</p> <p style="text-align: center;">8</p> <p style="text-align: center;">8</p>	<p style="text-align: center;">uimsbf</p> <p style="text-align: center;">uimsbf</p> <p style="text-align: center;">uimsbf</p>

C.9 Utilización de anchura de banda y tiempo de adquisición de señal

Todas las implementaciones de un tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 deben tener exigencias de anchura de banda razonables para información PSI y, en aplicaciones en las que se considera el acceso aleatorio, deben facilitar una adquisición rápida de la señal. Esta subcláusula analiza este aspecto y da algunos ejemplos de aplicaciones en radiodifusión.

Como el tren de transporte está basado en paquetes, tiene en cuenta la dispersión de la información PSI con fina granularidad en los datos multiplexados. Esto proporciona una flexibilidad importante en la construcción y transmisión de PSI.

El tiempo de adquisición de la señal en un decodificador real depende de muchos factores, que comprenden: tiempo de sintonización FDM, tiempo de demultiplexación, encabezamiento de secuencia, velocidad de aparición de tramas I y extracción y procesamiento de claves de aleatorización.

En esta subcláusula se examinan las repercusiones de la velocidad binaria y del tiempo de adquisición de señal de la sintaxis PSI (véanse 2.4.4.4 y 2.4.4.9). Se supone que la tabla de acceso condicional no tiene que ser recibida dinámicamente en cada cambio de programa. Se supone lo mismo para los trenes EMM privados, porque estos trenes no contienen los componentes ECM que varían rápidamente que son utilizados para la aleatorización de elementos de programa (criptación).

En el siguiente análisis, se ha despreciado también el tiempo para adquirir y procesar mensajes de permiso de control (ECM).

Los cuadros C.3 y C.4 proporcionan valores de utilización de anchura de banda para una gama de condiciones de trenes de transporte. Un eje de la tabla es el número de programas contenidos en un tren de transporte. El otro eje es la frecuencia con la cual se transmite la información PSI en el tren de transporte.

**Cuadro C.3 – Utilización de anchura de banda de la tabla de asociación de programas (bit/s)
Número de programas por tren de transporte**

Frecuencia de información de la tabla PA (s ⁻¹)		1	5	10	32	128
	1	1504	1504	1504	1504	4512
	10	15040	15040	15040	15040	45120
	25	37600	37600	37600	37600	112800
	50	75200	75200	75200	75200	225600
	100	150400	150400	150400	150400	451200

NOTA – Como 46 secciones de asociación de programas entran en un paquete, los números en el cuadro no cambian hasta la última columna.

**Cuadro C.4 – Utilización de anchura de banda de la tabla de correspondencia de programas (bit/s)
Número de programas por tren de transporte**

Frecuencia de información de la tabla PM (s ⁻¹)		1	5	10	32	128
	1	1504	1504	3008	7520	28576
	10	15040	15040	30080	75200	285760
	25	37600	37600	75200	188000	714400
	50	75200	75200	150400	376000	1428800
	100	150400	150400	300800	601600	2857600

Esta frecuencia será una clave determinante del componente de tiempo de adquisición de la señal debido a estructuras PSI.

En ambos cuadros de utilización de anchura de banda se supone que sólo se proporciona la información de correspondencia de programas mínima. Esto significa que los valores de PID y tipos de trenes se proporcionan sin descriptores adicionales. Todos los programas del ejemplo están compuestos por dos trenes elementales. Las asociaciones de programas tienen una longitud de 2 bytes, mientras que la correspondencia de programas mínima tiene una longitud de 26 bytes. Hay una tara adicional asociada con números de versión, longitudes de sección, etc. Esta será del orden de 1-3% de la utilización de la velocidad binaria PSI total en secciones de longitud moderada a máxima (de algunos centenares de bytes a 1024 bytes) por lo que se pasarán por alto en este caso.

Las anteriores hipótesis permiten la correspondencia de 46 asociaciones de programas con un paquete de transporte de tabla de asociación de programas (si no está presente el campo de adaptación). De manera similar, siete secciones de correspondencia de programas TS entran en un paquete de transporte. Cabe señalar que para proporcionar la "supresión/adición" fácil, es posible transmitir solamente una sección de correspondencia de programas TS por cada PMT_PID. No obstante, esto puede originar un aumento indeseable de la utilización de velocidad binaria PSI.

La utilización de una frecuencia de 25 Hz para las dos tablas PSI da una contribución del caso más desfavorable al tiempo de adquisición de la señal de aproximadamente 80 ms. Esto sólo ocurriría cuando los datos PAT requeridos "faltasen" y después, una vez adquirida y decodificada la PAT, los datos PMT requeridos "faltasen" también. Esta duplicación del tiempo de adquisición del caso más desfavorable es una desventaja del nivel suplementario introducido por la estructura PAT. Este efecto podrá ser reducido por la transmisión coordinada de paquetes PAT y PMT conexos. Se supone que la ventaja que este método ofrece para "suprimir/añadir" operaciones de remultiplexación es compensatoria.

Con la frecuencia PSI de 25 Hz, se pueden construir los siguientes ejemplos (todos los ejemplos dejan un amplio margen para distintos enlaces de datos, corrección de errores sin canal de retorno, CA y taras de encaminamiento):

Canal CATV de 6 MHz

- cinco programas de 5,2 Mbit/s: 26,5 Mbit/s (incluye tara de transporte)
 - anchura de banda PSI total: 5,2 kbit/s
 - anchura de banda CA: 500 kbit/s
- anchura de banda de transporte total de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 27,1 Mbit/s*
- tara PSI: 0,28%

Canal de fibra OC-3 (155 Mbit/s)

- 32 programas de 3,9 Mbit/s: 127,5 Mbit/s (incluye tara de transporte)
- anchura de banda PSI total: 225,6 kbit/s
- anchura de banda CA: 500 kbit/s
anchura de banda de transporte total de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 128,2 Mbit/s
- tara PSI: 0,18%

Transpondedor de satélite banda C

- 128 programas de audio de 256 kbit/s: 33,5 Mbit/s (incluye tara de transporte)
- anchura de banda PSI total: 826,4 kbit/s
- anchura de banda CA: 500 kbit/s
anchura de banda de transporte total de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1: 34,7 Mbit/s
- tara PSI: 2,4% (realmente sería más baja si se utilizase sólo un PID por programa)

Según lo previsto, el porcentaje de tara aumenta para servicios de velocidad más baja, porque son posibles muchos más servicios por tren de transporte. Sin embargo, en todos los casos la tara no es excesiva. Se pueden utilizar velocidades de transmisión más altas (que 25 Hz) para los datos PSI con miras a disminuir la repercusión en el tiempo de adquisición de canal con un aumento muy modesto de velocidad binaria.

Anexo D**Modelo de temporización de sistemas de la presente Recomendación | Norma Internacional y repercusiones de aplicación**

(Este anexo no es parte integrante de la presente Recomendación | Norma Internacional)

D.0 Introducción

La especificación de sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 incluye un modelo de temporización específico para el muestreo, codificación, almacenamiento en memoria tampón del codificador, transmisión, recepción, almacenamiento en memoria tampón del decodificador, decodificación y presentación de audio y vídeo digital en combinación. Este modelo se ha materializado directamente en la especificación de los requisitos de sintaxis y semántica de los trenes de datos conformes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Dado que un sistema de decodificación recibe un tren de bits conforme que es entregado correctamente de acuerdo con el modelo de temporización, es obvio implementar el decodificador de modo que produzca como salida audio y vídeo de alta calidad propiamente sincronizados. Sin embargo, no hay un requisito normativo de que los decodificadores se implementen de modo que proporcionen esta salida de presentación de alta calidad. En las aplicaciones en las que los datos no son entregados al decodificador con temporización correcta, puede ser posible producir la salida de presentación deseada, aunque estas capacidades no están garantizadas en general. Este anexo informativo describe el modelo de temporización de sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 en detalle, y da algunas sugerencias para implementar sistemas de decodificadores adaptados a algunas aplicaciones típicas.

D.0.1 Modelo de temporización

Los sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 materializan un modelo de temporización en el cual todas las imágenes y muestras de audio digitalizadas que entran en el decodificador son presentadas exactamente una vez cada una, después de un retardo constante de extremo a extremo, a la salida del decodificador. Como tal, las velocidades de muestra, es decir, la velocidad de la trama vídeo y la velocidad de la muestra de audio son precisamente las mismas en el decodificador que en el codificador. En la figura D.1 se presenta un diagrama de este modelo de temporización.

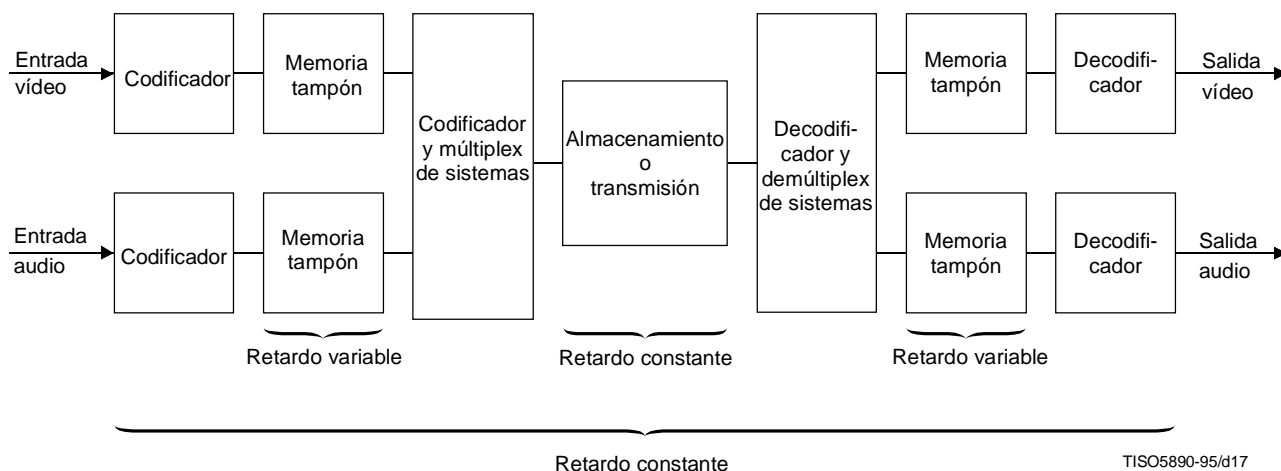


Figura D.1 – Modelo con retardo constante

Como se indica en la figura D.1, el retardo desde la entrada en el decodificador hasta la salida o presentación del decodificador es constante en este modelo¹, si bien el retardo a través de cada memoria tampón del codificador y del decodificador es variable. No sólo el retardo a través de cada una de estas memorias tampón es variable dentro del trayecto de un tren elemental, sino que difieren también los retardos de cada memoria tampón en los trayectos de vídeo y audio. Por consiguiente, la ubicación relativa de los bits codificados que representan audio o vídeo en el tren combinado no indica información de sincronización. La ubicación relativa de audio y vídeo codificados está constreñida solamente por el modelo del decodificador objetivo de sistema (STD) de modo que las memorias tampón del decodificador deben comportarse adecuadamente. Por consiguiente, el audio y vídeo codificados que representan sonido e imágenes que han de ser presentados simultáneamente pueden estar separados temporalmente hasta un segundo dentro del tren de bits codificado, que es el retardo máximo de la memoria tampón del codificador autorizado en el modelo STD.

Las velocidades de muestras de audio y de vídeo en el codificador son significativamente diferentes entre sí, y pueden o no tener una relación exacta y fija entre sí, dependiendo de si el tren combinado es un tren de programa o un tren de transporte y de si en el tren de programa están puestas las banderas `System_audio_locked` y `System_video_locked` (audio de sistema enganchado y vídeo de sistema enganchado). La duración de un bloque de muestras de audio (una unidad de presentación de audio) generalmente no es igual que la duración de una imagen vídeo.

Hay un solo reloj de sistema común en el codificador, y este reloj se utiliza para crear indicaciones de tiempo que muestran la temporización correcta de presentación y de codificación de audio y vídeo, así como para crear indicaciones de tiempo que señalan los valores instantáneos del propio reloj de sistema a intervalos muestreados. Las indicaciones de tiempo del tiempo de presentación de audio y vídeo se denominan indicaciones de tiempo de presentación (PTS); las que indican el tiempo de decodificación se denominan indicaciones de tiempo de decodificación (DTS) y las que indican el valor del reloj de sistema se denominan la referencia de reloj de sistema (SCR) en trenes de programa, y la referencia de reloj de programa (PCR) en trenes de transporte. La presencia de este reloj de sistema común en el codificador, las indicaciones de tiempo que son creadas a partir del mismo y la recreación del reloj en el decodificador así como la utilización correcta de las indicaciones de tiempo proporcionan la facilidad de sincronizar adecuadamente el funcionamiento del decodificador.

Las implementaciones de codificador pueden no seguir este modelo exactamente, aunque el tren de datos resultante del codificador real, sistema de almacenamiento, red y uno o más multiplexores debe seguir el modelo con precisión. (La entrega de los datos puede desviarse algo, según la aplicación.) Por consiguiente, en este anexo el término "reloj de sistema del codificador" se utiliza para indicar el reloj de sistema común real descrito en este modelo o la función equivalente, lo que pueda estar implementado.

Como el retardo de extremo a extremo a través de todo el sistema es constante, las presentaciones de audio y vídeo están sincronizadas precisamente. La construcción de trenes de bits de sistemas está constreñida de modo que cuando son decodificados por un decodificador que sigue este modelo con memorias tampón de decodificador de tamaño

¹ Se requiere el retardo constante indicado para todo el sistema para la sincronización correcta; sin embargo, son posibles algunas desviaciones. El retardo de red se analiza como constante; se pueden admitir ligeras desviaciones y la adaptación de red puede permitir variaciones mayores de retardo de red. Ambos aspectos se examinan posteriormente.

apropiado se garantiza que estas memorias tampón nunca estarán desbordadas ni subutilizadas, con excepciones específicas que permiten la subutilización intencional.

Para que el sistema del decodificador produzca la cantidad de retardo precisa que hace que todo el retardo de extremo a extremo sea constante, es necesario que el decodificador tenga un reloj de sistema cuya frecuencia de trabajo y valor instantáneo absoluto concuerden con los del codificador. La información necesaria para transmitir el reloj de sistema del codificador está codificada en la SCR o PCR; esta función se explica a continuación.

Los decodificadores que se implementan de acuerdo con este modelo de temporización de manera que presentan muestras de audio e imágenes de vídeo exactamente una vez (con excepciones específicas codificadas intencionalmente), a una velocidad constante, y cuyas memorias tampón se comportan como en el modelo, se denominan en este anexo temporizadores precisamente temporizados, o que producen salida precisamente temporizada. Según esta Norma Internacional, las implementaciones de decodificadores no tienen que presentar audio y vídeo de acuerdo con este modelo, por lo que es posible construir decodificadores que no tengan retardo constante, o equivalentemente, no presenten cada imagen o muestra de vídeo exactamente una vez. Sin embargo, en tales implementaciones, la sincronización entre el audio y vídeo presentados puede no ser precisa, y el comportamiento de las memorias tampón del decodificador pueden no seguir el modelo de decodificador de referencia; es importante evitar el desbordamiento en las memorias tampón del decodificador, porque el desbordamiento origina una pérdida de datos que puede tener efectos importantes sobre el proceso de decodificación resultante. Este anexo trata principalmente del funcionamiento de estos decodificadores precisamente temporizados y de alguna de las opciones disponibles al implementar estos decodificadores.

D.0.2 Sincronización de presentación de audio y vídeo

Dentro de la codificación de los datos de sistemas de la presente Recomendación | Norma Internacional hay indicaciones de tiempo relativas a la presentación y decodificación de imágenes vídeo y bloques de muestras de audio. Las imágenes y bloques se denominan "unidades de presentación", PU abreviadamente. Los conjuntos de bits codificados que representan las PU y que se incluyen dentro del tren de bits de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se denominan "unidades de acceso", o AU abreviadamente. La abreviatura de una unidad de acceso de audio es AAU y la de una unidad de acceso de vídeo es VAU. En ISO/CEI 13818-3, el término "trama de audio" tiene el mismo significado que AAU o unidad de presentación de audio (APU, *audio presentation unit*) según el contexto. Una unidad de presentación de vídeo (VPU, *video presentation unit*) es una imagen, y una VAU es una imagen codificada.

Algunas AAU y VAU, pero no necesariamente todas, tienen PTS asociadas con ellas. Una PTS indica el tiempo en que la PU resultante de la decodificación de la AU que está asociada con la PTS debe ser presentada al usuario. Las PTS de audio y las PTS de vídeo son muestras de un reloj de tiempo común, que se denomina reloj de tiempo del sistema o STC. Con los valores correctos de PTS de audio y vídeo incluidos en el tren de datos, y con la presentación de las PU de audio y de vídeo en el momento indicado por las PTS apropiadas desde el punto de vista de STC común, se logra la sincronización precisa del audio y vídeo presentados en el sistema de decodificación. Si bien el STC no forma parte del contenido normativo de esta Recomendación | Norma Internacional, y la información equivalente se indica en esta Recomendación | Norma Internacional mediante términos tales como frecuencia de reloj de sistema, el STC es un elemento importante y conveniente para explicar el modelo de temporización y en general es práctico para implementar codificadores y decodificadores que incluyen un STC de alguna forma.

Las PTS se requieren para el transporte de temporización relativa exacta entre audio y vídeo, porque las PU de audio y vídeo en general tienen duraciones considerablemente diferentes y no relacionadas esencialmente. Por ejemplo, las PU de audio de 1152 muestras cada una a una velocidad de muestra de 44 100 muestras por segundo tienen una duración de aproximadamente 26,12 ms, y las PU vídeo a una velocidad de trama de 29,97 Hz tienen una duración de aproximadamente 33,76 ms. En general, las fronteras temporales de las APU y de las VPU no coinciden casi nunca. Por tanto, PTS separadas para audio y vídeo proporcionan la información que indica la relación temporal precisa de las PU de audio y vídeo sin requerir ninguna relación específica entre la duración e intervalo de las PU de audio y vídeo.

