



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**H.222.0**

(07/95)

**TRANSMISIÓN DE SEÑALES NO TELEFÓNICAS**

---

**TECNOLOGÍA DE LA INFORMACIÓN –  
CODIFICACIÓN GENÉRICA DE IMÁGENES EN  
MOVIMIENTO E INFORMACIÓN DE AUDIO  
ASOCIADA: SISTEMAS**

**Recomendación UIT-T H.222.0**

(Anteriormente «Recomendación del CCITT»)

---

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

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. El texto de la Recomendación UIT-T H.222.0 se aprobó el 10 de julio de 1995. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 13818-1.

---

### 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.

© UIT 1996

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

# ÍNDICE

|   | <i>Página</i> |
|---|---------------|
| Resumen .....   | vii           |
| Introducción.....   | vii           |
| Intro. 1    Tren de transporte .....  | viii          |
| Intro. 2    Tren de programa .....  | x             |
| Intro. 3    Conversión entre tren de transporte y tren de programa .....  | xi            |
| Intro. 4    Tren elemental paquetizado.....   | xii           |
| Intro. 5    Modelo de temporización.....  | xii           |
| Intro. 6    Acceso condicional .....  | xii           |
| Intro. 7    Operaciones de múltiplex.....   | xii           |
| Intro. 8    Operaciones de tren (capa de paquete PES).....  | xiii          |
| Intro. 8.1 Demultiplexación.....  | xiii          |
| Intro. 8.2 Sincronización.....  | xiii          |
| Intro. 8.3 Relación con la capa de compresión .....   | xiv           |
| Intro. 9    Decodificador de referencia de sistemas .....   | xiv           |
| Intro. 10   Aplicaciones .....  | xiv           |
| SECCIÓN 1 – GENERALIDADES.....  | 1             |
| 1.1 Alcance .....   | 1             |
| 1.2 Referencias normativas .....  | 1             |
| 1.3 Recomendaciones   Normas Internacionales idénticas.....   | 1             |
| 1.4 Referencias adicionales.....  | 2             |
| SECCIÓN 2 – ELEMENTOS TÉCNICOS.....   | 2             |
| 2.1 Definiciones .....  | 2             |
| 2.2 Símbolos y abreviaturas .....   | 5             |
| 2.2.1 Operadores aritméticos .....  | 5             |
| 2.2.2 Operadores lógicos.....   | 6             |
| 2.2.3 Operaciones relacionales.....   | 6             |
| 2.2.4 Operadores para bits .....  | 6             |
| 2.2.5 Asignación .....  | 6             |
| 2.2.6 Mnemónicos.....   | 6             |
| 2.2.7 Constantes .....  | 7             |
| 2.3 Método para describir la sintaxis del tren de bits.....   | 7             |
| 2.4 Requisitos del tren de bits del tren de transporte .....  | 8             |
| 2.4.1 Estructura y parámetros de codificación del tren de transporte .....                                      | 8             |
| 2.4.2 Decodificador-objetivo de sistemas del tren de transporte .....   | 9             |
| 2.4.2.1 Frecuencia de reloj de sistema .....  | 10            |
| 2.4.2.2 Entrada al decodificador-objetivo de sistemas del tren de transporte .....                              | 11            |
| 2.4.2.3 Almacenamiento en memoria tampón.....   | 12            |
| 2.4.2.4 Decodificación .....  | 17            |
| 2.4.2.5 Presentación .....  | 17            |
| 2.4.2.6 Gestión de las memorias tampón .....  | 17            |
| 2.4.3 Especificación de la sintaxis y semántica del tren de transporte.....                                     | 18            |
| 2.4.3.1 Tren de transporte .....  | 18            |
| 2.4.3.2 Capa de paquete del tren de transporte .....  | 18            |
| 2.4.3.3 Definición semántica de campos en la capa de paquete del tren de transporte .....                       | 18            |
| 2.4.3.4 Campo de adaptación.....  | 21            |
| 2.4.3.5 Definición semántica de campos en el campo de adaptación .....  | 21            |
| 2.4.3.6 Paquete PES .....   | 30            |
| 2.4.3.7 Definición semántica de campos en el paquete PES.....   | 30            |
| 2.4.3.8 Transmisión de trenes de programa y de trenes de sistemas ISO/CEI 11172-1 en el tren de transporte..... | 40            |

|          | <i>Página</i>  |
|----------|--|
| 2.4.4    | Información específica de programa ..... 41  |
| 2.4.4.1  | Puntero ..... 43   |
| 2.4.4.2  | Definición semántica de campos en la sintaxis de puntero ..... 43  |
| 2.4.4.3  | Tabla de asociación de programas ..... 43  |
| 2.4.4.4  | Asignaciones de identificador de tabla ..... 43  |
| 2.4.4.5  | Definición semántica de campos en la sección de asociación de programas .. 43                                |
| 2.4.4.6  | Tabla de acceso condicional ..... 45   |
| 2.4.4.7  | Definición semántica de campos en la sección de acceso condicional ..... 45                                  |
| 2.4.4.8  | Tabla de correspondencia de programas ..... 46   |
| 2.4.4.9  | Definición semántica de campos en la sección de correspondencia de programas del tren de transporte ..... 46 |
| 2.4.4.10 | Sintaxis de la sección privada ..... 47  |
| 2.4.4.11 | Definición semántica de campos en la sección privada ..... 48  |
| 2.5      | Requisitos del tren de bits del tren de programa ..... 48  |
| 2.5.1    | Estructura de codificación y parámetros del tren de programa ..... 48  |
| 2.5.2    | Decodificador-objetivo de sistemas del tren de programas ..... 50  |
| 2.5.2.1  | Frecuencia de reloj de sistema ..... 51  |
| 2.5.2.2  | Entrada al decodificador-objetivo de sistemas del tren de transporte ..... 51                                |
| 2.5.2.3  | Almacenamiento en memoria tampón..... 52   |
| 2.5.2.4  | Trenes PES..... 53   |
| 2.5.2.5  | Decodificación y presentación..... 54  |
| 2.5.3    | Especificación de la sintaxis y semántica del tren de programa ..... 54                                      |
| 2.5.3.1  | Tren de programa ..... 54  |
| 2.5.3.2  | Definición semántica de campos en el tren de programa..... 54  |
| 2.5.3.3  | Capa de paca del tren de programa ..... 54   |
| 2.5.3.4  | Definición semántica de campos en la paca de trenes de programa..... 54                                      |
| 2.5.3.5  | Encabezamiento de sistema..... 54  |
| 2.5.3.6  | Definición semántica de campos en el encabezamiento de sistema..... 56                                       |
| 2.5.3.7  | Capa de paquete del tren de programa ..... 58  |
| 2.5.4    | Correspondencia de trenes de programa ..... 58   |
| 2.5.4.1  | Sintaxis de la correspondencia de trenes de programa..... 58   |
| 2.5.4.2  | Definición semántica de campos en la correspondencia de trenes de programa ..... 58                          |
| 2.5.5    | Directorio del tren de programa ..... 59   |
| 2.5.5.1  | Sintaxis del paquete para el directorio del tren de programa ..... 60  |
| 2.5.5.2  | Definición semántica de campos en el directorio del tren de programa..... 60                                 |
| 2.6      | Descriptores de programas y de elementos de programa ..... 62  |
| 2.6.1    | Definición semántica de campos de descriptores de programas y de elementos de programa ..... 62              |
| 2.6.2    | Descriptor de tren de vídeo ..... 63   |
| 2.6.3    | Definición semántica de campos en el descriptor de tren de vídeo..... 63                                     |
| 2.6.4    | Descriptor de tren de audio ..... 64   |
| 2.6.5    | Definición semántica de campos en el descriptor de tren de audio..... 64                                     |
| 2.6.6    | Descriptor de jerarquía..... 64  |
| 2.6.7    | Definición semántica de campos en el descriptor de jerarquía ..... 65  |
| 2.6.8    | Descriptor de registro..... 65   |
| 2.6.9    | Definición semántica de campos en el descriptor de registro ..... 66   |
| 2.6.10   | Descriptor de alineación de tren de datos ..... 66   |
| 2.6.11   | Definición semántica de campos en el descriptor de alineación de tren de datos ..... 66                      |
| 2.6.12   | Descriptor de cuadrícula de fondo objetivo ..... 67  |
| 2.6.13   | Definición semántica de campos en el descriptor de cuadrícula de fondo objetivo..... 67                      |
| 2.6.15   | Definición semántica de campos en el descriptor de ventana de vídeo ..... 68                                 |
| 2.6.16   | Descriptor de acceso condicional..... 68   |
| 2.6.17   | Definición semántica de campos en el descriptor de acceso condicional ..... 69                               |
| 2.6.18   | Descriptor de idioma ISO 639 ..... 69  |
| 2.6.19   | Definición semántica de campos en el descriptor de idioma ISO 639 ..... 69                                   |
| 2.6.20   | Descriptor de reloj de sistema..... 70   |
| 2.6.21   | Definición semántica de campos en el descriptor de reloj de sistema ..... 70                                 |
| 2.6.22   | Descriptor de utilización de memoria tampón múltiplex ..... 70   |

|         |  |    |
|---------|--|----|
| 2.6.23  | Definición semántica de campos en el descriptor de utilización de memoria tampón múltiplex .....     | 71 |
| 2.6.24  | Descriptor de derechos de autor .....  | 71 |
| 2.6.25  | Definición semántica de campos en el descriptor de derechos de autor .....                           | 71 |
| 2.6.26  | Descriptor de velocidad binaria máxima.....  | 72 |
| 2.6.27  | Definición semántica de campos en el descriptor de velocidad binaria máxima .....                    | 72 |
| 2.6.28  | Descriptor de indicador de datos privados .....  | 72 |
| 2.6.29  | Definición semántica de campos en el descriptor de indicador de datos privados .....                 | 72 |
| 2.6.30  | Descriptor de memoria tampón alisadora.....  | 72 |
| 2.6.31  | Definición semántica de los campos del descriptor de memoria tampón alisadora .....                  | 73 |
| 2.6.32  | Descriptor de STD .....  | 73 |
| 2.6.33  | Definición semántica de los campos del descriptor de STD .....                                       | 73 |
| 2.6.34  | Descriptor de IBP.....   | 74 |
| 2.6.35  | Definición semántica de los campos del descriptor IBP .....  | 74 |
| 2.7     | Restricciones impuestas a la semántica del tren multiplexado.....                                    | 74 |
| 2.7.1   | Frecuencia de codificación de la referencia de reloj de sistema .....                                | 74 |
| 2.7.2   | Frecuencia de codificación de la referencia de reloj de programa .....                               | 74 |
| 2.7.3   | Frecuencia de codificación de la referencia de reloj de tren elemental.....                          | 75 |
| 2.7.4   | Frecuencia de codificación de indicación de tiempo de presentación.....                              | 75 |
| 2.7.5   | Codificación condicional de indicaciones de tiempo .....   | 75 |
| 2.7.6   | Constricciones de temporización para la codificación escalonable .....                               | 75 |
| 2.7.7   | Frecuencia de codificación de tamaño de memoria tampón P-STD en encabezamientos de paquete PES ..... | 76 |
| 2.7.8   | Codificación de encabezamiento de sistema en el tren de programa .....                               | 76 |
| 2.7.9   | Tren de programa de parámetros de sistema constreñidos.....  | 76 |
| 2.7.10  | Tren de transporte .....   | 77 |
| 2.8     | Compatibilidad con la Norma ISO/CEI 11172 .....  | 78 |
| Anexo A | – Modelo de decodificador CRC .....  | 79 |
| A.0     | Modelo de decodificador CRC .....  | 79 |
| Anexo B | – Instrucciones y control del medio de almacenamiento digital (DSM-CC).....                          | 80 |
| B.0     | Introducción .....   | 80 |
| B.0.1   | Finalidad .....  | 80 |
| B.0.2   | Aplicaciones futuras.....  | 80 |
| B.0.3   | Ventajas .....   | 80 |
| B.0.4   | Funciones básicas.....   | 81 |
| B.0.4.1 | Selección de trenes.....   | 81 |
| B.0.4.2 | Extracción .....   | 81 |
| B.0.4.3 | Almacenamiento .....   | 81 |
| B.1     | Elementos generales.....   | 81 |
| B.1.1   | Alcance .....  | 81 |
| B.1.2   | Visión de conjunto y aplicación del protocolo DSM-CC .....   | 81 |
| B.1.3   | Transmisión de instrucciones y acuses de recibo DSM-CC .....   | 82 |
| B.2     | Elementos técnicos.....  | 83 |
| B.2.1   | Definiciones .....   | 83 |
| B.2.2   | Especificación de la sintaxis DSM-CC .....   | 84 |
| B.2.3   | Semántica de campos en la especificación de sintaxis DSM-CC.....                                     | 84 |
| B.2.4   | Capa de control .....  | 85 |
| B.2.5   | Semántica de campos en la capa de control .....  | 85 |
| B.2.6   | Capa de acuse de recibo .....  | 87 |
| B.2.7   | Semántica de campos en la capa de acuse de recibo.....   | 87 |
| B.2.8   | Código de tiempo.....  | 88 |
| B.2.9   | Semántica de campos en el código de tiempo.....  | 88 |
| Anexo C | – Información específica de programa .....   | 89 |
| C.0     | Explicación de la información específica de programa en trenes de transporte.....                    | 89 |
| C.1     | Introducción .....   | 89 |
| C.2     | Mecanismo funcional.....   | 89 |

|         |   |     |
|---------|---|-----|
| C.3     | Correspondencia de secciones con paquetes del tren de transporte .....  | 90  |
| C.4     | Velocidades de repetición y acceso aleatorio.....   | 91  |
| C.5     | ¿Qué es un programa?.....   | 91  |
| C.6     | Atribución de número de programa .....  | 91  |
| C.7     | Utilización de PSI en un sistema típico.....  | 92  |
| C.8     | Relaciones de estructuras PSI .....   | 92  |
| C.8.1   | Tabla de asociación de programas .....  | 92  |
| C.8.2   | Tabla de correspondencia de programas .....   | 93  |
| C.8.3   | Tabla de acceso condicional .....   | 93  |
| C.8.4   | Tabla de información de red .....   | 93  |
| C.8.5   | Sección privada.....  | 93  |
| C.8.6   | Descriptores .....  | 93  |
| C.9     | Utilización de anchura de banda y tiempo de adquisición de señal .....  | 95  |
| Anexo D | – Modelo de temporización de sistemas de la presente Recomendación   Norma Internacional y repercusiones de aplicación..... | 98  |
| D.0     | Introducción .....  | 98  |
| D.0.1   | Modelo de temporización.....  | 98  |
| D.0.2   | Sincronización de presentación de audio y vídeo .....   | 100 |
| D.0.3   | Recuperación del reloj de tiempo de sistema en el decodificador .....   | 101 |
| D.0.4   | Fluctuación de fase de SCR y PCR.....   | 104 |
| D.0.5   | Recuperación de reloj en presencia de fluctuación de fase de red .....  | 105 |
| D.0.6   | Reloj de sistema utilizado para la generación de una subportadora de crominancia .....                                      | 105 |
| D.0.7   | Reconstrucción de vídeo y audio de componentes .....  | 107 |
| D.0.8   | Deslizamiento de trama.....   | 107 |
| D.0.9   | Mitigación de la fluctuación de fase de red .....   | 107 |
| Anexo E | – Aplicaciones de transmisión de datos.....   | 109 |
| E.0     | Consideraciones .....   | 109 |
| E.1     | Sugerencia.....   | 109 |
| Anexo F | – Gráficos de sintaxis para la presente Recomendación   Norma Internacional .....   | 110 |
| F.0     | Introducción .....  | 110 |
| F.0.1   | Sintaxis del tren de transporte.....  | 110 |
| F.0.2   | Paquete PES .....   | 111 |
| F.0.3   | Sección de asociación de programas .....  | 112 |
| F.0.4   | Sección CA .....  | 112 |
| F.0.5   | Sección de correspondencia de programas de tren de transporte.....  | 113 |
| F.0.6   | Sección privada.....  | 113 |
| F.0.7   | Tren de programa .....  | 114 |
| F.0.8   | Correspondencia de trenes de programa .....   | 115 |
| Anexo G | – Información general .....   | 116 |
| G.0     | Información general.....  | 116 |
| G.0.1   | Emulación de byte de sincronización.....  | 116 |
| G.0.2   | Estado de imágenes omitidas y proceso de decodificación.....  | 116 |
| G.0.3   | Selección de valores de PID.....  | 116 |
| G.0.4   | Emulación de código de comienzo de PES.....   | 116 |
| Anexo H | – Datos privados .....  | 117 |
| H.0     | Datos privados .....  | 117 |
| Anexo I | – Conformidad de sistemas e interfaz en tiempo real.....  | 119 |
| I.0     | Conformidad de sistemas e interfaz en tiempo real .....   | 119 |
| Anexo J | – Interfaz entre redes que inducen fluctuación de fase y decodificadores MPEG-2 .....                                       | 120 |
| J.0     | Introducción .....  | 120 |
| J.1     | Modelos de conformidad de red.....  | 120 |
| J.2     | Especificación de red para suprimir la fluctuación de fase .....  | 121 |
| J.3     | Ejemplo de realizaciones de decodificador.....  | 122 |
| J.3.1   | Adaptador de red seguido de un decodificador MPEG-2 .....   | 122 |
| J.3.2   | Decodificador integrado.....  | 122 |

|  | <i>Página</i> |
|--|---------------|
| Anexo K – Empalme de trenes de transporte .....              | 124           |
| K.0 Introducción .....                                       | 124           |
| K.1 Diferentes tipos de punto de empalme .....               | 124           |
| K.1.1 Puntos de empalme ordinario.....                       | 124           |
| K.1.2 Puntos de empalme liso.....                            | 125           |
| K.2 Comportamiento del codificador en caso de empalmes ..... | 125           |
| K2.1 Caso de empalmes no lisos .....                         | 125           |
| K.2.2 Caso de empalmes lisos .....                           | 125           |
| K.2.3 Desbordamiento de las memorias tampón .....            | 126           |





## Resumen

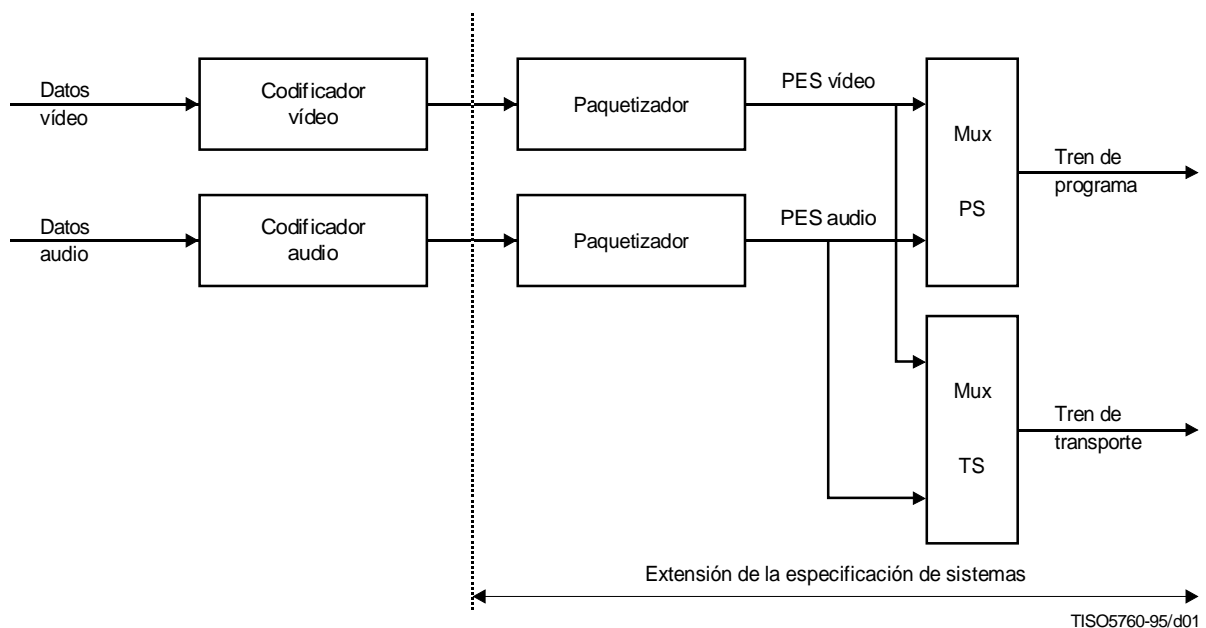
La presente Recomendación | Norma Internacional especifica métodos genéricos para la multiplexación, sincronización y recuperación de la base de tiempos en un entorno multimedia. Las especificaciones prevén una multiplexación multimedia basada en paquete, donde cada tren de bits elemental es segmentado en trenes elementales paquetizados, y después los respectivos paquetes son multiplexados en uno de los dos trenes siguientes: el tren de programa, que es un múltiplex de paquetes del tren elemental paquetizado de longitud variable y que está diseñado para utilización en entornos sin errores, y el tren de transporte, que consiste en paquetes de longitud fija de 188 bytes, que tiene la funcionalidad de multiplexación de múltiples programas así como la multiplexación de diversos paquetes del tren elemental paquetizado y que está diseñado para utilización en entornos propensos a errores. La sincronización y la recuperación de la base de tiempos multimedia se logran mediante indicaciones de tiempo para el reloj de temporización del sistema y la presentación/decodificación.