Los valores de los campos PTS se definen desde el punto de vista del decodificador-objetivo de sistema (o STD) que es una construcción normativa fundamental en todos los trenes de bits de sistema. El STD es un modelo matemático de un codificador ideal que especifica precisamente el movimiento de todos los bits dentro y fuera de las memorias tampón del decodificador, y la construcción semántica básica impuesta al tren de bits es que las memorias tampón dentro del STD nunca deben estar desbordadas ni subutilizadas, con excepciones específicas proporcionadas para la subutilización en casos especiales. En el modelo STD, el decodificador virtual está siempre sincronizado exactamente con la fuente de datos, y la decodificación y presentación de audio y vídeo están sincronizadas exactamente. Aunque el STD es exacto y coherente, se ha simplificado algo con respecto a las implementaciones físicas de decodificadores para aclarar su especificación y facilitar su amplia aplicación a una variedad de implementaciones de decodificador. En particular, en el modelo STD cada una de las operaciones realizadas en el tren de bits en el decodificador se realiza instantáneamente, con la excepción obvia del tiempo que emplean los bits en las memorias tampón del decodificador. En un sistema de decodificador real, los decodificadores de audio y vídeo no funcionan simultáneamente, y hay que tener en cuenta sus retardos en el diseño de la implementación. Por ejemplo, si se decodifican imágenes de vídeo exactamente en un

intervalo de presentación de una imagen I/P, donde P es la velocidad de imagen, y los datos de vídeo comprimido están llegando al decodificador a una velocidad binaria R, la compleción de suprimir los bits asociados con cada imagen se retarda con respecto al tiempo indicado en los campos PTS y DTS por I/P, y la memoria tampón del decodificador vídeo debe ser mayor que la especificada en el modelo STD por R/P. Es probable que la presentación de vídeo se retrarde con respecto al STD, y la PTS debe ser tratada en consecuencia. Como el vídeo se retarda, la decodificación y presentación de audio debe retardarse por un periodo similar para proporcionar la sincronización correcta. El retardo de la decodificación y presentación de audio y vídeo en un decodificador se puede implementar, por ejemplo, añadiendo una constante a los valores PTS cuando se utilizan dentro del decodificador.

Otra diferencia entre el STD y una implementación real de decodificador preciso es que en el modelo STD se supone explícitamente que la salida final de audio y vídeo se presenta al usuario instantáneamente y sin ulterior demora. Este puede no ser el caso en la práctica, en particular con pantallas de rayos de tubos catódicos, y este retardo adicional debe tenerse en cuenta en el diseño. Los codificadores tienen que codificar el vídeo de modo que se logre la sincronización correcta cuando los datos son decodificados con el STD. Los retardos en la entrada y muestreo de audio y vídeo, tales como la integración de carga óptica de la cámara vídeo, deben tenerse en cuenta en el codificador.

En el modelo STD se supone una sincronización apropiada y las indicaciones de tiempo y comportamiento de la memoria tampón se prueban con respecto a esta hipótesis como una condición de la validez del tren de bits. Naturalmente, en un decodificador físico, la sincronización precisa no se realiza automáticamente, en particular al comienzo y en presencia de fluctuación de fase de temporización. La temporización precisa del decodificador es un objetivo al que deben tender los diseños de decodificador. La inexactitud en la temporización del decodificador afecta al comportamiento de las memorias tampón del decodificador. Estos temas se tratan más detalladamente en otras subcláusulas de este anexo.

El STD incluye campos DTS y PTS. La DTS se refiere al tiempo en que una AU ha de ser extraída de la memoria tampón del decodificador y decodificada en el modelo STD. Como los decodificadores de trenes elementales de audio y vídeo son instantáneos en el STD, el tiempo de decodificación y presentación son idénticos en la mayoría de los casos; la única excepción se produce con imágenes vídeo que han sido reordenadas dentro del tren de bits codificado, es decir, imágenes I y P en el caso de secuencias vídeo sin bajo retardo. Cuando ha habido esta reordenación, la memoria tampón de retardo temporal en el decodificador vídeo se utiliza para almacenar las imágenes I o P decodificadas apropiadas hasta que sean presentadas. En todos los casos, cuando los tiempos de decodificación y presentación son idénticos en el STD, es decir, todas las AAU, VAU de imágenes B y VAU de imágenes I y P en secuencias vídeo de bajo retardo, la DTS no está codificada, como si tuviera el mismo valor que la PTS. Cuando los valores difieren, ambos están codificados si cualquiera de los dos está codificado. Para todas las AU en las que sólo la PTS está codificada, este campo se puede interpretar como ambos campos PTS y DTS.

Como no se necesitan valores de PTS y de DTS para cada AAU y VAU, el decodificador puede elegir interpolar valores que no están codificados. Se requieren valores de PTS con intervalos que no excedan de 700 ms en cada tren elemental de audio y vídeo. Estos intervalos de tiempo se miden en tiempo de presentación, es decir, en el mismo contexto que los valores de los campos, no en términos de los tiempos en que se transmiten y reciben los campos. En los casos de trenes de datos en los que los relojes de audio y vídeo y del sistema están enganchados, como se define en la parte normativa de esta Recomendación | Norma Internacional, cada AU siguiente, para la cual una DTS o PTS está específicamente codificada, tiene un tiempo de decodificación efectivo que es la suma de la anterior AU más una diferencia especificada y fija en valor del STC. Por ejemplo, en vídeo codificado a 29,97 Hz, cada imagen tiene una diferencia en tiempo de 3003 ciclos de la porción de 90 kHz del STC con respecto a la imagen anterior cuando los relojes de vídeo y de sistema están enganchados. La misma relación temporal existe para decodificar las AU sucesivas, aunque el retardo de reordenación en el decodificador afecta a la relación entre las AU del decodificador y las PU presentadas. Cuando el tren de datos está codificado de modo que el reloj de audio o vídeo no está enganchado al reloj de sistema, la diferencia de tiempo entre la decodificación de las AU sucesivas se puede estimar utilizando los mismos valores indicados anteriormente; sin embargo, estas diferencias de tiempo no son exactas debido a que las relaciones entre la velocidad de trama, la velocidad de muestra de audio y la frecuencia del reloj de sistema no han sido exactas en el codificador.

Obsérvese que los campos PTS y DTS no indican, por sí mismos, la ocupación correcta de las memorias tampón del decodificador al comienzo ni en ningún otro momento, y de manera equivalente, no indican el retardo de tiempo que debe transcurrir al recibir los bits iniciales de un tren de datos antes que comience la decodificación. Esta información es extraída combinando las funciones de los campos PTS y DTS y la recuperación de reloj correcta, que se examina a continuación. En el modelo STD y, por tanto, en los decodificadores realizados según este modelo, el comportamiento de la memoria tampón del decodificador es determinada completamente por los valores SCR (o PCR), los tiempos en que son recibidos, y los valores PTS y DTS, suponiendo que los datos se entregan de acuerdo con el modelo de temporización. Esta información especifica el tiempo que los datos codificados pasan en las memorias tampón del decodificador. La cantidad de datos que está en las memorias tampón de datos codificados no se especifica explícitamente, y esta información no es necesaria, porque la temporización se especifica completamente. Obsérvese también que la ocupación de las memorias tampón de datos puede variar considerablemente en función del tiempo, de manera que no puede ser previsible por el decodificador, salvo mediante la utilización adecuada de las indicaciones de tiempo.

Para que las PTS de audio y vídeo se refieran correctamente a un STC común, se debe disponer de un reloj común correctamente temporizado dentro del sistema decodificador. Esto se trata en la siguiente subcláusula.

D.0.3 Recuperación del reloj de tiempo de sistema en el decodificador

En los trenes de datos de sistemas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 hay, además de los campos PTS y DTS, indicaciones de tiempo de referencia de reloj. Estas referencias son muestras del reloj de tiempo de sistema, aplicables al decodificador y al codificador. Tienen una resolución de una parte en 27 000 000 por segundo, y se producen a intervalos de hasta 100 ms en trenes de transporte o de hasta 700 ms en trenes de programa. Por tanto, se pueden utilizar para implementar bucles de control de reconstrucción de reloj en los decodificadores con precisión suficiente para todas las aplicaciones identificadas.

En el tren de programa, el campo de referencia de reloj se denomina la referencia de reloj de sistema (SCR). En el tren de transporte, el campo de referencia de reloj se denomina la referencia de reloj de programa (PCR). En general, se puede considerar que las definiciones de SCR y PCR son equivalentes, aunque hay distinciones. En el resto de esta subcláusula se utiliza el término SCR para mayor claridad; las mismas explicaciones se aplican a la PCR, salvo cuando se indique otra cosa. La PCR en trenes de transporte proporciona la referencia de reloj para un programa, donde un programa es un conjunto de trenes elementales que tienen una base de tiempos común y están destinados para decodificación y presentación sincronizadas. Puede haber múltiples programas en un tren de transporte y cada uno puede tener una base de tiempos independiente y un conjunto separado de PCR.

El campo SCR indica el valor correcto del STC cuando la SCR se recibe en el decodificador. Como el SCR ocupa más de un byte de datos, y los trenes de datos de sistemas se definen como trenes de bytes, la SCR se define para llegar al decodificador cuando se recibe el último byte del `system_clock_reference_base` en el decodificador. Como otra posibilidad, la SCR puede ser interpretada como el tiempo en que el campo SCR debe llegar al decodificador, suponiendo que ya se sabe que el STC es correcto. La interpretación que se ha de utilizar depende de la estructura del sistema de aplicación. En las aplicaciones en las que la fuente de datos puede ser controlada por el decodificador, como un DSM conectado localmente, es posible que el decodificador tenga una frecuencia STC autónoma, y que el STC no tenga que ser recuperado. Sin embargo, en muchas aplicaciones importantes, esta hipótesis no se puede aplicar correctamente. Por ejemplo, considérese el caso cuando un tren de datos es entregado simultáneamente a múltiples decodificadores. Si cada decodificador tiene su propio STC autónomo con su propia frecuencia de reloj independiente, no se puede asegurar que las SCR lleguen en el momento correcto a todos los decodificadores; un decodificador necesitará en general las SCR más pronto que la fuente las entrega, mientras que otro las necesitará más tarde. Esta diferencia no se puede establecer con una memoria tampón de datos de tamaño finito en una cantidad de tiempo limitada de recepción de datos. Por tanto, a continuación se examina precisamente el caso cuando el STC debe subordinar su temporización a las SCR (o PCR) recibidas.

En un tren de datos de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 correctamente construido y entregado, cada SCR llega al decodificador precisamente en el instante indicado por el valor de esa SCR. En este contexto, "tiempo" significa valor correcto del STC. En teoría, este valor STC es el mismo valor que tenía el STC del codificador cuando se almacenó o transmitió la SCR. Sin embargo, es posible que la codificación no se haya realizado en tiempo real o que el tren de datos haya sido modificado desde que se codificó originalmente, y en general el codificador o fuente de datos puede estar implementado de diversas maneras, de modo que el STC del codificador puede ser una magnitud teórica.

Si la frecuencia del reloj del decodificador concuerda exactamente con la del codificador, la decodificación y presentación de audio y vídeo tendrá automáticamente la misma velocidad que en el codificador, y el retardo de extremo a extremo será constante. Con frecuencias de reloj de codificador y decodificador concordantes, se puede utilizar cualquier valor SCR correcto para fijar el valor instantáneo del STC del decodificador y a partir de esto, el tiempo en el STC del decodificador concordará con el del codificador sin necesidad de otro ajuste. Esta condición es verdadera hasta que haya una discontinuidad de temporización, como el fin de un tren de programa o la presencia de un indicador de discontinuidad en el tren de transporte.

En la práctica, una frecuencia de reloj de sistema del decodificador que funciona libremente no concordará con la frecuencia de reloj de sistema del codificador muestreada e indicada en los valores SCR. Se puede hacer que el STC del decodificador subordine su temporización al codificador utilizando las SRC recibidas. El método de prototipo de subordinación del reloj del decodificador al tren de datos recibidos se efectúa mediante un bucle enganchado en fase (PLL, *phase-locked loop*). Las variaciones de un bucle enganchado en fase, básico, u otros métodos, pueden ser apropiados, según las necesidades de la aplicación específica.

A continuación se muestra en diagrama y se describe un bucle enganchado en fase directo que recupera el STC en un decodificador.

La figura D.2 muestra un bucle enganchado en fase clásico, salvo que los términos de referencia y realimentación son números (valores STC y SCR o PCR) en vez de eventos de señales tales como bordes.

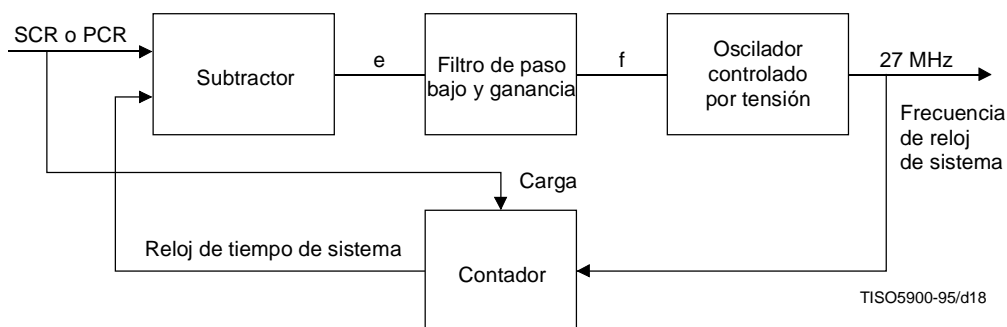


Figura D.2 – Recuperación del STC mediante un bucle enganchado en fase (PLL)

Después de la adquisición inicial de una nueva base de tiempos, es decir, un nuevo programa, el STC se fija al valor vigente codificado en las SCR. Típicamente, la primera SCR se carga directamente en el contador STC y el bucle enganchado en fase se aplica subsiguientemente como un bucle cerrado. Puede ser apropiado efectuar variaciones de este método, es decir, cuando los valores de las SCR son sospechosos debido a fluctuación de fase o errores.

La acción de bucle cerrado del bucle enganchado de fase es la siguiente. En el momento en que cada SCR (o PCR) llega al decodificador, el valor se compara con el valor vigente del STC. La diferencia es un número, que tiene una parte en unidades de 90 kHz y una parte en términos de 300 veces esta frecuencia, es decir, 27 MHz. El valor de diferencia se hace lineal para que sea un espacio de número, típicamente unidades de 27 MHz, y se denomina "e", el término error del bucle. La secuencia de términos e se introduce en el filtro de paso bajo y la etapa de ganancia, que se diseñan de acuerdo con las necesidades de la aplicación. La salida de esta etapa es una señal de control "f" que controla la frecuencia instantánea del oscilador controlado por tensión (VCO, *voltage controlled oscillator*). La salida del VCO es una señal de oscilador con una frecuencia nominal de 27 MHz; esta señal se utiliza como la frecuencia de reloj de sistema dentro del decodificador. El reloj de 27 MHz se introduce en un contador que produce los valores STC vigentes, que consisten en una extensión de 27 MHz, producida por la división por 300 y un valor básico de 90 kHz que se deriva calculando los resultados de los 90 kHz en un contador de 33 bits. La porción de 90 kHz, de 33 bits, de la salida del STC se utiliza según sea necesario para la comparación con los valores PTS y DTS. El STC completo es también la entrada al subtractor.

El intervalo máximo limitado entre SCR sucesivas (700 ms) o PCR (100 ms) permite el diseño y construcción de bucles enganchados en fase que se sabe son estables. La anchura de banda del bucle enganchado en fase tiene un límite superior impuesto por este intervalo. Como se muestra a continuación, en muchas aplicaciones, el bucle enganchado en fase requerido tiene una anchura de banda muy baja, y esta limitación no impone en general una limitación importante al diseño y funcionamiento del decodificador.

Si la frecuencia de funcionamiento libre o inicial del VCO está suficientemente próxima a la frecuencia de reloj de sistema correcta del decodificador, éste puede funcionar satisfactoriamente tan pronto el STC sea inicializado correctamente, antes de que el bucle enganchado en fase haya alcanzado un estado enganchado definitivo. Para una frecuencia STC de decodificador dada, que difiere por una magnitud limitada de la frecuencia codificada en las SCR y que está dentro de los límites de frecuencia absolutos requeridos por la aplicación del codificador, el efecto de la desadaptación entre las frecuencias STC del codificador y el decodificador, si no estuviesen enganchadas en fase, es el aumento o disminución gradual e inevitable de la ocupación de las memorias tampón del decodificador, de modo que se produciría eventualmente desbordamiento o subutilización con cualquier tamaño finito de memorias tampón de decodificador. En consecuencia, la cantidad de tiempo admisible antes que la frecuencia del STC del decodificador sea enganchada a la del codificador es determinado por la cantidad admisible de tamaño y retardo adicional de la memoria tampón del decodificador.

Si las SCR son recibidas por el decodificador con valores y temporización que reflejan muestras instantáneamente correctas de un STC de frecuencia constante en el codificador, el término de error e converge a un valor esencialmente constante después que el bucle ha alcanzado el estado enganchado. Esta condición de valores SCR correctos es sinónima de almacenamiento y transmisión con retardo constante de los datos del codificador al decodificador, o si este retardo no es constante, el equivalente efectivo de almacenamiento y transmisión con retardo constante con los valores SCR que han sido corregidos para reflejar las variaciones de retardo. Con los valores de e que convergen a una constante, las variaciones de la frecuencia VCO instantánea se hacen esencialmente cero después que el bucle está enganchado; se dice que VCO tiene muy poca fluctuación de fase o desviación de frecuencia. Mientras el bucle está en el proceso de enganche, la velocidad de cambio de la frecuencia VCO, la velocidad de desviación de frecuencia, puede ser controlada estrictamente por el diseño del filtro de paso bajo y la etapa de ganancia. En general, la velocidad de

desviación de VCO se puede diseñar para que satisfaga los requisitos de la aplicación, a reserva de las constricciones de tamaño y retardo de la memoria tampón del decodificador.

D.0.4 Fluctuación de fase de SCR y PCR

Si una red o un remultiplexor de trenes de transporte varía el retardo al entregar el tren de datos del codificador o sistema de almacenamiento al decodificador, esta variación tiende a causar una diferencia entre los valores de las SCR (o PCR) y los valores que deben tener cuando se reciben realmente. Esto se denomina como una fluctuación de fase de SCR o PCR. Por ejemplo, si el retardo en la entrega de una SCR es mayor que el retardo experimentado por otros campos similares en el mismo programa, esa SCR es tardía. De manera similar, si el retardo es menor que para otros campos de referencia de reloj en el programa, el campo es prematuro.

La fluctuación de fase de temporización a la entrada de un decodificador se refleja en la combinación de los valores de las SCR y los tiempos en que se reciben. Suponiendo una estructura de recuperación de reloj como se ilustra en la figura D.2, cualquier fluctuación de fase de temporización de este tipo se reflejará en los valores del término de error e , y los valores no cero de e inducen variaciones en los valores de f , resultando en variaciones de la frecuencia del reloj de sistema de 27 MHz. La variación de la frecuencia del reloj recuperado puede ser o no aceptable dentro de sistemas de decodificador, dependiendo de las necesidades de la aplicación específica. Por ejemplo, en decodificadores temporizados precisamente que producen salida de vídeo compuesta, la frecuencia de reloj recuperada se utiliza típicamente para generar el reloj de muestra de vídeo compuesto y la subportadora de crominancia. Las aplicaciones aplicables para la estabilidad de la frecuencia de la subportadora puede permitir solamente un ajuste muy lento de la frecuencia de reloj del sistema. En las aplicaciones en las que está presente una magnitud importante de fluctuación de fase SCR o PCR a la entrada del decodificador y hay constricciones estrictas de la velocidad de desviación de frecuencia del STC, las constricciones de tamaño y retardo adicionales razonables del decodificador pueden impedir un funcionamiento adecuado.

La presencia de fluctuación de fase SCR o PCR puede ser causada, por ejemplo, por transmisión de red que incorpora multiplexación de paquetes o de células o retardo variable de paquetes a través de la red, y puede ser causada por retardos de puesta en cola o por tiempo de acceso de red variable en sistemas de medios compartidos.

La multiplexación o remultiplexación de trenes de transporte o de programa cambia el orden y ubicación temporales relativos de los paquetes de datos y, por tanto, también de las SCR o PCR. El cambio de la ubicación temporal de las SCR hace que el valor de las SCR correctas previamente sea incorrecto, porque en general el tiempo en el cual son entregadas por una red de retardo constante no está representado correctamente por sus valores. De manera similar, un tren de programa o un tren de transporte con SCR o PCR correctas puede ser entregado por una red que impone un retardo variable del tren de datos, sin corregir los valores de SCR o PCR. El efecto es de nuevo la fluctuación de fase SCR o PCR, con efectos en el diseño y funcionamiento del decodificador. La cantidad de fluctuación de fase del caso más desfavorable impuesta por una red en la SCR o PCR recibidas en un decodificador depende de varios factores que rebasan el alcance de esta Recomendación | Norma Internacional, incluida la profundidad de las colas en cada uno de los conmutadores de red y el número total de conmutadores de red u operaciones de remultiplexación que actúan en cascada sobre el tren de datos.

En el caso de un tren de transporte, la corrección de las PCR es necesaria en una operación de remultiplexación, que crea un nuevo tren de transporte a partir de uno o más trenes de transporte. Esta corrección se realiza añadiendo un término de corrección a la PCR. Este término se puede calcular como sigue:

$$\Delta PCR = del_{act} - del_{const}$$

donde del_{act} es el retardo real experimentado por la PCR y del_{const} es una constante que se utiliza para todas las PCR de ese programa. El valor que se debe utilizar para del_{const} dependerá de la estrategia utilizada por el codificador/multiplexor original. Esta estrategia podrá ser, por ejemplo, planificar los paquetes lo más pronto posible, para que los enlaces de transmisión posteriores puedan retardarlos. En el cuadro D.1 se muestran tres estrategias múltiplex diferentes junto con el valor apropiado para del_{const} .

Cuadro D.1 – Estrategia de remultiplexación

Estrategia	del_{const}
Prematura	$del_{mín}$
Tardía	$del_{máx}$
Media	del_{avg}

ISO/CEI 13818-1:2007 (S)

Al diseñar un sistema, se pueden necesitar acuerdos privados respecto de la estrategia que han de utilizar los codificadores/multiplexores, porque esto tendrá un efecto sobre la capacidad de realizar cualquier remultiplexación adicional.

La cantidad de fluctuación de fase de múltiplex permitida no está limitada normativamente en esta Recomendación | Norma Internacional. Sin embargo, se pretende que 4 ms sea el máximo de fluctuación de fase en un sistema que funciona bien.

En sistemas que incluyen remultiplexores, es necesario tener especial cuidado en asegurar que la información del tren de transporte es coherente. En particular, esto se aplica a la PSI y a los puntos de discontinuidad. Pudiera ser necesario insertar los cambios de las tablas PSI en un tren de transporte de modo que no se pase al siguiente paso del remultiplexor hasta que la información sea incorrecta. Por ejemplo, en algunos casos una nueva versión de sección PMT no debe ser enviada en un plazo de 4 ms con respecto a los datos afectados por el cambio.

De manera similar, puede ser necesario que un codificador/multiplexor evite insertar PTS o DTS en una ventana de ± 4 ms alrededor de un punto de discontinuidad.

D.0.5 Recuperación de reloj en presencia de fluctuación de fase de red

En las aplicaciones en las que hay cualquier magnitud importante de fluctuación de fase presente en las indicaciones de tiempo de referencia de reloj recibidas, hay varias opciones disponibles para diseñar el decodificador; el diseño del decodificador depende en gran parte de las características de la señal de salida del decodificador y de las características de los datos y fluctuación de fase a la entrada.

Los decodificadores para distintas aplicaciones pueden tener requisitos diferentes en cuanto a la exactitud y estabilidad del reloj del sistema recuperado y se puede considerar que el grado de esta estabilidad y exactitud requeridas está en un eje. Se puede considerar que un extremo de este eje son las aplicaciones en las que el reloj del sistema reconstruido se utiliza directamente para sintetizar una subportadora de crominancia para uso en vídeo compuesto. Este requisito generalmente existe cuando el vídeo presentado es del tipo temporizado precisamente, según se ha descrito antes, de modo que cada imagen codificada se presenta exactamente una vez, y cuando la salida es vídeo compuesto en cumplimiento de las especificaciones aplicables. En ese caso, la subportadora de crominancia, el reloj de píxels y la velocidad de imagen tienen relaciones exactamente especificadas y todas tienen una relación definida con el reloj de sistema. La subportadora de vídeo compuesto debe tener como mínimo exactitud y estabilidad suficientes para que cualquier bucle enganchado en fase de la subportadora de crominancia del receptor de televisión normal pueda enganchar la subportadora, y que las señales de crominancia demoduladas utilizando la subportadora recuperada no muestren artefactos de fase de crominancia visibles. El requisito de algunas aplicaciones es utilizar el reloj de sistema para generar una subportadora que cumpla plenamente las especificaciones NTSC, PAL o SECAM que suelen ser más estrictas que las impuestas por los receptores de televisión típicos. Por ejemplo, la especificación SMPTE para NTSC requiere una precisión de la subportadora de 3 ppm, con una fluctuación de fase a corto plazo máxima de 1 ns por tiempo de línea horizontal y una deriva a largo plazo máxima de 0,1 Hz por segundo.

En las aplicaciones en las que el reloj de sistema recuperado no se utiliza para generar una subportadora de crominancia, se puede emplear para generar un reloj de píxels para vídeo y para generar un reloj de muestras de audio. Estos relojes tienen sus propios requisitos de estabilidad que dependen de las hipótesis hechas sobre el monitor de visualización receptor y de la cantidad aceptable de deriva de frecuencia de audio o "fluctuaciones de velocidad" a la entrada del decodificador.

En las aplicaciones en las que cada imagen y cada muestra de audio no se presentan exactamente una vez, es decir, se permite el "deslizamiento" de imagen y muestras de audio, el reloj de sistema puede tener requisitos de exactitud y estabilidad relativamente relajados. Este tipo de decodificador puede no tener sincronización precisa de presentación audio-vídeo y la presentación de audio y vídeo resultante puede no tener la misma calidad que en el caso de decodificadores precisamente temporizados.

La elección de los requisitos de exactitud y estabilidad del reloj de sistema recuperado depende de la aplicación. A continuación se examina el requisito más estricto identificado anteriormente, es decir, cuando el reloj de sistema se ha de utilizar para generar una subportadora de crominancia.

D.0.6 Reloj de sistema utilizado para la generación de una subportadora de crominancia

El diseño del decodificador puede ser determinado a partir de los requisitos de la subportadora resultante y la cantidad máxima de fluctuación de fase de red que se puede aceptar. De manera similar, si se conocen los requisitos de funcionamiento del reloj de sistema y el diseño y las capacidades del diseño del decodificador, se puede determinar la fluctuación de fase de red máxima admisible. Aunque está fuera del ámbito de la presente Recomendación | Norma Internacional indicar estos requisitos, se identifican los valores que se necesitan para especificar diseños con el fin de aclarar la explicación del problema e ilustrar un método de diseño representativo.

Con un circuito de bucle enganchado en fase de recuperación de reloj como se ilustra en la figura D.2, el reloj de sistema recuperado debe satisfacer los requisitos de una desviación de frecuencia respecto a la nominal del caso más desfavorable, medida en unidades de partes por millón (ppm) y una velocidad de desviación de frecuencia del caso más desfavorable, medida en ppm/s (ppm por segundo). La fluctuación de fase de temporización de red no corregida cresta a cresta tiene un valor que se puede especificar en milisegundos. En este tipo de bucle enganchado en fase, la fluctuación de fase de temporización de la red aparece como el término de error e en el diagrama, y como el bucle enganchado en fase actúa como un filtro de paso bajo en la fluctuación de fase en su entrada, el efecto del caso más desfavorable sobre la frecuencia de salida de 27 MHz se produce cuando hay una función de paso de amplitud máxima de temporización PCR en la entrada. El valor e tiene entonces una amplitud máxima igual a la fluctuación de fase cresta a cresta que se representa numéricamente como la fluctuación de base 2^{**33} en la porción de base de la codificación SCR o PCR. La velocidad máxima de cambio de la salida del filtro de paso bajo (LPF, *low pass filter*), f , que es el valor máximo de e en su entrada, determina directamente la velocidad de desviación de frecuencia máxima de la salida de 27 MHz. Para cualquier valor máximo de e y velocidad máxima de cambio de f , dados, se puede especificar un filtro de paso bajo. Sin embargo, a medida que se reduce la ganancia o frecuencia de corte del filtro de paso bajo, aumenta el tiempo requerido para que el bucle enganchado en fase enganche la frecuencia representada por las SCR o PCR. La implementación de los bucles enganchados en fase con constantes de tiempo muy largos se puede lograr mediante la utilización de técnicas de filtro de paso bajo digital, y posiblemente técnicas de filtro analógico. Con la implementación de filtros de paso bajo digitales, cuando el término de frecuencia f es la entrada a un VCO analógico, f es cuantificado por un convertidor digital analógico, cuyo tamaño de pasos se debe considerar al calcular la velocidad de desviación máxima de la frecuencia de salida.

Para asegurar que e converge con un valor que se aproxima a cero, la ganancia de bucle abierto del bucle enganchado en fase debe ser muy alta, como pudiera implementarse en una función integradora en el filtro de paso bajo en el bucle enganchado en fase.

Con un requisito de exactitud dado, puede ser razonable construir el bucle enganchado en fase de modo que la frecuencia de funcionamiento inicial del bucle enganchado en fase satisfaga el requisito de exactitud. En este caso, la frecuencia de 27 MHz inicial antes de que se enganche el bucle es suficientemente exacta para satisfacer el requisito de frecuencia de salida indicado. Si no fuera porque a la larga las memorias tampón del decodificador estarían desbordadas o subutilizadas, esta frecuencia de reloj de sistema inicial sería suficiente para el funcionamiento a largo plazo. Sin embargo, a partir del momento en que el decodificador comienza a recibir y decodificar datos hasta que el reloj de sistema se engancha al tiempo y frecuencia de reloj representados por las SCR o PCR recibidas, los datos llegan a las memorias tampón a una velocidad diferente a la que son extraídos, o equivalentemente, el decodificador está extrayendo unidades de acceso en tiempos que difieren del modelo de decodificador-objetivo de sistema (STD). Las memorias tampón del decodificador continuarán estando más o menos llenas que las del STD de acuerdo con la trayectoria de la frecuencia de reloj de sistema recuperado con respecto a la frecuencia de reloj del decodificador. Según la frecuencia VCO inicial relativa y la frecuencia de reloj de sistema del codificador, la ocupación de la memoria tampón del decodificador aumenta o disminuye. Suponiendo que esta relación no se conoce, el decodificador tiene que almacenar en memoria tampón datos adicionales para tener en cuenta cualquiera de los dos casos. El decodificador se debe construir de modo que retarde todas las operaciones de decodificación por una cantidad de tiempo que sea por lo menos igual a la cantidad de tiempo representada por el almacenamiento adicional en memoria tampón asignado para el caso en que la frecuencia VCO inicial es mayor que la frecuencia de reloj del codificador, con el fin de evitar subutilización de la memoria tampón. Si la frecuencia VCO inicial no es suficientemente exacta para satisfacer los requisitos de exactitud indicados, entonces el bucle enganchado en fase debe alcanzar el estado enganchado antes de que comience la decodificación, y hay un conjunto diferente de consideraciones relativas al comportamiento del bucle durante este tiempo y la capacidad de almacenamiento en memoria tampón adicional y retardo estático que es apropiado.

Una función por pasos en la fluctuación de fase de temporización a la entrada que produce una función por pasos en el término de error e del bucle enganchado en fase en la figura D.2 debe producir un término de frecuencia de salida f , de modo que cuando es multiplicado por la ganancia VCO la velocidad máxima de cambio es menor que la velocidad de desviación de frecuencia especificada. La ganancia del VCO se indica en términos de la cantidad del cambio en la frecuencia de salida con respecto a un cambio en la entrada de control. Una restricción adicional del filtro de paso bajo en el bucle enganchado en fase es que el valor estático de e cuando el bucle está enganchado debe estar limitado para determinar la capacidad de almacenamiento en memoria tampón adicional y el retardo de codificación estático que se debe aplicar. Este término se minimiza cuando el filtro de paso bajo tiene una ganancia DC muy alta.

Puede ser práctico utilizar circuitos de recuperación de reloj que difieren algo de los mostrados en la figura D.2. Por ejemplo, puede ser posible aplicar un bucle de control con un oscilador controlado numéricamente (NCO, *numerically controlled oscillator*) en vez de un VCO, dado que el NCO utiliza un oscilador de frecuencia fijo y se insertan o suprimen ciclos de reloj de eventos normalmente periódicos a la salida con el fin de ajustar la decodificación y temporización de presentación. Pueden plantearse algunas dificultades con este tipo de método cuando se utiliza con vídeo compuesto, pues hay una tendencia a originar desplazamientos de fase problemáticos de la subportadora o fluctuación de fase en la temporización de exploración horizontal o vertical. Una solución posible es ajustar el periodo

de exploraciones horizontales al comienzo del intervalo de supresión de trama vertical, a la vez que se mantiene la fase de la subportadora de crominancia.

En resumen, según los valores especificados para los requisitos, puede ser práctico o no construir un decodificador que reconstruya el reloj de sistema con suficiente exactitud y estabilidad, a la vez que mantiene los tamaños de memoria tampón de decodificador deseados y mayor retardo de decodificación.

D.0.7 Reconstrucción de vídeo y audio de componentes

Si se produce vídeo de componentes a la salida del decodificador, los requisitos de exactitud y estabilidad de temporización son generalmente menos estrictos que en el caso de vídeo compuesto. Típicamente, la tolerancia de frecuencia es la que pueden aceptar los circuitos de reflexión de visualización y la tolerancia de estabilidad es determinada por la necesidad de evitar desplazamiento visible de la imagen en la pantalla.

Se aplican los mismos principios ilustrados anteriormente, aunque en general es más fácil cumplir los requisitos específicos.

También en este caso la reconstrucción de la velocidad de muestra de audio sigue los mismos principios, aunque el requisito de estabilidad es determinado por la magnitud de la variación de velocidad de muestras aceptable a largo plazo y a corto plazo. Con el método de bucle enganchado en fase ilustrado en la sección anterior, la desviación a corto plazo puede ser muy pequeña y la variación de frecuencia a largo plazo se manifiesta como variación en la altura percibida. También en este caso, una vez fijado los límites especificados de esta variación, se pueden determinar los requisitos específicos del diseño.

D.0.8 Deslizamiento de trama

En algunas aplicaciones en las que no se requiere temporización precisa del decodificador, el reloj de tiempo de sistema del decodificador puede no ajustar su frecuencia de trabajo para que concuerde con la frecuencia representada por las SCR (o PCR) recibidas; puede tener un reloj de 27 MHz de funcionamiento libre, a la vez que subordina el STC del decodificador a los datos recibidos. En este caso, el valor de STC se debe actualizar según sea necesario para que concuerde con las SCR recibidas. La actualización del STC al recibir las SCR origina discontinuidades en el valor de STC. La magnitud de estas discontinuidades depende de la diferencia entre la secuencia de 27 MHz del decodificador y la frecuencia de 27 MHz del codificador, es decir, la que está representada por las SCR recibidas, y en el intervalo de tiempo entre SCR o PCR recibidas sucesivas. Como la frecuencia de reloj de sistema de 27 MHz del decodificador no está enganchada a la de los datos recibidos, no se puede utilizar para generar relojes de muestras de vídeo o de audio a la vez que se mantiene las hipótesis de temporización precisa de presentar la unidad de presentación de vídeo y audio una vez y mantener la misma velocidad de presentación de imagen y audio en el decodificador y en el codificador, con sincronización precisa de vídeo y audio. Hay múltiples posibilidades de implementar sistemas de decodificación y de presentación con esta estructura.

En un tipo de implementación, las imágenes y muestras de audio son decodificadas en el momento indicado por el STC del decodificador, mientras son presentadas en tiempos ligeramente diferentes, de acuerdo con los relojes de muestra producidos localmente. Según las relaciones de los relojes de muestra del decodificador con el reloj de sistema del codificador, a veces las imágenes y muestras de audio pueden ser presentadas más de una vez o no ser presentadas en absoluto, esto se denomina "deslizamiento de trama" o "deslizamiento de muestras" en el caso de audio. Este mecanismo puede introducir artefactos perceptibles. La sincronización de audio y vídeo no será precisa en general, debido a las unidades de tiempo en las cuales las imágenes y quizás las unidades de presentación de audio se repiten o suprimen. Según la implementación específica, se necesita en general almacenamiento en memoria tampón adicional en el decodificador para datos codificados o datos de presentación decodificados. La decodificación se puede realizar inmediatamente antes de la presentación, y no en el momento indicado en el STC del decodificador, o las unidades de presentación decodificadas se pueden almacenar para presentación retardada y posiblemente repetida. Si la decodificación se realiza en el momento de la presentación, se requiere un mecanismo para soportar la supresión de la presentación de imágenes y muestras de audio sin causar problemas en la decodificación de datos codificados con predicción.

D.0.9 Mitigación de la fluctuación de fase de red

En algunas aplicaciones puede ser posible introducir un mecanismo entre una red y un decodificador para reducir el grado de fluctuación de fase introducido por la red. La viabilidad de este método depende del tipo de trenes recibidos y de la cantidad y tipo de fluctuación de fase prevista.

El tren de transporte y el tren de programa indican en su sintaxis la velocidad a la cual el tren debe ser introducido en un decodificador. Estas velocidades indicadas no son precisas y no se pueden utilizar para reconstruir exactamente la temporización del tren de datos. No obstante, pueden ser útiles como parte de un mecanismo de mitigación.

Por ejemplo, un tren de transporte puede ser recibido de una red de modo que los datos se entreguen en ráfagas. Es posible almacenar los datos recibidos y transmitir datos de la memoria tampón al decodificador a una velocidad aproximadamente constante de manera que la memoria tampón permanezca aproximadamente medio llena.

Sin embargo, un tren de velocidad variable no se debe entregar a velocidad constante y con trenes de velocidad variable la memoria tampón no debe estar siempre medio llena. Un retardo medio constante a través de la memoria tampón requiere que la ocupación de la memoria tampón varíe en función de la velocidad de los datos. La velocidad a la cual se deben extraer los datos de la memoria tampón e introducirlos en el decodificador se puede aproximar utilizando la información de velocidad presente en el tren de datos. En trenes de transporte, la velocidad prevista es determinada por los valores de los campos PCR y el número de bytes del tren de transporte entre ellos. En trenes de programa, la velocidad prevista se especifica explícitamente como la velocidad del múltiplex de programa, aunque como se especifica en la presente Recomendación | Norma Internacional, la velocidad puede caer a cero en ubicaciones de SCR, es decir, si la SCR llega antes del momento previsto cuando se entregan datos a la velocidad indicada.