## 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 la Norma 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 la Norma 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 realización 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 realización 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 realización 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 y la Norma 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.

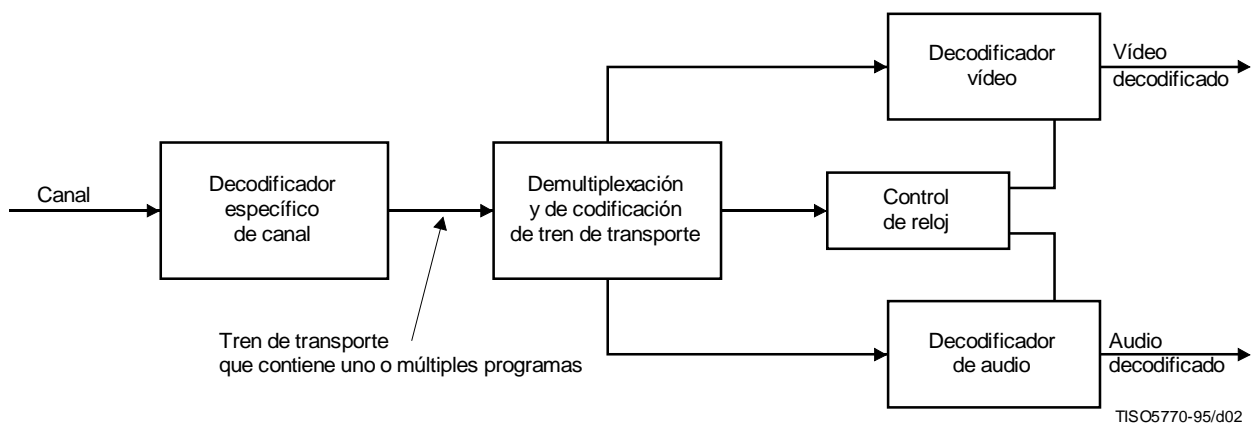
Las Figuras Intro. 2 e Intro. 3 ilustran prototipos de sistemas de demultiplexación y decodificación, que toman como entrada un tren de transporte. La Figura Intro. 2 ilustra el primer caso, cuando un tren de transporte es demultiplexado y decodificado directamente. Los trenes de transporte se construyen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

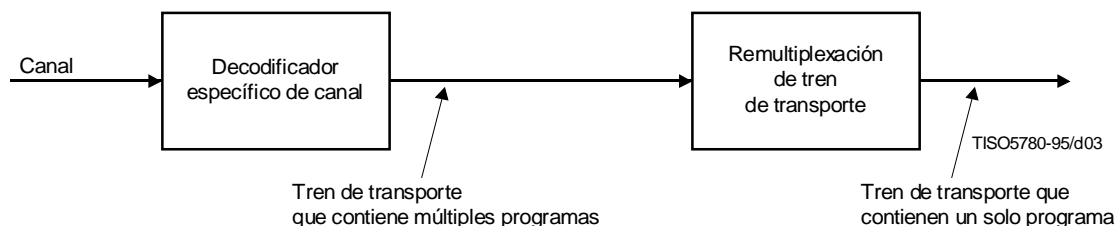
Las operaciones realizadas por el prototipo de decodificador que acepta trenes de transporte se aplican a todo el tren de transporte («operaciones de múltiplex»), o a trenes elementales («operaciones específicas de tren»). La capa de sistemas del tren de transporte se divide en dos subcapas, una para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

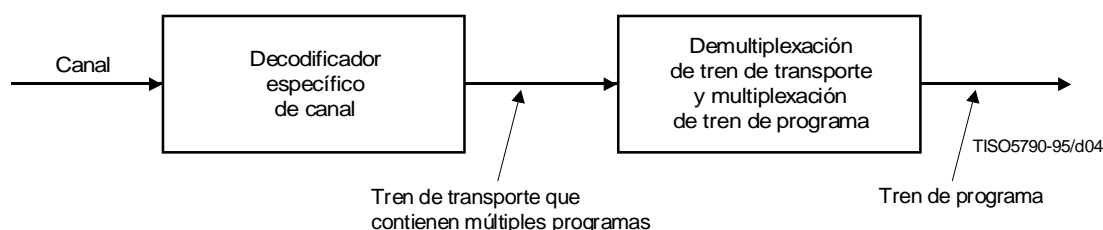
La Figura Intro. 3 ilustra el segundo caso, en el que un tren de transporte que contiene múltiples programas se convierte en un tren de transporte que contiene un solo programa. En este caso, la operación de remultiplexación puede necesitar la corrección de valores de la referencia de reloj de programa (PCR, *programme clock reference*) 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 realizaciones específicas de demultiplexores o decodificadores no tienen que incluir todas estas funciones.



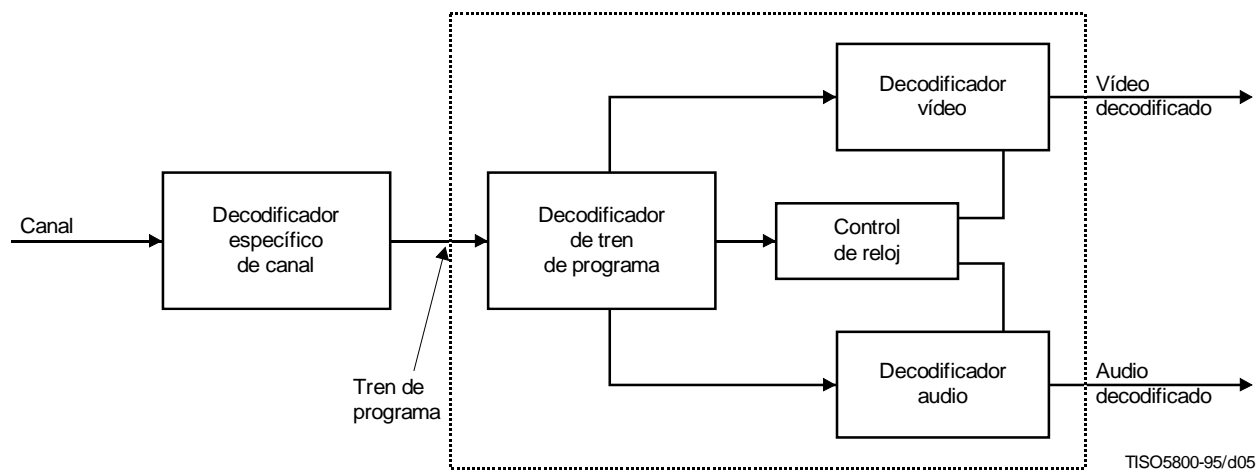
**Figura Intro. 4 – Prototipo de conversión de tren de transporte a tren de programa**

## 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 las referencias de reloj de sistemas (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 admiten 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 la Norma 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 paca (*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 paca, 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 realizaciones 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 realizar sistemas con retardo de extremo a extremo constante. Es posible realizar 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 sistemas 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**

El cifrado y la aleatorización para acceso condicional a programas codificados en los trenes de programa y de transporte son sustentados 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 realización 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. 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 la Norma 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 la Norma 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 la Norma ISO/CEI 13818 emplea un «decodificador de sistemas» que se considera como un modelo u objetivo (STD, *system target decoder*), 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 sustentar el vídeo de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 y el audio de la Norma 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 multimedios 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 la Norma 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 sustentar la combinación de los métodos de codificación de vídeo y audio definidos en las Partes 2 y 3 de la Norma ISO/CEI 13818. La capa de sistemas sustenta cinco 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; y
- 5) la identificación de tiempo.

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 sustentan la multiplexación de trenes comprimidos de vídeo y audio de un programa con una base de tiempos común. El **tren de transporte** sustenta 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 admite 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 admitida 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 Internacionales son objeto de revisiones, por lo que se preconiza que las partes en acuerdos basados en esta Recomendación | Norma Internacional investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones | Normas Internacionales indicadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales vigentes. La Oficina de Normalización de Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T vigentes.

**1.3 Recomendaciones | Normas Internacionales idénticas**

- Recomendación UIT-T H.262 (1995) | ISO/CEI 13818-2:…<sup>1)</sup>, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada – Vídeo*.

---

<sup>1)</sup> Por publicar.

## 1.4 Referencias adicionales

- ISO 639-2:...<sup>2)</sup>, *Codes for the representation of names of languages – Part 2: Alpha-3 code*.
- ISO 8859-1:1987, *Information processing – 8 bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*.
- 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 13522-1:...<sup>2)</sup>, *Information technology – Coding of Multimedia and Hypermedia information – Part 1: MHEG object representation – Base notation (ASN.1)*.
- ISO/CEI 13818-3:1995, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*.
- Recomendación UIT-R BT.601-3, *Parámetros de codificación de televisión digital para estudios*.
- Recomendación UIT-R BT.470-2, *Sistemas de televisión*.
- Recomendación UIT-R BR.648, *Grabación digital de señales de audio*.
- Informe UIT-R BO.955-2, *Radiodifusión sonora por satélite destinada a receptores en vehículos, portátiles y fijos en la gama 500-3000 MHz*.
- Recomendación J.17 del CCITT (1988), *Preacentuación utilizada en los circuitos radiofónicos*.
- IEEE Standard 1180:1990, *Standard Specification for the Implementations of 8 by 8 Inverse Discrete Cosine Transform*.
- IEC Publication 908:1987, *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.

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

**2.1.3 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.

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

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

---

<sup>2)</sup> Por publicar.

- 2.1.6 compresión:** Reducción del número de bits utilizados para representar un ítem de datos.
- 2.1.7 velocidad binaria constante:** Funcionamiento en el cual la velocidad binaria es constante desde el principio al fin del tren de bits comprimido.
- 2.1.8 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.9 verificación por redundancia cíclica (CRC, cyclic redundancy check):** La verificación por redundancia cíclica para verificar la corrección de los datos.
- 2.1.10 elemento de datos:** Un ítem de datos como se representa antes de la codificación y después de la decodificación.
- 2.1.11 tren decodificado:** La reconstrucción decodificada de un tren de bits comprimido.
- 2.1.12 decodificador:** Una materialización de un proceso de decodificación.
- 2.1.13 decodificación (proceso):** El proceso definido en la Rec. UIT-T 222.0 ISO/CEI 13818-1 que lee un tren de bits codificado de entrada y produce como salida imágenes o muestras de audio decodificadas.
- 2.1.14 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.
- 2.1.15 medio de almacenamiento digital DSM (DSM, digital storage media):** Un dispositivo o sistema de almacenamiento o transmisión digital.
- 2.1.16 DSM-CC:** Instrucciones y control de medios de almacenamiento digital (*digital storage media command and control*).
- 2.1.17 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.
- 2.1.18 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.
- 2.1.19 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.
- 2.1.20 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.
- 2.1.21 referencia de reloj de tren elemental; ESCR (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.
- 2.1.22 codificador:** Una materialización de un proceso de codificación.
- 2.1.23 codificación (proceso):** Un proceso, no especificado en la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, que lee un tren de imágenes o muestras de audio de entrada y produce un tren de bits codificado conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.24 codificación de entropía:** Codificación sin pérdida de la longitud variable de la representación digital de una señal para reducir la redundancia.
- 2.1.25 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.
- 2.1.26 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.
- 2.1.27 prohibido:** El término «prohibido», cuando se utiliza en las cláusulas de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 que definen el tren de bits codificado, indica que el valor especificado no se utilizará nunca.
- 2.1.28 tren (multiplexado) (sistema):** Un tren de bits compuesto de ninguno o varios trenes elementales combinados en una manera conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

- 2.1.29 capa (vídeo y sistema):** Uno de los niveles en la jerarquía de datos de las especificaciones de vídeo y sistemas definidas en las Partes 1 y 2 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.30 paca (sistema):** Una paca está constituida por un encabezamiento de paca seguido de ninguno o más paquetes. Es una capa en la sintaxis de codificación de sistema descrita en 2.5.3.3 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.31 datos de paquete (sistema):** Bytes de datos contiguos de un tren elemental presente en un paquete.
- 2.1.32 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.33 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.
- 2.1.34 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.
- 2.1.35 PES (sistema) (PES, *packetized elementary stream*):** Abreviatura de tren elemental paquetizado.
- 2.1.36 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.37 encabezamiento de paquete (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.
- 2.1.38 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.
- 2.1.39 indicación de tiempo de presentación (PTS, *presentation time-stamp*) (sistema):** 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.
- 2.1.40 unidad de presentación; PU (sistema) (PU, *presentation unit*):** Una unidad de acceso de audio decodificado o una imagen decodificada.
- 2.1.41 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.
- 2.1.42 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.
- 2.1.43 elemento de programa (sistema):** Un término genérico que designa un tren elemental u otro tren de datos que puede estar incluido en un programa.
- 2.1.44 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 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. Un ejemplo de PSI, la tabla de información de red no obligatoria, se define privadamente.
- 2.1.45 acceso aleatorio:** Proceso de comenzar a leer y decodificar el tren de bits codificado en un punto arbitrario.
- 2.1.46 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 la Rec. UIT-T H.222.0 | ISO/CEI 13818-1, todos los bits reservados se pondrán a '1'.

**2.1.47 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.

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

**2.1.49 empalme (sistema):** La concatenación, realizada en el nivel de sistema, de dos trenes elementales diferentes. El tren de sistema resultante es totalmente conforme a la Rec. UIT-T H.222.0 | ISO/CEI 13818-1. El empalme puede resultar en discontinuidades en la base de tiempo, el contador de continuidad, la PSI y la decodificación.

**2.1.50 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 2-18 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1.

**2.1.51 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.

**2.1.52 imagen fija:** Una imagen fija codificada consiste en una secuencia vídeo que contiene exactamente una imagen codificada que está intracodificada. Esta imagen tiene una PTS asociada, y el tiempo de presentación de las imágenes siguientes, si las hubiere, es posterior al de la imagen fija, por lo menos, dos periodos de imagen.

**2.1.53 encabezamiento de sistema (sistema):** El encabezamiento de sistema es una estructura de datos definida en 2.5.3.5 de la Rec. UIT-T H.222.0 | ISO/CEI 13818-1 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.

**2.1.54 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.

**2.1.55 decodificador-objetivo de sistemas; 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.

**2.1.56 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.

**2.1.57 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.

**2.1.58 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 $-\infty$   |
| %                 | Operador de módulo. Se define solamente para números positivos  |
| Signo( )          | $\text{Signo}(x) = \begin{cases} 1 & x > 0 \\ 0 & x == 0 \\ -1 & x < 0 \end{cases}$   |
| 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 <sub>10</sub> | Logaritmo con base diez   |
| log <sub>e</sub>  | Logaritmo con base e  |

### 2.2.2 Operadores lógicos

|    |            |
|----|------------|
|    | OR lógico  |
| && | AND lógico |
| !  | NOT lógico |

### 2.2.3 Operaciones relacionales

|           |   |
|-----------|---|
| >         | Mayor que                                 |
| ≥         | Mayor o igual que                         |
| <         | Menor que                                 |
| ≤         | Menor o igual que                         |
| ==        | Igual a                                   |
| !=        | No igual a                                |
| máx [...] | el valor máximo en la lista de argumentos |
| min [...] | 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. |
| 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.   |
| scfsi          | Información de selector de factor de escala.  |
| switch_point_1 | 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.  |
| tcimbsf        | Entero de complemento de dos, bit menos significativo (signo) primero.  |
| uimbsf         | Entero sin signo, bit más significativo primero.  |
| vlclbf         | 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, $0 \geq \text{window} \geq 2$ .   |

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

### 2.2.7 Constantes

|       |               |
|-------|---------------|
| $\pi$ | 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.

|  |   |
|--|---|
| while (condition) {<br><b>data_element</b><br>...<br>}                   | 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. |
| do {<br><b>data_element</b><br>...<br>}                                  | El elemento de datos siempre aparece por lo menos una vez. El elemento de datos se repite hasta que la condición no es verdadera.                               |
| while (condition)<br>if (condition) {<br><b>data_element</b><br>...<br>} | Si la condición es verdadera, el primer grupo de elementos de datos aparece a continuación en el tren de datos.   |

```

else {
    data_element
    ...
}
for (i = 0; i < n, i++) {
    data_element
    ...
}

```

Si la condición no es verdadera, el segundo grupo de elementos de datos aparece a continuación en el tren de datos.

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 aplica 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 cuatro tablas PSI:

- tabla de asociación de programas;
- tabla de correspondencia de programas;
- tabla de acceso condicional;
- tabla de información de red.



Estas tablas contienen la información necesaria y suficiente para demultiplexar y presentar programas. La tabla de correspondencia de programas del Cuadro 2-28 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.