En el caso de trenes de velocidad variable, la ocupación correcta de la memoria tampón varía en función del tiempo, y no puede ser determinada exactamente a partir de la información de velocidad. Como un método alternativo, la SCR o PCR se pueden utilizar para medir el tiempo en que los datos entran en la memoria y controlar el tiempo en que los datos salen de la memoria tampón. Se puede diseñar un bucle de control para proporcionar retardo medio constante a través de la memoria tampón. Se puede observar que este diseño es similar al bucle de control ilustrado en la figura D.2. La calidad de funcionamiento que es posible obtener insertando este mecanismo de mitigación antes de un decodificador, se puede lograr también poniendo en cascada múltiples bucles enganchados en fase de recuperación de reloj; el rechazo de la fluctuación de fase de la temporización recibida se beneficiará del efecto combinado en el filtro de paso bajo de los bucles enganchados en fase en cascada.

Anexo E

Aplicaciones de transmisión de datos

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

E.0 Consideraciones

- El múltiplex de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se utilizará para transmitir datos así como vídeo y audio.
- Los trenes elementales de datos no son continuos pues pueden aparecer trenes de vídeo y audio en aplicaciones de radiodifusión.
- Aunque ya es posible identificar el principio de un paquete PES, no siempre es posible identificar el fin de un paquete PES por el principio del siguiente paquete PES, pues se pueden perder uno o varios paquetes de transporte que llevan paquetes PES.

E.1 Sugerencia

Una solución adecuada es transmitir el siguiente paquete PES, exactamente después de un paquete PES asociado. Cuando no hay otros paquetes PES para enviar, se puede enviar un paquete PES sin cabida útil.

En el cuadro E.1 se da un ejemplo de este paquete PES.

Cuadro E.1 – Ejemplo de encabezamiento de paquete PES

Campos de encabezamientos de paquete PES	Valores
packet_start_code_prefix	0x000001
stream_id	asignado
PES_packet_length	0x0003
'10'	'10'
PES_scrambling_control	'00'
PES_priority	'0'
data_alignment_indicator	'0'
copyright	'0'
original_or_copy	'0'
PTS_DTS_flags	'00'
ESCR_flag	'0'
ES_rate_flag	'0'
DSM_trick_mode_flag	'0'
additonal_copy_info_flag	'0'
PES_CRC_flag	'0'
PES_extension_flag	'0'
PES_header_data_length	0x00

Anexo F

Gráficos de sintaxis para la presente Recomendación | Norma Internacional

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

F.0 Introducción

Este anexo es un anexo informativo que presenta gráficamente la sintaxis de tren de transporte y de tren de programa. Este anexo no sustituye las cláusulas normativas precedentes.

Con el fin de producir diagramas claros, no todos los campos se han descrito o representado completamente. Los campos reservados pueden ser omitidos o señalados mediante zonas sin detalles. La longitud de los campos se indica en bits.

F.0.1 Sintaxis del tren de transporte

Véase la figura F.1.

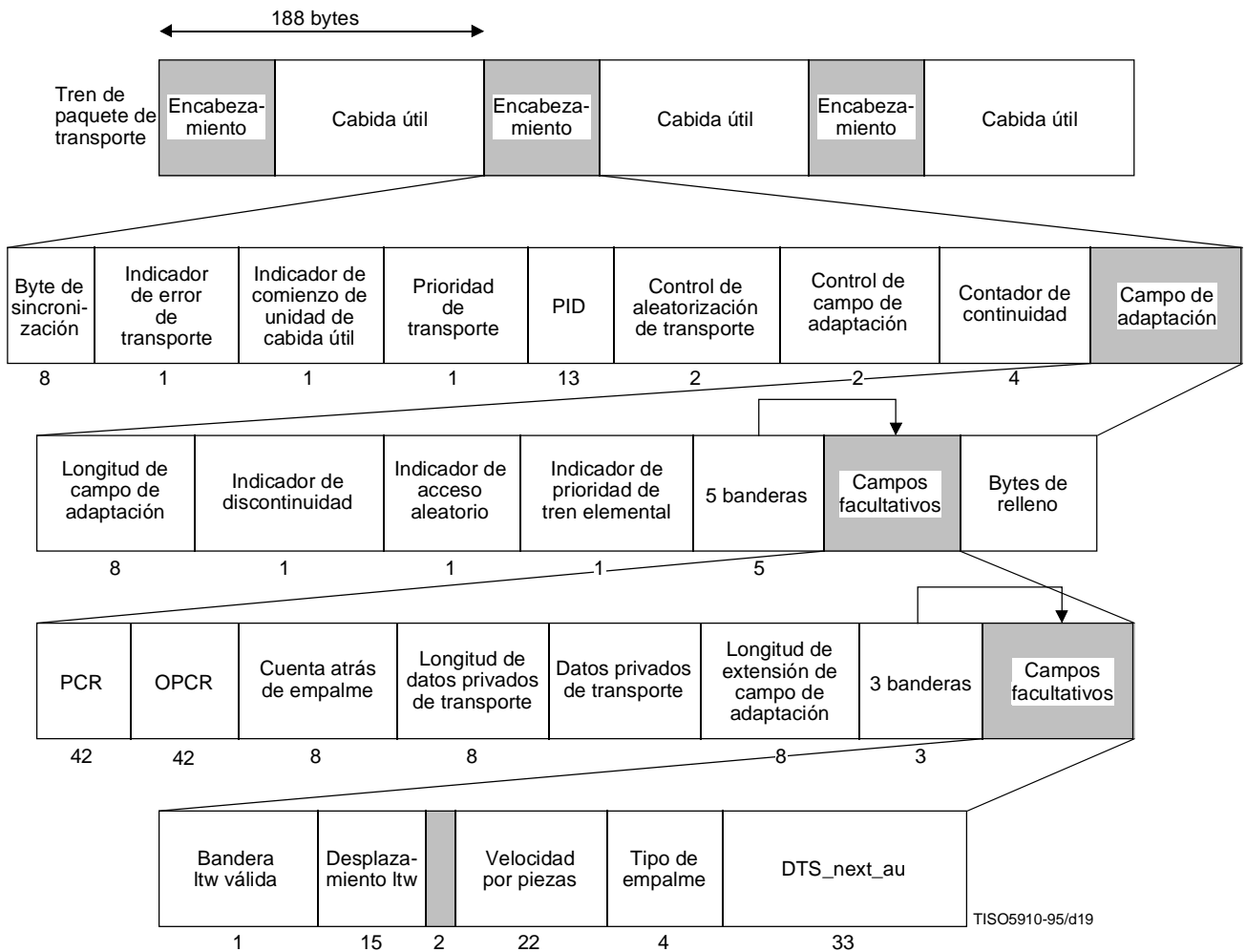


Figura F.1 – Diagrama de sintaxis de tren de transporte

F.0.2 Paquete PES

Véase la figura F.2.

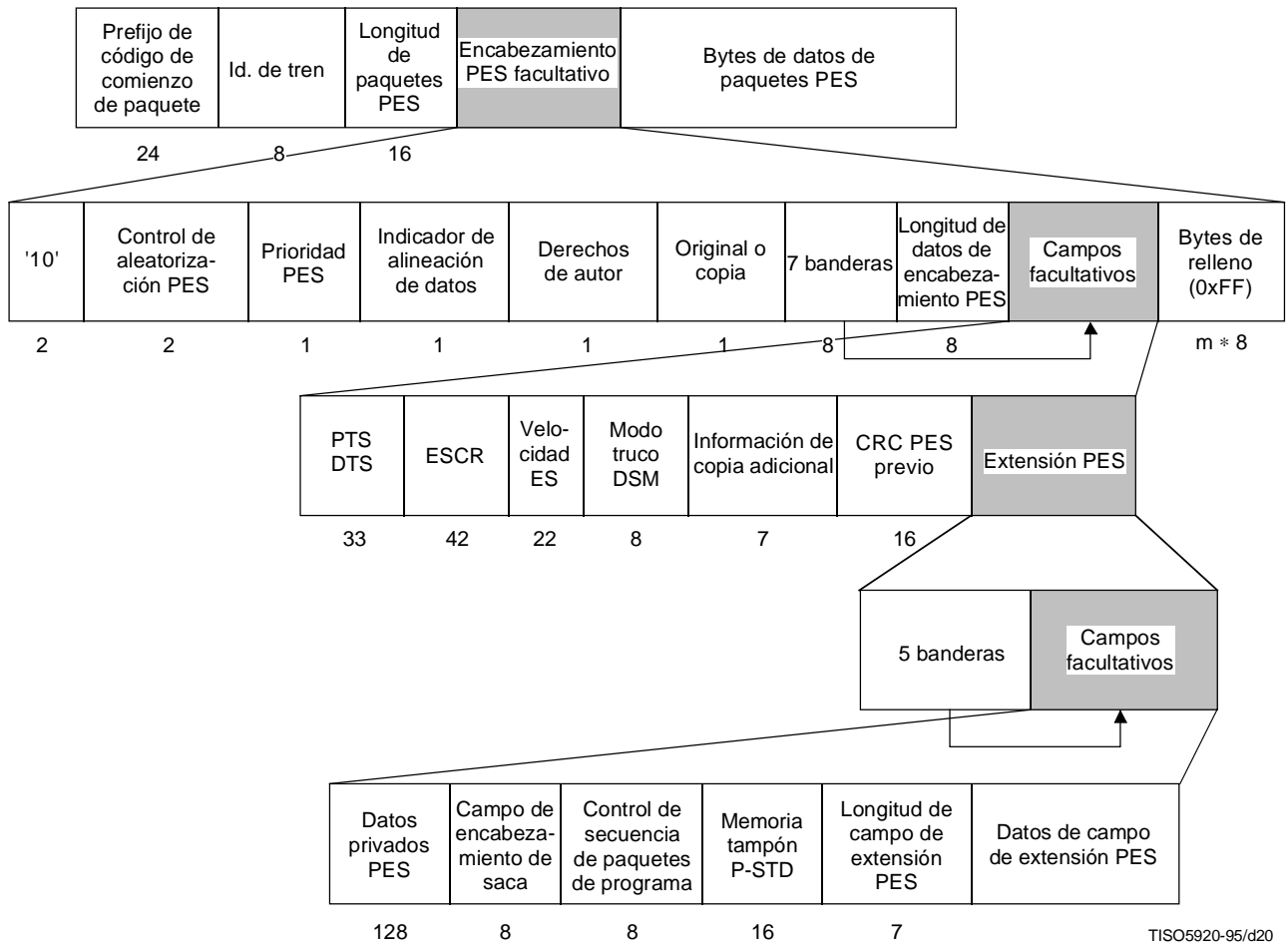


Figura F.2 – Diagrama de sintaxis de paquete PES

F.0.3 Sección de asociación de programas

Véase la figura F.3.

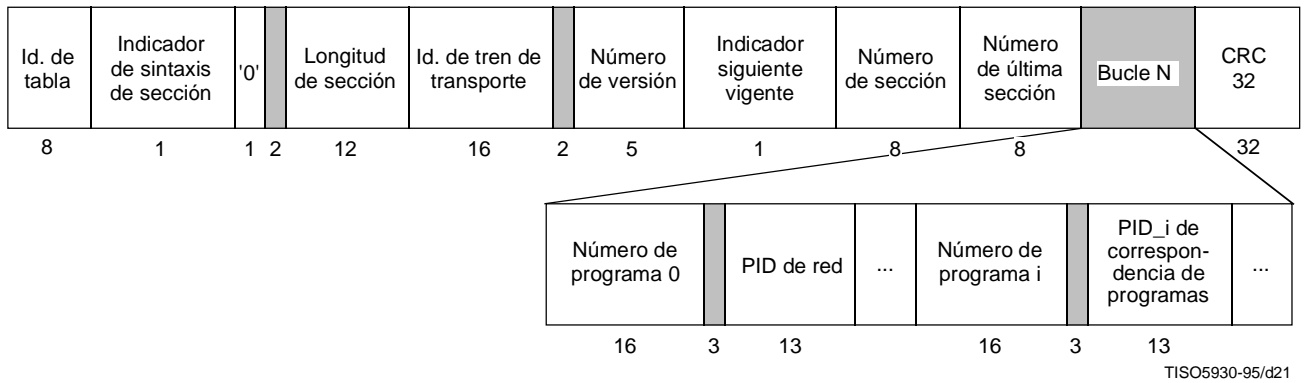


Figura F.3 – Diagrama de sección de asociación de programas

F.0.4 Sección CA

Véase la figura F.4.

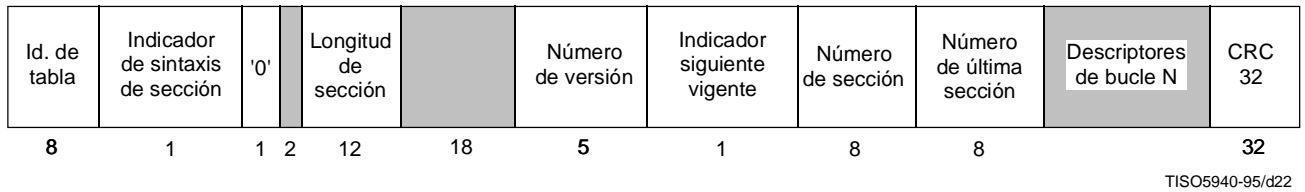


Figura F.4 – Diagrama de sección de acceso condicional

F.0.5 Sección de correspondencia de programas de tren de transporte

Véase la figura F.5.

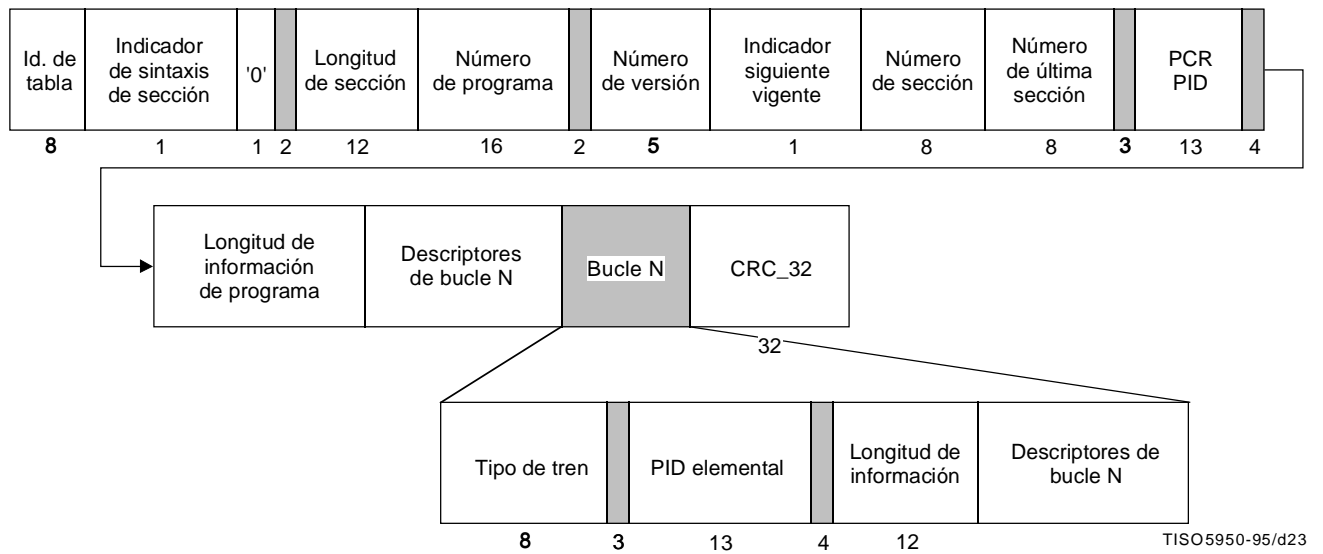


Figura F.5 – Diagrama de sección de correspondencia de programas de tren de transporte

F.0.6 Sección privada

Véase la figura F.6.

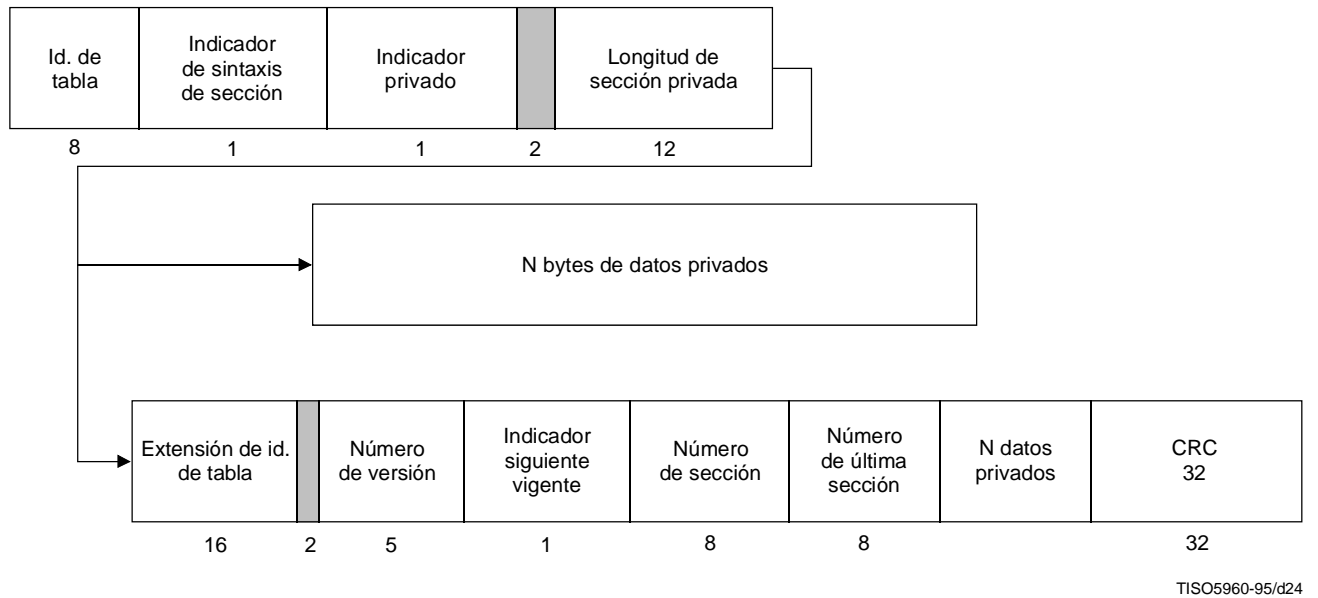


Figura F.6 – Diagrama de sección privada

F.0.7 Tren de programa

Véase la figura F.7.

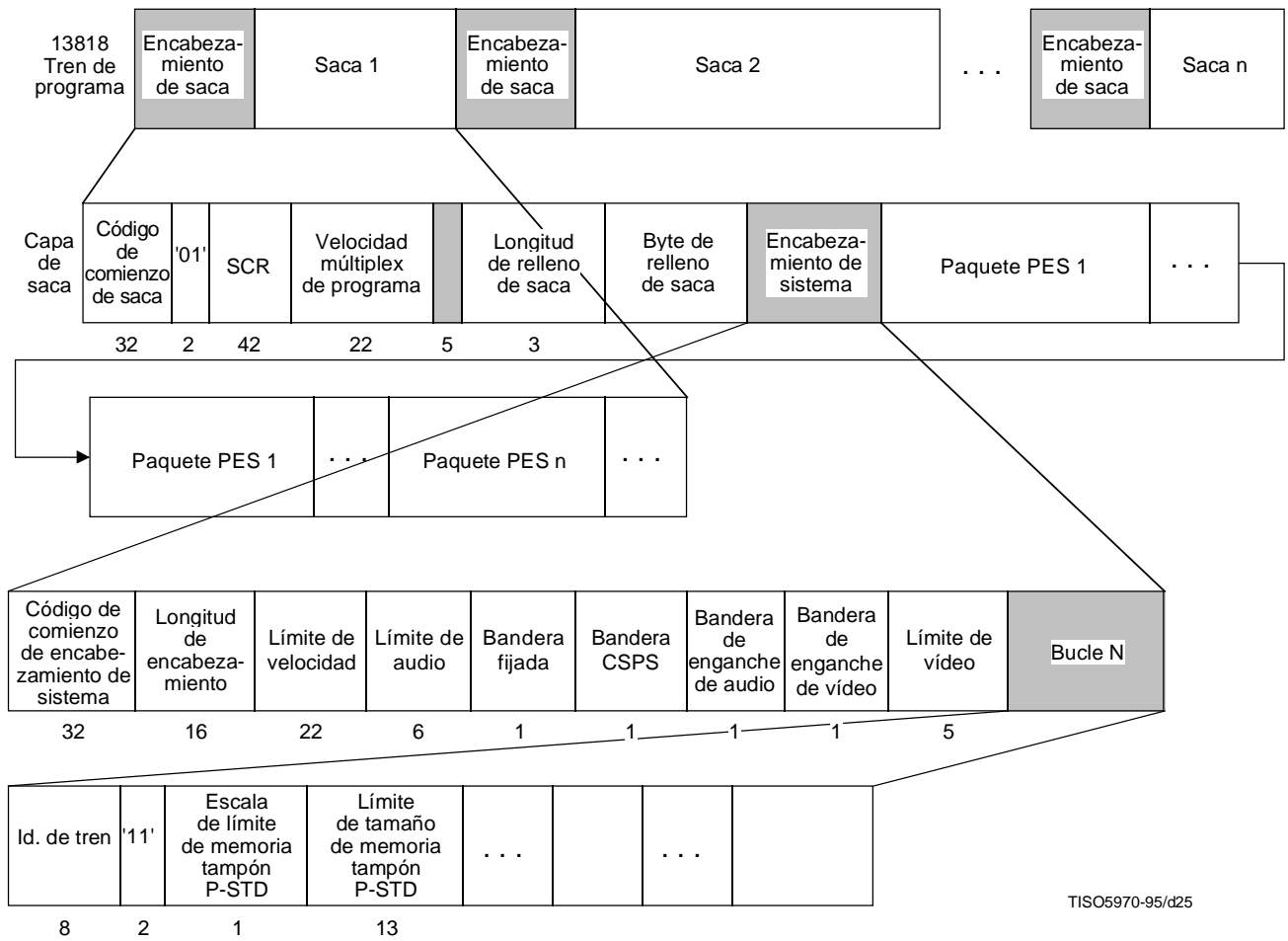


Figura F.7 – Diagrama de tren de programa

F.0.8 Correspondencia de trenes de programa

Véase la figura F.8.

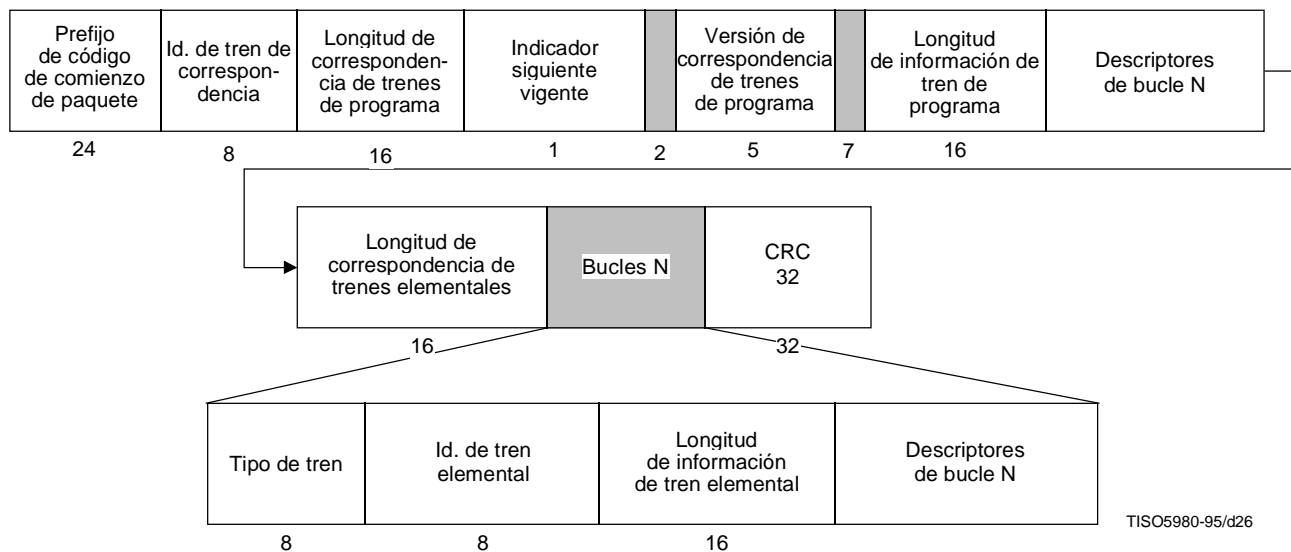


Figura F.8 – Diagrama de correspondencia de trenes de programa

Anexo G

Información general

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

G.0 Información general

G.0.1 Emulación de byte de sincronización

Al elegir valores de PID se recomienda que se evite la emulación periódica de los bytes de sincronización. Esta emulación puede ocurrir dentro del campo PID o como una combinación del campo PID y fijaciones de banderas adyacentes. Se recomienda que se permita la emulación del byte de sincronización en la misma posición del encabezamiento de paquete para un máximo de 4 paquetes de transporte consecutivos.

G.0.2 Estado de imágenes omitidas y proceso de decodificación

Se supone que la secuencia que se visualiza contiene solamente tramas I y P. Indíquese la siguiente imagen que se ha de decodificar por picture_next (imagen siguiente) y la imagen que se está visualizando por picture_current (imagen vigente). Como el decodificador vídeo puede omitir imágenes, es posible que no todos los bits de picture_next estén presentes en la memoria tampón STD, EB_n o B_n, cuando llega el momento de suprimir estos bits para la decodificación instantánea y visualización. Cuando esto ocurre, no se suprime ningún bit en la memoria tampón y picture_current se visualiza de nuevo. Cuando llega el tiempo de visualización de la siguiente imagen, si el resto de los bits correspondientes a picture_next están en ese momento en la memoria tampón, EB_n o B_n se suprimen todos los bits de picture_next y picture_next se visualiza. Si todos los bits de picture_next no están en la memoria tampón, se repite el proceso anterior de visualizar de nuevo picture_current. Este proceso se repite hasta que se puede visualizar picture_next. Obsérvese que si una PTS precede a picture_next en el tren de bits, será errónea por algún múltiplo del intervalo de visualización de imagen (el cual, por su parte, puede depender de algunos parámetros) y se debe pasar por alto.

Cuando se produce la situación de imágenes omitidas descrita anteriormente, el codificador tiene que insertar una PTS antes de la imagen que ha de ser decodificada después de `picture_next`. Esto permite al decodificador verificar inmediatamente que ha visualizado correctamente la secuencia de imágenes recibida.

G.0.3 Selección de valores de PID

Se insta a las aplicaciones a que utilicen valores de PID con números bajos (evitando valores reservados como los especificados en el cuadro 2-4) y valores de grupo juntos en la mayor medida posible.

G.0.4 Emulación de código de comienzo de PES

Tres bytes consecutivos que tienen el valor de un prefijo de código de comienzo de paquete (0x000001), que cuando están concatenados con un cuarto byte puede emular a un encabezamiento de paquete PES en un lugar no previsto en el tren.

Esto, denominado emulación de código de comienzo, no es posible en trenes elementales de vídeo. Es posible en trenes elementales de audio y datos. Es también posible en la frontera de un encabezamiento de paquete PES y una cabida útil de paquete PES, incluso si la cabida útil del paquete PES es vídeo.

Anexo H

Datos privados

(Este anexo no este parte integrante de esta Recomendación | Norma Internacional)

H.0 Datos privados

Datos privados son cualesquiera datos de usuario que no están codificados con una norma especificada por UIT-T | ISO/CEI y se mencionan en esta Especificación. El contenido de estos datos no está especificado ni se especificará en esta Recomendación | Norma Internacional en el futuro. El STD definido en esta Especificación no abarca otros datos privados que los del proceso de demultiplexación. Un participante privado puede definir cada STD para trenes privados.

Los datos privados pueden ser transportados en los siguientes lugares dentro de la sintaxis de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

1) *Paquete de tren de transporte – Cuadro 2-2*

Los bytes de datos de la sintaxis de paquetes de transporte pueden contener datos privados. Los datos privados transportados en este formato se denominan privados de usuario en el cuadro 2-34 sobre tipos de tren. Está permitido que los paquetes de tren del transporte contengan datos privados hasta los campos de adaptación inclusive.

2) *Campos de adaptación del tren de transporte – Cuadro 2-6*

La presencia de cualesquiera bytes de datos privados facultativos en el campo de adaptación es señalada por la bandera de datos privados de transporte. El número de los bytes de datos privados está restringido inherentemente por la semántica del campo de longitud de campo de adaptación, donde el valor de la longitud del campo de adaptación no excederá de 183 bytes.

3) *Paquetes PES – Cuadro 2-21*

Hay dos posibilidades de transportar datos privados dentro de paquetes PES. La primera posibilidad es dentro del encabezamiento de paquete PES, dentro de los 16 bytes facultativos de datos privados PES. La presencia de este campo es señalada por la bandera de datos privados PES. La presencia de la bandera de datos privados PES es señalada por la bandera de extensión PES. Si están presentes, estos bytes, cuando se consideran con los campos adyacentes, no emularán el prefijo de código de comienzo de paquete.

La segunda posibilidad es dentro del campo de bytes de datos de paquete PES. Esto puede denominarse datos privados dentro de paquetes PES en el tipo de tren del cuadro 2-34. Esta categoría de datos privados se puede dividir en dos: tren privado 1 se refiere a datos privados dentro de los paquetes PES que siguen la sintaxis de paquete PES de modo que están presentes todos los campos hasta longitud de datos de encabezamiento PES inclusive, aunque no limitados a éstos; tren privado 2 se refiere a datos

privados dentro de paquetes PES cuando sólo los tres primeros campos están presentes seguidos de los bytes de datos de paquetes PES que contienen datos privados.

Obsérvese que los paquetes PES existen dentro de trenes de programa y de trenes de transporte, por lo que tren privado 1 y tren privado 2 existen también dentro de los trenes de programa y los trenes de transporte.

4) *Descriptores*

Los descriptores existen en trenes de programa y en trenes de transporte. El usuario puede definir una gama de descriptores privados. Estos descriptores comenzarán con los campos de rótulo de descriptor y de longitud de descriptor. Para descriptores privados, el valor de rótulo de descriptor puede tomar los valores 64 a 255 como se indica en el cuadro 2-45. Estos descriptores se pueden colocar dentro de la tabla de correspondencia de trenes de programa (cuadro 2-34); en la tabla de sección CA (cuadro 2-32), en la sección de correspondencia de programas TS (cuadro 2-33) y en cualquier sección privada (cuadro 2-35).

Específicamente, los bytes de datos privados deben aparecer en el descriptor CA.

5) *Sección privada*

La sección privada, cuadro 2-35, proporciona otro medio de transportar datos privados también en dos formas. Este tipo de tren elemental puede ser identificado en el tipo de tren, cuadro 2-34, como datos privados en secciones PSI. Un tipo de sección privada incluye solamente los primeros cinco campos definidos, seguidos por datos privados. Para esta estructura, el indicador de sintaxis de sección se pondrá a un valor de '0'. Para el otro tipo, el indicador de sintaxis de sección se pondrá a un valor de '1' y toda la sintaxis estará presente hasta el último número de sección inclusive, seguido de bytes de datos privados y terminando con CRC_32.

Anexo I

Conformidad de sistemas e interfaz en tiempo real

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

I.0 Conformidad de sistemas e interfaz en tiempo real

La conformidad de los trenes de programa y de los trenes de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 se especifica desde el punto de vista de las especificaciones normativas de esta Recomendación | Norma Internacional. Estas especificaciones incluyen, entre otros requisitos, un decodificador-objetivo de sistema (T-STD y P-STD) que indica el comportamiento de un decodificador ideal cuando el tren es la entrada a este decodificador. Este modelo, y la verificación asociada no incluye información relativa al funcionamiento con entrega del tren en tiempo real, salvo en lo que concierne a la exactitud de la frecuencia del reloj de sistema representada por el tren de transporte y el tren de programa. Todos los trenes de transporte y trenes de programa deben cumplir esta Recomendación | Norma Internacional.

Además, hay una especificación de interfaz en tiempo real para la entrada de trenes de transporte y trenes de programa a un decodificador. Esta Recomendación | Norma Internacional permite la normalización de la interfaz entre decodificadores MPEG y adaptadores a redes, canales o medios de almacenamiento. Los efectos de temporización de canales, y la incapacidad de los adaptadores de eliminar completamente en la práctica estos defectos producen desviaciones con respecto a la planificación ideal de la entrega de bytes. Aunque no es necesario que todos los decodificadores MPEG tengan esta interfaz, las implementaciones que lo incluyen cumplen las especificaciones. Esta Recomendación | Norma Internacional trata del comportamiento de la entrega de trenes de transporte y trenes de programa en tiempo real a los decodificadores, de modo que se garantice que las memorias tampón de datos codificados en los decodificadores no estén desbordadas ni subutilizadas, y que los decodificadores son capaces de realizar la recuperación de reloj con la calidad de funcionamiento requerida por sus aplicaciones.

La interfaz en tiempo real MPEG especifica la magnitud de desviación admisible máxima con respecto al plan de entrega de bytes ideal indicado por los campos de referencia de reloj de programa (PCR) y referencia de reloj de sistema (SCR) codificados en el tren.

Anexo J

Interfaz entre redes que inducen fluctuación de fase y decodificadores MPEG-2

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

J.0 Introducción

En este anexo la expresión "tren de sistemas" se utilizará para referirse a trenes de transporte y trenes de programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Cuando se utiliza el término "STD" se entiende que significa P-STD (decodificador-objetivo de sistemas de programas) para trenes de programa y T-STD (decodificador-objetivo de sistemas de transporte) para trenes de transporte.

El plan previsto de entrega de bytes de un tren de sistemas se puede deducir analizando el tren. Un tren de sistemas es conforme si puede ser decodificado por el STD, que es un modelo matemático de un decodificador ideal. Si se transmite un tren de sistemas conforme por una red que induce fluctuación de fase, el plan verdadero de entrega de bytes puede diferir considerablemente del plan previsto de entrega de bytes. En estos casos, puede no ser posible decodificar el tren de sistemas en este decodificador ideal, porque la fluctuación de fase puede ocasionar desbordamientos o subutilización de la memoria tampón y puede hacer difícil recuperar la base de tiempos. Un ejemplo importante de una red que induce fluctuación de fase es la red en modo de transferencia asíncrono (ATM).

La finalidad de este anexo es proporcionar orientación a las entidades que envían trenes de sistemas por redes que inducen fluctuación de fase. Es probable que se elaboren modelos de conformidad específicos de red para transportar trenes de sistemas, para diversos tipos de red, incluida ATM. El STD más una definición de interfaz en tiempo real puede desempeñar una función importante en la definición de estos modelos. En J.1 se presenta un marco para elaborar modelos conformes de red.

En J.2 se presentan tres ejemplos de codificación de red para permitir que los adaptadores de red mitiguen la fluctuación de fase. En el primer ejemplo, se supone un tren de sistemas de velocidad binaria constante y se utiliza el sistema primero en llegar, primero en salir (FIFO, *first in, first out*) para la fluctuación de fase. En el segundo ejemplo, la capa de adaptación de red incluye indicaciones de tiempo para facilitar la mitigación de la fluctuación de fase. En el último ejemplo, se supone que se dispone de un reloj de red común de extremo a extremo, y se explota para paliar la fluctuación de fase.

En J.3 se presentan dos ejemplos de implementaciones de codificador en los cuales se puede acomodar la fluctuación de fase inducida por la red. En el primer ejemplo, se inserta un adaptador de red para mitigar la fluctuación de fase entre una salida de red y el decodificador MPEG-2. Se supone que el decodificador MPEG-2 se conforma con una especificación de interfaz MPEG-2 en tiempo real. Esta interfaz requiere un decodificador MPEG-2 con más tolerancia de la fluctuación de fase que el decodificador ideal del STD. El adaptador de red procesa el tren de bits con fluctuación de fase entrante y el tren de sistemas saliente es un tren cuyo plan verdadero de entrega de bytes se conforma a la especificación en tiempo real. El ejemplo 1 se examina en J.3.1. Para algunas aplicaciones, el método de adaptador de red será muy costoso porque requiere dos etapas de procesamiento. Por consiguiente, en el segundo ejemplo, se integran funciones de supresión de fluctuación de fase y de decodificación MPEG-2. Se omite el proceso intermedio del dispositivo que suprime la fluctuación de fase, de modo que sólo se requiere una etapa de la recuperación de reloj. Los decodificadores que realizan la supresión de fluctuación de fase y decodificación integradas se denominan en este anexo como decodificadores específicos de red integrados, o simplemente decodificadores integrados. Estos decodificadores se examinan en J.3.2.

Para construir adaptadores de red o decodificadores integrados, se debe suponer un valor máximo para la fluctuación de fase de red cresta a cresta. Con el fin de fomentar la interoperabilidad, se debe especificar un límite de la fluctuación de fase cresta a cresta para cada tipo de red.

J.1 Modelos de conformidad de red

En la figura J.1 se muestra una manera de modelar la transmisión de un tren de sistema a través de una red que induce fluctuación de fase.

El tren de sistemas entra en un dispositivo de codificación específico de red que convierte el tren de sistemas en un formato específico de red. La información para ayudar a suprimir la fluctuación de fase a la salida de la red puede formar parte de este formato. El decodificador de red comprende un decodificador específico de red y un decodificador de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Se supone que el decodificador de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 es conforme a una especificación de interfaz en tiempo real y puede tener la misma arquitectura que el STD con memorias tampón apropiadas ampliadas para proporcionar la tolerancia a la fluctuación de fase.

El decodificador específico de red suprime los datos que no son de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 añadidos por el codificador específico de red y suprime la fluctuación de fase a la salida de la red. La salida del decodificador específico de red es un tren de sistemas que se conforma a la especificación en tiempo real.