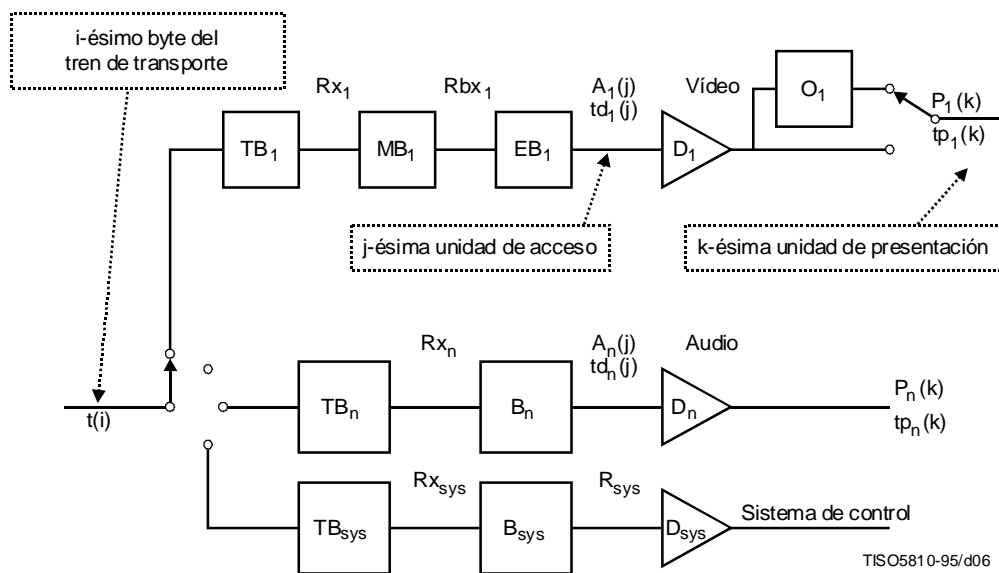
Los paquetes del tren de transporte pueden ser paquetes nulos. Los paquetes nulos están destinados a rellenar trenes de transporte. Se pueden insertar o suprimir mediante procesos de remultiplexación y, por consiguiente, no se puede suponer la entrega de la cabida útil de paquetes nulos al decodificador.

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 2-27 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 del equipo de grabación para el tren elemental de vídeo $n$ .   |
| $R_{sys}$     | es la velocidad a la que se suprimen datos de $B_{sys}$ .  |
| $R_{x_n}$     | es la velocidad a la que se suprimen datos de $TB_n$ .   |
| $Rb_{x_n}$    | 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.                      |
| $Rb_{x_n}(j)$ | 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.                      |
| $R_{x_{sys}}$ | velocidad a la que se suprimen datos de $TB_{sys}$ .   |
| $R_{es}$      | 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:

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

velocidad de cambio de la frecuencia de reloj de sistema en función del tiempo:  $\leq 75 \times 10^{-3}$  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 *i*-ésimo byte entra en el instante  $t(i)$ . El instante en el cual este byte entra en el T-STD puede ser recuperado del tren de entrada decodificando los campos de referencia de reloj de programa (PCR) de entrada, codificados en el campo de adaptación de paquete de tren de transporte del programa que se ha de decodificar y contando los bytes en el tren de transporte completo entre PCR sucesivas para el programa que se va a decodificar. La 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 (que da 90 kHz), denominada base de referencia de reloj de programa (ecuación 2-2), y otra parte, denominada extensión de referencia de reloj de programa (ecuación 2-3), en unidades del periodo de la frecuencia de reloj de sistema. Los valores codificados en ellas vienen dados por PCR-base(*i*), (véase la ecuación 2-2) y 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:

$$PCR(i) = PCR\_base(i) \times 300 + PCR\_ext(i) \quad (2-1)$$

donde:

$$PCR\_base(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (2-2)$$

$$PCR\_ext(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{ DIV } 1) \% 300 \quad (2-3)$$

Para todos los bytes, el tiempo de llegada de entrada  $t(i)$  mostrado en la siguiente ecuación 2-4, se calcula a partir de 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.

$$t(i) = \frac{PCR(i'')}{system\_clock\_frequency} + \frac{i - i''}{transport\_rate(i)} \quad (2-4)$$

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  $\pm 500$  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 datos de tren elemental n, indicados por su PID, son pasados a la memoria tampón de transporte para el tren n,  $TB_n$ . Este incluye paquetes de tren de transporte duplicados y paquetes sin cabida útil. La transferencia del i-ésimo byte de la entrada del decodificador-objetivo de sistemas a  $TB_n$  es instantánea, de modo que el i-ésimo byte entra en la memoria tampón para el tren n, de tamaño  $TBS_n$ , en el instante t(i).

Todos los bytes que entran en la memoria tampón  $TB_n$  son suprimidos a la velocidad  $R_{xn}$  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  $B_{sys}$ .

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:

$$R_{x_n} = 1, 2 \times R_{m\acute{a}x}[\text{perfil, nivel}]$$

donde

$R_{m\acute{a}x}[\text{perfil, nivel}]$  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  $1, 2 * R_{m\acute{a}x}$  para los trenes vídeo con parámetros constreñidos de la Norma ISO/CEI 11172-2, donde  $R_{m\acute{a}x}$  indica la velocidad binaria máxima para un tren de bits de parámetros constreñidos de dicha | Norma ISO/CEI 11172-2;

- para audio:

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

- para datos de sistemas:

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

$R_{x_n}$  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 ó 1, y todos los paquetes del tren de transporte identificados mediante la tabla de asociación de programas (Tabla 2-25) 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  $R_{x_{sys}}$  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 la Norma 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:

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

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

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

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

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

y donde  $VBV_{m\acute{a}x}[\text{perfil, nivel}]$  se define en el Cuadro 8-14 y  $R_{m\acute{a}x}[\text{perfil, nivel}]$  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).

## ISO/CEI 13818-1 : 1996 (S)

Para nivel alto 1440 y nivel alto:

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

donde  $BS_{oh}$  se define como:

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

y  $BS_{mux}$  se define como:

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

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

Para trenes de bits conformes a la Norma 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 la Norma ISO/CEI 11172-2, respectivamente.

Una porción  $BS_{mux} = 4ms \times R_{m\acute{a}x}[\text{perfil, nivel}]$  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 1 – La ocupación de la memoria tampón por la tara de paquetes PES está directamente limitada en los trenes PES por el P-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

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

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:

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

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.

Lo siguiente es aplicable a los trenes elementales de vídeo.

## 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 2.4.3.6).

Para el nivel bajo y el nivel principal:

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

Para el nivel alto 1440 y el nivel alto:

$$R_{bx_n} = \text{Min} \{ 1.05 \times R_{es}, R_{m\acute{a}x}[\text{perfil, nivel}] \}$$

Para un tren de bits para parámetros constreñidos según la Norma 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 la Norma 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  $R_{bx}$ . 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 (véase la Norma 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:

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

## ISO/CEI 13818-1 : 1996 (S)

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 2 –  $vbv\_delay(j + 1)$  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 2-29), 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 la Norma 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, la Norma ISO/CEI 13818-3, o la Norma 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  $B_{sys}$ .

$$R_{sys} = \text{m\acute{a}x} (80\,000 \text{ bit/s}, \text{transport\_rate}(i) * 8 \text{ bits/byte/500}) \quad (2-7)$$

NOTA 3 – 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 se está en el modo de bajo retardo, 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) ó «010» (trama congelada), ó «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 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; este 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/IEC 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/IEC 13818-2) ó 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.

## ISO/CEI 13818-1 : 1996 (S)

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. Específicamente:  $td_n(j) - t(i) \leq 1$  segundo para todas las  $j$  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 todas las  $j$ , 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  $n$ .

### 2.4.3 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.

#### 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).

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 &lt; 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> |

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 2-29).

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 2-29).

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 (Cuadro 2-25). El valor de PID 0x0001 se reserva para la tabla de acceso condicional (Cuadro 2-27). Los valores de PID 0x0002 - 0x000F están reservados. El valor de PID 0x1FFF está reservado para paquetes nulos (véase el Cuadro 2-3).

**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).

**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 2-3 – Tabla de PID**

| Valor  | Descripción  |
|--|--|
| 0x0000   | Tabla de asociación de programas   |
| 0x0001   | Tabla de acceso condicional  |
| 0x0002 - 0x000F  | Reservado  |
| 0x00010<br>...<br>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. |  |

**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 |

**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 PDI, 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.

#### 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.

Este 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 un 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 un 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 el mismo 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.

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

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

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 un 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 fines de esta subcláusula, un punto de acceso de tren elemental se define como sigue:

- Vídeo – el primer byte de un encabezamiento de secuencia de vídeo.
- Audio – el primer byte de una trama de audio.

Después de una discontinuidad del contador de continuidad en un paquete de transporte que según está designado contiene datos de tren elemental, el primer byte de datos de tren elemental en un paquete de tren de transporte del mismo PID será el primer byte de un punto de acceso de tren elemental o, en el caso de vídeo, el primer byte de un punto de acceso de tren elemental o un código de fin de secuencia seguido por un punto de acceso. Cada paquete de tren de transporte que contiene datos de tren elemental con un PID no designado como un 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' y '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 transporte vigente y subsiguiente con el mismo PID contienen alguna información para ayudar al acceso aleatorio en ese punto. Específicamente, cuando el bit está puesto a '1', el siguiente paquete PES que empezará en la cabida útil de paquetes de transporte (véase el Cuadro 2-29) con el PID vigente, contendrá el primer byte de un encabezamiento de secuencia de vídeo si el tipo de tren PES es 1 ó 2 (vídeo), o contendrá el primer byte de una trama de audio si el tipo de tren PES es 3 ó 4 (audio). Además, en estos casos estará presente una indicación de tiempo de presentación en este paquete PES o en un paquete subsiguiente para la primera imagen que siga al encabezamiento de secuencia, o para la trama de audio. Cuando el PID vigente es igual a PCR\_PID, la bandera PCR se pondrá a '1'.

**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, este campo se puede poner a '1' solamente si la cabida útil contiene uno o más bytes de una rebanada intracodificada. 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) 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

$$OPCR\_base(i) = ((system\_clock\_frequency \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (2-9)$$

$$OPCR\_ext(i) = ((system\_clock\_frequency \times t(i)) \text{ DIV } 1) \% 300 \quad (2-10)$$

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).

A los fines de esta subcláusula, un punto de acceso se define como sigue:

- Vídeo – El primer byte de un encabezamiento de secuencia de vídeo.
- Audio – El primer byte de una trama de audio.

**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 esta bandera está fijada, si el tren elemental llevado en este PID es un tren de audio, el campo `splice_type` se pondrá a 0000. Si el tren elemental llevado en este PID es un tren de vídeo, 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:

$$\text{offset} = t_1(i) - t(i)$$

$$\text{ltw\_offset} = \text{offset}/1$$

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 (*ltw\_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:

$$t_1(A_i + j) = t_1(A_i) + j * 188 * 8\text{-bits/byte} / R$$

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:

$$\text{offset} = t_1(A_i) - t(A_i)$$

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 es un tren de vídeo, este campo tendrá el valor 0000. Si el tren elemental transportado en ese PID es un tren de audio, este campo indica las condiciones que debe cumplir este 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 2-16.

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)**

| 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) – 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 SNR  
(todas las capas), nivel alto-1440 de perfil alto  
(capas media + 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) – 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**

| <b>splice_type</b> | 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**

| <b>splice_type</b> | 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 – Vídeo**

| <b>splice_type</b> | 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) – 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) – 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   |

**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 2-17.

#### 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' (0x000001).

**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 2-18. 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 2-18. 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 2-19).

**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 el encabezamiento de paquete PES va seguido inmediatamente por el código de comienzo de vídeo o la palabra de sincronización de audio indicados en el descriptor de alineación de tren de datos en 2.6.10, si este descriptor está 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 2-47 y 2-48. Cuando se pone a '0' no está definida, 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 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 puesta a '1' indica la presencia de un campo de modo truco de 8 bits. 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 puesta a '1' indica la presencia del campo de información de copia adicional. 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.

Cuadro 2-17 – Paquete PES

| Sintaxis  | N.º de bits | Mnemónico     |
|---|-------------|---------------|
| PES_packet() {  |             |               |
| <b>packet_start_code_prefix</b>   | <b>24</b>   | <b>bslbf</b>  |
| <b>stream_id</b>  | <b>8</b>    | <b>uimsbf</b> |
| <b>PES_packet_length</b>  | <b>16</b>   | <b>uimsbf</b> |
| if (stream_id != program_stream_map<br>&& stream_id != padding_stream<br>&& stream_id != private_stream_2<br>&& stream_id != ECM<br>&& stream_id != EMM<br>&& stream_id != program_stream_directory<br>&& stream_id != DSM-CC_stream<br>&& stream_id != ITU-T Rec. H.222.1 type E stream) { |             |               |
| '10'  | 2           | bslbf         |
| <b>PES_scrambling_control</b>   | 2           | bslbf         |
| <b>PES_priority</b>   | 1           | bslbf         |
| <b>data_alignment_indicator</b>   | 1           | bslbf         |
| <b>copyright</b>  | 1           | bslbf         |
| <b>original_or_copy</b>   | 1           | bslbf         |
| <b>PTS_DTS_flags</b>  | 2           | bslbf         |
| <b>ESCR_flag</b>  | 1           | bslbf         |
| <b>ES_rate_flag</b>   | 1           | bslbf         |
| <b>DSM_trick_mode_flag</b>  | 1           | bslbf         |
| <b>additional_copy_info_flag</b>  | 1           | bslbf         |
| <b>PES_CRC_flag</b>   | 1           | bslbf         |
| <b>PES_extension_flag</b>   | 1           | bslbf         |
| <b>PES_header_data_length</b>   | 8           | uimsbf        |
| if (PTS_DTS_flags == '10') {  |             |               |
| '0010'  | 4           | bslbf         |
| <b>PTS [32..30]</b>   | 3           | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>PTS [29..15]</b>   | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>PTS [14..0]</b>  | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| }   |             |               |
| if (PTS_DTS_flags == '11') {  |             |               |
| '0011'  | 4           | bslbf         |
| <b>PTS [32..30]</b>   | 3           | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>PTS [29..15]</b>   | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>PTS [14..0]</b>  | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| '0001'  | 4           | bslbf         |
| <b>DTS [32..30]</b>   | 3           | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>DTS [29..15]</b>   | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>DTS [14..0]</b>  | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| }   |             |               |
| if (ESCR_flag == '1') {   |             |               |
| <b>reserved</b>   | 2           | bslbf         |
| <b>ESCR_base[32..30]</b>  | 3           | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>ESCR_base[29..15]</b>  | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>ESCR_base[14..0]</b>   | 15          | bslbf         |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>ESCR_extension</b>   | 9           | uimsbf        |
| <b>marker_bit</b>   | 1           | bslbf         |
| }   |             |               |
| if (ES_rate_flag == '1') {  |             |               |
| <b>marker_bit</b>   | 1           | bslbf         |
| <b>ES_rate</b>  | 22          | uimsbf        |
| <b>marker_bit</b>   | 1           | bslbf         |
| }   |             |               |



Cuadro 2-17 – Paquete PES (continuación)

| Sintaxis   | N.º de bits | Mnemónico     |
|--|-------------|---------------|
| if (DSM_trick_mode_flag == '1') {                  |             |               |
| <b>trick_mode_control</b>                          | <b>3</b>    | <b>uimsbf</b> |
| if (trick_mode_control == fast_forward) {          |             |               |
| <b>field_id</b>                                    | <b>2</b>    | <b>bslbf</b>  |
| <b>intra_slice_refresh</b>                         | <b>1</b>    | <b>bslbf</b>  |
| <b>frequency_truncation</b>                        | <b>2</b>    | <b>bslbf</b>  |
| }  |             |               |
| else if (trick_mode_control == slow_motion) {      |             |               |
| <b>rep_cntrl</b>                                   | <b>5</b>    | <b>uimsbf</b> |
| }  |             |               |
| else if (trick_mode_control == freeze_frame) {     |             |               |
| <b>field_id</b>                                    | <b>2</b>    | <b>uimsbf</b> |
| <b>reserved</b>                                    | <b>3</b>    | <b>bslbf</b>  |
| }  |             |               |
| else if (trick_mode_control == fast_reverse) {     |             |               |
| <b>field_id</b>                                    | <b>2</b>    | <b>bslbf</b>  |
| <b>intra_slice_refresh</b>                         | <b>1</b>    | <b>bslbf</b>  |
| <b>frequency_truncation</b>                        | <b>2</b>    | <b>bslbf</b>  |
| else if (trick_mode_control == slow_reverse) {     |             |               |
| <b>rep_cntrl</b>                                   | <b>5</b>    | <b>uimsbf</b> |
| }  |             |               |
| else   |             |               |
| <b>reserved</b>                                    | <b>5</b>    | <b>bslbf</b>  |
| }  |             |               |
| if (additional_copy_info_flag == '1') {            |             |               |
| <b>marker_bit</b>                                  | <b>1</b>    | <b>bslbf</b>  |
| <b>additional_copy_info</b>                        | <b>7</b>    | <b>bslbf</b>  |
| }  |             |               |
| if (PES_CRC_flag == '1') {                         |             |               |
| <b>previous_PES_packet_CRC</b>                     | <b>16</b>   | <b>bslbf</b>  |
| }  |             |               |
| if (PES_extension_flag == '1') {                   |             |               |
| <b>PES_private_data_flag</b>                       | <b>1</b>    | <b>bslbf</b>  |
| <b>pack_header_field_flag</b>                      | <b>1</b>    | <b>bslbf</b>  |
| <b>program_packet_sequence_counter_flag</b>        | <b>1</b>    | <b>bslbf</b>  |
| <b>P-STD_buffer_flag</b>                           | <b>1</b>    | <b>bslbf</b>  |
| <b>reserved</b>                                    | <b>3</b>    | <b>bslbf</b>  |
| <b>PES_extension_flag_2</b>                        | <b>1</b>    | <b>bslbf</b>  |
| if (PES_private_data_flag == '1') {                |             |               |
| <b>PES_private_data</b>                            | <b>128</b>  | <b>bslbf</b>  |
| }  |             |               |
| if (pack_header_field_flag == '1') {               |             |               |
| <b>pack_field_length</b>                           | <b>8</b>    | <b>uimsbf</b> |
| pack_header()                                      |             |               |
| }  |             |               |
| if (program_packet_sequence_counter_flag == '1') { |             |               |
| <b>marker_bit</b>                                  | <b>1</b>    | <b>bslbf</b>  |
| <b>program_packet_sequence_counter</b>             | <b>7</b>    | <b>uimsbf</b> |
| <b>marker_bit</b>                                  | <b>1</b>    | <b>bslbf</b>  |
| <b>MPEG1_MPEG2_identifier</b>                      | <b>1</b>    | <b>bslbf</b>  |
| <b>original_stuff_length</b>                       | <b>6</b>    | <b>uimsbf</b> |
| }  |             |               |
| if (P-STD_buffer_flag == '1') {                    |             |               |
| '01'   | <b>2</b>    | <b>bslbf</b>  |
| <b>P-STD_buffer_scale</b>                          | <b>1</b>    | <b>bslbf</b>  |
| <b>P-STD_buffer_size</b>                           | <b>13</b>   | <b>uimsbf</b> |
| }  |             |               |
| if (PES_extension_flag_2 == '1') {                 |             |               |
| <b>marker_bit</b>                                  | <b>1</b>    | <b>bslbf</b>  |
| <b>PES_extension_field_length</b>                  | <b>7</b>    | <b>uimsbf</b> |
| for (i = 0; i < PES_extension_field_length; i++) { |             |               |
| <b>reserved</b>                                    | <b>8</b>    | <b>bslbf</b>  |
| }  |             |               |
| }  |             |               |
| for (i = 0; i < N1; i++) {                         |             |               |
| <b>stuffing_byte</b>                               | <b>8</b>    | <b>bslbf</b>  |
| }  |             |               |
| for (i = 0; i < N2; i++) {                         |             |               |
| <b>PES_packet_data_byte</b>                        | <b>8</b>    | <b>bslbf</b>  |
| }  |             |               |