Se puede definir un decodificador-objetivo de red (NTD, *network target decoder*) basado en la arquitectura anterior. Un tren de bits de red conforme sería uno que pudiese ser decodificado por el NTD. Un decodificador de red sería conforme si pudiese decodificar cualquier tren de bits de red que pueda ser decodificado por el NTD. Un decodificador de red real pudiera o no tener la arquitectura del NTD.

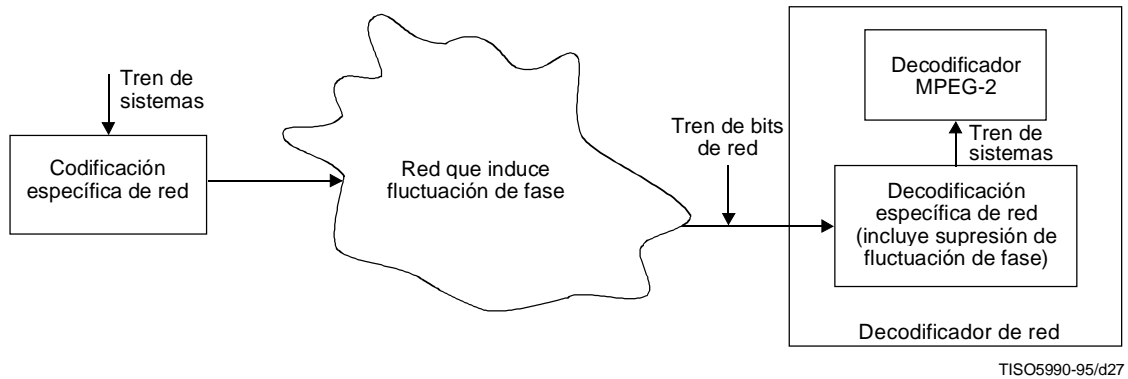


Figura J.1 – Envío de trenes de sistemas por una red que induce fluctuación de fase

J.2 Especificación de red para suprimir la fluctuación de fase

En el caso de trenes de sistemas de velocidad binaria constante, la supresión de la fluctuación de fase se puede realizar con FIFO (primero en entrar, primero en salir); los datos adicionales que proporcionan soporte específico para la supresión de fluctuación de fase no se requieren en la capa de adaptación de red. Después que se suprimen los bytes añadidos por la codificación de red, los datos del tren de sistema se colocan en FIFO. Un bucle enganchado en fase mantiene la memoria tampón aproximadamente medio llena ajustando la velocidad de salida en respuesta a los cambios de la ocupación de la memoria tampón. En este ejemplo, la magnitud de fluctuación de fase suprimida dependerá del tamaño de FIFO y de las características del bucle enganchado en fase.

La figura J.2 ilustra una segunda manera de realizar la supresión de la fluctuación de fase. En este ejemplo, se supone el soporte de indicaciones de tiempo de una capa de adaptación de red. Con esta técnica se puede suprimir la fluctuación de fase de trenes de sistemas de velocidad binaria constante y de velocidad binaria variable.

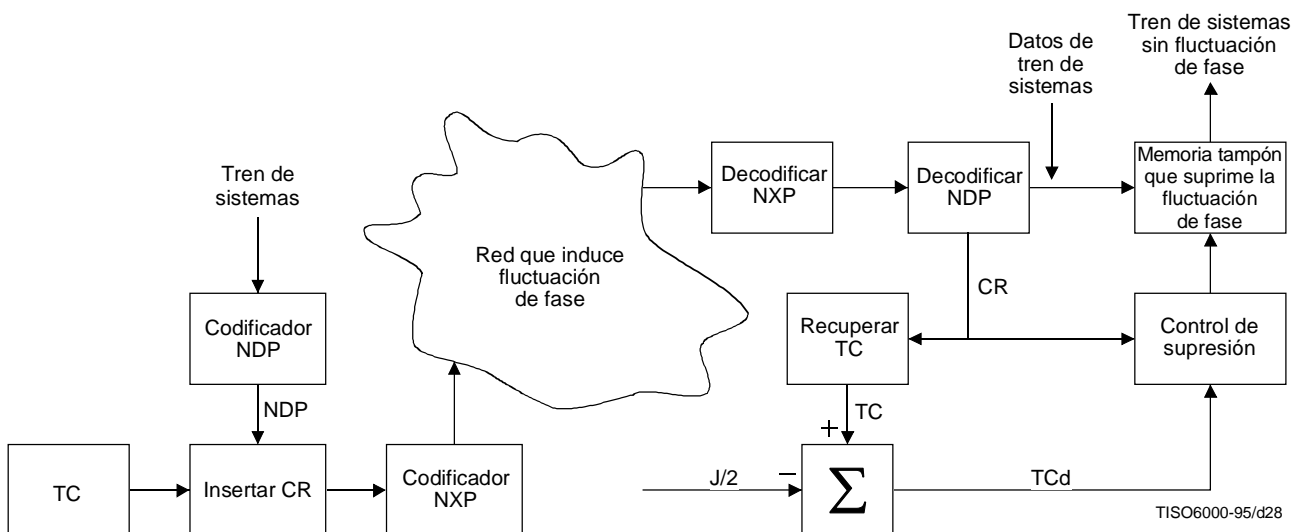


Figura J.2 – Supresión de fluctuación de fase mediante indicaciones de tiempo de la capa de red

Se supone que se diseña un adaptador de red para compensar la fluctuación de fase cresta a cresta de J segundos. El plan previsto de entrega de bytes se reconstruye utilizando muestras de referencia de reloj (CR, *clock reference*) de un reloj de tiempo (TC, *time clock*). Las CR y el TC son análogos a las PCR y el STC. El codificador de paquetes de datos de red (NDP, *network data packet*) convierte cada paquete de tren de sistemas en un paquete de datos de red (NDP). Los paquetes de datos de red contienen un campo para transportar valores CR, y el valor vigente del TC se inserta en este campo cuando el NDP deja el codificador NDP. La función de paquetización de transporte de red (NXP, *network transport packetization*) encapsula los NDP en paquetes de transporte de red. Después de la transmisión a través de la red, las CR son extraídas por el decodificador NDP cuando los NDP entran en el decodificador NDP. Las CR se utilizan para reconstruir el TC, por ejemplo, utilizando un bucle enganchado en fase. El primer paquete MPEG-2 es suprimido de la memoria tampón de supresión de fase cuando el TC retardado (TCd) es igual a la CR del primer paquete MPEG-2. Los siguientes paquetes MPEG-2 son suprimidos cuando sus valores CR equivalen al valor de TCd.

Cuando no se conocen los detalles de la implementación, tales como la velocidad del bucle de recuperación de reloj TC y la pureza espectral del TC, el tamaño de la memoria tampón de supresión de fase depende solamente de la fluctuación de fase cresta a cresta máxima que se ha de suprimir y de la velocidad de transporte mayor en el tren de sistemas. El tamaño de la memoria tampón de supresión de fase, B_{dj} viene dado por

$$B_{dj} = JR_{\text{máx}}$$

donde $R_{\text{máx}}$ es la velocidad de datos máxima del tren de sistemas en bits por segundo. Cuando los paquetes que atraviesan la red experimentan un retardo nominal, la memoria tampón está medio llena. Cuando experimentan un retardo de $J/2$ segundos, la memoria tampón está vacía, y cuando experimentan un retardo (adelanto) de $-J/2$ segundos, la memoria tampón está llena.

Como un último ejemplo, en algunos casos se dispondrá de un reloj de red común de extremo a extremo, y puede ser factible enganchar la frecuencia de reloj del sistema al reloj común. El adaptador de red puede paliar la fluctuación de fase con FIFO. El adaptador utiliza las PCR o SCR para reconstruir el plan original de entrega de bytes.

J.3 Ejemplo de implementaciones de decodificador

J.3.1 Adaptador de red seguido de un decodificador MPEG-2

En esta implementación, se conecta un adaptador de red conforme a la especificación de red a un decodificador MPEG-2 conforme a la especificación de interfaz en tiempo real.

J.3.2 Decodificador integrado

El ejemplo presentado en J.3.1 requiere dos etapas de procesamiento. La primera etapa es necesaria para suprimir la fluctuación de fase a la salida de la red; la segunda etapa, que recupera el STC mediante el procesamiento de PCR o SCR, se requiere para la decodificación STD. El ejemplo presentado en esta subcláusula es un decodificador que integra las funciones de supresión de fluctuación de fase y de decodificación en un solo sistema. El reloj STC es recuperado directamente utilizando los valores de PCR o SCR con fluctuación de fase. Para presentar este ejemplo, se supondrá un tren de transporte MPEG-2.

La figura J.3 ilustra el funcionamiento del decodificador integrado. La entrada del tren de paquetes de red en el decodificador se supone que es igual a la mostrada en la figura J.2.

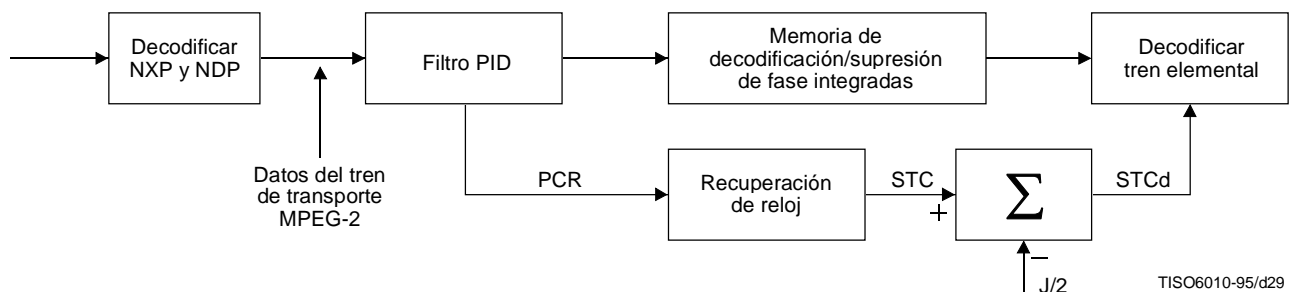


Figura J.3 – Supresión de fluctuación de fase y decodificación MPEG-2 integradas

ISO/CEI 13818-1:2007 (S)

Los paquetes de red entrantes son reensamblados en datos del tren de transporte MPEG-2 por las funciones de decodificación NXP y NDP. Los paquetes del tren de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 con fluctuación de fase son filtrados para extraer paquetes con el PID deseado. Para el caso ilustrado, el PID que se decodifica transporta también las PCR. Los valores de PCR se envían a un bucle enganchado en fase para recuperar el STC. Todos los paquetes para el PID seleccionado se colocan en la memoria tampón integrada. Un valor positivo de $J/2$ segundos es sustraído del STC para obtener el STC retardado, STCd. También en este caso, J es la fluctuación de fase cresta a cresta que el decodificador de red puede admitir. Se introduce el retardo para garantizar que todos los datos requeridos para una unidad de acceso han llegado a la memoria tampón cuando la PTS/DTS de la unidad de acceso equivale al valor vigente de STCd.

Cuando no conocen los detalles de la implementación, tales como la velocidad del bucle de recuperación de reloj STC y la pureza espectral del STC:

$$\begin{aligned} B_{size} &= B_{dec} + B_{mix} + B_{OH} + 512 + B_j \\ &= B_n + 512 + B_j \end{aligned}$$

donde $B_j = R_{\text{máx}} J$ y $R_{\text{máx}}$ es la velocidad máxima a la cual los datos entran en el filtro PID. Según la implementación, la memoria integrada podrá dividirse en dos componentes como en el STD de transporte.

Anexo K

Empalme de trenes de transporte

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

K.0 Introducción

A los efectos de este anexo, por 'empalme' ha de entenderse la concatenación, efectuada en el nivel de transporte, de dos trenes elementales diferentes, de modo que el tren de transporte resultante se ajuste íntegramente a esta Recomendación | Norma Internacional. Los dos trenes elementales pueden haber sido generados en lugares y/o tiempos diferentes, y que no se hubiera previsto su empalme cuando fueron generados. En lo sucesivo llamaremos el tren 'antiguo' a un tren elemental continuo (de vídeo o de audio) que, a partir de un determinado punto ha sido reemplazado por otro (el tren 'nuevo'). Este punto se denomina el empalme. Es la frontera entre los datos pertenecientes al tren 'antiguo' y los pertenecientes al 'nuevo'.

Un empalme puede ser liso y no liso:

- Un empalme liso es un empalme que no provoca discontinuidad en la decodificación (véase 2.7.6). Esto significa que el instante de decodificación de la primera unidad de acceso del tren 'nuevo' es consecuente con el instante de decodificación de la unidad de acceso del tren 'antiguo' que precede al empalme, es decir, es igual al que tendría la siguiente unidad de acceso si el tren 'antiguo' hubiera continuado. En lo sucesivo, este instante de decodificación se llamará 'instante de decodificación lisa'.
- Un empalme no liso es un empalme que produce una discontinuidad en la decodificación, es decir, el instante de decodificación de la primera unidad de acceso del tren 'nuevo' es mayor que el instante de decodificación lisa.

NOTA – Un instante de decodificación menor que el instante de decodificación lisa está prohibido.

El empalme está permitido en cualquier frontera de paquete de tren de transporte, ya que el tren resultante es legal. Sin embargo, en el caso general, si no se sabe nada en cuanto al lugar en que comienzan el paquete PES y la unidad de acceso, esta constricción obliga a analizar, no sólo la capa de transporte, sino también la capa de paquete PES y la capa de tren elemental, y en algunos casos podría ser necesario cierto procesamiento de la cabida útil de los paquetes del tren de transporte. Si se desea evitar esas complejas operaciones, el empalme debe efectuarse en lugares donde el tren de transporte tiene propiedades favorables, las que se indican por la presencia de un punto de empalme.

La presencia de un punto de empalme se señala por los campos bandera de empalme (`splice_flag`) y cuenta inversa de empalme (`splice_countdown`) (para la semántica de estos campos, véase 2.4.3.4). En lo sucesivo, el paquete de tren de transporte en el que el campo cuenta inversa de empalme llega al valor cero se llamará 'paquete de empalme'. El punto de empalme está situado inmediatamente después del último byte del paquete de empalme.

K.1 Diferentes tipos de punto de empalme

Un punto de empalme puede ser un punto de empalme ordinario o un punto de empalme liso.

K.1.1 Puntos de empalme ordinario

Si el campo de bandera de empalme liso no está presente, o si su valor es cero, el punto de empalme es ordinario. La presencia de un punto de empalme ordinario sólo señala propiedades de alineación del tren elemental: el paquete de empalme termina en el último byte de una unidad de acceso, y la cabida útil del siguiente paquete de transporte del mismo PID comenzará con el encabezamiento de un paquete PES cuya cabida útil empezará con un punto de acceso de tren elemental (o con un código de fin de secuencia (`sequence_end_code()`) seguido inmediatamente de un punto de acceso de tren elemental, en el caso de vídeo). Estas propiedades permiten efectuar fácilmente operaciones de 'cortar y pegar' en el nivel de transporte, al mismo tiempo que se respetan las constricciones semánticas y se asegura la coherencia del tren de bits. Sin embargo, no proporciona información sobre las características de la temporización ni sobre las propiedades de la memoria tampón. En consecuencia con esos puntos de empalme, un empalme liso sólo puede efectuarse por convenios privados, o estudiando la cabida útil de los paquetes de tren de transporte, y rastreando los valores del estado de la memoria tampón y de las indicaciones de tiempo.

K.1.2 Puntos de empalme liso

Si el campo de bandera de empalme liso está presente y su valor es '1', la información es dada por el punto de empalme, indicando algunas propiedades del tren 'antiguo'. Esta información no está destinada a los decodificadores; su finalidad principal es facilitar el empalme liso. Tal punto de empalme se denomina punto de empalme liso. Se da la siguiente información:

- el tiempo de decodificación lisa, que se codifica como un valor DTS en el campo `DTS_next_AU`. Este valor DTS se expresa en la base de tiempo que es válida en el paquete de empalme;
- en el caso de un tren elemental de vídeo, las constricciones aplicadas al tren 'antiguo' cuando fue generado, tendientes a facilitar el empalme liso. Estas condiciones se dan por el valor del campo de tipo de empalme en la tabla correspondiente al perfil y el nivel del tren de vídeo.

Obsérvese que un punto de empalme liso puede utilizarse como un punto de empalme ordinario, para lo cual se descarta la información adicional. Esta información puede utilizarse también si se considera de interés para efectuar empalmes no lisos, o para otros fines que sean de empalme.

K.2 Comportamiento del codificador en caso de empalmes

K.2.1 Caso de empalmes no lisos

Como se ha dicho anteriormente, un empalme no liso es un empalme que produce una discontinuidad en la decodificación.

Debe señalarse que, con tal empalme, hay que cumplir las constricciones relativas a la discontinuidad de la codificación (véase 2.7.6). En particular:

- se codificará una PTS para la primera unidad de acceso del tren 'nuevo' (excepto durante el funcionamiento en modo truco o cuando `low_delay = '1'`);
- el instante de decodificación derivado de esta PTS (o de la DTS asociada) no llegará antes que el instante de decodificación lisa;
- en el caso de un tren elemental de vídeo, si el paquete de empalme no termina en un código de fin de secuencia, el tren 'nuevo' empezará con un código de fin de secuencia seguido inmediatamente de un encabezamiento() de secuencia.

Teóricamente, como estos empalmes introducen discontinuidades de decodificación, dan por resultado una presentación no continua de unidades de presentación (esto es, hay un tiempo muerto de duración variable entre las visualizaciones de dos imágenes consecutivas, o entre dos tramas de audio consecutivas). En la práctica, el resultado dependerá de la implementación del decodificador, especialmente en vídeo. Con algunos decodificadores de vídeo, la congelación de una o más imágenes puede ser la solución preferida. Véase la parte 4 de ISO/CEI 13818.

K.2.2 Caso de empalmes lisos

La finalidad de que no haya discontinuidades en la decodificación es que permite no tener discontinuidades en la presentación. En el caso de audio, esto siempre puede asegurarse. En cambio, debe señalarse que en caso de vídeo la continuidad de la presentación es teóricamente imposible en los dos casos indicados a continuación:

- 1) El tren 'antiguo' termina al final de una secuencia de retardo bajo y el tren 'nuevo' comienza al principio de una secuencia cuyo retardo no es bajo.
- 2) El tren 'antiguo' termina al final de una secuencia de retardo no bajo y el tren 'nuevo' comienza al principio de una secuencia de retardo bajo.

Los efectos inducidos por estas situaciones dependen de la implementación. Por ejemplo, en el primer caso, puede que haya que presentar una imagen durante dos periodos de trama, y, en el segundo caso, puede que haya que saltar una imagen. Sin embargo, es técnicamente posible que algunas implementaciones puedan soportar en esas situaciones sin efectos indeseables.

Además, en lo tocante a 6.1.1.6 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-2, un código de fin de secuencia deberá estar presente antes del primer encabezamiento de secuencia del tren 'nuevo' si por lo menos un parámetro (es decir, un parámetro definido en el encabezamiento de secuencia o en una extensión de encabezamiento de secuencia) tiene un valor diferente en los dos trenes, con la sola excepción de los que definen la matriz de cuantificación. Por ejemplo, si el campo de velocidad binaria no tiene el mismo valor en el tren 'nuevo' que en el 'antiguo', deberá estar presente un código de fin de secuencia. Por tanto, si el paquete de empalme no termina en un código de fin de secuencia, el tren 'nuevo' empezará con un código de fin de secuencia, seguido por un encabezamiento de secuencia.

De acuerdo con el párrafo anterior, un código de fin de secuencia será obligatorio en todos los empalmes, incluso en los lisos. Debe señalarse que la Rec. UIT-T H.262 | ISO/CEI 13818-2 especifica el proceso de decodificación de secuencias de vídeo [es decir, de los datos comprendidos entre un encabezamiento de secuencia (`sequence_header()`) y un código de fin de secuencia (`sequence_end_code()`)], y no se especifica nada en cuanto a la manera de tratar un cambio de secuencia. Por tanto, para el comportamiento de los decodificadores cuando aparecen esos empalmes, véase la parte 4 de ISO/CEI 13818.

K.2.3 Desbordamiento de las memorias tampón

Aunque dos trenes elementales respondan individualmente al modelo T-STD, ello no garantiza necesariamente que, después de empalmados uno al otro, ninguna de las memorias tampón STD desbordará en el intervalo de tiempo en el que los bits de ambos trenes se encuentran en ellas.

En el caso de vídeo de velocidad constante, si no se han aplicado condiciones particulares al tren 'antiguo', y si no se han tomado precauciones especiales para el empalme, el desbordamiento es posible cuando la velocidad binaria de vídeo del tren 'nuevo' es mayor que la velocidad binaria de vídeo del 'antiguo'. En realidad, es cierto que las memorias tampón MB_n y EB_n del T-STD no desbordan si se entregan bits al T-STD a la velocidad 'antigua'. Pero si la velocidad de entrega se conmuta a un valor mayor a la entrada de TB_n antes de que se hayan suprimido bits 'antiguos' en el T-STD, las memorias tampón STD estarán más llenas que si el tren 'antiguo' hubiera seguido sin empalme, y se puede producir el desbordamiento de EB_n y/o MB_n . En el caso de vídeo de velocidad variable, el mismo problema puede presentarse si la velocidad de entrega del tren 'nuevo' es mayor que la prevista cuando se creó el tren 'antiguo'. Esta situación está prohibida.

No obstante, el codificador que genera el tren 'antiguo' tiene la posibilidad de imponer condiciones adicionales a la gestión de la memoria tampón VBV en las proximidades del punto de empalme en previsión de cualquier velocidad binaria de vídeo 'nueva' que sea menor que un valor elegido. Por ejemplo, en el caso de un punto de empalme liso, estas condiciones adicionales pueden indicarse por un valor 'splice_type' (tipo de empalme) al que corresponden entradas en los cuadros 2-7 a 2-20 para 'splice-decoding-delay' (retardo de codificación con empalme) y 'max-splice-rate' (retardo máximo con empalme). En este caso, si la velocidad binaria de vídeo del tren 'nuevo' es menor que la velocidad máxima de empalme, queda asegurado que el tren empalmado no provocará desbordamiento en el intervalo de tiempo durante el cual bits de ambos trenes se encuentran en la memoria tampón T-STD.

Cuando no se han impuesto esas condiciones, este problema puede evitarse introduciendo un tiempo muerto para la entrega de bits entre el tren 'antiguo' y el 'nuevo', para dar tiempo a que las memorias tampón T-STD se vacíen suficientemente antes de entregarles bits del tren 'nuevo'. Si designamos por t_{in} el instante en que el último byte de la última unidad de acceso del tren 'antiguo' entra en el STD, y por t_{out} el instante en que sale del STD, basta con asegurar que en el intervalo de tiempo $[t_{in}, t_{out}]$ no entran T-STD más bits con el tren empalmado que si el tren 'antiguo' hubiera seguido sin empalme. Por ejemplo, en el caso de que el tren 'antiguo' tenga una velocidad binaria constante R_{old} y el 'nuevo' una velocidad binaria constante R_{new} , para evitar este riesgo de desbordamiento basta con introducir un tiempo muerto T_d que satisfaga las siguientes condiciones:

$$T_d \geq 0 \text{ y } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$

Anexo L

Procedimiento de registro (véase 2.9)

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

L.1 Procedimiento para solicitar un identificador registrado (RID)

Los interesados en solicitar un RID deberán dirigirse a la autoridad de registro, que pondrá a disposición los formularios correspondientes. La información que el solicitante debe proporcionar se indica en L.3. Pueden presentar solicitudes tanto empresas como organizaciones.

L.2 Responsabilidades de la autoridad de registro

En esta subcláusula se describen las principales responsabilidades de la autoridad de registro encargada de registrar identificadores de derechos de autor; en las directrices del JTC 1 se indican algunas otras responsabilidades. La autoridad de registro deberá:

- a) implementar un procedimiento de registro de solicitudes de RID único, de conformidad con el anexo H a las directrices del JTC 1;
- b) recibir y procesar las solicitudes de asignación de identificador de código de tipo de trabajo procedentes de la autoridad de registro de derechos de autor;
- c) precisar cuáles de las solicitudes recibidas se ajustan a dicho procedimiento de registro, y comunicar al solicitante el RID que le fue asignado en un plazo de 30 días a partir de la recepción de la solicitud;
- d) si la petición fuese rechazada, informar por escrito al respecto al solicitante en un plazo de 30 días a partir de la recepción de su solicitud, e instruirle sobre cómo ha de proceder en caso de apelación;
- e) mantener un registro preciso de RID asignados. La autoridad de registro aceptará y conservará cualquier revisión en relación con los datos de referencia y a las especificaciones técnicas de la organización;
- f) hacer que el contenido de ese registro esté a disposición de cualquier parte interesada que solicite su consulta;
- g) conservar en una base de datos las solicitudes de RID concedidas y denegadas. Aquellas partes que busquen información técnica sobre los formatos de datos privados que detenten un identificador de derechos de autor tendrán acceso a la información que forma parte de la base de datos mantenida por la autoridad de registro;
- h) elevar un informe de sus actividades al JTC 1, ITTF, y a la secretaría del JTC 1/SC 29 o a sus respectivos cesionarios, una vez al año y en función de un programa concertado de común acuerdo.

L.2.1 Datos de referencia para la autoridad de registro

Nombre de la organización:

Dirección:

Teléfono:

Fax:

L.3 Responsabilidades de las partes que soliciten un RID

A los fines de la identificación de los derechos de autor aquellas que soliciten un RID deberán:

- a) presentar su solicitud empleando el formulario suministrado y siguiendo los procedimientos indicados a tal efecto por la autoridad de registro;
- b) aportar datos de referencia en los que se puntualice la manera de obtener una descripción completa de la organización de registro de derechos de autor de forma no discriminatoria;
- c) incluir los detalles técnicos de la sintaxis y la semántica de los formatos de datos empleados para describir los trabajos audiovisuales u otros trabajos protegidos por derechos de autor dentro del campo información de derechos de autor adicional (**additional_copyright_info**). Una vez registrada, la sintaxis de la información de derechos de autor adicional no debe cambiar;

- d) convenir en que el empleo pretendido del identificador de derechos de autor concedido se instituya en un plazo de tiempo razonable;
- e) mantener un registro permanente del formulario de solicitud y de la notificación enviada por la autoridad de registro relativos a cada identificador de derechos de autor otorgado.

L.4 Procedimiento de apelación contra denegaciones de solicitudes

Se formará un grupo de gestión de registros que tendrá jurisdicción sobre las apelaciones relativas a las peticiones de RID denegadas. El grupo de gestión de registros (RMG) estará constituido por miembros designados por los miembros P y L del órgano técnico de la ISO responsable de este aspecto de la Recomendación | Norma Internacional. El grupo tendrá un presidente y una secretaria elegidos entre sus miembros. La autoridad de registro tiene la facultad de nombrar un miembro observador sin derecho a voto.

Las responsabilidades del RMG serán:

- a) examinar y dar curso a todas las apelaciones en un plazo de tiempo razonable;
- b) informar por escrito a las organizaciones que presenten una apelación para que se reconsideren sus peticiones, acerca de las disposiciones del RMG sobre la materia;
- c) examinar el informe anual en el que se resumen las actividades de la autoridad de registro;
- d) proporcionar a los organismos miembros de la ISO la información sobre las competencias de la autoridad de registro.

Anexo M**Formulario de solicitud de registro (véase 2.9)**

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

M.1 Datos de referencia de la organización que solicita un identificador registrado (RID)

Nombre de la organización:

Dirección:

Teléfono:

Fax:

Correo electrónico:

M.2 Declaración de intención de aplicar el RID asignado

Campo de aplicación del RID: según directrices que la autoridad de registro proporcionará a esos efectos.

M.3 Fecha prevista de aplicación del RID**M.4 Representante autorizado**

Nombre:

Título:

Dirección:

Firma: _____

M.5 Sólo para uso oficial de la autoridad de registro

Registro denegado: _____

Razón por la que se denegó la solicitud:

Registro concedido: _____ Valor de registro: _____

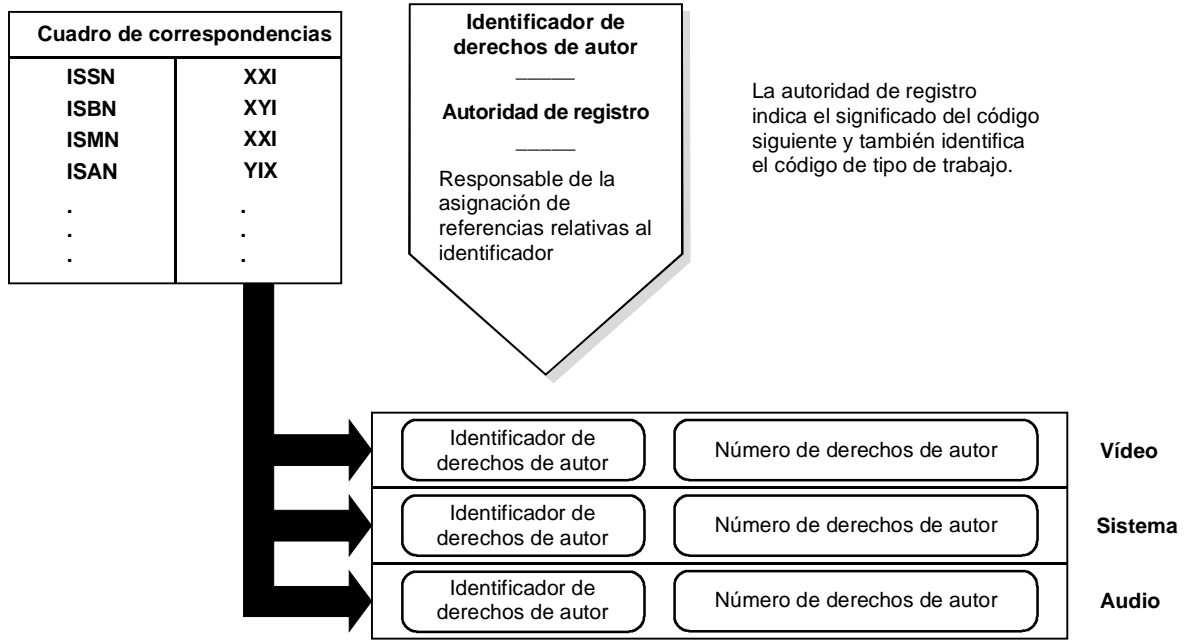
Adjunto 1 – Detalles técnicos del formato de datos registrado.

Adjunto 2 – Notificación del procedimiento de apelación contra denegación de solicitudes.

Anexo N

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

Autoridad de registro Organigrama de la administración (véase 2.9)



TISO8190-97/d30

Ejemplos

Identificador de derechos de autor	Número de derechos de autor
I.S.B.N. (para libros)	2-11-0725 575 (número ISBN)
I.S.A.N. (para trabajos audiovisuales)	1234567890123456 (número ISAN)

TISO8200-97/d31

Todos los identificadores de derechos de autor son registrados por la autoridad de registro, aunque sólo cuando se trata de los números de derechos de autor normalizados por la ISO. Cada organización que asigna números de derechos de autor solicita a la autoridad de registro un identificador de derechos de autor específico. Por ejemplo, el Staatsbibliothek Preussischer Kulturbesitz, designado por la ISO para gestionar el I.S.B.N., pide a la autoridad de registro un identificador de derechos de autor específico para la numeración de libros.

Anexo O

Procedimiento de registro (véase 2.10)

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

O.1 Procedimiento para solicitar un identificador registrado (RID)

Los interesados en solicitar un RID deberán dirigirse a la autoridad de registro, que pondrá a disposición los formularios correspondientes. La información que el solicitante debe proporcionar se indica en O.4. Pueden presentar solicitudes tanto empresas como organizaciones.

O.2 Responsabilidades de la autoridad de registro

En este anexo se describen las principales responsabilidades de la autoridad de registro encargada de registrar identificadores de formatos de datos privados; en las directrices del JTC 1 se indican algunas otras responsabilidades. La autoridad de registro deberá:

- a) implementar un procedimiento de registro de solicitudes de RID único de conformidad con las directrices del JTC 1;
- b) recibir y procesar las solicitudes de asignación de identificador que envíen los solicitantes;
- c) precisar cuáles de las solicitudes recibidas se ajustan a dicho procedimiento de registro, y comunicar al solicitante el RID que le fue asignado en un plazo de 30 días a partir de la recepción de la solicitud;
- d) si la petición de RID fuese rechazada informar por escrito al respecto al solicitante en un plazo de 30 días a partir de la recepción de su solicitud, y considerar la presentación de nuevas solicitudes oportunamente;
- e) mantener un registro preciso de identificadores asignados. La autoridad de registro aceptará y conservará cualquier revisión en relación con las especificaciones de formato;
- f) hacer que el contenido de este registro esté a disposición de los organismos nacionales del JTC 1 miembros de la ISO o de la CEI, las organizaciones de enlace de la ISO o de la CEI así como cualquier parte interesada que solicite su consulta;
- g) conservar en una base de datos las solicitudes de RID concedidas y denegadas. Aquellas partes que busquen información técnica sobre los formatos de datos privados que detentan un RID tendrán acceso a la información que forma parte de la base de datos mantenida por la autoridad de registro;
- h) elevar un informe de sus actividades al JTC 1, ITTF, y a la secretaría del SC 29 o a sus respectivos cesionarios, una vez al año;
- i) cuando sea posible, facilitar el empleo de los RID existentes.

O.3 Datos de referencia para la autoridad de registro

O.4 Responsabilidades de las partes que solicitan un RID

Las partes que soliciten un identificador de formato deberán:

- a) presentar su solicitud empleando el formulario suministrado y siguiendo los procedimientos indicados por la autoridad de registro;
- b) incluir una descripción del propósito del tren binario registrado, así como los detalles técnicos requeridos según lo especificado en el formulario de solicitud;
- c) aportar datos de referencia en los que se puntualice la manera de obtener una descripción completa de forma no discriminatoria;
- d) convenir en que el empleo pretendido del RID concedido se instituya en un plazo de tiempo razonable;
- e) mantener un registro permanente del formulario de solicitud y de la notificación enviada por la autoridad de registro relativos a cada RID otorgado.

O.5 Procedimiento de apelación contra denegaciones de solicitudes

Se formará un grupo de gestión de registros que tendrá jurisdicción sobre las apelaciones relativas a peticiones de RID denegadas. El grupo de gestión de registros (RMG) estará constituido por miembros designados por los miembros P y L del comité técnico de la ISO responsable de esta Especificación. El grupo tendrá un presidente y una secretaria elegidos entre sus miembros. La autoridad de registro tiene la facultad de nombrar un miembro observador sin derecho a voto.

ISO/CEI 13818-1:2007 (S)

Las responsabilidades del RMG serán:

- a) examinar y dar curso a todas las apelaciones en un plazo de tiempo razonable;
- b) informar por escrito a las organizaciones que presenten una apelación para que se reconsideren sus peticiones acerca de las disposiciones del RMG sobre la materia;
- c) examinar el informe anual en el que se resumen las actividades de la autoridad de registro;
- d) proporcionar a los organismos miembros de la ISO y a los Comités Nacionales del CEI información sobre las competencias de la autoridad de registro.

Anexo P**Formulario de solicitud de registro**

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

P.1 Datos de referencia de la organización que solicita un identificador registrado (RID)

Nombre de la organización:

Dirección:

Teléfono:

Fax:

Correo electrónico:

Télex:

P.2 Solicitud de un RID específico

NOTA – Si el sistema ya está establecido y se encuentra en uso, complete este ítem, siga con el ítem P.3 y pase directamente al ítem P.6. De no ser así, deje este espacio en blanco y siga con el ítem P.4.

P.3 Breve descripción del RID en uso y fecha en que se estableció el sistema**P.4 Declaración de intención de aplicar el RID asignado****P.5 Fecha prevista de aplicación del RID****P.6 Representante autorizado**

Nombre:

Título:

Dirección:

Firma: _____

P.7 Para uso oficial de la autoridad de registro

Registro denegado: _____

Razón por la que se denegó la solicitud:

Registro concedido: _____ Valor de registro: _____

Adjunto 1 – Detalles técnicos del formato de datos registrado.

Adjunto 2 – Notificación del procedimiento de apelación contra denegación de solicitudes.

Anexo Q

Modelos de memoria tampón de T-STD y P-STD para ADTS de ISO/CEI 13818-7

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

Q.1 Introducción

En 2.4.2 se define el modelo de decodificador-objetivo de sistemas del tren de transporte para trenes audio. En el presente anexo se describe el modelo de memoria tampón para ADTS de ISO/CEI 13818-7.

Los trenes de audio ADTS conforme a ISO/CEI 13818-7 se pueden reconocer en un multiplex de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 mediante la presencia de `stream_id=0x110yyyyy` ['y' = intrascendente (don't care)] y de `stream_type=0x0F`, como se define en los cuadros 2-22 y 2-23.

Q.2 Velocidad de fuga de la memoria tampón de transporte

Para audio distinto del de ISO/CEI 13818-7 con ADTS, la velocidad de fuga de la memoria tampón de transporte es de 2 Mbit/s. Esta velocidad es, no obstante, inferior a la velocidad máxima de ISO/CEI 13818-7 con ADTS. Por ello, la velocidad de fuga de tren de ISO/CEI 13818-7 con ADTS se fija a un valor distinto del de los trenes de audio de ISO/CEI 11172-3 e ISO/CEI 13818-3.

El tren elemental de ISO/CEI 13818-7 con ADTS consta de uno o más canales. La velocidad máxima de cada canal es de 576 kbit/s, siendo la frecuencia de muestreo de 96 kHz. La velocidad de fuga para ADTS de ISO/CEI 13818-7 se calcula, por consiguiente, aplicando la siguiente ecuación:

$$R_{x_n} = 1,2 \times R_{\text{m\acute{a}x}} \times N \text{ bits por segundo}$$

donde:

$R_{\text{m\acute{a}x}}$ es una velocidad constante de 576 kbit/s que se define en 3.2.2 de ISO/CEI 13818-7. Es el límite superior de la velocidad binaria por canal del tren de codificación audio de alta calidad (ACC, *advanced audio coding*) con ADTS correspondiente al valor máximo de la velocidad de muestreo (a saber, $F_s = 96$ kHz);

y donde:

N es el número de canales de audio que requieren su propia memoria tampón del decodificador (es decir, trenes de canales individuales en un solo elemento de canal o elemento de par de canales y elementos de canal con acoplamiento conmutado).