**Cuadro 2-17 – Paquete PES (fin)**

| Sintaxis  | N.º de bits | Mnemónico |
|---|-------------|-----------|
| <pre> 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 == DSM-CC_stream    stream_id == ITU-T Rec. H.222.1 type E stream) {     for (i = 0; i &lt; PES_packet_length; i++) {         PES_packet_data_byte     } } else if (stream_id == padding_stream) {     for (i = 0; i &lt; PES_packet_length; i++) {         padding_byte     } } </pre> | 8           | bslbf     |
|   | 8           | bslbf     |

**Cuadro 2-18 – Asignaciones de 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               |      | ISO/IEC 13818-3 or ISO/IEC 11172-3 audio<br>stream number x xxxx                  |
| 1110 xxxx               |      | ITU-T Rec. H.262   ISO/IEC 13818-2 or ISO/IEC 11172-2<br>video stream number xxxx |
| 1111 0000               | 3    | ECM_stream  |
| 1111 0001               | 3    | EMM_stream  |
| 1111 0010               | 5    | ITU-T Rec. H.222.0   ISO/IEC 13818-1 Annex B or<br>ISO/IEC 13818-6_DSM-CC_stream  |
| 1111 0011               | 2    | ISO/IEC_13522_stream  |
| 1111 0100               | 6    | ITU-T Rec. H.222.1 type A   |
| 1111 0101               | 6    | ITU-T Rec. H.222.1 type B   |
| 1111 0110               | 6    | ITU-T Rec. H.222.1 type C   |
| 1111 0111               | 6    | ITU-T Rec. H.222.1 type D   |
| 1111 1000               | 6    | ITU-T Rec. H.222.1 type E   |
| 1111 1001               | 7    | ancillary_stream  |
| 1111 1010 ... 1111 1110 |      | Reserved data stream  |
| 1111 1111               | 4    | program_stream_directory  |

El símbolo x significa que está permitido 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.

**NOTAS**

- Los paquetes PES de tipo program\_stream\_map tienen una sintaxis única, que se especifica en 2.5.4.1.
- Los paquetes PES de tipo private\_stream\_1 e ISO/IEC\_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 de audio de la Norma ISO/CEI 13818-3.
- Los paquetes PES de tipo private\_stream\_2, ECM\_stream y EMM\_stream son similares a los de tipo private\_stream\_1, excepto que no se especifica sintaxis después del campo de longitud de paquete PES.
- Los paquetes PES de tipo program\_stream\_directory tienen una sintaxis única, especificada en 2.5.5.
- Los paquetes PES de tipo DSM-CC\_stream tienen una sintaxis única, especificada en la Norma ISO/CEI 13818-6.
- Este identificador de tren está asociado con el tipo de tren 0x09 del Cuadro 2-29.
- Este identificador de tren sólo se utiliza en paquetes PES que llevan datos de un tren de programa o de un tren de sistema de la Norma ISO/CEI 11172-1, en un tren de transporte (véase 2.4.3.7).

Cuadro 2-19 – 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 |

**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.

$$PTS(k) = ((system\_clock\_frequency \times tp_n(k)) DIV 300) \% 2^{33} \quad (2-11)$$

donde  $tp_n(k)$  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, si una PTS está presente en un encabezamiento de paquete PES 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 el primer byte del código de comienzo de imagen está presente en el paquete PES.

Para unidades de presentación de audio (PU), PU de vídeo en secuencias de bajo retardo e imágenes B, el tiempo de presentación  $tp_n(k)$  será igual al tiempo de decodificación  $td_n(k)$ .

Para imágenes I/P en secuencias que no tienen bajo retardo, y cuando no hay discontinuidad de decodificación entre unidades de acceso (AU)  $k$  y  $k'$ , el tiempo de presentación  $tp_n(k)$  será igual al tiempo de decodificación  $td_n(k')$  de la siguiente imagen transmitida (véase 2.7.5). Si hay una discontinuidad de decodificación, o el tren termina, la diferencia entre  $tp_n(k)$  y  $td_n(k)$  será la misma que si el tren original hubiese continuado sin una discontinuidad y sin fin.

NOTA 1 – Una secuencia de bajo retardo es una secuencia de vídeo en la cual la bandera de bajo retardo está puesta (véase 6.2.2.3 de la Rec. UIT-T H.262 | ISO/CEI 13818-2).

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 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  $A_n(j)$ .

## ISO/CEI 13818-1 : 1996 (S)

En el caso de vídeo, si una DTS está presente en un encabezamiento de paquete PES 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 el primer byte del código de comienzo de imagen está presente en el paquete PES.

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 (ESCR, *elementary stream clock reference*) 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

$$ESCR\_base(i) = ((system\_clock\_frequency * t(i)) DIV 300) \% 2^{33} \quad (2-14)$$

$$ESCR\_ext(i) = ((system\_clock\_frequency * t(i)) DIV 1) \% 300 \quad (2-15)$$

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 positivo 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 la Norma 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 2-20 – 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 2-21, 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 2-22.

**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  $N * rep\_cntrl$  veces, donde  $N$  se define como se ha indicado antes.

En el caso de secuencias progresivas de vídeo Norma ISO/CEI 11172-2 y Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de imagen de  $N * rep\_cntrl$ .

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  $N * rep\_cntrl$ . 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  $N * rep\_cntrl / 2$ . El otro campo de la imagen se visualiza entonces por una duración de campo de  $N - N * rep\_cntrl / 2$ .

**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 se define en el Cuadro 2-21, 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 2-21, 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 2-22 «Valores de selección de coeficientes».

**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 Norma ISO/CEI 11172-2 y Rec. UIT-T H.262 | ISO/CEI 13818-2, la imagen debe visualizarse por una duración de imagen de  $N * rep\_cntrl$ , 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  $N * rep\_cntrl$ . 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  $N * rep\_cntrl / 2$ . El otro campo de la imagen se visualiza entonces por una duración de campo de  $N - N * rep\_cntrl / 2$ .

**field\_id (identificador de campo)** – Un campo de dos bits que indica qué campo o campos se han de visualizar. Se codifica de acuerdo con el Cuadro 2-21.

**Cuadro 2-21 – 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 2-22.

**Cuadro 2-22 – 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 B, 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 2 – 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 paca)** – Una bandera de 1 bit que cuando se pone a '1' indica que un encabezamiento de paca de la Norma ISO/CEI 11172-1 o un encabezamiento de paca 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 paca 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 '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 '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 paca)** – Es un campo de 8 bits que indica la longitud en bytes, del campo de encabezamiento de paca.

**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\_idenfier (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 la Norma 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 la Norma 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' ó '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:

$$\text{si } (P\text{-STD\_buffer\_scale} == 0) \quad (2-16)$$

$$BS_n = \bar{P}\text{-STD\_buffer\_size} \times 128$$

si no,

$$BS_n = P\text{-STD\_buffer\_size} \times 1024 \quad (2-17)$$

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.

**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.

**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 | 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 la Norma 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 2-18). 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 la Norma ISO/CEI 11172-1 se llevan en trenes de transporte sustituyendo primeramente los encabezamientos de paquete de la Norma ISO/CEI 11172-1 por encabezamientos de paquete PES de la Rec. UIT-T H.262 | ISO/CEI 13818-1. Los valores de campo de encabezamiento de paquete Norma ISO/CEI 11172-1 se copian en campos equivalentes de encabezamiento de paquete PES de la Rec. UIT-T H.262 | ISO/CEI 13818-1 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 la Norma 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 la Norma ISO/CEI 11172-1 original.

El campo pack\_header() de un tren de programa o de un tren de sistema de la Norma 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 cuatro estructuras de tablas como se muestra en el Cuadro 2-23. 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 predeterminados y otras con PID seleccionables por el usuario.

**Cuadro 2-23 – 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 | 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 | Parámetros de red físicos tales como frecuencias 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                     |

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 PDI 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 2-24.

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 después del último byte de una sección, en cuyo caso todos los bytes siguientes hasta el fin del paquete del tren de transporte serán también bytes de relleno de valor 0xFF. Estos bytes pueden ser descartados por 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) 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 '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 inválida 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) 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 '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.

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 2-24.

**Cuadro 2-24 – Puntero de información específica de programa**

| Sintaxis             | N.º de bits | Mnemónico     |
|----------------------|-------------|---------------|
| <b>pointer_field</b> | <b>8</b>    | <b>uimsbf</b> |

#### 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 2-25).

#### 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 2-26.

#### 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 2-26.

**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 multiplex dentro de una red. Su valor es definido por el usuario.

Cuadro 2-25 – Sección de asociación de programas

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

Cuadro 2-26 – Valores de asignación de identificador de tabla

| Valor       | Descripción                                    |
|-------------|--|
| 0x00        | Sección de asociación de programas             |
| 0x01        | Sección de acceso condicional                  |
| 0x02        | Sección de correspondencia de programas TS     |
| 0x03 - 0x3F | Reservado Rec. UIT-T H.222.0   ISO/CEI 13818-1 |
| 0x40 - 0xFE | Privado de usuario                             |
| 0xFF        | Prohibido                                      |

**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 2-27, véase 2.6.16.

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

**Cuadro 2-27 – Sección de acceso condicional**

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

#### 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 2-26.

**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.

**Cuadro 2-28 – Sección de correspondencia de programas del tren de transporte**

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

**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 2-26.

**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 2-29.

**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 2-30).

#### 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 2-26 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 B después de procesar toda la sección privada.

## 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.



Cuadro 2-29 – Asignaciones de tipos de tren

| Valor       | Descripción  |
|-------------|--|
| 0x00        | UIT-T   ISO/CEI reservado  |
| 0x01        | ISO/CEI 11172-2 Vídeo  |
| 0x02        | Rec. UIT-T H.262   ISO/CEI 13818-2 Video o ISO/CEI 11172-2, tren de vídeo de parámetros constreñidos |
| 0x03        | ISO/CEI 11172-3 Audio  |
| 0x04        | ISO/CEI 13818-3 Audio  |
| 0x05        | Rec. UIT-T 222.0   ISO/CEI 13818-1 secciones privadas  |
| 0x06        | Rec. UIT-T H.222.0   ISO/CEI 13818-1 paquetes PES con datos privados                                 |
| 0x07        | ISO/CEI 13522 MHEG   |
| 0x08        | Annexo 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        | ISO/CEI 13818-1 auxiliar   |
| 0x0F - 0x7F | Rec. UIT-T H.222.0   ISO/CEI 13818-1 reservado   |
| 0x80 - 0xFF | Usuario privado  |

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 2-30 – Sección privada

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

El encabezamiento de paquete PES empieza con un código de comienzo de 32 bits que identifica también al tren (véase el Cuadro 2-18) al cual pertenecen los datos del paquete. El encabezamiento de paquete PES puede contener solamente una indicación de tiempo de presentación o una indicación de tiempo de presentación y una indicación de tiempo de codificación. 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 pacas. Una paca comienza con un encabezamiento de paca y es seguida por ninguno o varios paquetes PES. El encabezamiento de paca empieza con un código de comienzo de 32 bits. El encabezamiento de paca se utiliza para almacenar información de temporización y de velocidad binaria.

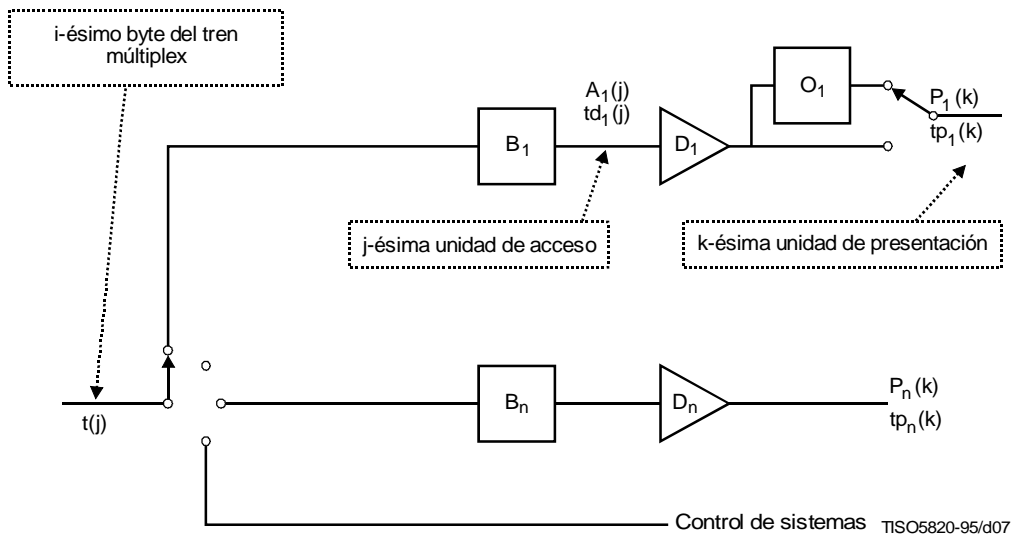
El tren de programa empieza con un encabezamiento de sistema que puede repetirse facultativamente. El encabezamiento de sistema transporta un resumen de los parámetros de sistema definidos en el tren.

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.



**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 <sub>n</sub> (j)  | es la j-ésima unidad de acceso en un tren elemental n. A <sub>n</sub> (j) está indexada según el orden de decodificación.  |
| td <sub>n</sub> (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 <sub>n</sub> (k)  | es la k-ésima unidad de presentación en un tren elemental n. P <sub>n</sub> (k) está indexada en el orden de presentación.   |
| tp <sub>n</sub> (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 <sub>n</sub> (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 <sub>n</sub>      | es la memoria tampón de entrada en el decodificador-objetivo de sistemas para el tren elemental n.   |
| BS <sub>n</sub>     | es el tamaño de la memoria tampón de entrada del decodificador-objetivo de sistemas, medido en bytes, para el tren elemental n.  |
| D <sub>n</sub>      | es el decodificador para el tren elemental n.  |
| O <sub>n</sub>      | 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.

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 Especificación 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 system\_clock\_reference\_base (ecuación 2-19) y otra, denominada system\_clock\_reference\_ext (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 SCR\_base(i) y SCR\_ext(i). El valor codificado en el campo SCR(i) 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:

$$SCR(i) = SCR\_base(i) \times 300 + SCR\_ext(i) \quad (2-18)$$

donde

$$SCR\_base(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (2-19)$$

$$SCR\_ext(i) = ((\text{system\_clock\_frequency} \times t(i)) \text{ DIV } 1) \% 300 \quad (2-20)$$

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

$$t(i) = \frac{SCR(i')}{system\_clock\_frequency} + \frac{i - i'}{program\_mux\_rate \times 50} \quad (2-21)$$

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 paca. |
| $i$                           | es el índice de cualquier byte en la paca, incluido el encabezamiento de paca.   |
| $SCR(i')$                     | es el tiempo codificado en los campos de base y extensión de referencia de reloj de sistema en unidades del reloj de sistema.        |
| <code>program_mux_rate</code> | es un campo definido en 2.5.3.3.   |

Después de la entrega del último byte de una paca 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 paca, 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. 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, este retardo está constreñido por:

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

y cuando hay datos de vídeo de imagen fija, por:

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

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 paca subsiguiente.

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:

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

$BS_{\text{oh}} = (1/750) \text{ segundos} * R_{\max}[\text{perfil, nivel}]$ , 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.  $BS_{\text{oh}}$  está atribuida para tara de encabezamiento de paquetes PES.

- Para vídeo de la Norma ISO/CEI 11172-2:

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

$BS_{\text{oh}} = (1/750) \text{ segundos} * R_{\max}$ , 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 la Norma ISO/CEI 11172-2.

- Para audio de la Norma ISO/CEI 11172-3 ó ISO/CEI 13818-3:

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

**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.

**2.5.3 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 2-31.

**Cuadro 2-31 – 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> | <b>32</b>   | <b>bslbf</b> |

**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 paca del tren de programa**

Véase el Cuadro 2-32.

**2.5.3.4 Definición semántica de campos en la paca de trenes de programa**

**pack\_start\_code (código de comienzo de paca)** - Es la cadena de bits '0000 0000 0000 0000 0000 0001 1011 1010' (0x000001BA). Identifica el principio de una paca.

**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, *system\_clock\_reference*) 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 multiplex 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 paca en la cual está incluido. El valor de la velocidad de multiplex de programa se mide en unidades de 50 bytes/segundo. El valor 0 está prohibido. El valor representado en la velocidad de multiplex 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 multiplex de programa puede variar de una paca 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 paca)** – 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 paca no estarán presentes más de 7 bytes de relleno.

**2.5.3.5 Encabezamiento de sistema**

Véase el Cuadro 2-34.

Cuadro 2-32 – Paca de tren de programa

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

Cuadro 2-33 – Encabezamiento de paca de tren de programa

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

Cuadro 2-34 – Encabezamiento de sistema de tren de programa

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

**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 paca 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 la Norma 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:

$$SCR\_base(i) = ((c1 * i + c2) DIV 300) \% 2^{33} \tag{2-22}$$

$$SCR\_ext(i) = ((c1 * i + c2) DIV 300) \% 300 \tag{2-23}$$

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 la Norma 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.

$$SCASR = \frac{system\_clock\_frequency}{audio\_sample\_rate\_in\_the\_P-STD} \tag{2-24}$$

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

|   |                               |                               |                               |                               |                               |                               |
|---|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| Frecuencia de muestreo de audio nominal (kHz) | 16                            | 32                            | 22,05                         | 44,1                          | 24                            | 48                            |
| SCASR   | 27 000 000<br>-----<br>16 000 | 27 000 000<br>-----<br>32 000 | 27 000 000<br>-----<br>22 050 | 27 000 000<br>-----<br>44 100 | 27 000 000<br>-----<br>24 000 | 27 000 000<br>-----<br>48 000 |



**system\_video\_lock\_flag (bandera de enganche de vídeo de sistema)** – Es un campo de 1 bit que indica que hay una relación racional constante especificada entre la velocidad de trama de vídeo 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 trama de vídeo se especifica en la Rec. UIT-T H.262 | ISO/CEI 13818-2. La bandera de enganche de vídeo de sistema sólo puede ponerse a '1' si, para todas las unidades de presentación en todos los trenes elementales de vídeo en el tren de programa de la presente Rec. UIT-T H.222.0 | ISO/CEI 13818-1, la relación de la frecuencia de reloj de sistema con la velocidad de trama de vídeo real, SCFR, es constante e igual al valor indicado en el siguiente cuadro a la velocidad de trama nominal indicada en el tren de vídeo.

$$SCFR = \frac{\text{system\_clock\_frequency}}{\text{frame\_rate\_in\_the\_P-STD}} \quad (2-25)$$

|                                 |           |           |           |         |         |         |         |         |
|---------------------------------|-----------|-----------|-----------|---------|---------|---------|---------|---------|
| 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)** – Es un entero de 5 bits en la gama comprendida de 0 a 16 inclusive y puesto a un mayor o igual que el número máximo de trenes de la Rec. UIT-T H.262 | Norma ISO/CEI 13818-2 e ISO/CEI 11172-2 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 vídeo de la Rec. UIT-T H.262 | Norma ISO/CEI 13818-2 e ISO/CEI 11172-2 está activo si la memoria tampón P-STD no está vacía, o si una unidad de presentación está siendo presentada en el modelo P-STD, o si la memoria tampón de reordenación no está vacía.

**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 constricción 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 vídeos 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 2-18.

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' ó '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:

$$\text{si } (P\text{-STD\_buffer\_bound\_scale} == 0)$$

$$BS_n \leq P\text{-STD\_buffer\_size\_bound} \times 128$$

si no

$$BS_n \leq P\text{-STD\_buffer\_size\_bound} \times 1024$$

### 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 2-18).

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 2-35.

Cuadro 2-35 – Correspondencia de trenes de programa

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

#### 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 2-29. El campo de tipo de tren sólo identificará trenes elementales contenidos en paquetes PES. Un valor 0x15 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 del 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 2-36.

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 la Norma 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 referencias del directorio a trenes de audio definidos en las Normas 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 del 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 2-36.

### 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 2-18).

**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 del 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 del 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 2-37).

**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 2-36 – 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     |
| }  |             |           |
| }  |             |           |

Cuadro 2-37 – Indicador intracodificado

| Valor | Significado        |
|-------|--------------------|
| 0     | No intracodificado |
| 1     | Intracodificado    |

**Cuadro 2-38 – 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 2-39 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.

**Cuadro 2-39 – Descriptores de programas y de 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                                  |
| 19-63          | n/a | n/a | Reservado por la Rec. H.222.0   ISO/CEI 13818-1 |
| 64-255         | n/a | n/a | Uso privado                                     |

**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.

### 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 la Norma ISO/CEI 11172-2 (véase el Cuadro 2-40).

**Cuadro 2-40 – Descriptor de tren de vídeo**

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

### 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 2-41.

**Cuadro 2-41 – 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\_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 Norma ISO/CEI 11172-2. Cuando se pone a '0', el tren de vídeo puede contener datos de vídeo Norma ISO/CEI 13818-2 y datos de vídeo Norma 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 Norma ISO/CEI 11172-2 no constreñidos. Cuando se pone a '0', el tren de vídeo puede contener parámetros constreñidos y trenes de vídeo Norma 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 la Norma ISO/CEI 11172-2 es un tren de perfil principal y nivel bajo (MP @ LL).

**chroma\_format (formato de crominancia)** – 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 crominancia en cualquier secuencia vídeo el tren de vídeo asociado. A los efectos de esta subcláusula, se considera que un tren de vídeo de la Norma ISO/CEI 11172-2 está constreñido a tener ambos campos puestos a cero.

**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 la Norma 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 las Normas ISO/CEI 13818-3 o ISO/CEI 11172-3 (véase el Cuadro 2-42).

**Cuadro 2-42 – Descriptor de tren de audio**

| Sintaxis                                | N.º de bits | Mnemónico     |
|---|-------------|---------------|
| <code>audio_stream_descriptor(){</code> |             |               |
| <b>descriptor_tag</b>                   | <b>8</b>    | <b>uimsbf</b> |
| <b>descriptor_length</b>                | <b>8</b>    | <b>uimsbf</b> |
| <b>free_format_flag</b>                 | <b>1</b>    | <b>bslbf</b>  |
| <b>ID</b>                               | <b>1</b>    | <b>bslbf</b>  |
| <b>layer</b>                            | <b>2</b>    | <b>bslbf</b>  |
| <b>variable_rate_audio_indicator</b>    | <b>1</b>    | <b>bslbf</b>  |
| <b>reserved</b>                         | <b>3</b>    | <b>bslbf</b>  |
| <code>}</code>                          |             |               |

### 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 la Norma 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 la Norma 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 las Normas ISO/CEI 13818-3 o ISO/CEI 11172-3 (véase 2.4.2.3 de la Norma 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, que se multiplexan en múltiples trenes como se describe en la presente Recomendación | Norma Internacional, en la Rec. UIT-T H.262 | ISO/CEI 13818-2 y en la ISO/CEI 13818-3 (véase el Cuadro 2-43).



Cuadro 2-43 – Descriptor de jerarquía

| Sintaxis                              | N.º de bits | Mnemónico     |
|---------------------------------------|-------------|---------------|
| hierarchy_descriptor() {              |             |               |
| <b>descriptor_tag</b>                 | <b>8</b>    | <b>uimsbf</b> |
| <b>descriptor_length</b>              | <b>8</b>    | <b>uimsbf</b> |
| <b>reserved</b>                       | <b>4</b>    | <b>bslbf</b>  |
| <b>hierarchy_type</b>                 | <b>4</b>    | <b>uimsbf</b> |
| <b>reserved</b>                       | <b>2</b>    | <b>bslbf</b>  |
| <b>hierarchy_layer_index</b>          | <b>6</b>    | <b>uimsbf</b> |
| <b>reserved</b>                       | <b>2</b>    | <b>bslbf</b>  |
| <b>hierarchy_embedded_layer_index</b> | <b>6</b>    | <b>uimsbf</b> |
| <b>reserved</b>                       | <b>2</b>    | <b>bslbf</b>  |
| <b>hierarchy_channel</b>              | <b>6</b>    | <b>uimsbf</b> |
| }                                     |             |               |

### 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 2-44.

Cuadro 2-44 – Valores del campo de tipo de jerarquía

| Valor | Descripción   |
|-------|---|
| 0     | Reservado   |
| 1     | Escalonabilidad espacial de la Rec. UIT-T H.262   ISO/CEI 13818-2 |
| 2     | Escalonabilidad SNR de la Rec. UIT-T H.262   ISO/CEI 13818-2      |
| 3     | Escalonabilidad temporal de la Rec. UIT-T H.262   ISO/CEI 13818-2 |
| 4     | Partición de datos de la Rec. UIT-T H.262   ISO/CEI 13818-2       |
| 5     | Tren de bits de extensión de la Norma ISO/CEI 13818-3             |
| 6     | Tren privado de la Rec. UIT-T H.222.0   ISO/CEI 13818-1           |
| 7-14  | Reservado   |
| 15    | Capa básica   |

**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.

### 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 2-45).

**Cuadro 2-45 – Descriptor de registro**

| Sintaxis   | N.º de bits  | Identificador  |
|--|--|--|
| <pre> registration_descriptor() {     descriptor_tag     descriptor_length     format_identifier     for (i = 0; i &lt; N; i++){         additional_identification_info     } }                     </pre> | <p><b>8</b></p> <p><b>8</b></p> <p><b>32</b></p> <p><b>8</b></p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>bslbf</b></p> |

**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 el SC29.

**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 2-46).

**Cuadro 2-46 – Descriptor de alineación de tren de datos**

| Sintaxis  | N.º de bits                                     | Mnemónico  |
|---|---|--|
| <pre> data_stream_alignment_descriptor() {     descriptor_tag     descriptor_length     alignment_type }                     </pre> | <p><b>8</b></p> <p><b>8</b></p> <p><b>8</b></p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> |

**2.6.11 Definición semántica de campos en el descriptor de alineación de tren de datos**

**alignment\_type (tipo de alineación)** – El cuadro 2-47 describe el tipo de alineación de vídeo cuando el indicador de alineación de datos en el encabezamiento del paquete PES tiene un valor de '1'.

En cada caso de valor de tipo de alineación, el primer byte de datos del paquete PES que sigue al encabezamiento de paquete PES será el primer byte de un código de comienzo del tipo indicado en el Cuadro 2-47. Al principio de una secuencia de vídeo, la alineación se producirá en el código de comienzo del primer encabezamiento de secuencia.

NOTA – La especificación del tipo de alineación '01' del Cuadro 2-47 no excluye la alineación desde el principio en un encabezamiento de GOP o SEQ.

La definición de unidad de acceso para datos de vídeo se indica en 2.1.1.

**Cuadro 2-47 – 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                             |

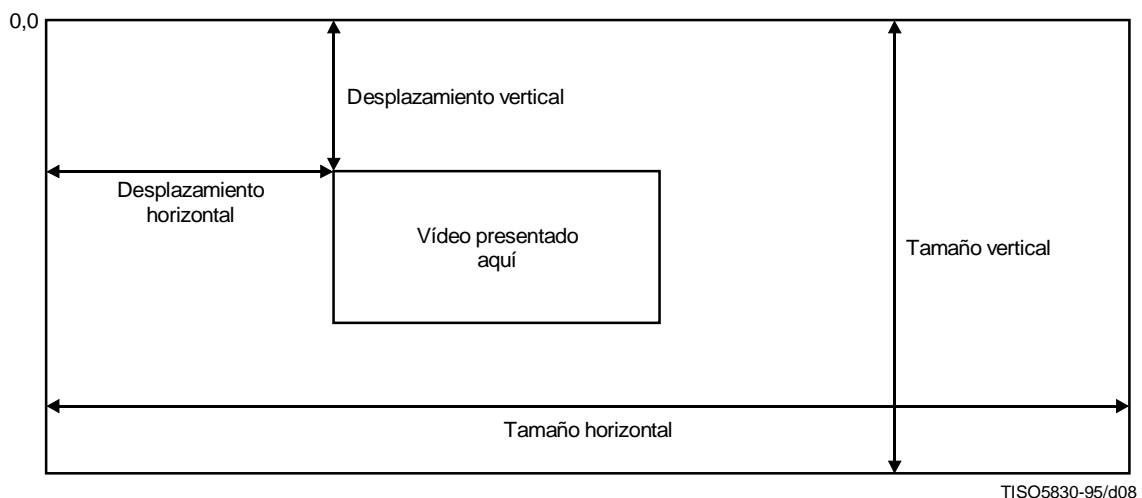
El Cuadro 2-48 describe el tipo de alineación de audio cuando el indicador de alineación de datos en el encabezamiento de paquete PES tiene un valor de '1'. En este caso, el primer byte de datos de paquete PES que sigue al encabezamiento PES es el primer byte de una palabra de sincronización de audio.

**Cadro 2-48 – 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 pixels 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 pixel superior izquierdo de la ventana de visualización o del rectángulo de visualización. Esto se representa en la Figura 2-3.



**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 pixels.

**vertical\_size (tamaño vertical)** – El tamaño vertical del modelo de cuadrícula de fondo en pixels.

**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 2-49).

Cuadro 2-49 – Descriptor de la cuadrícula de fondo de destino

| Sintaxis   | N.º de bits  | Mnemónico   |
|--|--|---|
| target_background_grid_descriptor() {<br><b>descriptor_tag</b><br><b>descriptor_length</b><br><b>horizontal_size</b><br><b>vertical_size</b><br><b>aspect_ratio_information</b><br>} | <b>8</b><br><b>8</b><br><b>14</b><br><b>14</b><br><b>4</b> | <b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b> |

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 2-50).

Cuadro 2-50 – Descriptor de ventana de vídeo

| Sintaxis  | N.º de bits  | Mnemónico   |
|---|--|---|
| video_window_descriptor() {<br><b>descriptor_tag</b><br><b>descriptor_length</b><br><b>horizontal_offset</b><br><b>vertical_offset</b><br><b>window_priority</b><br>} | <b>8</b><br><b>8</b><br><b>14</b><br><b>14</b><br><b>4</b> | <b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b><br><b>uimsbf</b> |

### 2.6.15 Definición semántica de campos en el descriptor de ventana de vídeo

**horizontal\_offset (desplazamiento horizontal)** – Su valor indica la posición del pixel 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 pixel superior izquierdo de la ventana de vídeo será uno de los pixel de la cuadrícula de fondo objetivo (véase la Figura 2-3).

**vertical\_offset (desplazamiento vertical)** – Su valor indica la posición del pixel 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 pixel superior izquierdo de la ventana de vídeo será uno de los pixel 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 2-51).

**Cuadro 2-51 – Descriptor de acceso condicional**

| Sintaxis  | N.º de bits   | Mnemónico  |
|---|---|--|
| <pre>CA_descriptor() {   descriptor_tag   descriptor_length   CA_system_ID   reserved   CA_PID   for (i = 0; i &lt; N; i++) {     private_data_byte   } }</pre> | <p>8</p> <p>8</p> <p>16</p> <p>3</p> <p>13</p> <p>8</p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>bslbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> |

### 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.

### 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 2-52).

**Cuadro 2-52 – Descriptor de idioma ISO 639**

| Sintaxis  | N.º de bits                          | Mnemónico   |
|---|--------------------------------------|---|
| <pre>ISO_639_language_descriptor() {   descriptor_tag   descriptor_length   for (i = 0; i &lt; N; i++) {     ISO_639_language_code     audio_type   } }</pre> | <p>8</p> <p>8</p> <p>24</p> <p>8</p> | <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>bslbf</b></p> <p><b>bslbf</b></p> |

### 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 la Norma ISO 639. Cada carácter está codificado en 8 bits de acuerdo con la Norma 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 2-53.

**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.

**Cuadro 2-53 – Valores de tipo de audio**

| Valor       | Descripción                                 |
|-------------|---|
| 0x00        | No definido                                 |
| 0x01        | En blanco                                   |
| 0x02        | Degradación auditiva                        |
| 0x03        | Comentario para casos de degradación visual |
| 0x04 - 0xFF | Reservado                                   |

### 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 2-54).

**Cuadro 2-54 – Descriptor de reloj de sistema**

| Sintaxis                                  | N.º de bits | Mnemónico     |
|---|-------------|---------------|
| <code>system_clock_descriptor() {</code>  |             |               |
| <b>descriptor_tag</b>                     | <b>8</b>    | <b>uimsbf</b> |
| <b>descriptor_length</b>                  | <b>8</b>    | <b>uimsbf</b> |
| <b>external_clock_reference_indicator</b> | <b>1</b>    | <b>bslbf</b>  |
| <b>reserved</b>                           | <b>1</b>    | <b>bslbf</b>  |
| <b>clock_accuracy_integer</b>             | <b>6</b>    | <b>uimsbf</b> |
| <b>clock_accuracy_exponent</b>            | <b>3</b>    | <b>uimsbf</b> |
| <b>reserved</b>                           | <b>5</b>    | <b>bslbf</b>  |
| <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 2-55).

Cuadro 2-55 – Descriptor de utilización de memoria tampón múltiplex

| Sintaxis  | N.º de bits   | Mnemónico   |
|---|---|---|
| <pre> multiplex_buffer_utilization_descriptor() {     descriptor_tag     descriptor_length     bound_valid_flag     LTW_offset_lower_bound     reserved     LTW_offset_upper_bound } </pre> | <p>8</p> <p>8</p> <p>1</p> <p>15</p> <p>1</p> <p>14</p> | <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> |

### 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 períodos 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 trabajos audiovisuales. Este descriptor de derechos de autor se aplica a programas o trenes elementales dentro del programa (véase el Cuadro 2-56).

Cuadro 2-56 – Descriptor de derechos de autor

| Sintaxis  | N.º de bits                          | Identificador  |
|---|--------------------------------------|--|
| <pre> copyright_descriptor() {     descriptor_tag     descriptor_length     copyright_identifier     for (i = 0; i &lt; N; i++){         additional_copyright_info     } } </pre> | <p>8</p> <p>8</p> <p>32</p> <p>8</p> | <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> |

### 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 2-57.

**Cuadro 2-57 – Descriptor de velocidad binaria máxima**

| Sintaxis  | N.º de bits   | Identificador   |
|---|---|---|
| <pre> maximum_bitrate_descriptor() {     descriptor_tag     descriptor_length     reserved     maximum_bitrate }                     </pre> | <p style="text-align: center;">8<br/>8<br/>2<br/>22</p> | <p style="text-align: center;"><b>uimsbf</b><br/><b>uimsbf</b><br/><b>bslbf</b><br/><b>uimsbf</b></p> |

**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 2-58.

**Cuadro 2-58 – Descriptor de indicador de datos privados**

| Sintaxis   | N.º de bits                                       | Identificador  |
|--|---|--|
| <pre> private_data_indicator_descriptor() {     descriptor_tag     descriptor_length     private_data_indicator }                     </pre> | <p style="text-align: center;">8<br/>8<br/>32</p> | <p style="text-align: center;"><b>uimsbf</b><br/><b>uimsbf</b><br/><b>uimsbf</b></p> |

**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 («smoothing buffer»), SB<sub>n</sub>, asociada con este descriptor, y sobre la correspondiente velocidad a que salen los datos de esa memoria tampón (velocidad de fuga, «leak rate»), 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<sub>n</sub> 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<sub>n</sub> 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 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, y
- también todos los paquetes de tren de transporte del PID que es igual PMT\_PID de esta sección.

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 2-59 – Descriptor de memoria tampón alisadora**

| Sintaxis                         | N.º de bits | Mnemónico     |
|----------------------------------|-------------|---------------|
| smoothing_buffer_descriptor () { |             |               |
| <b>descriptor_tag</b>            | <b>8</b>    | <b>uimsbf</b> |
| <b>descriptor_length</b>         | <b>8</b>    | <b>uimsbf</b> |
| <b>reserved</b>                  | <b>2</b>    | <b>bslbf</b>  |
| <b>sb_leak_rate</b>              | <b>22</b>   | <b>uimsbf</b> |
| <b>reserved</b>                  | <b>2</b>    | <b>bslbf</b>  |
| <b>sb_size</b>                   | <b>22</b>   | <b>uimsbf</b> |
| }                                |             |               |

### 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<sub>n</sub> 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<sub>n</sub> para el tren elemental asociado u otros datos, en unidades de 1 byte (véase el Cuadro 2-59).

### 2.6.32 Descriptor de STD

Este descriptor (STD\_descriptor) es facultativo y sólo se aplica al modelo de T-STD y a los trenes elementales de vídeo, y se utiliza como se indica en 2.4.2. Este descriptor no se aplica a trenes de programa (véase el Cuadro 2-60).

**Cuadro 2-60 – Descriptor de STD**

| Sintaxis                 | N.º de bits | Mnemónico     |
|--------------------------|-------------|---------------|
| STD_descriptor () {      |             |               |
| <b>descriptor_tag</b>    | <b>8</b>    | <b>uimsbf</b> |
| <b>descriptor_length</b> | <b>8</b>    | <b>uimsbf</b> |
| <b>reserved</b>          | <b>7</b>    | <b>bslbf</b>  |
| <b>leak_valid_flag</b>   | <b>1</b>    | <b>bslbf</b>  |
| }                        |             |               |

### 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<sub>n</sub> a la memoria tampón EB<sub>n</sub> 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<sub>n</sub> a la memoria tampón EB<sub>n</sub> 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 la secuencia de vídeo (véase el Cuadro 2-61).

**Cuadro 2-61 – Descriptor de IBP**

| Sintaxis   | N.º de bits  | Mnemónico  |
|--|--|--|
| <pre> ibp_descriptor() {     descriptor_tag     descriptor_length     closed_gop_flag     identical_gop_flag     max_gop_length }                     </pre> | <p><b>8</b><br/><b>8</b><br/><b>1</b><br/><b>1</b><br/><b>14</b></p> | <p><b>uimsbf</b><br/><b>uimsbf</b><br/><b>uimsbf</b><br/><b>uimsbf</b><br/><b>uimsbf</b></p> |

**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.

**2.7 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 segundos. De este modo:

$$| t(i) - t(i') | \leq 0,7 \text{ seg}$$

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 segundos. De este modo:

$$| t(i) - t(i') | \leq 0,1 \text{ seg}$$

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 PCE\_PID para cada programa.

Debe haber por lo menos dos 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 segundos. En trenes PES, la codificación de ESCR se requiere con el mismo intervalo. De este modo:

$$|t(i) - t(i')| \leq 0,7 \text{ seg}$$

para todas la  $i$  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 segundos. De este modo:

$$|tp_n(k) - tp_n(k'')| \leq 0,7 \text{ seg}$$

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 segundos no se aplica en el caso de imágenes fijas.

### 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.

### 2.7.6 Constricciones de temporización para la codificación escalonable

Si una secuencia de audio se codifica utilizando un tren de bits de extensión de la Norma ISO/CEI 13818-3, las correspondientes unidades de codificación/presentación 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.

## ISO/CEI 13818-1 : 1996 (S)

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 el bajo retardo puesto 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 paca que sigue inmediatamente al encabezamiento de paca. El encabezamiento de sistema estará presente en la primera paca 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 bits/segundo 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 pacas p en el tren de programa cuando la bandera de restricción de velocidad de paquetes está puesta a '1' (véase 2.5.3.5):

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[ 1, \frac{R_{\max}}{2,0 \times 10^6} \right] \quad (2-27)$$

y cuando la bandera de restricción de velocidad de paquetes está puesta a '0':

$$NP \leq (t(i') - t(i)) \times 300 \times \max \left[ 1, \frac{R_{\max}}{2,0 \times 10^6} \right] \quad (2-27)$$

donde

$$R_{\max} = 8 \times 50 \times \text{rate\_bound} \quad \text{bit/s} \quad (2-29)$$

NP es el número de prefijos de código de comienzo de paquete y de códigos de comienzo de encabezamiento entre códigos de comienzo de paca adyacente o entre el último código de comienzo de paca y el código de fin de programa MPEG según se define en el Cuadro 2-31 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 paca p;

t(i') es el tiempo, medido en segundos, codificado en la referencia de reloj de sistema para la paca p + 1, que sigue inmediatamente a la paca p o en el caso de la paca 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 en un CSPS, se aplica lo siguiente:

BS<sub>n</sub> 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, y una cantidad adicional de almacenamiento BS<sub>add</sub>. BS<sub>add</sub> se especifica como:

$$BS_{add} \leq \text{MAX} [6 \times 1024, R_{v\max} \times 0,001] \text{ bytes}$$

R<sub>vmax</sub> es la velocidad binaria de vídeo máxima del tren elemental de vídeo.

En el caso de un tren elemental de audio en un CSPS, se aplica lo siguiente:

$$BS_n \leq 4096 \text{ bytes}$$

## 2.7.10 Tren de transporte

### Enganche 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 la Norma ISO 11172-2. La velocidad de muestreo de audio es la especificada en las Normas ISO/CEI 13818-3 ó 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:

$$SCASR = \frac{\text{system\_clock\_frequency}}{\text{audio\_sample\_rate\_in\_the\_T-STD}} \quad (2-30)$$

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 trenes elementales de vídeo en el tren de transporte, la relación de la frecuencia de reloj de sistema, con la velocidad de trama de vídeo real, SCFR, es contante e igual al valor indicado en la siguiente tabla a la velocidad de trama 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 |

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.

## 2.8 Compatibilidad con la Norma ISO/CEI 11172

El tren de programa de la presente Recomendación | Norma Internacional se define para que sea compatible hacia adelante con la Norma ISO/CEI 11172-1. Los decodificadores del tren de programa definidos en la presente Recomendación | Norma Internacional admitirán también la decodificación de la Norma ISO/CEI 11172-1.

## 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.

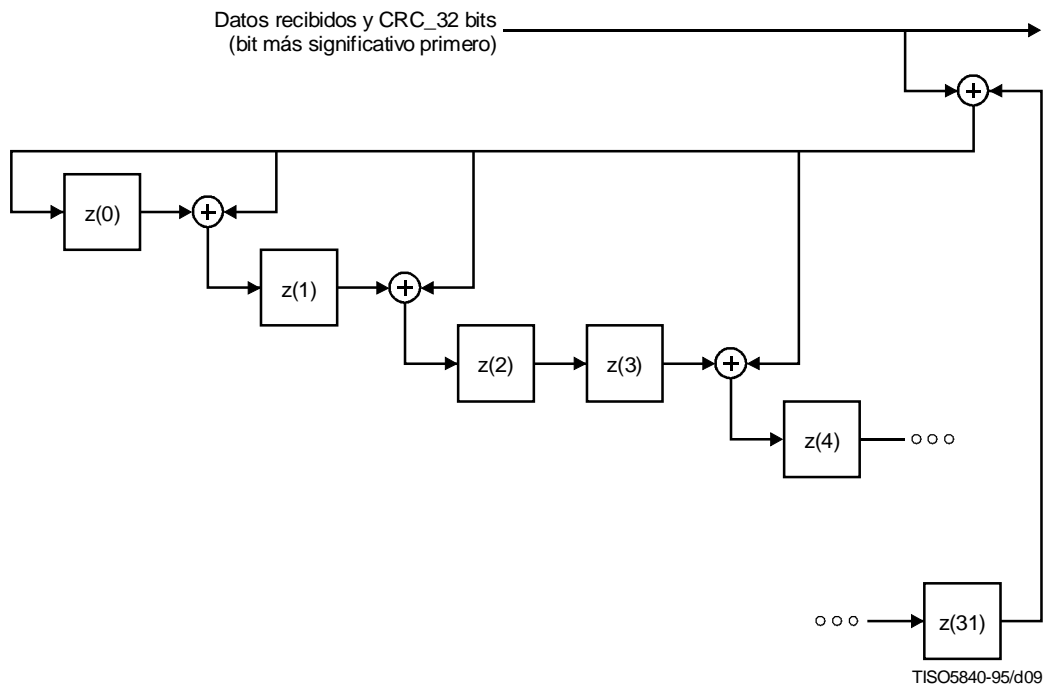


Figura A.1 – Modelo de decodificador CRC de 32 bits

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.

Esta 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 ceros 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, *digital storage medium command and control*) 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 | Norma 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 la ISO 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. La Norma ISO/CEI 13818-6 define la extensión DSM-CC con un alcance más amplio.

#### B.0.2 Aplicaciones futuras

Además de las aplicaciones inmediatas sustentadas 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 petición

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 multimedia 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 la Norma 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 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.

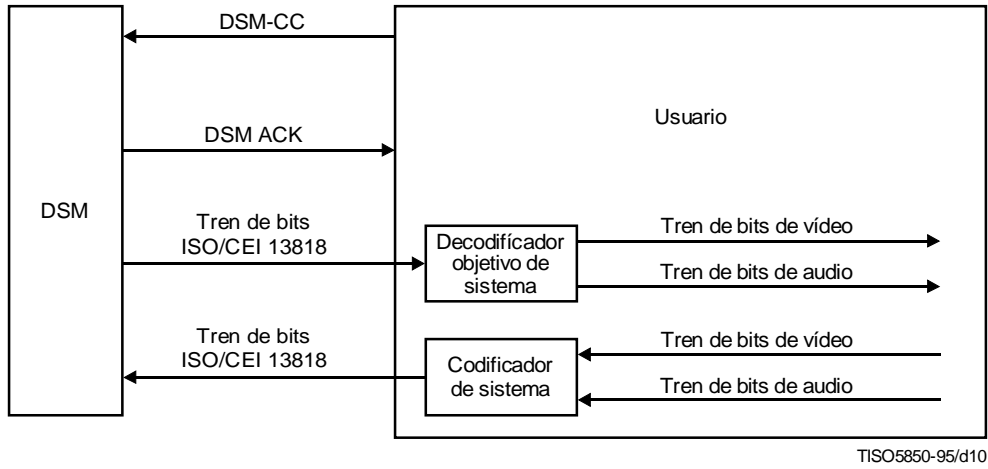


Figura B.1 – Configuración de una aplicación DSM-CC

**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 A.2.2 a A.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.

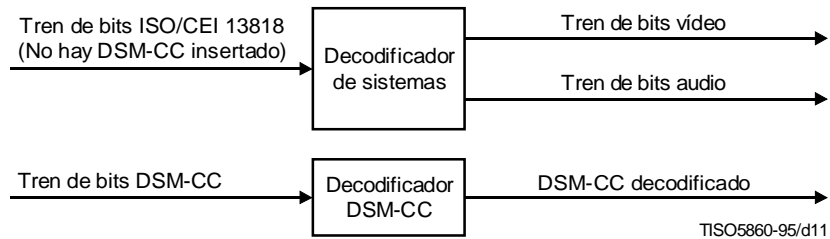
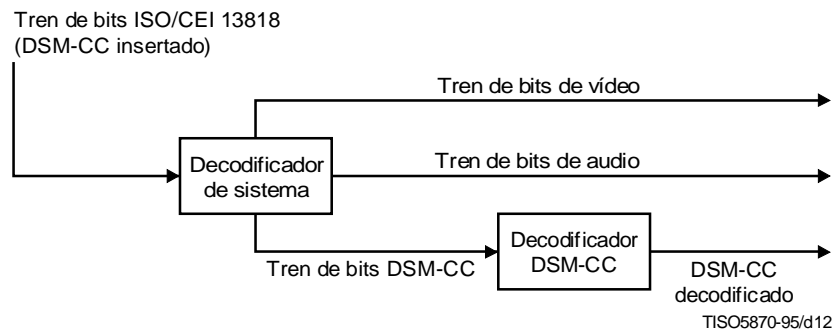


Figura B.2 – Tren de bits DSM-CC decodificado como un tren de bits autónomo

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 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 2-18 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, son aplicables las definiciones siguientes.

**B.2.1.1 DSM-CC:** Las instrucciones y control de medios de almacenamiento digital especificadas por esta Recomendación | Norma Internacional 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 la Norma 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 contruidos 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  |
|--|---|--|
| <pre> DSM_CC() {   packet_start_code_prefix   stream_id   packet_length   command_id   If (command_id == '01') {     control()   } else if (command_id == '02') {     ack()   } }                     </pre> | <p><b>24</b></p> <p><b>8</b></p> <p><b>16</b></p> <p><b>8</b></p> | <p><b>bslbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> <p><b>uimsbf</b></p> |

**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 2-19.

**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>     |             |              |
| <b>select_flag</b>           | <b>1</b>    | <b>bslbf</b> |
| <b>retrieval_flag</b>        | <b>1</b>    | <b>bslbf</b> |
| <b>storage_flag</b>          | <b>1</b>    | <b>bslbf</b> |
| <b>reserved</b>              | <b>12</b>   | <b>bslbf</b> |
| <b>marker_bit</b>            | <b>1</b>    | <b>bslbf</b> |
| If (select_flag == '1') {    |             |              |
| <b>bitstream_id</b> [31..17] | <b>15</b>   | <b>bslbf</b> |
| <b>marker_bit</b>            | <b>1</b>    | <b>bslbf</b> |
| <b>bitstream_id</b> [16..2]  | <b>15</b>   | <b>bslbf</b> |
| <b>marker_bit</b>            | <b>1</b>    | <b>bslbf</b> |
| <b>bitstream_id</b> [1..0]   | <b>2</b>    | <b>bslbf</b> |
| <b>select_mode</b>           | <b>5</b>    | <b>bslbf</b> |
| <b>marker_bit</b>            | <b>1</b>    | <b>bslbf</b> |
| }                            |             |              |
| if (retrieve_flag == '1') {  |             |              |
| <b>jump_flag</b>             | <b>1</b>    | <b>bslbf</b> |
| <b>play_flag</b>             | <b>1</b>    | <b>bslbf</b> |
| <b>pause_mode</b>            | <b>1</b>    | <b>bslbf</b> |
| <b>resume_mode</b>           | <b>1</b>    | <b>bslbf</b> |
| <b>stop_mode</b>             | <b>1</b>    | <b>bslbf</b> |
| <b>reserved</b>              | <b>10</b>   | <b>bslbf</b> |
| <b>marker_bit</b>            | <b>1</b>    | <b>bslbf</b> |
| if (jump_flag == '1') {      |             |              |
| <b>reserved</b>              | <b>7</b>    | <b>bslbf</b> |
| <b>direction_indicator</b>   | <b>1</b>    | <b>bslbf</b> |
| time_code()                  |             |              |
| }                            |             |              |
| if (play_flag == '1'){       |             |              |
| <b>speed_mode</b>            | <b>1</b>    | <b>bslbf</b> |
| <b>direction_indicator</b>   | <b>1</b>    | <b>bslbf</b> |
| <b>reserved</b>              | <b>6</b>    | <b>bslbf</b> |
| time_code()                  |             |              |
| }                            |             |              |
| }                            |             |              |
| if (storage_flag == '1') {   |             |              |
| <b>reserved</b>              | <b>6</b>    | <b>bslbf</b> |
| <b>record_flag</b>           | <b>1</b>    | <b>bslbf</b> |
| <b>stop_mode</b>             | <b>1</b>    | <b>bslbf</b> |
| 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() {   |             |              |
| <b>select_ack</b>   | <b>1</b>    | <b>bslbf</b> |
| <b>retrieval_ack</b>  | <b>1</b>    | <b>bslbf</b> |
| <b>storage_ack</b>  | <b>1</b>    | <b>bslbf</b> |
| <b>error_ack</b>  | <b>1</b>    | <b>bslbf</b> |
| <b>reserved</b>   | <b>10</b>   | <b>bslbf</b> |
| <b>marker_bit</b>   | <b>1</b>    | <b>bslbf</b> |
| <b>cmd_status</b>   | <b>1</b>    | <b>bslbf</b> |
| If (cmd_status == '1' &&<br>(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 están puestos a '1', especifican 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 inválido, un salto que rebasa el fin de un fichero, o una función no admitida, 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 inválido, 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>       |             |              |
| <b>reserved</b>                  | <b>7</b>    | <b>bslbf</b> |
| <b>infinite_time_flag</b>        | <b>1</b>    | <b>bslbf</b> |
| if (infinite_time_flag == '0') { |             |              |
| <b>reserved</b>                  | <b>4</b>    | <b>bslbf</b> |
| <b>PTS [32..30]</b>              | <b>3</b>    | <b>bslbf</b> |
| <b>marker</b>                    | <b>1</b>    | <b>bslbf</b> |
| <b>PTS [29..15]</b>              | <b>15</b>   | <b>bslbf</b> |
| <b>marker_bit</b>                | <b>1</b>    | <b>bslbf</b> |
| <b>PTS [14..0]</b>               | <b>15</b>   | <b>bslbf</b> |
| <b>marker_bit</b>                | <b>1</b>    | <b>bslbf</b> |
| }                                |             |              |
| }                                |             |              |

### 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 cuatro 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*).

El contenido de PAT, PMT y CAT se especifican en esta Recomendación | Norma Internacional NIT es una tabla privada, pero el valor PID de los paquetes del tren de transporte que la llevan se especifica en la PAT. Sin embargo, se debe 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 sustenta 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.

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 2.1.42 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). 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 sustentar 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 seudoindependientes 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 EMM 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.

### 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.

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 cifrado.

### 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.

#### NOTAS

- 1 Los paquetes del tren de transporte que contienen la tabla de correspondencia de programa se transmiten sin cifrado.
- 2 Es posible transmitir información sobre eventos en descriptores privados transportados dentro de TS\_program\_map\_section(s).

### 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, cifrado (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).

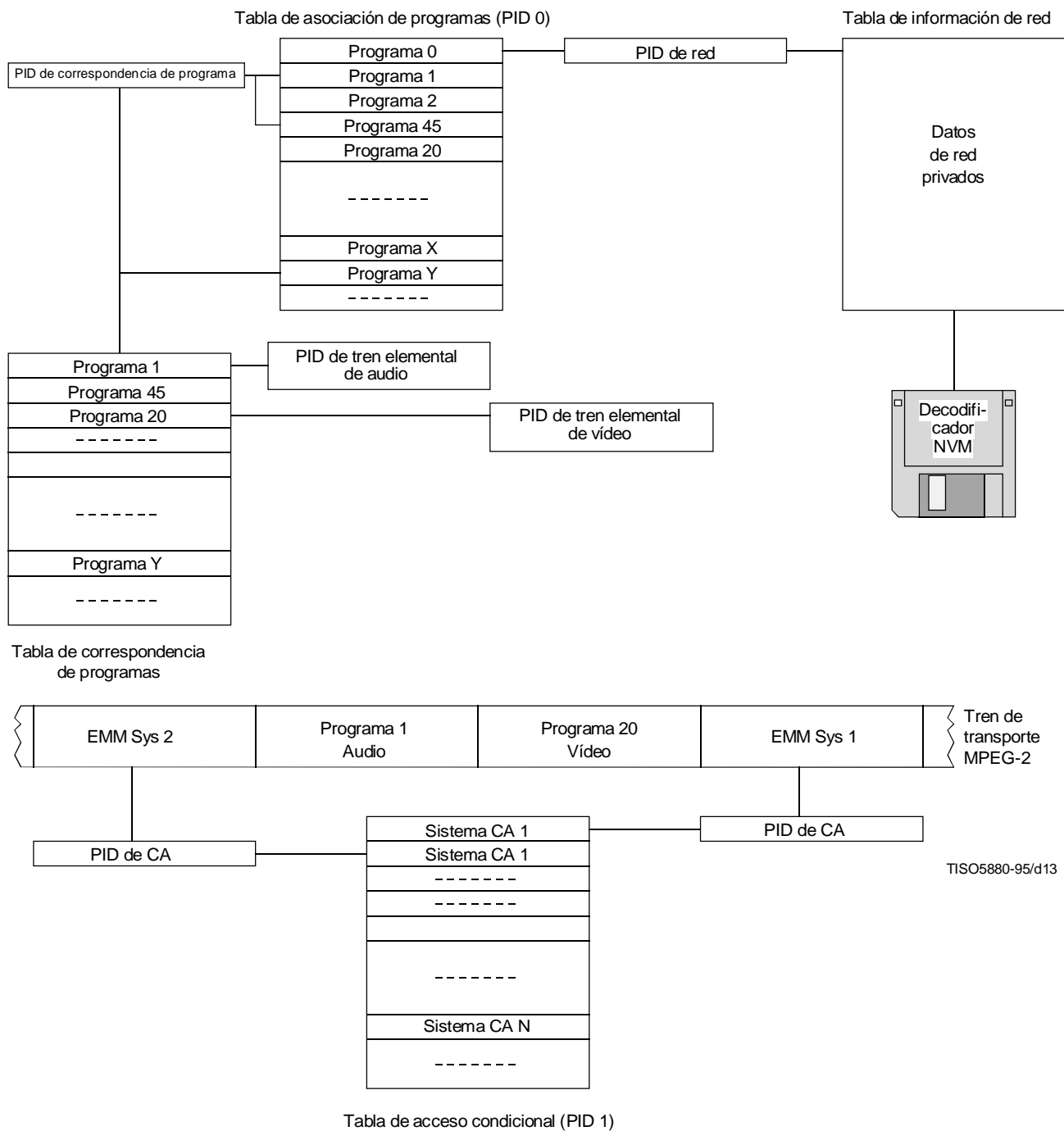
Las secciones privadas pueden aparecer en PID que están etiquetados como PMT\_PID o en paquetes del tren de transporte con otros valores de PID que contienen exclusivamente secciones privadas, incluido el PID asignado a la NIT. Si los paquetes de tren de transporte del PID que transporta las secciones 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.



TISO5880-95/d13

Figura C.1 – Relaciones de correspondencia de programa y red

**Cuadro C.1 – Descriptor compuesto**

| Sintaxis  | N.º de bits  | Identificador  |
|---|--|--|
| <pre>Composite_descriptor() {   <b>descriptor_tag</b>(privately defined)   <b>descriptor_length</b>   for (i = 0; i &lt; N; i++) {     sub_descriptor()   } }</pre> | <p style="text-align: center;"><b>8</b><br/><b>8</b></p> | <p style="text-align: center;"><b>uimsbf</b><br/><b>uimsbf</b></p> |

**Cuadro C.2 – Subdescriptor**

| Sintaxis   | N.º de bits   | Identificador  |
|--|---|--|
| <pre>sub_descriptor() {   <b>sub_descriptor_tag</b>   <b>sub_descriptor_length</b>   for (i = 0; i &lt; N; i++) {     <b>private_data_byte</b>   } }</pre> | <p style="text-align: center;"><b>8</b><br/><b>8</b><br/><b>8</b></p> | <p style="text-align: center;"><b>uimsbf</b><br/><b>uimsbf</b><br/><b>uimsbf</b></p> |

### C.9 Utilización de anchura de banda y tiempo de adquisición de señal

Todas las realizaciones 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 MDF, 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 (cifrado).

En el siguiente análisis, se ha despreciado también el tiempo para adquirir y procesar mensajes de permiso de control (ECM).

Las siguientes tablas 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.

Esta frecuencia será una clave determinante del componente de tiempo de adquisición de la señal debido a estructuras PSI.

En ambas tablas 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.

**Cuadro C.3 – Utilización de anchura de banda de la tabla de asociación de programas (bit/s)**

**Número de programas por tren de transporte**

|  |     |        |        |        |        |        |
|--|-----|--------|--------|--------|--------|--------|
|  |     | 1      | 5      | 10     | 32     | 128    |
| Frecuencia de información de la tabla PA<br>(Hz) | 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**

|  |     |        |        |        |        |         |
|--|-----|--------|--------|--------|--------|---------|
|  |     | 1      | 5      | 10     | 32     | 128     |
| Frecuencia de información de la tabla PM<br>(Hz) | 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 |

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

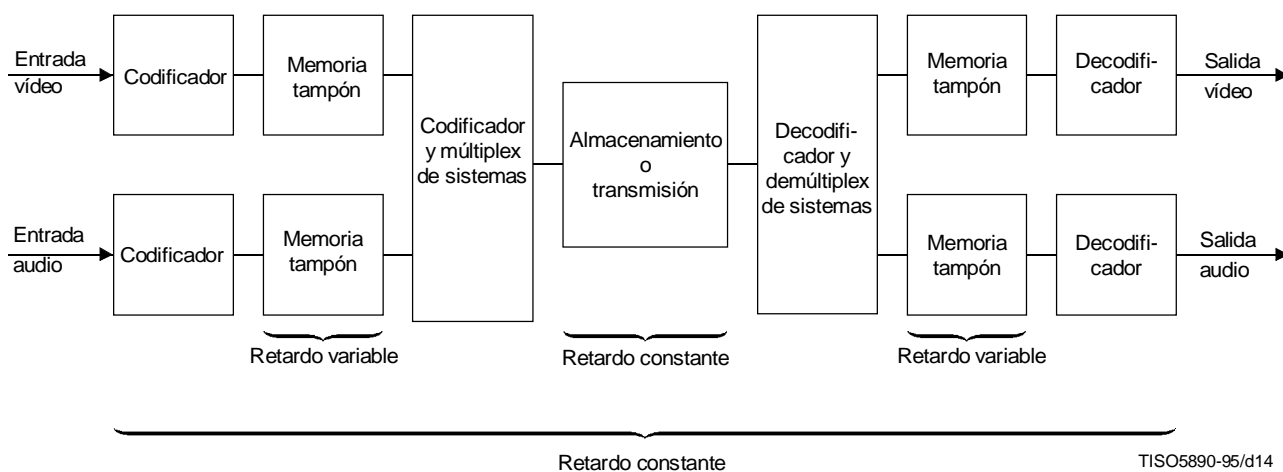
(Este anexo no es parte integrante de la presente Recomendación | Norma Internacional)

#### 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 realizar 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 realicen 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 realizar 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:



TISO5890-95/d14

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 modelo<sup>2)</sup>, 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 realizaciones de decodificador 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 realizado.

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 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 realizan 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 realizaciones 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 realizaciones, 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 realizar estos decodificadores.

<sup>2)</sup> 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.

## 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 la Norma ISO/CEI 13818-3, el término «trama de audio» tiene el mismo significado que AAU o APU según el contexto. Una VPU 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 especificación, 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 realizar 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 (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 restricción 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 realizaciones físicas de decodificadores para aclarar su especificación y facilitar su amplia aplicación a una variedad de realizaciones 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 realización. Por ejemplo, si se decodifican imágenes de vídeo exactamente en un intervalo de presentación de una imagen  $1/P$ , 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  $1/P$ , 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 retarde 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 realizar, por ejemplo, añadiendo una constante a los valores PTS cuando se utilizan dentro del decodificador.

Otra diferencia entre el STD y una realización 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 realizar 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 de la SCR\_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 realizado de diversas maneras, de modo que el STC del codificador puede ser una magnitud teórica.

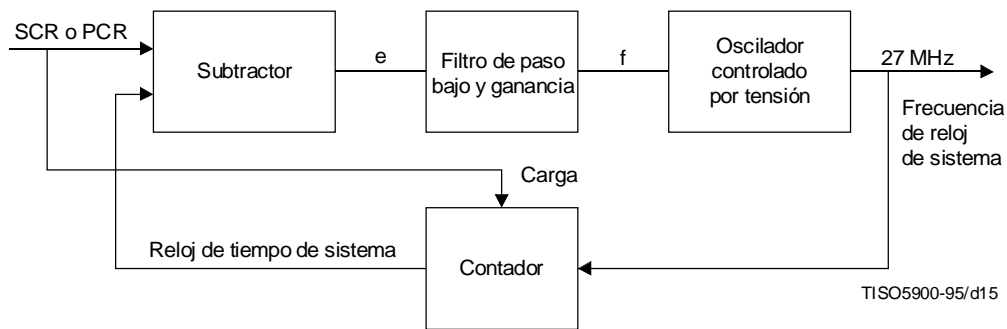
Si la frecuencia del reloj del decodificador concuerda exactamente con la del codificador, la decodificación y presentación de audio y vídeo tendrá automáticamente la misma velocidad que en el codificador, y el retardo de extremo a extremo será constante. Con frecuencias de reloj de codificador y decodificador 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. 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.

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.



**Figura D.2 – Recuperación del STC mediante un bucle enganchado en fase**

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). 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 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<sub>act</sub>* es el retardo real experimentado por la PCR y *del<sub>const</sub>* es una constante que se utiliza para todas las PCR de ese programa. El valor que se debe utilizar para *del<sub>const</sub>* 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<sub>const</sub>*.

**Cuadro D.1 – Estrategia de remultiplexación**

| Estrategia | <i>del<sub>const</sub></i> |
|------------|----------------------------|
| Prematura  | <i>del<sub>mín</sub></i>   |
| Tardía     | <i>del<sub>máx</sub></i>   |
| Media      | <i>del<sub>avg</sub></i>   |



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  $\pm 4$  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 y una velocidad de desviación de frecuencia del caso más desfavorable, medida en ppm/s. 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,  $f$ , 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 realización 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 realización 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 realizarse 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 deflexión 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 realizar sistemas de decodificación y de presentación con esta estructura.

En un tipo de realización, 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 realización 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 sustentar 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 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       |
| additional_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.

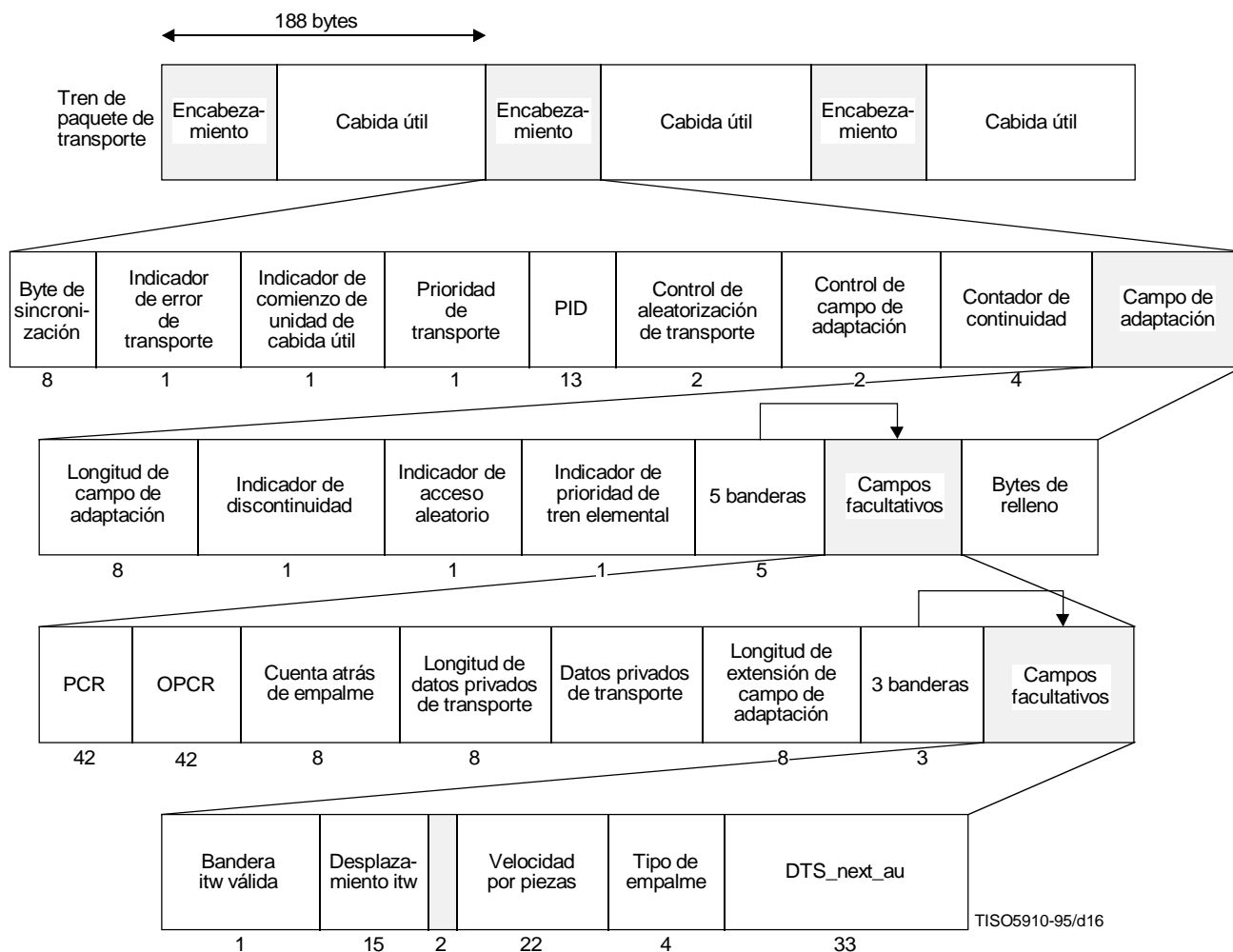
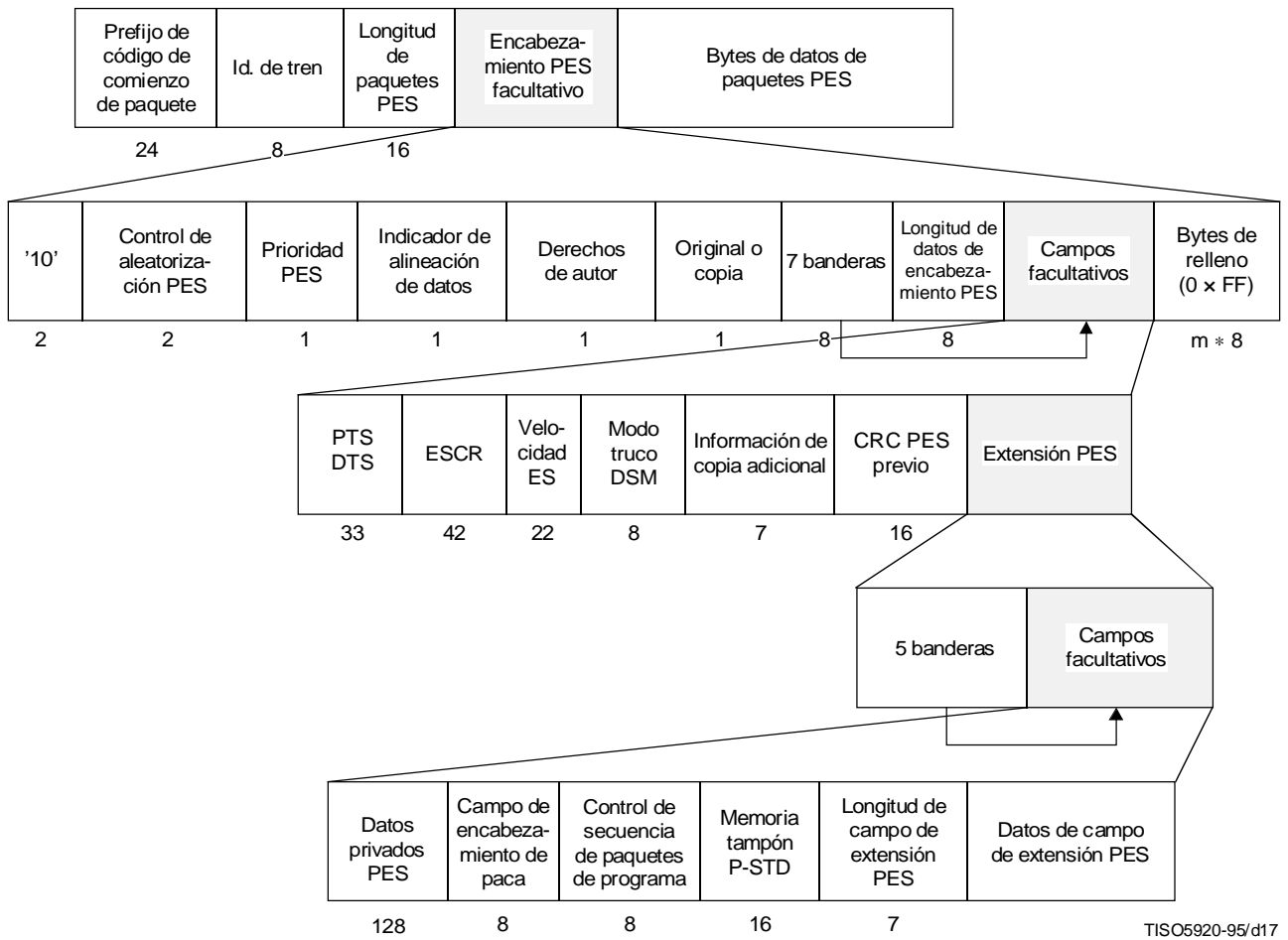


Figura F.1 – Diagrama de sintaxis de tren de transporte

**F.0.2 Paquete PES**

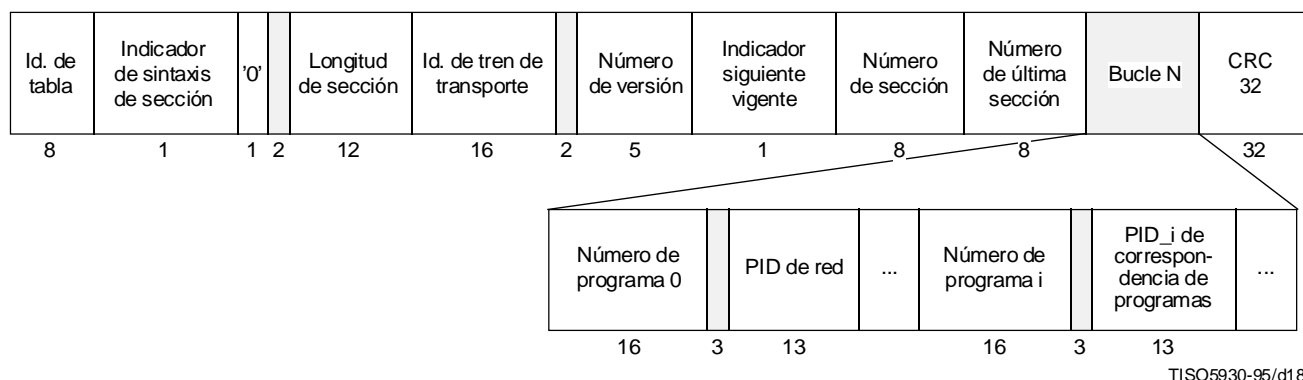
Véase la Figura F.2.



**Figura F.2 – Diagrama de sintaxis de paquete PES**

**F.0.3 Sección de asociación de programas**

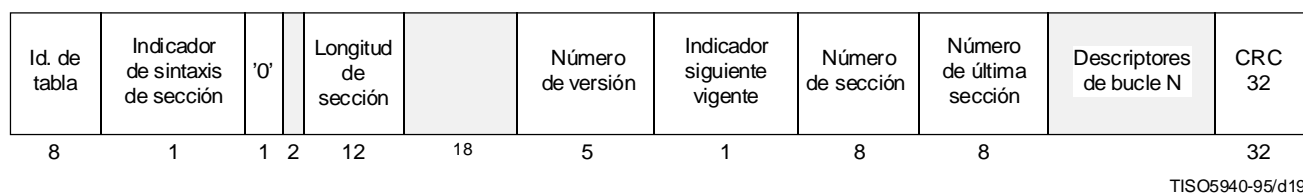
Véase la Figura F.3.



**Figura F.3 – Diagrama de sección de asociación de programas**

**F.0.4 Sección CA**

Véase la Figura F.4.

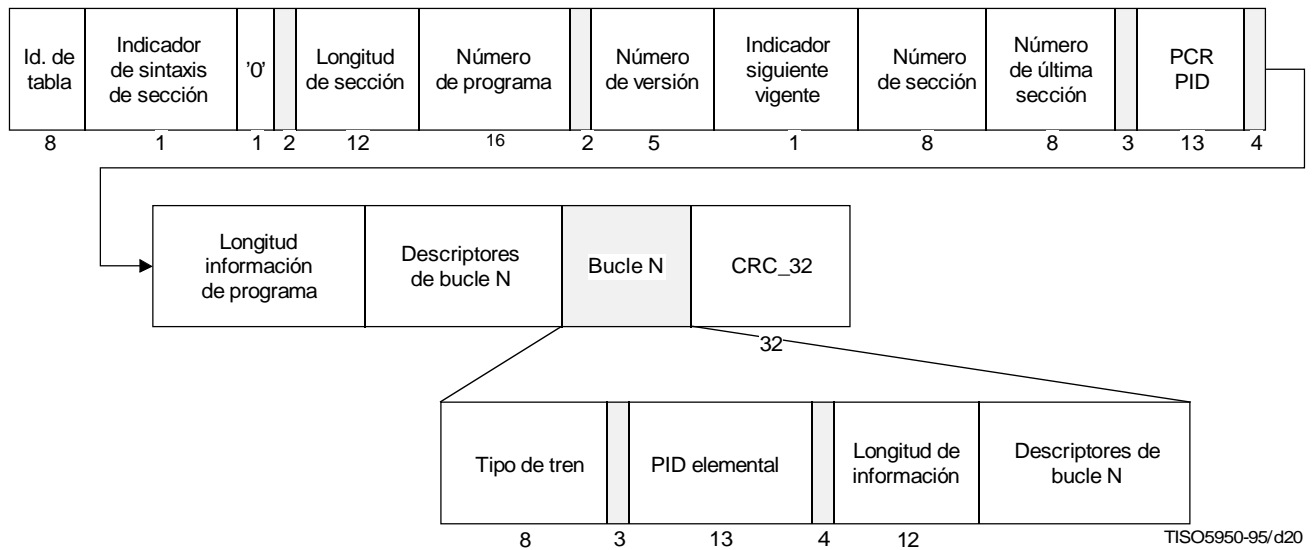


**Figura F.4 – Diagrama de sección de acceso condicional**



**F.0.5 Sección de correspondencia de programas de tren de transporte**

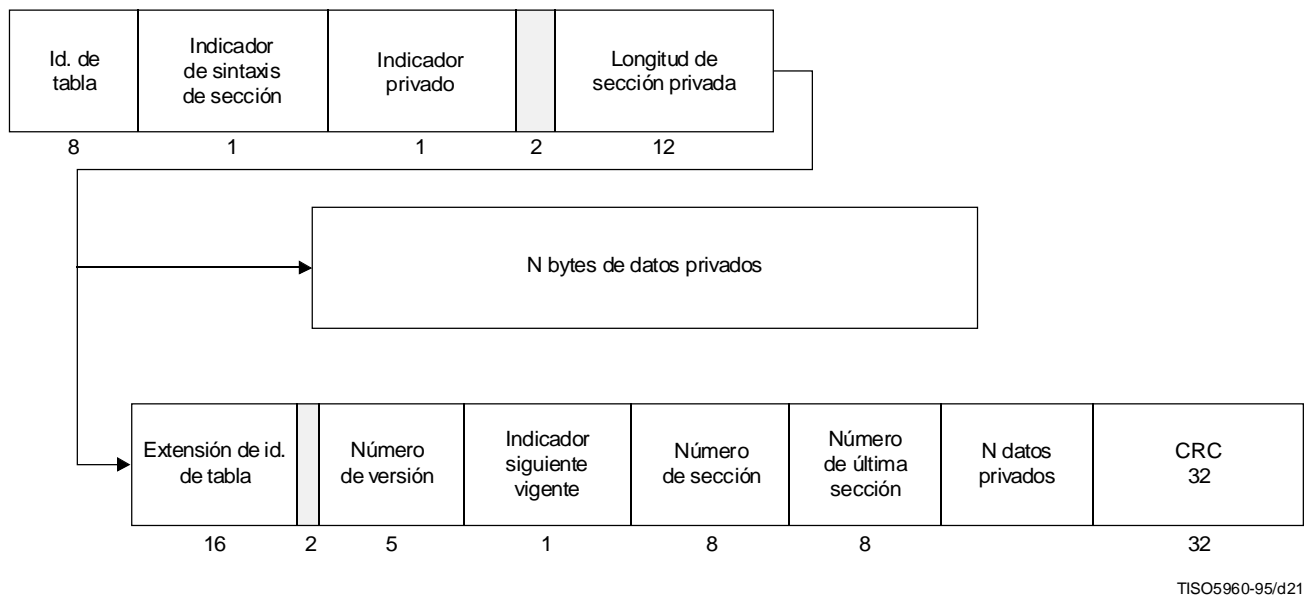
Véase la Figura F.5.



**Figura F.5 – Diagrama de sección de correspondencia de programas de tren de transporte**

**F.0.6 Sección privada**

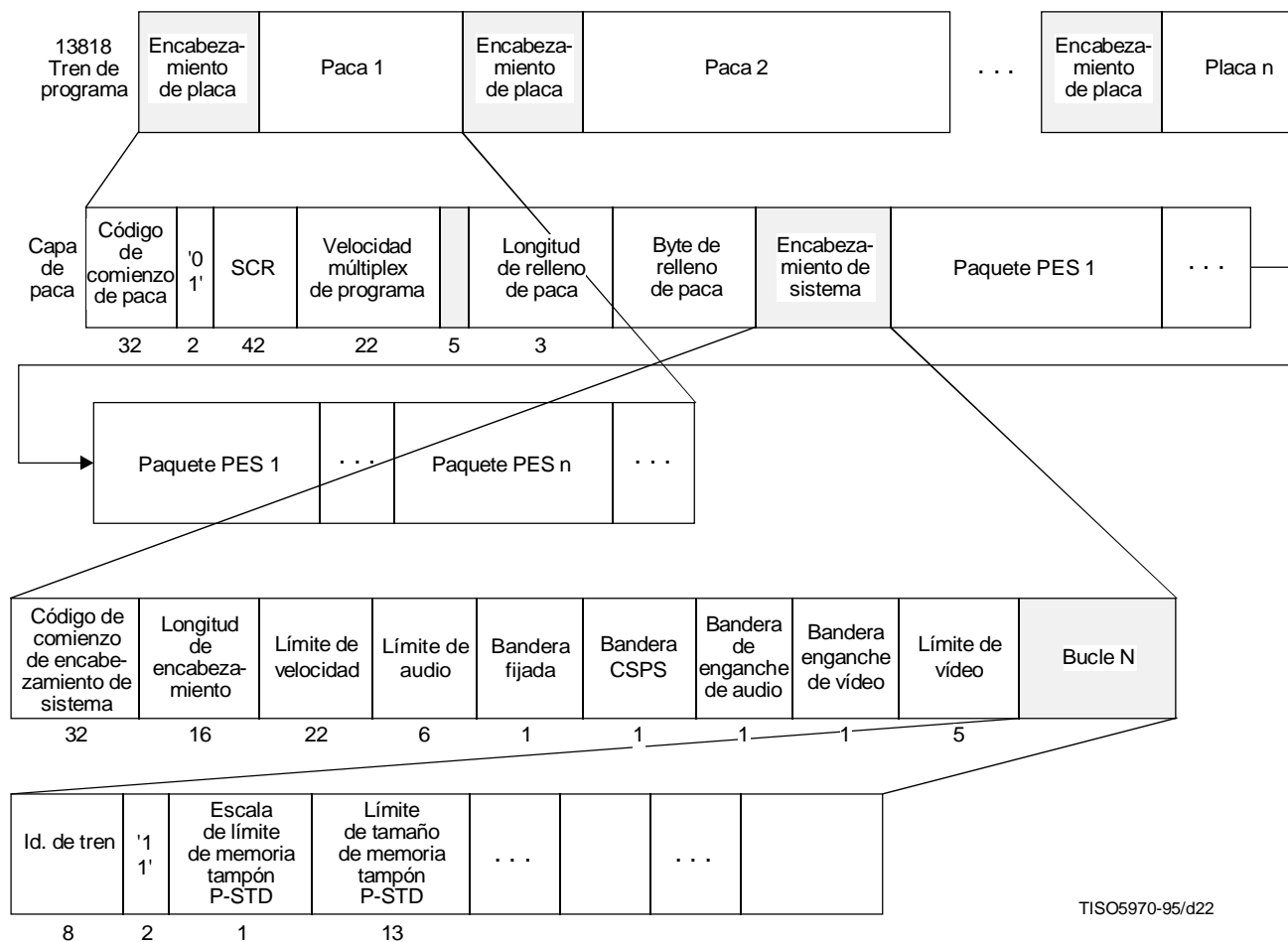
Véase la Figura F.6.



**Figura F.6 – Diagrama de sección privada**

F.0.7 Tren de programa

Véase la Figura F.7.

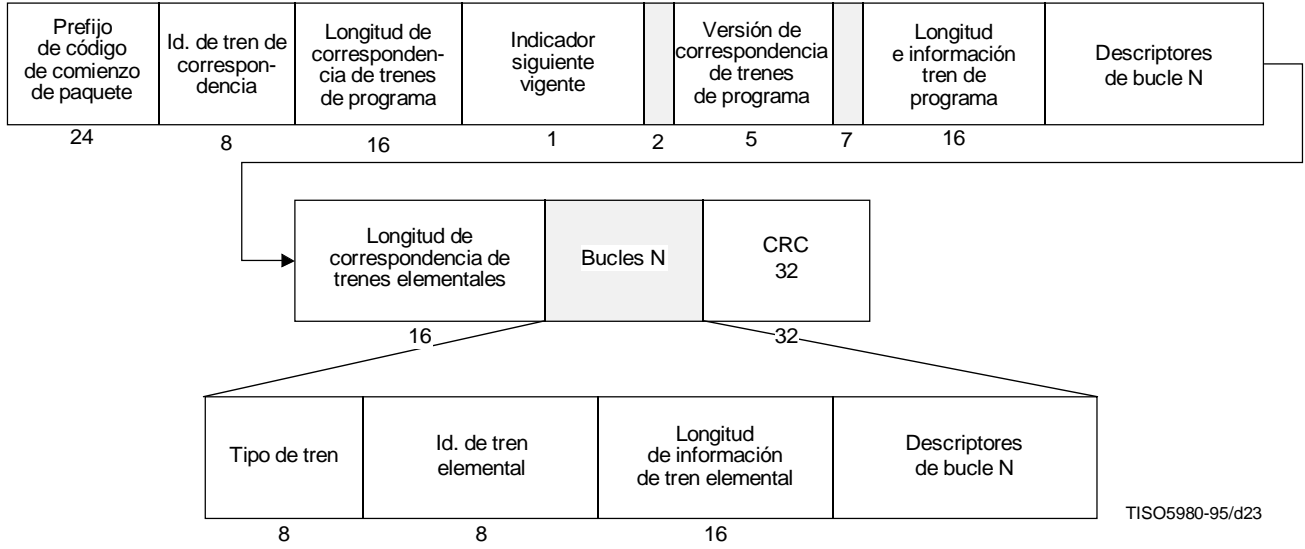


TISO5970-95/d22

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 2-29 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 2-17*

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 señalizada por la bandera de datos privados PES. La presencia de la bandera de datos privados PES es señalizada 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 2-29. 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 2-39. Estos descriptores se pueden colocar dentro de la tabla de correspondencia de trenes de programa (Cuadro 2-29); en la tabla de sección CA (Cuadro 2-27), en la sección de correspondencia de programas TS (Cuadro 2-28) y en cualquier sección privada (Cuadro 2-30).

Específicamente, los bytes de datos privados deben aparecer en el descriptor CA.

5) *Sección privada*

La sección privada, Cuadro 2-30, 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 2-29, 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 '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 realizaciones 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.2 se presenta un marco para elaborar modelos conformes de red.

En J.3 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.4 se presentan dos ejemplos de realizaciones 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.4.1. 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.4.2.

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.

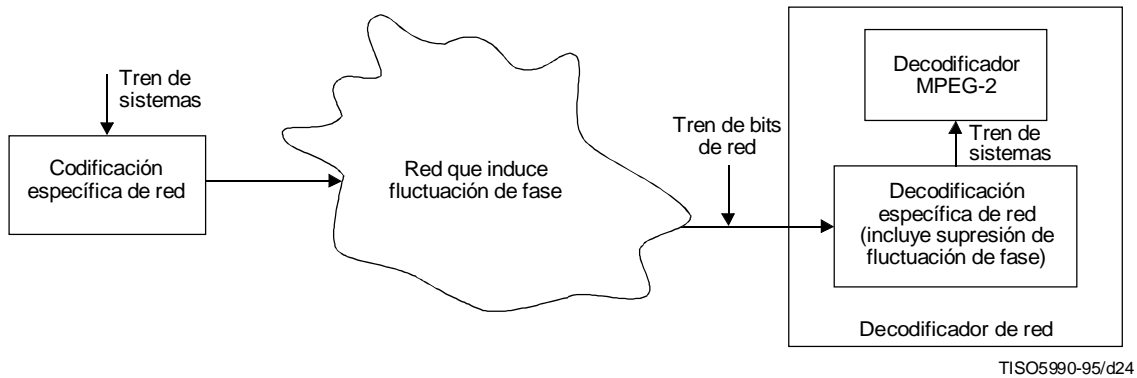


Figura J.1 – Envío de trenes de sistemas por una red que induce fluctuación de fase

### 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.

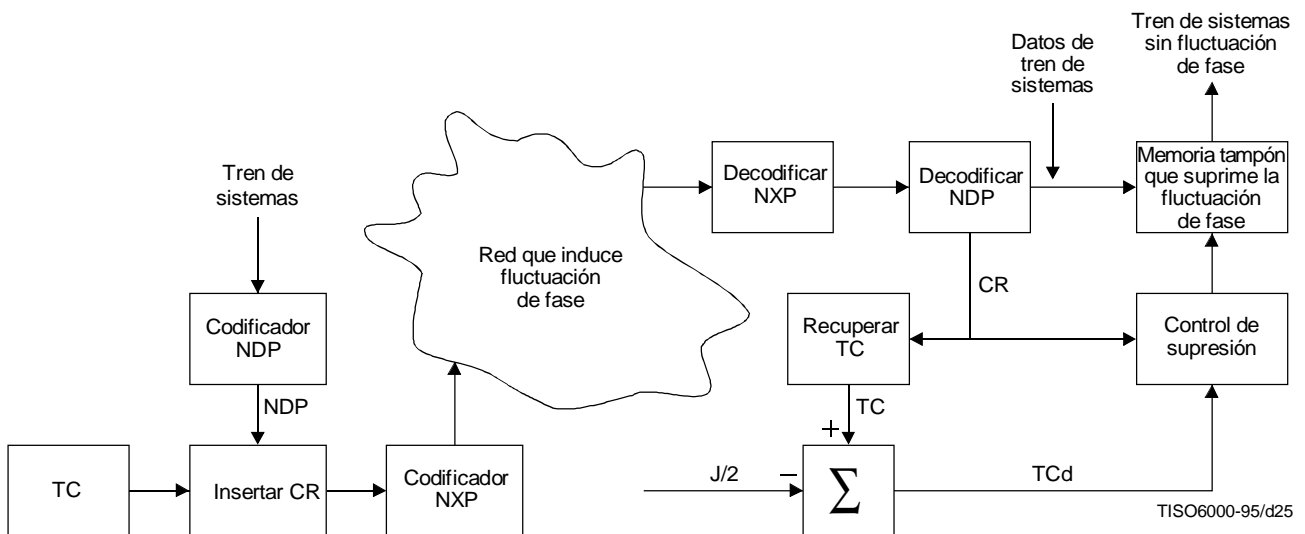


Figura J.2 – Supresión de fluctuación de fase mediante indicaciones de tiempo de la capa de red

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) 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) 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 realización, 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

$$B_{dj} = JR_{m\acute{a}x}$$

donde  $R_{m\acute{a}x}$  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 realizaciones de decodificador

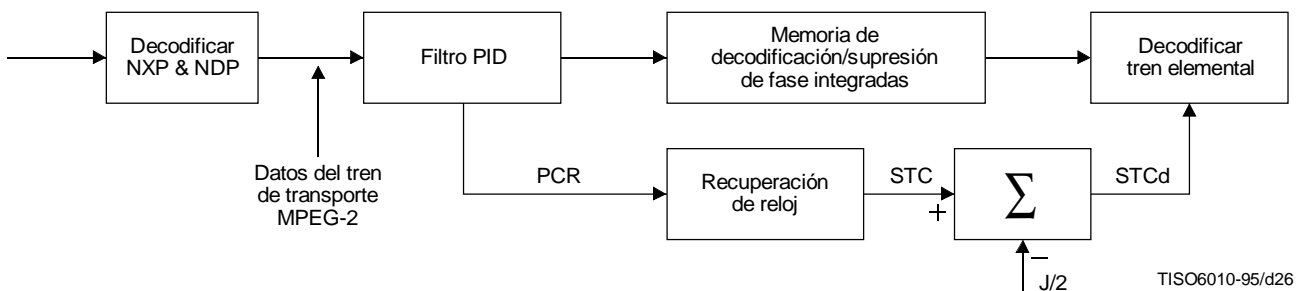
#### J.3.1 Adaptador de red seguido de un decodificador MPEG-2

En esta realización, 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.4.1 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 sección 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.



TISO6010-95/d26

Figura J.3 – Supresión de fluctuación de fase y decodificación MPEG-2 integradas

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 realización, tales como la velocidad del bucle de recuperación de reloj STC y la pureza espectral del STC,

$$B_{size} = B_{dec} + B_{mux} + B_{OH} + 512 + B_J, \quad = B_n + 512 + B_J$$

donde  $B_J = R_{m\acute{a}x} J$  y  $R_{m\acute{a}x}$  es la velocidad máxima a la cual los datos entran en el filtro PID. Según la realización, 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 restricción 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' ('Cut and Paste') en el nivel de transporte, al mismo tiempo que se respetan las restricciones 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

### K2.1 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 la Norma 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 funcionar 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 la Norma 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 2-16 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:

$$T_d \geq 0 \text{ y } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old}/R_{new})$$