Q.3 Tamaño de memoria tampón

Para audio distinto del de ISO/CEI 13818-7 con ADTS, el tamaño de la memoria tampón principal es de 3584 bytes. Este tamaño es, no obstante, inferior al tamaño máximo de la memoria tampón de entrada del decodificador de audio ISO/CEI 13818-7 con ADTS. Por ello, el tamaño de la memoria tampón principal del tren de ISO/CEI 13818-7 con ADTS se fija a un valor distinto del de los trenes de audio del ISO/CEI 11172-3 e ISO/CEI 13818-3.

El tamaño de la memoria tampón principal para audio de ISO/CEI 13818-7 con ADTS se calcula como sigue:

$$BS_n = BS_{\text{mux}} + BS_{\text{dec}} + BS_{\text{oh}}$$

donde BS_{oh} , tamaño de la memoria tampón para tara de paquete PES, se define como:

$$BS_{\text{oh}} = 528 \text{ bytes}$$

y BS_{mux} , tamaño de la memoria tampón de multiplexación adicional, se define como:

$$BS_{\text{mux}} = 0,004 \text{ segundos} \times R_{\text{m\acute{a}x}} \times N$$

y BS_{dec} , tamaño de la memoria tampón de la unidad de acceso, se define como:

$$BS_{\text{dec}} = 6144 \text{ bits} \times N$$

donde:

$R_{\text{máx}}$ es una velocidad constante de 576 kbit/s que se define en 3.2.2 de ISO/CEI 13818-7. Es el límite superior de la velocidad binaria por canal del tren AAC con ADTS correspondiente al valor máximo de la frecuencia de muestreo (a saber, $F_s = 96$ kHz);

y donde:

N es el número de canales de audio que requieren su propia memoria tampón del decodificador (es decir, trenes de canales individuales en un solo elemento de canal o elemento de par de canales y elementos de canal con acoplamiento conmutado).

Q.3.1 TBS_n : igual que el de otro audio

Por lo que se refiere a la memoria tampón alisadora, no hay diferencia en la TB_n entre el tren ISO/CEI 13818-7 con ADTS y otros trenes de audio. En consecuencia, no es necesario cambiar el TBS_n , que es el tamaño de la TB_n .

Q.3.2 BS_{mux} : diferente del de otro audio

BS_{mux} , tamaño de la memoria tampón de multiplexación adicional deberá cambiarse para que sea acepte hasta 4 ms de fluctuación de fase. Este procedimiento es similar al aplicado para otros trenes en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

Q.3.3 BS_{dec} : diferente del de otro audio

BS_{dec} , tamaño de la memoria tampón de la unidad de acceso, se basa en el tamaño de la memoria tampón de entrada del decodificador para el tren elemental. Como se define en 3.2.2 de ISO/CEI 13818-7, el tamaño total de la memoria tampón de entrada del decodificador es de 6144 bits multiplicado por el número de canales que requiere cada una de sus memorias tampón de entrada de decodificador.

Q.3.4 BS_{oh} : diferente del de otro audio

El BS_{oh} corresponde a la tara de encabezamiento de paquetes PES.

En 2.4.2.6,

El tiempo empleado por cualquier dato para pasar por las memorias tampón de los decodificadores objetivos de sistemas será igual o menor que un segundo, salvo para los datos de vídeo de imagen fija.

Además, en 2.7.4,

El tren de programa y el tren de transporte se construirán de modo que la diferencia máxima entre las indicaciones de tiempo de presentación codificadas referentes a cada tren de vídeo o audio elemental sea 0,7 segundos.

El BS_{oh} deberá fijarse en el tamaño apropiado que corresponda a la tara de encabezamiento de paquetes PES cuando el tren AAC se paquetice aplicando las reglas anteriores. El tamaño máximo del encabezamiento de paquetes PES es de 264 bytes. Por ello, $BS_{\text{oh}} = 528$ bytes, es decir, dos veces el tamaño máximo del encabezamiento de paquetes PES, garantiza que al menos dos encabezamientos de paquetes PES pueden entrar en la memoria tampón principal con independencia del tamaño del encabezamiento de los paquetes PES. Esto significa que se puede insertar un encabezamiento de paquetes PES con PTS a intervalos de menos de 0,7 segundos incluso cuando los datos de un segundo estén en la memoria tampón principal.

Ejemplo: frecuencia de muestreo de 48 kHz

El tamaño del encabezamiento de paquete PES sin campos opcionales, salvo el PTS, es de 18 bytes. El número de unidades de acceso de un segundo es de unas 47. Cuando los datos de un segundo se hallan en la memoria principal (es decir, el caso más desfavorable) la tara de encabezamiento de paquetes PES puede caber en el BS_{oh} paquetizando más de dos, o bien dos, unidades de acceso en un paquete.

$$\text{number_of_AU} = 48 \text{ kHz}/1024 = 46 \text{ 875 por segundo}$$

$$(\text{number_of_AU}/2) \times 18[\text{byte}] = 421 \text{ 875 bytes} < BS_{\text{oh}}$$

Encabezamientos de paquetes PES más frecuentes pueden caber en el BS_{oh} , si el tiempo empleado por cualquier dato para pasar por la memoria tampón principal es inferior a un segundo.

Q.4 Conclusión

El modelo de memoria tampón de decodificador deberá abarcar el tamaño máximo de la memoria tampón, no obstante, la codificación audio de alta calidad (AAC) puede tratar hasta 48 canales y velocidades binarias muy elevadas. Por ello,

ISO/CEI 13818-1:2007 (S)

se utilizan 3 niveles de número de canales, a saber, 2, 8 y 48, para definir la velocidad de fuga y el tamaño de la memoria tampón principal. En el caso de 2 canales, se utilizan la misma velocidad de fuga y el mismo tamaño de memoria tampón principal como valores convencionales para mantener la compatibilidad. En los otros casos (8 y 48 canales), se aplican las fórmulas propuestas.

Velocidad de fuga de T-STD para audio de ISO/CEI 13818-7 con ADTS

Número de canales	R _{xn} [bit/s]
1-2	2 000 000
3-8	5 529 600
9-12	8 294 400
13-48	33 177 600

Canales: El número de canales de salida de audio de anchura de banda completa más el número de elementos de canal con acoplamiento conmutado independientes dentro del mismo tren de audio elemental. Por ejemplo, en el caso típico en que no hay elementos de canal con acoplamiento conmutado independientes, mono es 1 canal, estéreo es 2 canales y canal ambifónico 5.1 es 5 canales (no se cuenta el canal LFE).

Tamaño de memoria tampón principal de T-STD para audio de ISO/CEI 13818-7 con ADTS

Número de canales	BS _n [bytes]
1-2	3 584
3-8	8 976
9-12	12 804
13-48	51 216

Canales: El número de canales de salida de audio de anchura de banda completa más el número de elementos de canal con acoplamiento conmutado independientes dentro del mismo tren de audio elemental. Por ejemplo, en el caso típico en que no hay elementos de canal con acoplamiento conmutado independientes, mono es 1 canal, estéreo es 2 canales y canal ambifónico 5.1 es 5 canales (no se cuenta el canal LFE).

Para tren de programa, el tamaño de memoria tampón principal anterior deberá fijarse en la P-STD_buffer_scale y el P-STD_buffer_size como sigue.

Número de canales	P-STD_buffer_scale	P-STD_buffer_size
1-2	0	28
3-8	0	71
9-48	0	401

Anexo R

Transporte de escenas ISO/CEI 14496 en trenes de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

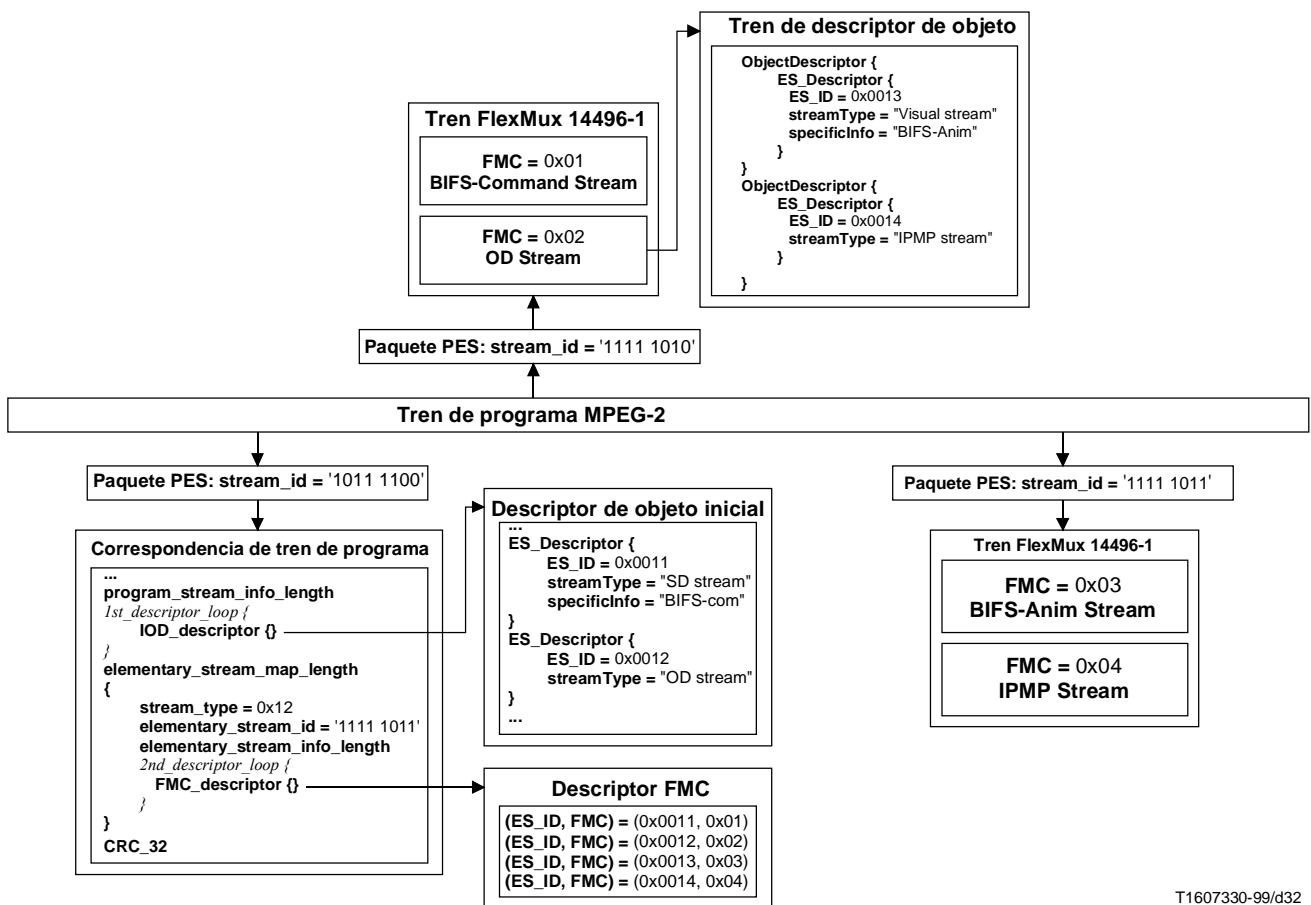
R.1 Procedimiento de acceso de contenido para componentes de programa de ISO/CEI 14496 dentro de un tren de programa

A continuación se proporciona un procedimiento de adquisición de receptor de referencia para acceder a elementos de programa ISO/CEI 14496 en un tren de programa de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Se supone aquí que el tren de programa incluye una correspondencia de trenes de programa (transportada en un paquete PES que tiene un stream_id igual a 0xBC):

- Adquirir la correspondencia de tren de programa.
- Identificar el descriptor IOD en el primer bucle de descriptor.

- Identificar los ES_ID de los trenes de descriptor de objeto, de descripción de escena y otros trenes descritos dentro del descriptor de objeto inicial.
- Adquirir el descriptor SL y el descriptor FMC en el segundo bucle de descriptor para elementary_stream_ids 0xFA y 0xFB, según proceda.
- Generar, a partir de estos descriptores, una tabla de correspondencia de trenes entre los ES_ID y los identificadores de tren elemental asociados, más el canal FlexMux, si procede.
- Localizar el tren de descriptor de objeto utilizando su ES_ID y la tabla de correspondencia de trenes.
- Localizar otros trenes descritos en el descriptor de objeto inicial utilizando su ES_ID y la tabla de correspondencia de trenes.
- Supervisar continuamente el tren de descriptor de objeto e identificar los ES_ID de trenes adicionales.
- Localizar los trenes adicionales utilizando su ES-D y la tabla de correspondencia de trenes.

La figura R.1 da un ejemplo de contenido ISO/CEI 14496 en un tren de programa, formado por trenes de descriptor de objeto, de descripción de escena (instrucción BIFS), de animación BIFS e IPMP. Todos los trenes ISO/CEI 14496 son multiplexados en un solo tren FlexMux.



T1607330-99/d32

Figura R.1 – Ejemplo de contenido ISO/CEI 14496 en un tren de programa

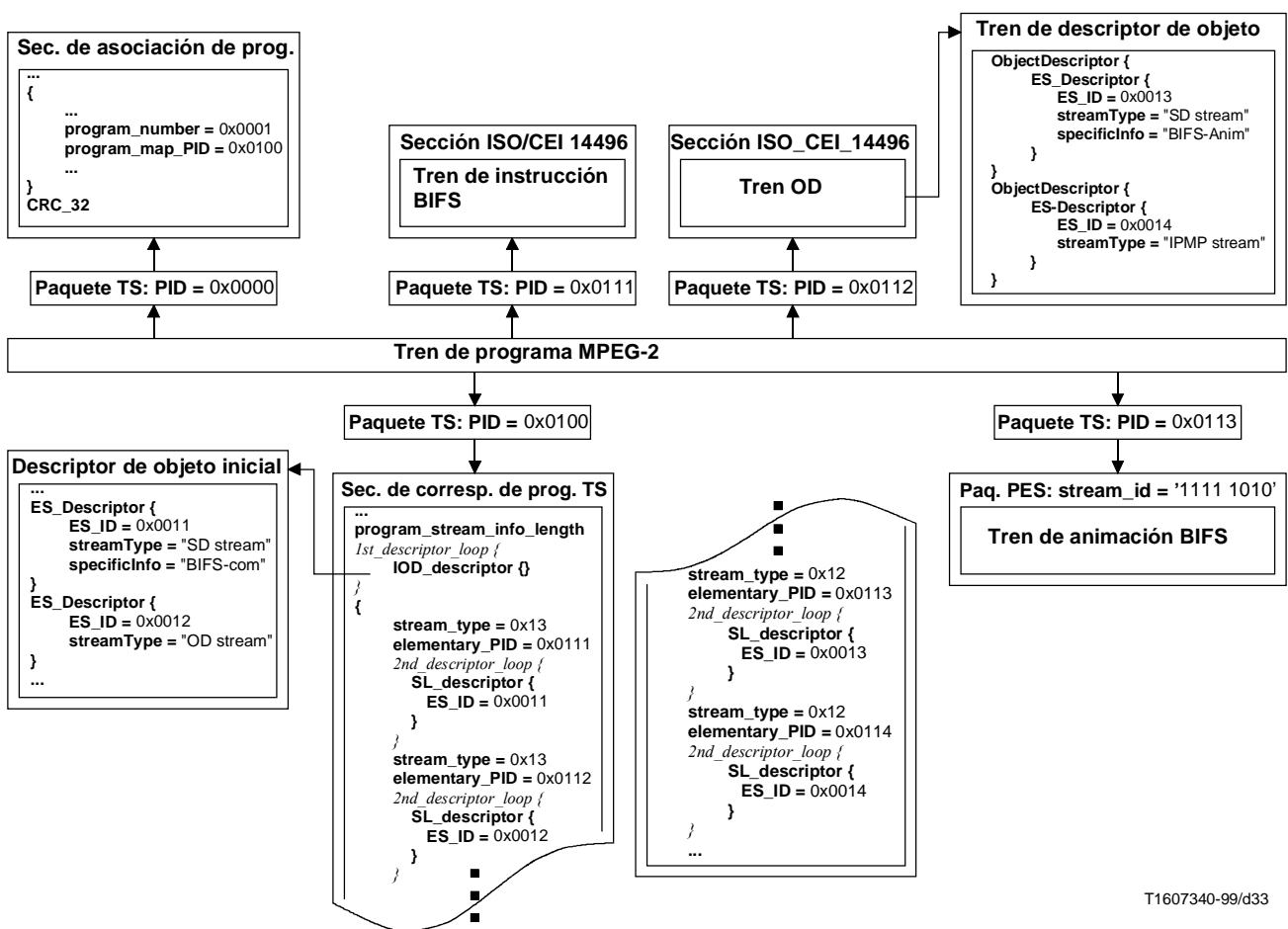
R.2 Procedimiento de acceso de contenido para componentes de programa de ISO/CEI 14496 dentro de un tren de transporte

A continuación se proporciona un procedimiento de adquisición de receptor de referencia para acceder a elementos de programa ISO/CEI 14496 en un tren de transporte de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1:

- Adquirir la tabla de correspondencia de programa para el programa deseado.
- Identificar el descriptor IOD en el primer bucle de descriptor.
- Identificar los ES_ID de los trenes de descriptor de objeto, de descripción de escena y otros trenes descritos dentro del descriptor de objeto inicial.

- Adquirir el descriptor SL y el descriptor FMC en el segundo bucle de descriptor para cualquiera de los PID elementales.
- Generar, a partir de estos descriptores, una tabla de correspondencia de trenes entre los ES_ID y los PID elementales asociados, más el canal FlexMux, si procede.
- Localizar el tren de descriptor de objeto utilizando su ES_ID y la tabla de correspondencia de trenes.
- Localizar otros trenes descritos en el descriptor de objeto inicial utilizando su ES_ID y la tabla de correspondencia de trenes.
- Supervisar continuamente el tren de descriptor de objeto e identificar los ES_ID de trenes adicionales.
- Localizar los trenes adicionales utilizando su ES-D y la tabla de correspondencia de trenes.

La figura R.2 da un ejemplo de elementos de programa ISO/CEI 14496 en un tren de transporte, formado por trenes de descriptor de objeto, de descripción de escena (instrucción BIFS), y trenes elementales de animación BIFS e IPMP. La instrucción BIFS y el tren OD son transportados por medio de secciones ISO_CEI_14496, mientras que los trenes elementales de animación BIFS e IPMP son transportados en paquetes PES referenciados por dos valores distintos de PID elemental, sin utilizar la herramienta FlexMex de ISO/CEI 14496-1.



T1607340-99/d33

Figura R.2 – Ejemplo de contenido ISO/CEI 14496 en un tren de transporte

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
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	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
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, comunicaciones de sistemas abiertos y seguridad
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet y Redes de la próxima generación
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación