

Unión Internacional de Telecomunicaciones

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**H.245**

(01/2005)

SERIE H: SISTEMAS AUDIOVISUALES Y  
MULTIMEDIOS

Infraestructura de los servicios audiovisuales –  
Procedimientos de comunicación

---

**Protocolo de control para comunicación  
multimedia**

Recomendación UIT-T H.245

UIT-T



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

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

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

## **Recomendación UIT-T H.245**

### **Protocolo de control para comunicación multimedia**

#### **Resumen**

La presente Recomendación especifica la sintaxis y la semántica de los mensajes de información de terminales, así como los procedimientos para utilizarlos en la negociación dentro de banda al comienzo de la comunicación o durante ésta. Los mensajes comprenden capacidades receptoras y transmisoras y la preferencia de modos desde el extremo receptor, la señalización de canal lógico y control e indicación. Se especifican procedimientos de señalización con acuse de recibo para garantizar la fiabilidad de las comunicaciones audiovisuales y de datos.

Con referencia a la versión 10 de esta Recomendación (07/2003), esta versión incluye:

- Soporte para cabida útil RTP conforme al documento RFC 3389 para ruido de confort (ruido nivelador).
- Soporte para velocidades binarias en tren de cabida útil múltiple.
- Definición de capacidad L16.
- Resolución de conflictos de canal lógico abierto H.245.
- Definición de tasa de paquetes de audio mínima.
- Rec. UIT-T H.235 Anexo G "Utilización del protocolo de gestión de clases MIKEY para el protocolo de transporte en tiempo real seguro (SRTP) en H.235".
- Expresiones de capacidades H.245 compactas.
- Capacidad genérica para retransmisión de circuitos Nx64K sobre IP.
- Actualización de la resolución de conflictos de canal lógico abierto H.245.
- Procedimiento de reiniciación de llamada en la Rec. UIT-T H.324.

Los productos que alegan conformidad con esta versión 11 de la Rec. UIT-T H.245 deben satisfacer todos los requisitos obligatorios de esta Recomendación. Los productos de la versión 11 se pueden identificar a través de los mensajes H.245 TerminalCapabilitySet que contienen un valor protocolIdentifier de {itu-t (0) recommendation (0) h (8) 245 version (0) 11}.

#### **Orígenes**

La Recomendación UIT-T H.245 fue aprobada el 8 de enero de 2005 por la Comisión de Estudio 16 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

## PREFACIO

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

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

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

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

## NOTA

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

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

## PROPIEDAD INTELECTUAL

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

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

© UIT 2005

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

## ÍNDICE

	<b>Página</b>
1 Alcance .....	1
2 Referencias .....	1
2.1 Referencias normativas .....	1
2.2 Referencia informativa .....	5
3 Definiciones.....	5
4 Abreviaturas, siglas o acrónimos.....	7
5 Generalidades .....	8
5.1 Determinación principal-subordinado .....	9
5.2 Intercambio de capacidades.....	9
5.3 Procedimientos de señalización de canal lógico .....	10
5.4 Petición de cierre de canal lógico por el terminal receptor .....	10
5.5 Modificación de entrada en la tabla múltiplex H.223 .....	10
5.6 Petición de modo audiovisual y de modo datos .....	11
5.7 Determinación del retardo de ida y vuelta.....	11
5.8 Bucles de mantenimiento .....	11
5.9 Instrucciones e indicaciones .....	11
Anexo A – Mensajes: Sintaxis.....	12
Anexo B – Mensajes: Definiciones semánticas .....	78
B.1 Mensajes de determinación principal-subordinado .....	78
B.2 Mensajes de capacidad de terminal .....	79
B.3 Mensajes de señalización de canal lógico .....	112
B.4 Mensajes de señalización de tabla múltiplex.....	122
B.5 Mensajes de señalización de petición de tabla múltiplex .....	124
B.6 Mensajes de petición de modo .....	124
B.7 Mensajes de retardo de ida y vuelta .....	129
B.8 Mensajes de bucle de mantenimiento.....	129
B.9 Mensajes de modo de comunicación.....	130
B.10 Mensajes de petición y respuesta de conferencia .....	131
B.11 Mensajes multienlace .....	134
B.12 Mensajes de cambio de velocidad binaria de canal lógico.....	136
B.13 Instrucciones.....	137
B.14 Indicaciones .....	144
B.15 Mensajes genéricos.....	152
Anexo C – Procedimientos .....	152
C.1 Introducción.....	152
C.2 Procedimientos de determinación principal-subordinado .....	155
C.3 Procedimientos de intercambio de capacidad.....	165
C.4 Procedimientos de señalización de canal lógico unidireccional.....	173

	<b>Página</b>
C.5	Procedimientos de señalización de canal lógico bidireccional..... 186
C.6	Procedimientos de cierre de canal lógico ..... 201
C.7	Procedimientos de la tabla múltiplex H.223..... 207
C.8	Procedimientos de petición de entrada múltiplex..... 218
C.9	Procedimientos de petición de modo..... 225
C.10	Procedimientos de retardo de ida y vuelta..... 235
C.11	Procedimientos de bucle de mantenimiento ..... 239
Anexo D	– Asignaciones de identificadores de objetos..... 250
Anexo E	– Definiciones de capacidades visuales ISO/CEI 14496-2..... 253
Anexo F	– Definiciones de capacidades de gestión de velocidad binaria de canal lógico ..... 255
Anexo G	– Definiciones de capacidades ISO/CEI 14496-1 ..... 257
G.1	Identificador de capacidades ..... 257
G.2	Parámetros de capacidad utilizados para negociaciones de capacidades y señalización de canal lógico ..... 257
G.3	Parámetros de capacidad utilizados para la señalización de canal lógico ..... 259
Anexo H	– Definiciones de capacidades ISO/CEI 14496-3 ..... 260
Anexo I	– Definiciones de la capacidad de multivelocidad adaptable GSM..... 264
I.1	Definición de señalización de modo y relleno de bits para lograr la alineación de octetos..... 266
Anexo J	– Definiciones del códec vocal TDMA ACELP..... 275
Anexo K	– Definiciones del códec vocal TDMA US1 ..... 276
Anexo L	– Definiciones del códec vocal EVRC CDMA..... 278
Anexo M	– Definiciones ISO/CEI 13818-7 e UIT-R BS.1196 ..... 279
Anexo N	– RFC 3389 – Cabida útil RTP para ruido de confort..... 280
Anexo O	– Identificador de capacidad L16 ..... 281
Anexo P	– Capacidad limitada del tren de audio ..... 282
Anexo Q	– Capacidad genérica para retransmisión de circuitos Nx64K por el protocolo Internet..... 283
Q.1	Introducción..... 283
Q.2	Descripción..... 283
Apéndice I	– Visión general de la sintaxis en ASN.1 ..... 287
I.1	Introducción a la ASN.1 ..... 287
I.2	Tipos de datos ASN.1 básicos ..... 287
I.3	Tipos de datos agregados..... 289
I.4	Tipo identificador de objeto ..... 290
Apéndice II	– Ejemplos de procedimientos H.245 ..... 291
II.1	Introducción..... 291
II.2	Entidad de señalización de determinación principal-subordinado ..... 291
II.3	Entidad de señalización de intercambio de capacidad ..... 295

	<b>Página</b>
II.4 Entidad de señalización de canal lógico .....	297
II.5 Entidad de señalización de cierre de canal lógico .....	299
II.6 Entidad de señalización de tabla múltiplex .....	300
II.7 Entidad de señalización de petición de modo.....	302
II.8 Entidad de señalización de retardo de ida y vuelta.....	304
II.9 Entidad de señalización de canal lógico bidireccional .....	305
Apéndice III – Sumario de temporizadores y contadores de procedimiento .....	307
III.1 Temporizadores .....	307
III.2 Contadores.....	308
Apéndice IV – Procedimiento de extensión de H.245 .....	309
Apéndice V – Procedimiento replacementFor .....	310
Apéndice VI – Ejemplos de fijaciones de parámetros de la estructura de capacidad H.263 ...	312
VI.1 Ejemplos de fijación de parámetros H.245 de capa de mejora.....	312
VI.2 Ejemplos de fijación de parámetros H.245 de canal de retorno de vídeo .....	313
Apéndice VII – Procedimiento y plantilla para definir nuevas capacidades con las capacidades genéricas H.245 .....	317
VII.1 Procedimiento.....	318
VII.2 Plantilla.....	318
VII.3 Ejemplo de plantilla – H.261 .....	319
Apéndice VIII – Lista de capacidades genéricas y mensajes genéricos definidos en Recomendaciones/Normas distintas de esta Recomendación .....	321
Apéndice IX – Utilización de la notación ASN.1 en la presente Recomendación .....	323
IX.1 Rotulación.....	324
IX.2 Tipos .....	324
IX.3 Limitaciones y gamas .....	324
IX.4 Extensibilidad .....	324
Apéndice X – Resolución de escenarios de conflictos unidireccionales y bidireccionales.....	324
X.1 Ambos terminales utilizan OLC bidireccional .....	324
X.2 El terminal principal propone OLC bidireccional y el terminal subordinado propone OLC unidireccional .....	325
X.3 El terminal principal propone OLC unidireccional y el terminal subordinado OLC bidireccional .....	326
X.4 El terminal principal propone OLC bidireccional con nullData y el terminal subordinado propone OLC unidireccional .....	326
X.5 Ambos terminales proponen OLC bidireccional con nullData .....	327





# Recomendación UIT-T H.245

## Protocolo de control para comunicación multimedia

### 1 Alcance

La presente Recomendación especifica la sintaxis y la semántica de los mensajes de información de terminal así como los procedimientos para utilizarlos en la negociación en banda al comienzo de la comunicación o durante ésta. Los mensajes comprenden capacidades de recepción y transmisión así como preferencia de modos desde el extremo de recepción, la señalización de canal lógico, y control e indicación. Se especifican procedimientos de señalización con acuse de recibo para garantizar comunicaciones fiables audiovisuales y de datos.

Esta Recomendación abarca una amplia gama de aplicaciones que incluyen servicios de almacenamiento/extracción, mensajería y distribución, así como servicios conversacionales. Se aplica a sistemas multimedia que utilizan múltiplex definidos en las Recs. UIT-T H.222.0, H.223 y H.225.0 aunque no se limita a tales sistemas únicamente. Estos sistemas comparten la misma sintaxis y semántica, por lo que son compatibles en términos de bits. Algunos de los procedimientos son aplicables a la totalidad de los sistemas, en tanto que otros son más específicos de sistemas concretos.

Los diferentes sistemas que utilizan esta Recomendación pueden especificar el empleo de diferentes protocolos de transporte. Sin embargo, se ha previsto su utilización con una capa de transporte fiable, es decir, que proporciona una entrega garantizada de datos correctos.

NOTA – No debería haber ninguna confusión con el sistema de gestión T.120, que es transportado dentro del tren de datos y comprende funcionalidades diferentes de las que aquí se describen; el tren H.245 y el tren de datos T.120 son complementarios.

### 2 Referencias

#### 2.1 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- [1] Recomendación UIT-T E.164 (1997), *Plan internacional de numeración de telecomunicaciones públicas*.
- [2] Recomendación UIT-T G.711 (1988), *Modulación por impulsos codificados (MIC) de frecuencias vocales*.
- [3] Recomendación UIT-T G.722 (1988), *Codificación de audio de 7 kHz dentro de 64 kbit/s*.
- [4] Recomendación UIT-T G.723.1 (1996), *Codificadores vocales: Códec de voz de doble velocidad para la transmisión en comunicaciones multimedios a 5,3 y 6,3 kbit/s*.
- [5] Recomendación UIT-T G.728 (1992), *Codificación de señales vocales a 16 kbit/s utilizando predicción lineal con excitación por código de bajo retardo*.
- [6] Recomendación UIT-T G.729 (1996), *Codificación de la voz a 8 kbit/s mediante predicción lineal con excitación por código algebraico de estructura conjugada*.

- [7] Recomendación UIT-T H.221 (1999), *Estructura de trama para un canal de 64 a 1920 kbit/s en teleservicios audiovisuales*.
- [8] Recomendación UIT-T H.222.0 (2000) | ISO/CEI 13818-1:2000, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Sistemas*.
- [9] Recomendación UIT-T H.222.1 (1996), *Multiplexación y sincronización multimedia para comunicación audiovisual en entornos del modo de transferencia asíncrono*.
- [10] Recomendación UIT-T H.223 (1996), *Protocolo de multiplexación para comunicación multimedios a baja velocidad binaria*.
- [11] Recomendación UIT-T H.224 (2000), *Protocolo de control en tiempo real para aplicaciones simplex que utilizan los canales de datos a baja velocidad, datos a alta velocidad y protocolo multicapa de la Recomendación H.221*.
- [12] Recomendación UIT-T H.225.0 (2003), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.
- [13] Recomendación UIT-T H.230 (1999), *Señales de control e indicación con sincronismo de trama para sistemas audiovisuales*.
- [14] Recomendación UIT-T H.233 (1995), *Sistemas con confidencialidad para servicios audiovisuales*.
- [15] Recomendación UIT-T H.234 (1994), *Sistema de gestión de claves de criptación y de autenticación para servicios audiovisuales*.
- [16] Recomendación UIT-T H.235 (2003), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245), más enmienda 1 (2004)*.
- [17] Recomendación UIT-T H.243 (2000), *Procedimientos para el establecimiento de comunicaciones entre tres o más terminales audiovisuales con utilización de canales digitales de hasta 1920 kbit/s*.
- [18] Recomendación UIT-T H.261 (1993), *Códec vídeo para servicios audiovisuales a  $p \times 64$  kbit/s*.
- [19] Recomendación UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Vídeo*.
- [20] Recomendación UIT-T H.263 (1998), *Codificación de vídeo para comunicación a baja velocidad binaria*.
- [21] Recomendación UIT-T H.310 (1998), *Sistemas y terminales para comunicaciones audiovisuales de banda ancha*.
- [22] Recomendación UIT-T H.320 (1999), *Sistemas y equipos terminales videotelefónicos de banda estrecha*.
- [23] Recomendación UIT-T H.323 (2003), *Sistemas de comunicación multimedios basados en paquetes*.
- [24] Recomendación UIT-T H.324 (2002), *Terminal para comunicación multimedios a baja velocidad binaria*.
- [25] Recomendaciones UIT-T de la serie I.363.x, *Especificación de la capa de adaptación del modo de transferencia asíncrono de la red digital de servicios integrados de banda ancha*.

- [26] Recomendación UIT-T Q.2931 (1995), *Sistema de señalización digital de abonado N.º 2 – Especificación de la capa 3 de la interfaz usuario-red para el control de llamada/conexión básica.*
- [27] Recomendación UIT-T T.30 (2003), *Procedimientos de transmisión de documentos por facsímil por la red telefónica general conmutada.*
- [28] Recomendación UIT-T T.35 (2000), *Procedimiento para la asignación de códigos definidos por el UIT-T para facilidades no normalizadas.*
- [29] Recomendación UIT-T T.38 (2003), *Procedimientos para la comunicación facsímil en tiempo real entre terminales facsímil del Grupo 3 por redes con protocolo Internet.*
- [30] Recomendación UIT-T T.51 (1992), *Juegos de caracteres codificados basados en el alfabeto latino para los servicios de telemática.*
- [31] Recomendación UIT-T T.84 (1996) | ISO/CEI 10918-3:1997, *Tecnología de la información – Compresión y codificación digitales de imágenes fijas de tonos continuos: Ampliaciones.*
- [32] Recomendación UIT-T T.120 (1996), *Protocolo de datos para conferencias multimedios.*
- [33] Recomendación UIT-T T.123 (1999), *Pilas de protocolos de datos específicos de la red para conferencias multimedios.*
- [34] Recomendación UIT-T T.140 (1998), *Protocolo de conversación mediante texto para aplicaciones multimedios.*
- [35] Recomendación UIT-T T.434 (1999), *Formato de transferencia de ficheros binarios para servicios de telemática.*
- [36] Recomendación UIT-T V.14 (1993), *Transmisión de caracteres arrítmicos por canales portadores síncronos.*
- [37] Recomendación UIT-T V.34 (1998), *Módem que funciona a velocidades de señalización de datos de hasta 33 600 bit/s para uso en la red telefónica general conmutada y en circuitos arrendados punto a punto a dos hilos de tipo telefónico.*
- [38] Recomendación UIT-T V.42 (2002), *Procedimientos de corrección de errores para los equipos de terminación del circuito de datos que utilizan la conversión de modo asíncrono a modo síncrono.*
- [39] Recomendación UIT-T V.140 (1998), *Procedimientos para el establecimiento de comunicaciones entre dos terminales audiovisuales multiprotocolo que utilizan canales digitales a un múltiplo de 64 ó 56 kbit/s.*
- [40] Recomendación UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- [41] Recomendación UIT-T X.681 (2002) | ISO/CEI 8824-2:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- [42] Recomendación UIT-T X.691 (2002) | ISO/CEI 8825-2:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno – Especificación de las reglas de codificación compactada.*
- [43] ISO/CEI 13239:2002, *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures.*
- [44] 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.*
- [45] 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.*

- [46] ISO/CEI 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio.*
- [47] ISO/CEI 13818-6:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC.*
- [48] ISO/CEI 14496-1:1999, *Information technology – Coding of audio-visual objects – Part 1: Systems.*
- [49] ISO/CEI 14496-2:1999, *Information technology – Coding of audio-visual objects – Part 2: Visual.*
- [50] ISO/CEI 14496-3:1999, *Information technology – Coding of audio-visual objects – Part 3: Audio.*
- [51] ISO/CEI 14496-3/Amd.1:2000, *Information technology – Coding of audio-visual objects – Part 3: Audio – Amendment 1: Audio extensions.*
- [52] ISO/CEI TR 9577:1999, *Information technology – Protocol identification in the network layer.*
- [53] ETSI ETS 300 961 (GSM 06.10), *Full rate speech transcoding.*
- [54] ETSI ETS 300 969 (GSM 06.20), *Half rate speech transcoding.*
- [55] ETSI ETS 300 726 (GSM 06.60), *Enhanced Full Rate (EFR) speech transcoding.*
- [56] ETSI ETS 300 963 (GSM 06.12), *Comfort noise aspect for full rate speech traffic channels.*
- [57] ETSI ETS 300 971 (GSM 06.22), *Comfort noise aspects for half rate speech traffic channels.*
- [58] ETSI ETS 300 728 (GSM 06.62), *Comfort noise aspects for Enhanced Full Rate (EFR) speech traffic channels.*
- [59] ETSI ETS 300 964 (GSM 06.31), *Discontinuous Transmission (DTX) for full rate speech traffic channels.*
- [60] ETSI ETS 300 972 (GSM 06.41), *Discontinuous transmission (DTX) for half rate speech traffic channels.*
- [61] ETSI ETS 300 729 (GSM 06.81), *Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels.*
- [62] ETSI ETS 300 962 (GSM 06.11), *Substitution and muting of lost frames for full rate speech traffic channels.*
- [63] ETSI ETS 300 970 (GSM 06.21), *Substitution and muting of lost frames for half rate speech traffic channels.*
- [64] ETSI ETS 300 727 (GSM 06.61), *Substitution and muting of lost frames for Enhanced Full Rate (EFR) speech traffic channels.*
- [65] ETSI ETS 300 965 (GSM 06.32), *Voice Activity Detector (VAD) for full rate speech traffic channels.*
- [66] ETSI ETS 300 973 (GSM 06.42), *Voice Activity Detector (VAD) for half rate speech traffic channels.*
- [67] ETSI ETS 300 730 (GSM 06.82), *Voice activity detection for enhanced full rate speech traffic channels.*
- [68] ETSI ETS 300 724 (GSM 06.53), *ANSI-C code for the GSM Enhanced Full Rate Speech (EFR) speech codec.*

- [69] ETSI EN 301 712 (GSM 06.73), *ANSI-C code for the AMR speech codec*.
- [70] ETSI EN 301 704 (GSM 06.90), *Adaptive Multi-Rate (AMR) speech transcoding*.
- [71] ETSI EN 301 705 (GSM 06.91), *Substitution and muting of lost frames for Adaptive Multi Rate (AMR) speech traffic channels*.
- [72] ETSI EN 301 706 (GSM 06.92), *Comfort noise aspects for Adaptive Multi-Rate (AMR) speech traffic channels*.
- [73] ETSI EN 301 708 (GSM 06.94), *Voice Activity Detection (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels*.
- [74] RCR STD-27H, *Personal Digital Cellular Telecommunication System RCR Standard*.
- [75] TIA/EIA – 136-Rev.A, Part 410, *TDMA Cellular/PCS – Radio Interface, Enhanced Full Rate Voice Codec (ACELP). Formerly IS-641. TIA published standard, 1998*.
- [76] TIA/EIA/IS 641-A (1998), *TDMA Cellular/PCS – Radio Interface, US1 Full Rate Voice Codec*.
- [77] Recomendación UIT-T H.239 (2003), *Gestión de funciones y canales de medios adicionales para terminales de la serie H.300*.
- [78] Recomendación UIT-T H.241 (2003), *Señales de control y procedimientos de vídeo extendidos para terminales de la serie H.300*.
- [79] Recomendación UIT-T H.235 anexo G (2005), *Utilización del protocolo de gestión de claves MIKEY para el protocolo de transporte en tiempo real seguro (SRTP) en H.235*.
- [80] Recomendación UIT-T Y.1413 (2004), *Interfuncionamiento de redes con conmutación por etiquetas multiprotocolo y multiplexación por división en el tiempo – Interfuncionamiento en el plano de usuario*.
- [81] IETF RFC 2198 (1997), *RTP Payload for Redundant Audio Data*.
- [82] IETF RFC 2733 (1999), *An RTP Payload Format for Forward Error Correction*.
- [83] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.
- [84] IETF RFC 3711 (2004), *The secure Real-time Transport Protocol*.

## 2.2 Referencia informativa

- [PWE3] BRYANT (S.), PATE (P.): PWE3 Architecture, <draft-ietf-pwe3-arch-07.txt>, marzo de 2003.

## 3 Definiciones

En esta Recomendación se definen los términos siguientes.

- 3.1 canal lógico bidireccional:** Un canal lógico bidireccional consiste en un par de trayectos de transmisión asociados, entre dos terminales, uno para cada sentido de la transmisión.
- 3.2 capacidad:** Un terminal tiene una determinada capacidad si puede codificar y transmitir, o recibir y decodificar, esa señal determinada.
- 3.3 canal:** Un canal es un enlace unidireccional entre dos puntos extremos.
- 3.4 instrucción:** Una instrucción es un mensaje que requiere una acción, pero no una respuesta explícita.
- 3.5 tren elemental:** Un tren elemental es un término genérico para trenes binarios de vídeo codificado, de audio codificado o trenes binarios codificados de otros tipos.

- 3.6 entrada:** El término "entrada" se utiliza para hacer referencia a elementos en conjuntos o tablas, como conjuntos de capacidades y tablas múltiplex.
- 3.7 hacia adelante:** Se utiliza la expresión hacia adelante, para hacer referencia a la transmisión del terminal que hace la petición, en un canal lógico bidireccional, al otro terminal.
- 3.8 en banda:** Son mensajes en banda los transportados dentro del canal, o dentro del canal lógico, a que se refieren esos mensajes.
- 3.9 de entrada:** Una entidad de señalización de entrada no puede iniciar un procedimiento, sino que responde a mensajes provenientes de la entidad de señalización distante y a las primitivas de su propio usuario.
- 3.10 indicación:** Una indicación es un mensaje que contiene información, pero que no requiere acción ni respuesta.
- 3.11 canal lógico:** Un canal lógico es un trayecto unidireccional o un trayecto bidireccional para la transmisión de información.
- 3.12 número de canal lógico:** Un número de canal lógico es un número que identifica un canal lógico individual.
- 3.13 señalización de canal lógico:** Señalización de canal lógico es un conjunto de procedimientos que se utilizan para abrir y cerrar canales lógicos.
- 3.14 terminal principal:** Un terminal principal es el terminal que ha sido determinado como el terminal principal por el procedimiento de determinación principal-subordinado, definido en esta Recomendación, o por algún otro procedimiento.
- 3.15 tipo de medio:** Un tipo de medio es una forma individual de información que se presenta a un usuario o los datos que representan esa información: vídeo, audio y texto son ejemplos de tipos de medio.
- 3.16 modo:** Un modo es un conjunto de trenes elementales que un terminal está transmitiendo, tiene el propósito de transmitir o quisiera recibir.
- 3.17 comunicación multimedia:** Por comunicación multimedia ha de entenderse la transmisión y/o recepción simultáneas de señales de dos o más tipos de medios.
- 3.18 no normalizado:** No conforme con una norma nacional o internacional a que se hace referencia en esta Recomendación.
- 3.19 de salida:** Una entidad de señalización de salida es una entidad que inicia un procedimiento.
- 3.20 multipunto:** Por multipunto ha de entenderse la interconexión simultánea de tres o más terminales para permitir la comunicación entre varios puntos mediante el uso de unidades (puentes) de control multipunto que dirigen el flujo de información de forma centralizada.
- 3.21 petición:** Una petición es un mensaje como resultado del cual el terminal distante ejecuta una acción y que requiere una respuesta inmediata de dicho terminal.
- 3.22 respuesta:** Una respuesta es un mensaje por el que se responde a una petición.
- 3.23 inverso:** Este calificativo se utiliza para hacer referencia a una transmisión dirigida del terminal que recibe una petición, en el caso de un canal lógico bidireccional, al terminal que hace la petición.
- 3.24 sesión:** Una sesión es un periodo de comunicación entre dos terminales, que puede ser o no conversacional (por ejemplo, la extracción de datos de una base de datos).

**3.25 terminal subordinado:** Un terminal subordinado es un terminal que se determina como el terminal subordinado por el procedimiento de determinación director-subordinado definido en esta Recomendación o por algún otro procedimiento.

**3.26 soporte:** La aptitud para funcionar en un determinado modo, si bien el requisito de soportar un modo no significa que el modo tiene que ser utilizado en todo momento: a menos que se prohíba, pueden utilizarse otros modos por negociación mutua.

**3.27 terminal:** Un terminal es cualquier punto extremo, y puede ser un terminal de usuario o de algún otro sistema de comunicación, como una MCU u otro servidor de información.

**3.28 identificador de TSAP:** El elemento de información utilizada para multiplexar varias conexiones de transporte del mismo tipo en una sola entidad H.323, donde todas las conexiones de transporte comparten la misma dirección LAN (por ejemplo, el número de puerto en un entorno TCP/UDP/IP). Los identificadores de TSAP pueden ser asignados (o preasignados) por alguna autoridad internacional o pueden ser atribuidos dinámicamente durante el establecimiento de una comunicación. Los identificadores de TSAP asignados dinámicamente son de naturaleza transitoria, es decir, sus valores sólo son válidos durante la existencia de una llamada individual.

**3.29 canal lógico unidireccional:** Un canal lógico unidireccional es un trayecto para la transmisión de un tren elemental individual de un terminal a otro.

#### 4 Abreviaturas, siglas o acrónimos

En esta Recomendación se utilizan las siguientes abreviaturas, siglas o acrónimos.

AAL	Capa de adaptación ATM ( <i>ATM adaptation layer</i> )
AL1,2,3	Capas de adaptación 1, 2 y 3 H.223 ( <i>H.223 adaptation layers 1, 2 and 3</i> )
ASN.1	Notación de sintaxis abstracta uno ( <i>abstract syntax notation one</i> )
ATM	Modo de transferencia asíncrono ( <i>asynchronous transfer mode</i> )
B-LCSE	Entidad de señalización de canal lógico bidireccional ( <i>bidirectional logical channel signalling entity</i> )
CESE	Entidad de señalización de intercambio de capacidad ( <i>capability exchange signalling entity</i> )
CIF	Formato intermedio común ( <i>common intermediate format</i> ) (de una imagen vídeo: véanse las Recs. UIT-T H.261 y H.263)
CLCSE	Entidad de señalización de cierre de canal lógico ( <i>close logical channel signalling entity</i> )
CPCS	Subcapa de convergencia de parte común ( <i>common part convergence sublayer</i> ) (de la capa 5 de adaptación ATM)
DSM-CC	Instrucción y control de medios de almacenamiento digital ( <i>digital storage media/command and control</i> )
DTMF	Multifrecuencia bitono ( <i>dual tone multi-frequency</i> )
GOB	Grupo de bloques ( <i>group of blocks</i> ) (de una imagen vídeo: véanse las Recs. UIT-T H.261 y H.263)
HDLC	Control de enlace de datos de alto nivel ( <i>high-level data link control</i> )
HRD	Decodificador hipotético de referencia ( <i>hypothetical reference decoder</i> ) (véanse las Recs. UIT-T H.261 y H.263)

IV	Vector de inicialización ( <i>initialization vector</i> ) (utilizado para criptación: véanse las Recs. UIT-T H.233 y H.234)
LAPM	Protocolo de acceso al enlace para modems ( <i>link access protocol for modems</i> )
LCSE	Entidad de señalización de canal lógico ( <i>logical channel signalling entity</i> )
MC	Entidad de control multipunto H.323 ( <i>H.323 multipoint control entity</i> )
MCU	Unidad de control multipunto ( <i>multipoint control unit</i> )
MLSE	Entidad de señalización del bucle de mantenimiento ( <i>maintenance loop signalling entity</i> )
MPI	Intervalo de imagen mínimo ( <i>minimum picture interval</i> )
MRSE	Entidad de señalización de petición de modo ( <i>mode request signalling entity</i> )
MSDSE	Entidad de señalización de determinación principal-subordinado ( <i>master-slave determination signalling entity</i> )
MTSE	Entidad de señalización de tabla múltiplex ( <i>multiplex table signalling entity</i> )
PCR	Referencia de reloj de programa ( <i>program clock reference</i> ) (véase la Rec. UIT-T H.222.0   ISO/CEI 13818-1)
PID	Identificador de paquete ( <i>packet identifier</i> ) (véase la Rec. UIT-T H.222.0   ISO/CEI 13818-1)
QCIF	Cuarto de CIF ( <i>quarter CIF</i> )
RMESE	Entidad de señalización de petición de entrada múltiplex ( <i>request multiplex entry signalling entity</i> )
RTCP	Protocolo de control de transporte en tiempo real ( <i>real-time transport control protocol</i> )
RTDSE	Entidad de señalización de retardo de ida y vuelta ( <i>round-trip delay signalling entity</i> )
RTGC	Red telefónica general conmutada
RTP	Protocolo de transporte en tiempo real ( <i>real-time transport protocol</i> )
SDL	Lenguaje de especificación y descripción ( <i>specification and description language</i> )
SDU	Unidad de datos de servicio ( <i>service data unit</i> )
SE	Mensaje de intercambio de sesión ( <i>session exchange message</i> ) (utilizado para criptación: véanse las Recs. UIT-T H.233 y H.234)
SQCIF	Sub QCIF ( <i>sub QCIF</i> )
STD	Decodificador-objetivo de sistema ( <i>system target decoder</i> ) (véase la Rec. UIT-T H.222.0   ISO/CEI 13818-1)
VC	Canal virtual de ATM ( <i>ATM virtual channel</i> )

## 5 Generalidades

Esta Recomendación proporciona diversos servicios diferentes, alguno de los cuales se espera que sean aplicables a todos los terminales que los utilizan, siendo otros más específicos de terminales concretos. Se definen procedimientos para permitir el intercambio de capacidad audiovisuales y de datos; para solicitar la transmisión de un modo audiovisual y de datos determinado; para gestionar los canales lógicos utilizados para transportar la información audiovisual y de datos; para establecer qué terminal es el terminal principal y cuál el subordinado con fines de gestión de los canales lógicos bidireccionales; para transportar distintas señales de control e indicación; para controlar la velocidad de bits de los canales lógicos individuales y de la totalidad del múltiplex; y para medir el



retardo de ida y vuelta entre un par de terminales. Estos procedimientos se explican con más detalle en lo que sigue.

Continuando con esta introducción general, hay cláusulas donde se detalla la sintaxis y la semántica de los mensajes, así como los procedimientos correspondientes. Se ha definido la sintaxis empleando la notación ASN.1 [40] y la semántica define el significado de los elementos sintácticos y proporciona asimismo las limitaciones de sintaxis no especificadas en ASN.1. La cláusula de procedimientos define los protocolos que utilizan los mensajes definidos en las demás cláusulas.

Aunque no todos los mensajes y procedimientos definidos en esta Recomendación se aplicarán a la totalidad de los terminales, no se indican aquí tales restricciones, las cuales competen a las recomendaciones que hagan uso de esta Recomendación.

Se ha definido esta Recomendación de forma que sea independiente del mecanismo de transporte subyacente, aunque se ha previsto su empleo con una capa de transporte fiable, es decir, aquella que proporcione una entrega garantizada de datos correctos.

### **5.1 Determinación principal-subordinado**

Se presentan conflictos cuando dos terminales que intervienen en una llamada inician simultáneamente eventos similares y sólo uno de esos eventos es posible o deseado, por ejemplo, cuando los recursos están disponibles solamente para una aparición del evento. Para resolver esas situaciones, un terminal actuará como terminal principal y el otro actuará como terminal subordinado. Las reglas especificarán el comportamiento del terminal principal y del terminal subordinado en caso de conflicto.

El procedimiento de determinación principal-subordinado permite a los terminales en una llamada determinar cuál es el principal y cuál es el subordinado. La categoría del terminal se puede determinar de nuevo en cualquier momento durante una llamada; sin embargo, un terminal sólo puede iniciar el proceso de determinación principal-subordinado si no está activo localmente ningún procedimiento que dependa de su resultado.

### **5.2 Intercambio de capacidades**

Los procedimientos de intercambio de capacidades tienen por finalidad asegurar que únicamente las señales multimedia que deben transmitirse son aquellas que el terminal de recepción puede recibir y manejar adecuadamente. Esto exige que las capacidades de cada terminal para recibir y decodificar sean conocidas por el otro terminal. No es necesario que un terminal comprenda o almacene todas las capacidades entrantes. Podrán ignorarse capacidades no comprendidas o no utilizadas, sin que esto implique la consideración de que existen errores cuando se reciba una capacidad que contenga extensiones no comprendidas por el terminal, la capacidad será aceptada como si se contuviera las extensiones.

Mediante la transmisión de su juego de capacidades, se pone en conocimiento de un terminal la capacidad total de otro terminal para recibir y decodificar diversas señales.

Las capacidades de recepción describen la aptitud del terminal para recibir y procesar los trenes de información entrantes. Los transmisores deberán limitar el contenido de la información transmitida al valor que el receptor haya indicado que es capaz de recibir. La ausencia de una capacidad de recepción indica que el terminal es incapaz de recibir información (se trata de un transmisor únicamente).

Las capacidades de transmisión describen la aptitud del terminal para la transmisión de trenes de información. Estas capacidades permiten ofrecer a los receptores la posibilidad de elección entre distintos modos de funcionamiento, de forma que el receptor pueda solicitar el modo en el que prefiere efectuar la recepción. La ausencia de una capacidad de transmisión indica que el terminal no ofrecerá al receptor la elección de modos preferidos (pero puede sin embargo transmitir alguna información compatible con la capacidad del receptor).

Estos conjuntos de capacidades permiten la transmisión simultánea de más de un tren de un tipo de medio determinado. Por ejemplo, un terminal puede declarar su aptitud para recibir (o enviar) simultáneamente dos flujos de vídeo H.262 independientes y dos trenes de audio G.722 independientes. Se han definido los mensajes de capacidad para permitir que un terminal indique que no posee capacidades fijas, sino que dependen de los demás modos que se estén utilizando simultáneamente. Por ejemplo, es posible indicar que puede decodificarse una señal vídeo de elevada resolución cuando se utiliza un algoritmo de audio más simple o que pueden decodificarse cualquiera de dos secuencias de vídeo de baja resolución o una sola de alta resolución. Es posible también expresar compromisos entre la capacidad de transmisión y la capacidad de recepción.

Pueden emitirse capacidades y mensajes de control no normalizados utilizando la estructura NonStandardParameter. Debe observarse que aunque el significado de los mensajes no normalizados lo definen organizaciones individuales, los equipos construidos por cualquier fabricante pueden señalar cualquier mensaje no normalizado si conocen su significado.

Los terminales pueden volver a enviar conjuntos de capacidades en cualquier momento.

### **5.3 Procedimientos de señalización de canal lógico**

Se define un protocolo de acuse de recibo para la apertura y el cierre de canales lógicos que transportan información audiovisual y de datos. La finalidad de estos procedimientos es garantizar que un terminal es capaz de recibir y decodificar los datos que se transmitirán por un canal lógico en el momento en que se abra tal canal, en vez de en el momento en que se transmita el primer dato por él y para asegurar que el terminal de recepción está preparado para recibir y decodificar los datos que se transmitirán por el canal lógico antes del comienzo de la transmisión. El mensaje de apertura del canal lógico incluye una descripción de los datos que se transportarán, por ejemplo H.262 MP@ML a 6 Mbit/s. Los canales lógicos únicamente se abrirán cuando exista capacidad suficiente para recibir datos sobre todos los canales lógicos abiertos simultáneamente.

Una parte de este protocolo se aplica a la apertura de canales bidireccionales. Para evitar problemas de temporización, se define un terminal como terminal principal y el otro como terminal subordinado. Únicamente el terminal principal puede iniciar la apertura de canales bidireccionales: sin embargo, el terminal subordinado puede solicitar al terminal principal que realice esa apertura. Se ha definido un protocolo para establecer qué terminal será el principal y cuál el subordinado. Sin embargo, los sistemas que hagan uso de esta Recomendación deberán especificar otros modos de determinar qué terminal es el principal y cuál el subordinado.

### **5.4 Petición de cierre de canal lógico por el terminal receptor**

Un canal lógico se abre y cierra desde el lado transmisor. Se define un mecanismo que permite a un terminal receptor solicitar el cierre de un canal lógico entrante. El terminal transmisor puede aceptar o rechazar la petición de cierre del canal lógico. Un terminal puede, por ejemplo, utilizar estos procedimientos para solicitar el cierre de un canal lógico entrante que, por una razón cualquiera, no puede decodificarse. Estos procedimientos pueden utilizarse también para solicitar el cierre de un canal lógico bidireccional, por el terminal que no abrió el canal.

### **5.5 Modificación de entrada en la tabla múltiplex H.223**

La tabla múltiplex H.223 asocia cada octeto dentro de un mensaje MUX H.223 con un determinado número de canal lógico. La tabla múltiplex H.223 puede tener hasta 15 entradas. Se proporciona un mecanismo que permite al terminal transmisor especificar e informar al receptor sobre las nuevas entradas en la tabla múltiplex H.223. Un terminal receptor puede también solicitar la retransmisión de una entrada en la tabla múltiplex.

## **5.6 Petición de modo audiovisual y de modo datos**

Cuando el protocolo de intercambio de capacidad ha concluido, ambos terminales tendrán conocimiento de la capacidad del otro para transmitir y recibir como se especifica en los descriptores de capacidad que han sido intercambiados. Un terminal no está obligado a declarar todas sus capacidades; sólo necesita declarar aquellas que desea utilizar.

Un terminal puede indicar sus capacidades para transmitir. Un terminal que recibe capacidades de transmisión del terminal distante puede pedir que se le transmitan en un determinado modo. Un terminal indica que no desea que su modo de transmisión sea controlado por el terminal distante no enviando capacidades de transmisión.

## **5.7 Determinación del retardo de ida y vuelta**

En algunas aplicaciones puede ser conveniente tener conocimiento del retardo de ida y vuelta entre un terminal transmisor y un terminal receptor. Se proporciona un mecanismo para medir este tiempo de ida y vuelta. Este mecanismo puede ser también útil como un medio para detectar si el terminal distante está funcionando todavía.

## **5.8 Bucles de mantenimiento**

Se especifican procedimientos para establecer bucles de mantenimiento. Es posible especificar el bucle de un canal lógico individual como un bucle digital o como un bucle decodificado y el bucle del múltiplex completo.

## **5.9 Instrucciones e indicaciones**

Se proporcionan instrucciones e indicaciones para diversas finalidades: señales de vídeo/audio activo/inactivo para informar al usuario; petición de actualización rápida para conmutación en la fuente, en aplicaciones multipunto, son algunos ejemplos. Ni las instrucciones ni las indicaciones provocan mensajes de respuesta del terminal distante. Las instrucciones fuerzan la ejecución de una acción en el terminal distante, mientras que las indicaciones se limitan a proporcionar información y no fuerzan a ejecutar ninguna acción.

Una instrucción, por definición, permite que la velocidad binaria de los canales lógicos y de la totalidad del múltiplex sean controladas desde el terminal distante. Esto tiene varias finalidades: interfuncionamiento con terminales que utilizan múltiplex en los cuales sólo hay un número finito de velocidades binarias disponibles; aplicaciones multipunto en que las velocidades de las diferentes fuentes deben ser adaptadas; y control de flujo en redes congestionadas.

## Anexo A

### Mensajes: Sintaxis

En este anexo se especifica la sintaxis de mensajes que utilizan la notación definida en ASN.1 [40]. Los mensajes se codificarán para transmisión aplicando las reglas de codificación compactada especificadas en [42] que emplea la variante alineada básica. El primer bit de cada octeto transmitido es el bit más significativo del octeto, como se especifica en la Rec. UIT-T X.691 | ISO/CEI 8825-2.

```
MULTIMEDIA-SYSTEM-CONTROL DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- Export all symbols

-- =====
-- Top level Messages
-- =====

MultimediaSystemControlMessage      ::=CHOICE
{
    request          RequestMessage,
    response         ResponseMessage,
    command          CommandMessage,
    indication       IndicationMessage,
    ...
}

-- A RequestMessage results in action and requires an immediate response

RequestMessage                      ::=CHOICE
{
    nonStandard      NonStandardMessage,

    masterSlaveDetermination  MasterSlaveDetermination,

    terminalCapabilitySet      TerminalCapabilitySet,

    openLogicalChannel         OpenLogicalChannel,
    closeLogicalChannel        CloseLogicalChannel,

    requestChannelClose       RequestChannelClose,

    multiplexEntrySend        MultiplexEntrySend,

    requestMultiplexEntry     RequestMultiplexEntry,

    requestMode               RequestMode,

    roundTripDelayRequest     RoundTripDelayRequest,

    maintenanceLoopRequest    MaintenanceLoopRequest,

    ...,
    communicationModeRequest  CommunicationModeRequest,
    conferenceRequest         ConferenceRequest,

    multilinkRequest          MultilinkRequest,
    logicalChannelRateRequest LogicalChannelRateRequest,
    genericRequest            GenericMessage
}

```

-- A *ResponseMessage* is the response to a *RequestMessage*

```
ResponseMessage ::= CHOICE
{
    nonStandard NonStandardMessage,

    masterSlaveDeterminationAck MasterSlaveDeterminationAck,
    masterSlaveDeterminationReject MasterSlaveDeterminationReject,

    terminalCapabilitySetAck TerminalCapabilitySetAck,
    terminalCapabilitySetReject TerminalCapabilitySetReject,

    openLogicalChannelAck OpenLogicalChannelAck,
    openLogicalChannelReject OpenLogicalChannelReject,
    closeLogicalChannelAck CloseLogicalChannelAck,

    requestChannelCloseAck RequestChannelCloseAck,
    requestChannelCloseReject RequestChannelCloseReject,

    multiplexEntrySendAck MultiplexEntrySendAck,
    multiplexEntrySendReject MultiplexEntrySendReject,

    requestMultiplexEntryAck RequestMultiplexEntryAck,
    requestMultiplexEntryReject RequestMultiplexEntryReject,

    requestModeAck RequestModeAck,
    requestModeReject RequestModeReject,

    roundTripDelayResponse RoundTripDelayResponse,

    maintenanceLoopAck MaintenanceLoopAck,
    maintenanceLoopReject MaintenanceLoopReject,

    ...,
    communicationModeResponse CommunicationModeResponse,

    conferenceResponse ConferenceResponse,

    multilinkResponse MultilinkResponse,

    logicalChannelRateAcknowledge LogicalChannelRateAcknowledge,
    logicalChannelRateReject LogicalChannelRateReject,
    genericResponse GenericMessage
}
```

-- A *CommandMessage* requires action, but no explicit response

```
CommandMessage ::= CHOICE
{
    nonStandard NonStandardMessage,

    maintenanceLoopOffCommand MaintenanceLoopOffCommand,

    sendTerminalCapabilitySet SendTerminalCapabilitySet,

    encryptionCommand EncryptionCommand,

    flowControlCommand FlowControlCommand,

    endSessionCommand EndSessionCommand,
```

```

miscellaneousCommand      MiscellaneousCommand,
...
communicationModeCommand  CommunicationModeCommand,
conferenceCommand         ConferenceCommand,
h223MultiplexReconfiguration  H223MultiplexReconfiguration,
newATMVCCCommand         NewATMVCCCommand,
mobileMultilinkReconfigurationCommand  MobileMultilinkReconfigurationCommand,
genericCommand           GenericMessage
}

-- An IndicationMessage is information that does not require action or response
IndicationMessage ::= CHOICE
{
    nonStandard          NonStandardMessage,
    functionNotUnderstood  FunctionNotUnderstood,
    masterSlaveDeterminationRelease  MasterSlaveDeterminationRelease,
    terminalCapabilitySetRelease  TerminalCapabilitySetRelease,
    openLogicalChannelConfirm  OpenLogicalChannelConfirm,
    requestChannelCloseRelease  RequestChannelCloseRelease,
    multiplexEntrySendRelease  MultiplexEntrySendRelease,
    requestMultiplexEntryRelease  RequestMultiplexEntryRelease,
    requestModeRelease      RequestModeRelease,
    miscellaneousIndication  MiscellaneousIndication,
    jitterIndication        JitterIndication,
    h223SkewIndication      H223SkewIndication,
    newATMVCIndication      NewATMVCIndication,
    userInput              UserInputIndication,
    ...
    h2250MaximumSkewIndication  H2250MaximumSkewIndication,
    mcLocationIndication    MCLocationIndication,
    conferenceIndication    ConferenceIndication,
    vendorIdentification    VendorIdentification,
    functionNotSupported    FunctionNotSupported,
    multilinkIndication     MultilinkIndication,
    logicalChannelRateRelease  LogicalChannelRateRelease,
    flowControlIndication   FlowControlIndication,

```

```

        mobileMultilinkReconfigurationIndication  MobileMultilinkReconfigurationIndication,
        genericIndication                        GenericMessage
    }

-- SequenceNumber is defined here as it is used in a number of Messages
SequenceNumber                                ::=INTEGER (0..255)

-- =====
-- Generic Message definitions
-- =====

GenericMessage                                ::=SEQUENCE
{
    messageIdentifier                          CapabilityIdentifier,
    subMessageIdentifier                       INTEGER(0..127) OPTIONAL,

    messageContent                             SEQUENCE OF GenericParameter OPTIONAL,
    ...
}

-- =====
-- Non-standard Message definitions
-- =====

NonStandardMessage                            ::=SEQUENCE
{
    nonStandardData                            NonStandardParameter,
    ...
}

NonStandardParameter                          ::=SEQUENCE
{
    nonStandardIdentifier                      NonStandardIdentifier,
    data                                       OCTET STRING
}

NonStandardIdentifier                          ::=CHOICE
{
    object                                     OBJECT IDENTIFIER,
    h221NonStandard                            SEQUENCE
    {
        t35CountryCode                        INTEGER (0..255),      -- country, per
                                                -- Annex A/T.35
        t35Extension                          INTEGER (0..255),
                                                -- assigned nationally unless
                                                -- t35CountryCode is binary
                                                -- 1111 1111, in which case it shall
                                                -- contain the country code
                                                -- according to Annex B/T.35
        manufacturerCode                      INTEGER (0..65535) -- assigned nationally
    }
}

-- =====
-- Master-slave determination definitions
-- =====

```

```

MasterSlaveDetermination                               ::=SEQUENCE
{
    terminalType                                     INTEGER (0..255),
    statusDeterminationNumber                       INTEGER (0..16777215),
    ...
}

MasterSlaveDeterminationAck                           ::=SEQUENCE
{
    decision                                         CHOICE
    {
        master                                       NULL,
        slave                                       NULL
    },
    ...
}

MasterSlaveDeterminationReject                       ::=SEQUENCE
{
    cause                                           CHOICE
    {
        identicalNumbers                           NULL,
        ...
    },
    ...
}

MasterSlaveDeterminationRelease                      ::=SEQUENCE
{
    ...
}

-- =====
-- Capability exchange definitions
-- =====

TerminalCapabilitySet                                ::=SEQUENCE
{
    sequenceNumber                                  SequenceNumber,

    protocolIdentifier                              OBJECT IDENTIFIER,
    -- shall be set to the value
    -- {itu-t (0) recommendation (0) h (8) 245
    -- version (0) 11}

    multiplexCapability                              MultiplexCapability OPTIONAL,

    capabilityTable                                 SET SIZE (1..256) OF CapabilityTableEntry OPTIONAL,

    capabilityDescriptors                           SET SIZE (1..256) OF CapabilityDescriptor OPTIONAL,

    ...
}

CapabilityTableEntry                                 ::=SEQUENCE
{
    capabilityTableEntryNumber                      CapabilityTableEntryNumber,
    capability                                       Capability OPTIONAL
}

CapabilityDescriptor                                 ::=SEQUENCE
{

```



```

        capabilityDescriptorNumber      CapabilityDescriptorNumber,
        simultaneousCapabilities         SET SIZE (1..256) OF AlternativeCapabilitySet OPTIONAL
    }

AlternativeCapabilitySet                ::=SEQUENCE SIZE (1..256) OF CapabilityTableEntryNumber

CapabilityTableEntryNumber             ::=INTEGER (1..65535)

CapabilityDescriptorNumber             ::=INTEGER (0..255)

TerminalCapabilitySetAck               ::=SEQUENCE
{
    sequenceNumber                     SequenceNumber,
    ...
}

TerminalCapabilitySetReject            ::=SEQUENCE
{
    sequenceNumber                     SequenceNumber,
    cause                              CHOICE
    {
        unspecified                    NULL,
        undefinedTableEntryUsed        NULL,
        descriptorCapacityExceeded     NULL,
        tableEntryCapacityExceeded     CHOICE
        {
            highestEntryNumberProcessed CapabilityTableEntryNumber,
            noneProcessed                NULL
        },
        ...
    },
    ...
}

TerminalCapabilitySetRelease           ::=SEQUENCE
{
    ...
}

-- =====
-- Capability exchange definitions: top level capability description
-- =====

Capability                             ::=CHOICE
{
    nonStandard                         NonStandardParameter,

    receiveVideoCapability              VideoCapability,
    transmitVideoCapability             VideoCapability,
    receiveAndTransmitVideoCapability   VideoCapability,

    receiveAudioCapability              AudioCapability,
    transmitAudioCapability             AudioCapability,
    receiveAndTransmitAudioCapability   AudioCapability,

    receiveDataApplicationCapability    DataApplicationCapability,
    transmitDataApplicationCapability    DataApplicationCapability,
    receiveAndTransmitDataApplicationCapability DataApplicationCapability,

    h233EncryptionTransmitCapability    BOOLEAN,
    h233EncryptionReceiveCapability     SEQUENCE
    {

```

```

        h233IVResponseTime          INTEGER (0..255),          -- units milliseconds
    },
    ...,
    conferenceCapability            ConferenceCapability,
    h235SecurityCapability          H235SecurityCapability,
    maxPendingReplacementFor       INTEGER (0..255),
    receiveUserInputCapability      UserInputCapability,
    transmitUserInputCapability     UserInputCapability,
    receiveAndTransmitUserInputCapability UserInputCapability,

    genericControlCapability        GenericCapability,
    receiveMultiplexedStreamCapability MultiplexedStreamCapability,
    transmitMultiplexedStreamCapability MultiplexedStreamCapability,
    receiveAndTransmitMultiplexedStreamCapability MultiplexedStreamCapability,
    receiveRTPAudioTelephonyEventCapability AudioTelephonyEventCapability,
    receiveRTPAudioToneCapability  AudioToneCapability,
    depFecCapability                DepFECCapability,
    -- Deprecated, do not use
    multiplePayloadStreamCapability MultiplePayloadStreamCapability,
    fecCapability                   FECCapability,
    redundancyEncodingCap           RedundancyEncodingCapability,
    oneOfCapabilities               AlternativeCapabilitySet
}

H235SecurityCapability            ::=SEQUENCE
{
    encryptionAuthenticationAndIntegrity EncryptionAuthenticationAndIntegrity,

    mediaCapability                CapabilityTableEntryNumber,
    -- NOTE - The mediaCapability shall refer to Capability Table Entries
    -- that do contain, directly or indirectly, one or more transmit,
    -- receive, or receiveAndTransmit AudioCapability, VideoCapability,
    -- DataApplicationCapability, or similar capabilities indicated by a
    -- NonStandardParameter or GenericCapability only

    ...
}

-- =====
-- Capability exchange definitions: Multiplex capabilities
-- =====

MultiplexCapability               ::=CHOICE
{
    nonStandard                    NonStandardParameter,
    h222Capability                  H222Capability,
    h223Capability                  H223Capability,
    v76Capability                   V76Capability,
    ...,
    h2250Capability                 H2250Capability,

    genericMultiplexCapability      GenericCapability
}

H222Capability                    ::= SEQUENCE
{
    numberOfVCs                     INTEGER (1..256),
    vcCapability                     SET OF VCCapability,
    ...
}

VCCapability                      ::=SEQUENCE
{

```

```

aall SEQUENCE
{
    nullClockRecovery          BOOLEAN,
    srtsClockRecovery          BOOLEAN,
    adaptiveClockRecovery      BOOLEAN,
    nullErrorCorrection         BOOLEAN,
    longInterleaver            BOOLEAN,
    shortInterleaver           BOOLEAN,
    errorCorrectionOnly         BOOLEAN,
    structuredDataTransfer      BOOLEAN,
    partiallyFilledCells       BOOLEAN,
    ...
} OPTIONAL,
aal5 SEQUENCE
{
    forwardMaximumSDUSize      INTEGER (0..65535), -- units octets
    backwardMaximumSDUSize     INTEGER (0..65535), -- units octets
    ...
} OPTIONAL,
transportStream               BOOLEAN,
programStream                 BOOLEAN,
availableBitRates             SEQUENCE
{
    type                       CHOICE
    {
        singleBitRate          INTEGER (1..65535), -- units 64 kbit/s
        rangeOfBitRates        SEQUENCE
        {
            lowerBitRate        INTEGER (1..65535), -- units 64 kbit/s
            higherBitRate       INTEGER (1..65535) -- units 64 kbit/s
        }
    },
    ...
},
...
aallViaGateway                SEQUENCE
{
    gatewayAddress             SET SIZE(1..256) OF Q2931Address,
    nullClockRecovery          BOOLEAN,
    srtsClockRecovery          BOOLEAN,
    adaptiveClockRecovery      BOOLEAN,
    nullErrorCorrection         BOOLEAN,
    longInterleaver            BOOLEAN,
    shortInterleaver           BOOLEAN,
    errorCorrectionOnly         BOOLEAN,
    structuredDataTransfer      BOOLEAN,
    partiallyFilledCells       BOOLEAN,
    ...
} OPTIONAL
}

H223Capability ::=SEQUENCE
{
    transportWithI-frames      BOOLEAN, -- I-frame transport
                                     -- of H.245

    videoWithAL1               BOOLEAN,
    videoWithAL2               BOOLEAN,
    videoWithAL3               BOOLEAN,
    audioWithAL1               BOOLEAN,
    audioWithAL2               BOOLEAN,
    audioWithAL3               BOOLEAN,
    dataWithAL1                BOOLEAN,

```

```

dataWithAL2                BOOLEAN,
dataWithAL3                BOOLEAN,

maximumAL2SDUSize          INTEGER (0..65535),  -- units octets
maximumAL3SDUSize          INTEGER (0..65535),  -- units octets

maximumDelayJitter         INTEGER (0..1023),   -- units milliseconds

h223MultiplexTableCapability CHOICE
{
  basic                     NULL,
  enhanced                   SEQUENCE
  {
    maximumNestingDepth     INTEGER (1..15),
    maximumElementListSize  INTEGER (2..255),
    maximumSubElementListSize INTEGER (2..255),
    ...
  }
},
...,
maxMUXPDUSizeCapability    BOOLEAN,
nsrpSupport                BOOLEAN,
mobileOperationTransmitCapability SEQUENCE
{
  modeChangeCapability      BOOLEAN,
  h223AnnexA                BOOLEAN,
  h223AnnexADoubleFlag      BOOLEAN,
  h223AnnexB                BOOLEAN,
  h223AnnexBwithHeader      BOOLEAN,
  ...
} OPTIONAL,
h223AnnexCCapability       H223AnnexCCapability OPTIONAL,

bitRate                    INTEGER (1..19200) OPTIONAL,  -- units of
                                                                -- 100 bit/s

mobileMultilinkFrameCapability SEQUENCE
{
  maximumSampleSize         INTEGER (1..255),   -- units octets
  maximumPayloadLength      INTEGER (1..65025), -- units octets
  ...
} OPTIONAL
}

H223AnnexCCapability ::= SEQUENCE
{
  videoWithAL1M             BOOLEAN,
  videoWithAL2M             BOOLEAN,
  videoWithAL3M             BOOLEAN,
  audioWithAL1M             BOOLEAN,
  audioWithAL2M             BOOLEAN,
  audioWithAL3M             BOOLEAN,
  dataWithAL1M              BOOLEAN,
  dataWithAL2M              BOOLEAN,
  dataWithAL3M              BOOLEAN,
  alpduInterleaving         BOOLEAN,

  maximumAL1MPDUSize        INTEGER (0..65535),  -- units octets
  maximumAL2MSDUSize        INTEGER (0..65535),  -- units octets
  maximumAL3MSDUSize        INTEGER (0..65535),  -- units octets
  ...,
  rsCodeCapability          BOOLEAN OPTIONAL
}

```

```

V76Capability ::=SEQUENCE
{
    suspendResumeCapabilitywAddress BOOLEAN,
    suspendResumeCapabilitywoAddress BOOLEAN,
    rejCapability BOOLEAN,
    sREJCapability BOOLEAN,
    mREJCapability BOOLEAN,
    crc8bitCapability BOOLEAN,
    crc16bitCapability BOOLEAN,
    crc32bitCapability BOOLEAN,
    uihCapability BOOLEAN,
    numOfDLCS INTEGER (2..8191),
    twoOctetAddressFieldCapability BOOLEAN,
    loopBackTestCapability BOOLEAN,
    n401Capability INTEGER (1..4095),
    maxWindowSizeCapability INTEGER (1..127),
    v75Capability V75Capability,
    ...
}

V75Capability ::=SEQUENCE
{
    audioHeader BOOLEAN,
    ...
}

H2250Capability ::=SEQUENCE
{
    maximumAudioDelayJitter INTEGER(0..1023), -- units in
                                                -- milliseconds

    receiveMultipointCapability MultipointCapability,
    transmitMultipointCapability MultipointCapability,
    receiveAndTransmitMultipointCapability MultipointCapability,
    mcCapability SEQUENCE
    {
        centralizedConferenceMC BOOLEAN,
        decentralizedConferenceMC BOOLEAN,
        ...
    },
    rtcpVideoControlCapability BOOLEAN, -- FIR and NACK
    mediaPacketizationCapability MediaPacketizationCapability,
    ...,
    transportCapability TransportCapability OPTIONAL,
    redundancyEncodingCapability SEQUENCE SIZE(1..256) OF RedundancyEncodingCapability OPTIONAL,
    logicalChannelSwitchingCapability BOOLEAN,
    tl20DynamicPortCapability BOOLEAN
}

MediaPacketizationCapability ::=SEQUENCE
{
    h261aVideoPacketization BOOLEAN,
    ...,
    rtpPayloadType SEQUENCE SIZE(1..256) OF RTPPayloadType OPTIONAL
}

RSVPPParameters ::=SEQUENCE
{
    qosMode QOSMode OPTIONAL,
    tokenRate INTEGER (1..4294967295) OPTIONAL,
        -- rate in bytes/s
    bucketSize INTEGER (1..4294967295) OPTIONAL,
        -- size in bytes
}

```

```

    peakRate                INTEGER (1..4294967295) OPTIONAL,
                           -- peak bandwidth bytes/s
    minPoliced              INTEGER (1..4294967295) OPTIONAL,
                           --
    maxPktSize              INTEGER (1..4294967295) OPTIONAL,
                           -- size in bytes
    ...
}

QOSMode                    ::=CHOICE
{
    guaranteedQOS          NULL,
    controlledLoad         NULL,
    ...
}

ATMParameters              ::=SEQUENCE
{
    maxNTUSize             INTEGER(0..65535), -- units in octets
    atmUBR                 BOOLEAN,         -- unspecified bit rate
    atmrtVBR               BOOLEAN,         -- real time variable
    atmnrVBR               BOOLEAN,         -- non real time
                                -- variable bit rate
    atmABR                 BOOLEAN,         -- available bit rate
    atmCBR                 BOOLEAN,         -- constant bit rate
    ...
}

QOSCapability              ::=SEQUENCE
{
    nonStandardData        NonStandardParameter OPTIONAL,
    rsvpParameters         RSVPParameters OPTIONAL,
    atmParameters          ATMParameters OPTIONAL,
    ...
}

MediaTransportType         ::=CHOICE
{
    ip-UDP                 NULL,
    ip-TCP                 NULL,
    atm-AAL5-UNIDIR        NULL, -- virtual circuits used as unidirectional
    atm-AAL5-BIDIR        NULL, -- virtual circuits used as bidirectional
    ...,
    atm-AAL5-compressed    SEQUENCE
    {
        variable-delta    BOOLEAN,
        ...
    }
}

MediaChannelCapability     ::=SEQUENCE
{
    mediaTransport         MediaTransportType OPTIONAL,
    ...
}

TransportCapability        ::=SEQUENCE
{
    nonStandard            NonStandardParameter OPTIONAL,
    qosCapabilities        SEQUENCE SIZE(1..256) OF QOSCapability OPTIONAL,
    mediaChannelCapabilities SEQUENCE SIZE(1..256) OF MediaChannelCapability OPTIONAL,
    ...
}

```

```

RedundancyEncodingCapability ::=SEQUENCE
{
    redundancyEncodingMethod RedundancyEncodingMethod,
    primaryEncoding           CapabilityTableEntryNumber,
    secondaryEncoding         SEQUENCE SIZE(1..256) OF CapabilityTableEntryNumber OPTIONAL,
    -- must be Audio, Video, or Data capabilities, not derived
    -- capabilities; redundancy order is inferred from number of
    -- secondary encodings
    ...
}

RedundancyEncodingMethod ::=CHOICE
{
    nonStandard NonStandardParameter,
    rtpAudioRedundancyEncoding NULL,
    ...,
    rtpH263VideoRedundancyEncoding RTPH263VideoRedundancyEncoding
}

RTPH263VideoRedundancyEncoding ::= SEQUENCE
{
    numberOfThreads INTEGER (1..16),
    framesBetweenSyncPoints INTEGER (1..256),
    frameToThreadMapping CHOICE
    {
        roundrobin NULL,
        custom SEQUENCE SIZE(1..256) OF
        RTPH263VideoRedundancyFrameMapping,
        -- empty SEQUENCE for capability negotiation
        -- meaningful contents only OpenLogicalChannel
        ...
    },
    containedThreads SEQUENCE SIZE(1..256) OF INTEGER (0..15) OPTIONAL,
    -- only used for opening of logical channels
    ...
}

RTPH263VideoRedundancyFrameMapping ::= SEQUENCE
{
    threadNumber INTEGER (0..15),
    frameSequence SEQUENCE SIZE(1..256) OF INTEGER (0..255),
    ...
}

MultipointCapability ::=SEQUENCE
{
    multicastCapability BOOLEAN,
    multiUniCastConference BOOLEAN,
    mediaDistributionCapability SEQUENCE OF MediaDistributionCapability,
    ...
}

MediaDistributionCapability ::=SEQUENCE
{
    centralizedControl BOOLEAN,
    distributedControl BOOLEAN, -- for further study in
    -- ITU-T Rec. H.323

    centralizedAudio BOOLEAN,
    distributedAudio BOOLEAN,
    centralizedVideo BOOLEAN,
    distributedVideo BOOLEAN,
    centralizedData SEQUENCE OF DataApplicationCapability OPTIONAL,
}

```

```

distributedData          SEQUENCE OF DataApplicationCapability OPTIONAL,
                          -- for further study in
                          -- ITU-T Rec. H.323
...
}

-- =====
-- Capability exchange definitions: Video capabilities
-- =====

VideoCapability          ::=CHOICE
{
    nonStandard          NonStandardParameter,
    h261VideoCapability  H261VideoCapability,
    h262VideoCapability  H262VideoCapability,
    h263VideoCapability  H263VideoCapability,
    is11172VideoCapability IS11172VideoCapability,
    ...,
    genericVideoCapability GenericCapability,
    extendedVideoCapability ExtendedVideoCapability
}

ExtendedVideoCapability ::= SEQUENCE
{
    videoCapability      SEQUENCE OF VideoCapability,
    videoCapabilityExtension SEQUENCE OF GenericCapability OPTIONAL,
    ...
}

H261VideoCapability     ::=SEQUENCE
{
    qcifMPI              INTEGER (1..4) OPTIONAL, -- units 1/29.97 Hz
    cifMPI               INTEGER (1..4) OPTIONAL, -- units 1/29.97 Hz
    temporalSpatialTradeOffCapability BOOLEAN,
    maxBitRate           INTEGER (1..19200),      -- units of
                                                -- 100 bit/s
    stillImageTransmission BOOLEAN,             -- Annex D/H.261
    ...,
    videoBadMBsCap       BOOLEAN
}

H262VideoCapability     ::=SEQUENCE
{
    profileAndLevel-SPatML    BOOLEAN,
    profileAndLevel-MPatLL    BOOLEAN,
    profileAndLevel-MPatML    BOOLEAN,
    profileAndLevel-MPatH-14  BOOLEAN,
    profileAndLevel-MPatHL    BOOLEAN,
    profileAndLevel-SNRatLL   BOOLEAN,
    profileAndLevel-SNRatML   BOOLEAN,
    profileAndLevel-SpatialatH-14 BOOLEAN,
    profileAndLevel-HPatML    BOOLEAN,
    profileAndLevel-HPatH-14  BOOLEAN,
    profileAndLevel-HPatHL    BOOLEAN,
    videoBitRate             INTEGER (0.. 1073741823) OPTIONAL, -- units 400 bit/s
    vbvBufferSize            INTEGER (0.. 262143) OPTIONAL,      -- units 16 384 bits
    samplesPerLine           INTEGER (0..16383) OPTIONAL,       -- units samples/line
    linesPerFrame            INTEGER (0..16383) OPTIONAL,       -- units lines/frame
    framesPerSecond          INTEGER (0..15) OPTIONAL,          -- frame_rate_code
}

```



```

luminanceSampleRate      INTEGER (0..4294967295) OPTIONAL, -- units samples/s
...
videoBadMBsCap           BOOLEAN
}

H263VideoCapability      ::=SEQUENCE
{
    sqcifMPI              INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    qcifMPI               INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cifMPI                INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cif4MPI              INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cif16MPI             INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    maxBitRate           INTEGER (1..192400),      -- units 100 bit/s
    unrestrictedVector   BOOLEAN,
    arithmeticCoding     BOOLEAN,
    advancedPrediction   BOOLEAN,
    pbFrames             BOOLEAN,
    temporalSpatialTradeOffCapability BOOLEAN,
    hrd-B                INTEGER (0..524287) OPTIONAL, -- units 128 bits
    bppMaxKb             INTEGER (0..65535) OPTIONAL, -- units 1024 bits
    ...

    slowSqcifMPI         INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowQcifMPI          INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCifMPI           INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCif4MPI          INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCif16MPI         INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    errorCompensation    BOOLEAN,

    enhancementLayerInfo EnhancementLayerInfo OPTIONAL,
    h263Options          H263Options OPTIONAL
}

EnhancementLayerInfo    ::=SEQUENCE
{
    baseBitRateConstrained BOOLEAN,
    snrEnhancement        SET SIZE(1..14) OF EnhancementOptions OPTIONAL,
    spatialEnhancement    SET SIZE(1..14) OF EnhancementOptions OPTIONAL,
    bPictureEnhancement   SET SIZE(1..14) OF BEnhancementParameters OPTIONAL,
    ...
}

BEnhancementParameters  ::=SEQUENCE
{
    enhancementOptions    EnhancementOptions,
    numberOfBPictures     INTEGER (1..64),
    ...
}

EnhancementOptions      ::=SEQUENCE
{
    sqcifMPI              INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    qcifMPI               INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cifMPI                INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cif4MPI              INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    cif16MPI             INTEGER (1..32) OPTIONAL, -- units 1/29.97 Hz
    maxBitRate           INTEGER (1..192400),      -- units 100 bit/s
    unrestrictedVector   BOOLEAN,
    arithmeticCoding     BOOLEAN,
    temporalSpatialTradeOffCapability BOOLEAN,
    slowSqcifMPI         INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowQcifMPI          INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCifMPI           INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCif4MPI          INTEGER (1..3600) OPTIONAL, -- units seconds/frame
    slowCif16MPI         INTEGER (1..3600) OPTIONAL, -- units seconds/frame
}

```

```

    errorCompensation          BOOLEAN,
    h263Options                H263Options OPTIONAL,
    ...
}

H263Options ::= SEQUENCE
{
    advancedIntraCodingMode    BOOLEAN,
    deblockingFilterMode      BOOLEAN,
    improvedPBframesMode      BOOLEAN,

    unlimitedMotionVectors    BOOLEAN,

    fullPictureFreeze         BOOLEAN,
    partialPictureFreezeAndRelease  BOOLEAN,
    resizingPartPicFreezeAndRelease  BOOLEAN,
    fullPictureSnapshot       BOOLEAN,
    partialPictureSnapshot    BOOLEAN,
    videoSegmentTagging       BOOLEAN,
    progressiveRefinement     BOOLEAN,

    dynamicPictureResizingByFour  BOOLEAN,
    dynamicPictureResizingSixteenthPel  BOOLEAN,
    dynamicWarpingHalfPel       BOOLEAN,
    dynamicWarpingSixteenthPel  BOOLEAN,

    independentSegmentDecoding  BOOLEAN,

    slicesInOrder-NonRect      BOOLEAN,
    slicesInOrder-Rect        BOOLEAN,
    slicesNoOrder-NonRect     BOOLEAN,
    slicesNoOrder-Rect       BOOLEAN,

    alternateInterVLCMode      BOOLEAN,
    modifiedQuantizationMode   BOOLEAN,
    reducedResolutionUpdate    BOOLEAN,

    transparencyParameters     TransparencyParameters OPTIONAL,
    separateVideoBackChannel   BOOLEAN,
    refPictureSelection        RefPictureSelection OPTIONAL,
    customPictureClockFrequency  SET SIZE (1..16) OF CustomPictureClockFrequency OPTIONAL,
    customPictureFormat        SET SIZE (1..16) OF CustomPictureFormat OPTIONAL,
    modeCombos                 SET SIZE (1..16) OF H263VideoModeCombos OPTIONAL,
    ...,
    videoBadMBsCap            BOOLEAN,
    h263Version3Options       H263Version3Options
}

TransparencyParameters ::= SEQUENCE
{
    presentationOrder          INTEGER(1..256),
    offset-x                   INTEGER(-262144..262143), -- 1/8 pixels
    offset-y                   INTEGER(-262144..262143), -- 1/8 pixels
    scale-x                    INTEGER(1..255),
    scale-y                    INTEGER(1..255),
    ...
}

RefPictureSelection ::=SEQUENCE
{
    additionalPictureMemory    SEQUENCE
    {
        sqcifAdditionalPictureMemory  INTEGER (1..256) OPTIONAL, -- units frame
        qcifAdditionalPictureMemory   INTEGER (1..256) OPTIONAL, -- units frame
    }
}

```

```

    cifAdditionalPictureMemory    INTEGER (1..256) OPTIONAL,      -- units frame
    cif4AdditionalPictureMemory    INTEGER (1..256) OPTIONAL,      -- units frame
    cif16AdditionalPictureMemory   INTEGER (1..256) OPTIONAL,      -- units frame
    bigCpfAdditionalPictureMemory  INTEGER (1..256) OPTIONAL,      -- units frame
    ...
} OPTIONAL,
videoMux                          BOOLEAN,
videoBackChannelSend              CHOICE
{
    none                          NULL,
    ackMessageOnly                NULL,
    nackMessageOnly               NULL,
    ackOrNackMessageOnly          NULL,
    ackAndNackMessage             NULL,
    ...
},
...,
enhancedReferencePicSelect        SEQUENCE
{
    subPictureRemovalParameters   SEQUENCE
    {
        mpuHorizMBs               INTEGER (1..128),
        mpuVertMBs                INTEGER (1..72),
        mpuTotalNumber             INTEGER (1..65536),
        ...
    } OPTIONAL,
    ...
}
}

CustomPictureClockFrequency       ::=SEQUENCE
{
    clockConversionCode            INTEGER(1000..1001),
    clockDivisor                   INTEGER(1..127),
    sqcifMPI                       INTEGER (1..2048) OPTIONAL,
    qcifMPI                        INTEGER (1..2048) OPTIONAL,
    cifMPI                          INTEGER (1..2048) OPTIONAL,
    cif4MPI                        INTEGER (1..2048) OPTIONAL,
    cif16MPI                       INTEGER (1..2048) OPTIONAL,
    ...
}

CustomPictureFormat               ::=SEQUENCE
{
    maxCustomPictureWidth          INTEGER(1..2048),      -- units 4 pixels
    maxCustomPictureHeight         INTEGER(1..2048),      -- units 4 pixels
    minCustomPictureWidth          INTEGER(1..2048),      -- units 4 pixels
    minCustomPictureHeight         INTEGER(1..2048),      -- units 4 pixels
    mPI                            SEQUENCE
    {
        standardMPI                INTEGER (1..31) OPTIONAL,
        customPCF                   SET SIZE (1..16) OF SEQUENCE
        {
            clockConversionCode      INTEGER (1000..1001),
            clockDivisor              INTEGER (1..127),
            customMPI                 INTEGER (1..2048),
            ...
        } OPTIONAL,
        ...
    },
    pixelAspectInformation         CHOICE
    {
        anyPixelAspectRatio         BOOLEAN,

```

```

        pixelAspectCode          SET SIZE (1..14) OF INTEGER(1..14),
        extendedPAR              SET SIZE (1..256) OF SEQUENCE
        {
            width                 INTEGER(1..255),
            height                INTEGER(1..255),
            ...
        },
        ...
    } ,
    ...
}

H263VideoModeCombos           ::= SEQUENCE
{
    h263VideoUncoupledModes    H263ModeComboFlags,
    h263VideoCoupledModes      SET SIZE (1..16) OF H263ModeComboFlags,
    ...
}

H263ModeComboFlags           ::= SEQUENCE
{
    unrestrictedVector         BOOLEAN,
    arithmeticCoding           BOOLEAN,
    advancedPrediction         BOOLEAN,
    pbFrames                   BOOLEAN,
    advancedIntraCodingMode    BOOLEAN,
    deblockingFilterMode      BOOLEAN,
    unlimitedMotionVectors    BOOLEAN,
    slicesInOrder-NonRect     BOOLEAN,
    slicesInOrder-Rect       BOOLEAN,
    slicesNoOrder-NonRect    BOOLEAN,
    slicesNoOrder-Rect      BOOLEAN,
    improvedPBFramesMode     BOOLEAN,
    referencePicSelect        BOOLEAN,
    dynamicPictureResizingByFour  BOOLEAN,
    dynamicPictureResizingSixteenthPel  BOOLEAN,
    dynamicWarpingHalfPel     BOOLEAN,
    dynamicWarpingSixteenthPel  BOOLEAN,
    reducedResolutionUpdate   BOOLEAN,
    independentSegmentDecoding  BOOLEAN,
    alternateInterVLCMode     BOOLEAN,
    modifiedQuantizationMode   BOOLEAN,
    ...,
    enhancedReferencePicSelect  BOOLEAN,
    h263Version3Options        H263Version3Options}

H263Version3Options          ::=SEQUENCE
{
    dataPartitionedSlices     BOOLEAN,
    fixedPointIDCT0          BOOLEAN,
    interlacedFields          BOOLEAN,
    currentPictureHeaderRepetition  BOOLEAN,
    previousPictureHeaderRepetition  BOOLEAN,
    nextPictureHeaderRepetition  BOOLEAN,
    pictureNumber             BOOLEAN,
    spareReferencePictures    BOOLEAN,
    ...
}

IS11172VideoCapability       ::=SEQUENCE
{
    constrainedBitstream      BOOLEAN,
    videoBitRate              INTEGER (0.. 1073741823) OPTIONAL, -- units 400 bit/s
    vbvBufferSize             INTEGER (0.. 262143) OPTIONAL, -- units 16 384 bits

```

```

    samplesPerLine          INTEGER (0..16383) OPTIONAL,    -- units samples/line
    linesPerFrame           INTEGER (0..16383) OPTIONAL,    -- units lines/frame
    pictureRate             INTEGER (0..15) OPTIONAL,
    luminanceSampleRate    INTEGER (0..4294967295) OPTIONAL, -- units samples/s
    . . . ,
    videoBadMBsCap         BOOLEAN
}

```

```

-- =====
-- Capability exchange definitions: Audio capabilities
-- =====

```

```

-- For an H.222 multiplex, the integers indicate the size of the STD buffer
-- in units of 256 octets
-- For an H.223 multiplex, the integers indicate the maximum number of audio
-- frames per AL-SDU
-- For an H.225.0 multiplex, the integers indicate the maximum number of audio
-- frames per packet

```

```

AudioCapability ::= CHOICE
{
    nonStandard NonStandardParameter,
    g711Alaw64k INTEGER (1..256),
    g711Alaw56k INTEGER (1..256),
    g711Ulaw64k INTEGER (1..256),
    g711Ulaw56k INTEGER (1..256),

    g722-64k INTEGER (1..256),
    g722-56k INTEGER (1..256),
    g722-48k INTEGER (1..256),

    g7231 SEQUENCE
    {
        maxAl-sduAudioFrames INTEGER (1..256),
        silenceSuppression BOOLEAN
    },

    g728 INTEGER (1..256),
    g729 INTEGER (1..256),
    g729AnnexA INTEGER (1..256),
    is11172AudioCapability IS11172AudioCapability,
    is13818AudioCapability IS13818AudioCapability,
    . . . ,
    g729wAnnexB INTEGER (1..256),
    g729AnnexAwAnnexB INTEGER (1..256),
    g7231AnnexCCapability G7231AnnexCCapability,
    gsmFullRate GSMAudioCapability,
    gsmHalfRate GSMAudioCapability,
    gsmEnhancedFullRate GSMAudioCapability,
    genericAudioCapability GenericCapability,
    g729Extensions G729Extensions,
    vbd VBDCapability,
    audioTelephonyEvent NoPTAudioTelephonyEventCapability,
    audioTone NoPTAudioToneCapability
}

```

```

G729Extensions ::= SEQUENCE
{
    audioUnit INTEGER (1..256) OPTIONAL,
    annexA BOOLEAN,
    annexB BOOLEAN,
    annexD BOOLEAN,
    annexE BOOLEAN,
    annexF BOOLEAN,
}

```

```

annexG          BOOLEAN,
annexH          BOOLEAN,
...
}

G7231AnnexCCapability ::= SEQUENCE
{
    maxAl-sduAudioFrames    INTEGER (1..256),
    silenceSuppression      BOOLEAN,
    g723AnnexCAudioMode    SEQUENCE
    {
        highRateMode0      INTEGER (27..78),    -- units octets
        highRateMode1      INTEGER (27..78),    -- units octets
        lowRateMode0       INTEGER (23..66),    -- units octets
        lowRateMode1       INTEGER (23..66),    -- units octets
        sidMode0           INTEGER (6..17),     -- units octets
        sidMode1           INTEGER (6..17),     -- units octets
        ...
    } OPTIONAL,
    ...
}

IS11172AudioCapability ::=SEQUENCE
{
    audioLayer1            BOOLEAN,
    audioLayer2            BOOLEAN,
    audioLayer3            BOOLEAN,

    audioSampling32k       BOOLEAN,
    audioSampling44k1      BOOLEAN,
    audioSampling48k       BOOLEAN,

    singleChannel          BOOLEAN,
    twoChannels            BOOLEAN,

    bitRate                INTEGER (1..448),    -- units kbit/s
    ...
}

IS13818AudioCapability ::=SEQUENCE
{
    audioLayer1            BOOLEAN,
    audioLayer2            BOOLEAN,
    audioLayer3            BOOLEAN,

    audioSampling16k       BOOLEAN,
    audioSampling22k05     BOOLEAN,
    audioSampling24k       BOOLEAN,
    audioSampling32k       BOOLEAN,
    audioSampling44k1      BOOLEAN,
    audioSampling48k       BOOLEAN,

    singleChannel          BOOLEAN,
    twoChannels            BOOLEAN,
    threeChannels2-1       BOOLEAN,
    threeChannels3-0       BOOLEAN,
    fourChannels2-0-2-0    BOOLEAN,
    fourChannels2-2        BOOLEAN,
    fourChannels3-1        BOOLEAN,
    fiveChannels3-0-2-0    BOOLEAN,
    fiveChannels3-2        BOOLEAN,

    lowFrequencyEnhancement    BOOLEAN,
}

```

```

        multilingual                BOOLEAN,

        bitRate                     INTEGER (1..1130),    -- units kbit/s
        ...
    }

GSMAudioCapability                ::= SEQUENCE
{
    audioUnitSize                   INTEGER (1..256),
    comfortNoise                     BOOLEAN,
    scrambled                         BOOLEAN,
    ...
}

VBDCapability                     ::= SEQUENCE
{
    type                             AudioCapability,    -- shall not be "vbd"
    ...
}

-- =====
-- Capability exchange definitions: Data capabilities
-- =====

DataApplicationCapability          ::= SEQUENCE
{
    application                      CHOICE
    {
        nonStandard                  NonStandardParameter,
        t120                          DataProtocolCapability,
        dsm-cc                        DataProtocolCapability,
        userData                       DataProtocolCapability,
        t84SEQUENCE
        {
            t84Protocol                DataProtocolCapability,
            t84Profile                  T84Profile
        },
        t434                          DataProtocolCapability,
        h224                            DataProtocolCapability,
        nlpid                           SEQUENCE
        {
            nlpidProtocol               DataProtocolCapability,
            nlpidData                   OCTET STRING
        },
        dsvdControl                    NULL,
        h222DataPartitioning           DataProtocolCapability,
        ...,
        t30fax                          DataProtocolCapability,
        t140                             DataProtocolCapability,
        t38fax                           SEQUENCE
        {
            t38FaxProtocol               DataProtocolCapability,
            t38FaxProfile                 T38FaxProfile
        },
        genericDataCapability           GenericCapability
    },
    maxBitRate                         INTEGER (0..4294967295), -- units 100 bit/s
    ...
}

DataProtocolCapability             ::= CHOICE
{
    nonStandard                       NonStandardParameter,
    v14buffered                        NULL,

```

```

v42lapm                NULL,                -- may negotiate to V.42 bis
hdlcFrameTunnelling    NULL,
h310SeparateVCStack    NULL,
h310SingleVCStack      NULL,
transparent             NULL,
...,
segmentationAndReassembly  NULL,
hdlcFrameTunnelingwSAR    NULL,
v120                   NULL,                -- as in H.230
separateLANStack        NULL,
v76wCompression        CHOICE
{
    transmitCompression    CompressionType,
    receiveCompression      CompressionType,
    transmitAndReceiveCompression  CompressionType,
    ...
},
tcp                     NULL,
udp                     NULL
}

CompressionType        ::=CHOICE
{
    v42bis               V42bis,
    ...
}

V42bis                 ::=SEQUENCE
{
    numberOfCodewords     INTEGER (1..65536),
    maximumStringLength   INTEGER (1..256),
    ...
}

T84Profile              ::=CHOICE
{
    t84Unrestricted      NULL,
    t84Restricted         SEQUENCE
    {
        qcif              BOOLEAN,
        cif                BOOLEAN,
        ccir601Seq        BOOLEAN,
        ccir601Prog       BOOLEAN,
        hdtvSeq           BOOLEAN,
        hdtvProg          BOOLEAN,

        g3FacsMH200x100   BOOLEAN,
        g3FacsMH200x200   BOOLEAN,
        g4FacsMMR200x100   BOOLEAN,
        g4FacsMMR200x200   BOOLEAN,
        jbig200x200Seq     BOOLEAN,
        jbig200x200Prog    BOOLEAN,
        jbig300x300Seq     BOOLEAN,
        jbig300x300Prog    BOOLEAN,

        digPhotoLow       BOOLEAN,
        digPhotoMedSeq     BOOLEAN,
        digPhotoMedProg    BOOLEAN,
        digPhotoHighSeq    BOOLEAN,
        digPhotoHighProg   BOOLEAN,
        ...
    }
}

```



```

T38FaxProfile ::=SEQUENCE
{
    fillBitRemoval          BOOLEAN,
    transcodingJBIG         BOOLEAN,
    transcodingMMR          BOOLEAN,
    ...,
    version                 INTEGER (0..255),
                           -- Version 0, the default, refers to
                           -- T.38 (1998)

    t38FaxRateManagement   T38FaxRateManagement,
                           -- The default Data Rate Management is
                           -- determined by the choice of
                           -- DataProtocolCapability

    t38FaxUdpOptions        T38FaxUdpOptions OPTIONAL,
                           -- For UDP, t38UDPRedundancy is the default

    t38FaxTcpOptions        T38FaxTcpOptions OPTIONAL
}

T38FaxRateManagement ::= CHOICE
{
    localTCF                NULL,
    transferredTCF          NULL,
    ...
}

T38FaxUdpOptions ::= SEQUENCE
{
    t38FaxMaxBuffer         INTEGER OPTIONAL,
    t38FaxMaxDatagram       INTEGER OPTIONAL,
    t38FaxUdpEC             CHOICE
    {
        t38UDPFEC           NULL,
        t38UDPRedundancy   NULL,
        ...
    }
}

T38FaxTcpOptions ::= SEQUENCE
{
    t38TCPBidirectionalMode BOOLEAN,
    ...
}

-- =====
-- Encryption Capability Definitions
-- =====

EncryptionAuthenticationAndIntegrity ::=SEQUENCE
{
    encryptionCapability    EncryptionCapability OPTIONAL,
    authenticationCapability AuthenticationCapability OPTIONAL,
    integrityCapability      IntegrityCapability OPTIONAL,
    ...,
    genericH235SecurityCapability GenericCapability OPTIONAL
}

EncryptionCapability ::=SEQUENCE SIZE(1..256) OF MediaEncryptionAlgorithm

MediaEncryptionAlgorithm ::=CHOICE
{

```

```

        nonStandard          NonStandardParameter,
        algorithm            OBJECT IDENTIFIER, -- many defined
                                -- in ISO/IEC 9979
        ...
    }

AuthenticationCapability ::=SEQUENCE
{
    nonStandard          NonStandardParameter OPTIONAL,
    ...,
    antiSpamAlgorithm   OBJECT IDENTIFIER OPTIONAL
}

IntegrityCapability ::=SEQUENCE
{
    nonStandard          NonStandardParameter OPTIONAL,
    ...
}

-----
-- Capability Exchange Definitions: UserInput
-----

UserInputCapability ::= CHOICE
{
    nonStandard          SEQUENCE SIZE(1..16) OF NonStandardParameter,
    basicString          NULL, -- alphanumeric
    ia5String            NULL, -- alphanumeric
    generalString        NULL, -- alphanumeric
    dtmf                 NULL, -- supports dtmf using signal
                        -- and signalUpdate
    hookflash            NULL, -- supports hookflash using signal
    ...,
    extendedAlphanumeric NULL,
    encryptedBasicString NULL, -- encrypted Basic string in
                        -- encryptedAlphanumeric
    encryptedIA5String  NULL, -- encrypted IA5 string in
                        -- encryptedSignalType
    encryptedGeneralString NULL, -- encrypted general string in
                        -- extendedAlphanumeric.encryptedalphanumeric
    securedTMF           NULL -- secure DTMF using encryptedSignalType
}

-----
-- Capability Exchange Definitions: Conference
-----

ConferenceCapability ::=SEQUENCE
{
    nonStandardData      SEQUENCE OF NonStandardParameter OPTIONAL,
    chairControlCapability BOOLEAN,
    ...,
    videoIndicateMixingCapability BOOLEAN,
    multipointVisualizationCapability BOOLEAN OPTIONAL -- same as H.230 MVC
}

-----
-- Capability Exchange Definitions: Generic Capability
-----

GenericCapability ::=SEQUENCE
{
    capabilityIdentifier CapabilityIdentifier,

```

```

maxBitRate                INTEGER (0..4294967295) OPTIONAL,
                           -- Units 100 bit/s
collapsing                 SEQUENCE OF GenericParameter OPTIONAL,
nonCollapsing              SEQUENCE OF GenericParameter OPTIONAL,
nonCollapsingRaw           OCTET STRING OPTIONAL,
                           -- Typically contains ASN.1
                           -- PER encoded data describing capability
transport                  DataProtocolCapability OPTIONAL,
...
}

```

```

CapabilityIdentifier       ::=CHOICE
{
  standard                 OBJECT IDENTIFIER,
                           -- e.g., { itu-t (0) recommendation (0) h (8) 267
                           -- version (0) 2 subIdentifier (0)}
  h221NonStandard          NonStandardParameter,
  uuid                     OCTET STRING ( SIZE (16) ),
  domainBased              IA5String ( SIZE (1..64) ),
  ...
}

```

-- NOTE - The ranges of parameter values have been selected to ensure that the  
-- GenericParameter preamble, standard part of ParameterIdentifier and the  
-- encoding of that choice, and the preamble of ParameterValue to fit into  
-- 2 octets.

```

GenericParameter          ::=SEQUENCE
{
  parameterIdentifier      ParameterIdentifier,
  parameterValue           ParameterValue,
  supersedes              SEQUENCE OF ParameterIdentifier OPTIONAL,
  ...
}

```

```

ParameterIdentifier       ::=CHOICE
{
  standard                 INTEGER (0..127), -- Assigned by
                           -- Capability
                           -- specifications
  h221NonStandard          NonStandardParameter, -- N.B.
                           -- NonStandardIdentifier
                           -- is not sufficient in
                           -- this case
  uuid                     OCTET STRING ( SIZE (16) ), -- For non-
                           -- standard
  domainBased              IA5String ( SIZE (1..64) ),
  ...
}

```

```

ParameterValue            ::=CHOICE
{
  logical                  NULL, -- Only acceptable if
                           -- all entities
                           -- include this option
  booleanArray             INTEGER (0..255), -- array of 8 logical
                           -- types
  unsignedMin              INTEGER (0..65535), -- Look for min
                           -- common value
  unsignedMax              INTEGER (0..65535), -- Look for max
                           -- common value
  unsigned32Min            INTEGER (0..4294967295), -- Look for min
                           -- common value
  unsigned32Max            INTEGER (0..4294967295), -- Look for max
                           -- common value
}

```

```

    octetString                OCTET STRING,                -- non-collapsing
                                -- octet string

    genericParameter           SEQUENCE OF GenericParameter,
    ...
}

-- =====
-- Capability Exchange Definitions: Multiplexed Stream Capability
-- =====

MultiplexedStreamCapability    ::=SEQUENCE
{
    multiplexFormat             MultiplexFormat,
    controlOnMuxStream         BOOLEAN,
    capabilityOnMuxStream       SET SIZE (1..256) OF
AlternativeCapabilitySet OPTIONAL,
    ...
}

MultiplexFormat                ::= CHOICE
{
    nonStandard                 NonStandardParameter,
    h222Capability              H222Capability,
    h223Capability              H223Capability,
    ...
}

-- =====
-- Capability Exchange Definitions: AudioTelephonyEventCapability and AudioToneCapability
-- =====

AudioTelephonyEventCapability  ::=SEQUENCE
{
    dynamicRTPPayloadType       INTEGER(96..127),
    audioTelephoneEvent         GeneralString, -- As per <list of values>
                                -- in 3.9/RFC 2833
    ...
}

AudioToneCapability            ::=SEQUENCE
{
    dynamicRTPPayloadType       INTEGER(96..127),
    ...
}

-- The following definitions are as above but without a Payload Type field.

NoPTAudioTelephonyEventCapability ::=SEQUENCE
{
    audioTelephoneEvent         GeneralString, -- As per <list of values>
                                -- in 3.9/RFC 2833
    ...
}

NoPTAudioToneCapability        ::=SEQUENCE
{
    ...
}

```

```

-- =====
-- Capability Exchange Definitions: MultiplePayloadStreamCapability
-- =====

```

```

MultiplePayloadStreamCapability ::=SEQUENCE
{
    capabilities                SET SIZE(1..256) OF AlternativeCapabilitySet,
    ...
}

```

```

-- =====
-- Capability Exchange Definitions: FECCapability
-- =====

```

```

DepFECCapability                ::=CHOICE -- Deprecated, do not use
{
    rfc2733                      SEQUENCE
    {
        redundancyEncoding       BOOLEAN,
        separateStream           SEQUENCE
        {
            separatePort         BOOLEAN,
            samePort              BOOLEAN,
            ...
        },
        ...
    },
    ...
}

```

```

FECCapability                   ::= SEQUENCE
{
    protectedCapability          CapabilityTableEntryNumber,
    fecScheme                    OBJECT IDENTIFIER OPTIONAL,
    -- identifies encoding scheme
    rfc2733Format                CHOICE
    {
        rfc2733rfc2198           MaxRedundancy, -- RFC 2198
        rfc2733sameport         MaxRedundancy,
        -- separate packet, same port
        rfc2733diffport         MaxRedundancy
        -- separate packet and port
    } OPTIONAL,
    ...
}

```

```

MaxRedundancy ::= INTEGER (1..MAX)

```

```

-- =====
-- Logical channel signalling definitions
-- =====

```

```

-- "Forward" is used to refer to transmission in the direction from the terminal
-- making the original request for a logical channel to the other terminal, and
-- "reverse" is used to refer to the opposite direction of transmission, in the
-- case of a bidirectional channel request.

```

```

OpenLogicalChannel              ::=SEQUENCE
{
    forwardLogicalChannelNumber  LogicalChannelNumber,
    forwardLogicalChannelParameters SEQUENCE
    {

```

```

portNumber                INTEGER (0..65535) OPTIONAL,
dataType                  DataType,
multiplexParameters       CHOICE
{
  h222LogicalChannelParameters H222LogicalChannelParameters,
  h223LogicalChannelParameters H223LogicalChannelParameters,
  v76LogicalChannelParameters  V76LogicalChannelParameters,
  ...,
  h2250LogicalChannelParameters H2250LogicalChannelParameters,
  none                          NULL -- for use with Separate Stack when
                                -- multiplexParameters are not
                                -- required or appropriate
},
...,
forwardLogicalChannelDependency LogicalChannelNumber OPTIONAL,
                                -- also used to refer to the primary
                                -- logical channel when using video
                                -- redundancy coding
replacementFor             LogicalChannelNumber OPTIONAL
},
-- Used to specify the reverse channel for bidirectional open request

reverseLogicalChannelParameters SEQUENCE
{
  dataType                  DataType,
  multiplexParameters       CHOICE
  {
    -- H.222 parameters are never present in reverse direction
    h223LogicalChannelParameters H223LogicalChannelParameters,
    v76LogicalChannelParameters  V76LogicalChannelParameters,
    ...,
    h2250LogicalChannelParameters H2250LogicalChannelParameters
  } OPTIONAL, -- Not present for H.222
  ...,
  reverseLogicalChannelDependency LogicalChannelNumber OPTIONAL,
    -- also used to refer to the primary logical channel when using
    -- video redundancy coding
  replacementFor             LogicalChannelNumber OPTIONAL
} OPTIONAL, -- Not present for unidirectional channel request
...,
separateStack               NetworkAccessParameters OPTIONAL,
    -- for Open responder to establish the stack
encryptionSync              EncryptionSync OPTIONAL -- used only by Master
}

LogicalChannelNumber ::= INTEGER (1..65535)

NetworkAccessParameters ::= SEQUENCE
{
  distribution               CHOICE
  {
    unicast                  NULL,
    multicast                 NULL, -- for further study in T.120
    ...
  } OPTIONAL,

```

```

networkAddress                               CHOICE
{
    q2931Address                             Q2931Address,
    e164Address                              IA5String(SIZE(1..128)) (FROM ("0123456789#*,")),
    localAreaAddress                         TransportAddress,
    ...
},
associateConference                          BOOLEAN,
externalReference                           OCTET STRING(SIZE(1..255)) OPTIONAL,
...
t120SetupProcedure                          CHOICE
{
    originateCall                           NULL,
    waitForCall                             NULL,
    issueQuery                              NULL,
    ...
} OPTIONAL
}

Q2931Address                                 ::=SEQUENCE
{
    address                                   CHOICE
    {
        internationalNumber                 NumericString(SIZE(1..16)),
        nsapAddress                        OCTET STRING (SIZE(1..20)),
        ...
    },
    subaddress                              OCTET STRING (SIZE(1..20)) OPTIONAL,
    ...
}

V75Parameters                               ::= SEQUENCE
{
    audioHeaderPresent                     BOOLEAN,
    ...
}

DataType                                     ::=CHOICE
{
    nonStandard                             NonStandardParameter,
    nullData                                NULL,
    videoData                               VideoCapability,
    audioData                               AudioCapability,
    data                                    DataApplicationCapability,
    encryptionData                         EncryptionMode,
    ...
    h235Control                             NonStandardParameter,
    h235Media                               H235Media,
    multiplexedStream                       MultiplexedStreamParameter,
    redundancyEncoding                      RedundancyEncoding,
    multiplePayloadStream                  MultiplePayloadStream,
    depFec                                  DepFECData, -- Deprecated, do not use
    fec                                     FECData
}

H235Media                                   ::=SEQUENCE
{
    encryptionAuthenticationAndIntegrity    EncryptionAuthenticationAndIntegrity,

    mediaType                               CHOICE
    {
        nonStandard                         NonStandardParameter,

```

```

        videoData          VideoCapability,
        audioData          AudioCapability,
        data               DataApplicationCapability,
        ...,
        redundancyEncoding RedundancyEncoding,
        multiplePayloadStream MultiplePayloadStream,
        depFec             DepFECData, -- Deprecated, do not use
        fec               FECDData
    },
    ...
}

MultiplexedStreamParameter ::=SEQUENCE
{
    multiplexFormat      MultiplexFormat,
    controlOnMuxStream  BOOLEAN,
    ...
}

H222LogicalChannelParameters ::=SEQUENCE
{
    resourceID          INTEGER (0..65535),
    subChannelID       INTEGER (0..8191),
    pcr-pid            INTEGER (0..8191) OPTIONAL,
    programDescriptors OCTET STRING OPTIONAL,
    streamDescriptors  OCTET STRING OPTIONAL,
    ...
}

H223LogicalChannelParameters ::=SEQUENCE
{
    adaptationLayerType CHOICE
    {
        nonStandard      NonStandardParameter,
        allFramed        NULL,
        allNotFramed     NULL,
        al2WithoutSequenceNumbers NULL,
        al2WithSequenceNumbers  NULL,
        al3              SEQUENCE
        {
            controlFieldOctets  INTEGER (0..2),
            sendBufferSize      INTEGER (0..16777215) -- units octets
        },
        ...,
        al1M              H223AL1MParameters,
        al2M              H223AL2MParameters,
        al3M              H223AL3MParameters
    },
    segmentableFlag     BOOLEAN,
    ...
}

H223AL1MParameters ::=SEQUENCE
{
    transferMode        CHOICE
    {
        framed           NULL,
        unframed        NULL,
        ...
    },
    headerFEC          CHOICE
    {

```



```

        sebch16-7                NULL,
        golay24-12              NULL,
        ...
    },
    crcLength                    CHOICE
    {
        crc4bit                 NULL,
        crc12bit                NULL,
        crc20bit                NULL,
        crc28bit                NULL,
        ...,
        crc8bit                 NULL,
        crc16bit                NULL,
        crc32bit                NULL,
        crcNotUsed              NULL
    },

    rcpcCodeRate                INTEGER (8..32),

    arqType                     CHOICE
    {
        noArq                   NULL,
        typeIArq                 H223AnnexCArqParameters,
        typeIIArq                H223AnnexCArqParameters,
        ...
    },
    alpduInterleaving           BOOLEAN,
    alsduSplitting              BOOLEAN,
    ...,
    rsCodeCorrection            INTEGER (0..127) OPTIONAL
}

H223AL2MParameters            ::=SEQUENCE
{
    headerFEC                   CHOICE
    {
        sebch16-5               NULL,
        golay24-12              NULL,
        ...
    },
    alpduInterleaving           BOOLEAN,
    ...
}

H223AL3MParameters            ::=SEQUENCE
{
    headerFormat                 CHOICE
    {
        sebch16-7               NULL,
        golay24-12              NULL,
        ...
    },
    crcLength                    CHOICE
    {
        crc4bit                 NULL,
        crc12bit                NULL,
        crc20bit                NULL,
        crc28bit                NULL,
        ...,
        crc8bit                 NULL,
        crc16bit                NULL,
        crc32bit                NULL,
        crcNotUsed              NULL
    },
}

```

```

rcpcCodeRate                INTEGER (8..32),

arqType                      CHOICE
{
  noArq                      NULL,
  typeIArq                   H223AnnexCArqParameters,
  typeIIArq                  H223AnnexCArqParameters,
  ...
},

alpduInterleaving           BOOLEAN,
...,
rsCodeCorrection            INTEGER (0..127) OPTIONAL
}

H223AnnexCArqParameters     ::=SEQUENCE
{
  numberOfRetransmissions    CHOICE
  {
    finite                   INTEGER (0..16),
    infinite                 NULL,
    ...
  },
  sendBufferSize            INTEGER (0..16777215),  -- units octets
  ...
}

V76LogicalChannelParameters ::=SEQUENCE
{
  hdlcParameters            V76HDLParameters,
  suspendResume             CHOICE
  {
    noSuspendResume         NULL,
    suspendResumewAddress   NULL,
    suspendResumewoAddress  NULL,
    ...
  },
  uIH                      BOOLEAN,
  mode                      CHOICE
  {
    eRM                     SEQUENCE
    {
      windowSize            INTEGER (1..127) ,
      recovery              CHOICE
      {
        rej                 NULL,
        sREJ               NULL,
        mSREJ              NULL,
        ...
      },
      ...
    },
    uNERM                   NULL,
    ...
  },
  v75Parameters            V75Parameters,
  ...
}

```

```

V76HDLParameters ::=SEQUENCE
{
    crcLength          CRCLength,
    n401               INTEGER (1..4095),
    loopbackTestProcedure
    ...
}

CRCLength ::=CHOICE
{
    crc8bit           NULL,
    crc16bit          NULL,
    crc32bit          NULL,
    ...
}

H2250LogicalChannelParameters ::=SEQUENCE
{
    nonStandard        SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID          INTEGER (0..255),
    associatedSessionID
    mediaChannel        TransportAddress OPTIONAL,
    mediaGuaranteedDelivery
    mediaControlChannel TransportAddress OPTIONAL, -- reverse
    ...
    mediaControlGuaranteedDelivery
    silenceSuppression
    destination        TerminalLabel OPTIONAL,
    ...
    dynamicRTPPayloadType
    mediaPacketization
    {
        h261aVideoPacketization
        ...
        rtpPayloadType
    } OPTIONAL,
    ...
    transportCapability
    redundancyEncoding
    source              TerminalLabel OPTIONAL
}

RTPPayloadType ::= SEQUENCE
{
    payloadDescriptor  CHOICE
    {
        nonStandardIdentifier
        rfc-number      INTEGER (1..32768, ...),
        oid             OBJECT IDENTIFIER,
        ...
    },
    payloadType        INTEGER (0..127) OPTIONAL,
    ...
}

RedundancyEncoding ::=SEQUENCE
{
    redundancyEncodingMethod
    secondaryEncoding  DataType OPTIONAL, -- depends on method
    ...
}

-- The sequence below may be used in place of the above secondaryEncoding field

```

```

rtpRedundancyEncoding          SEQUENCE
{
    primary                    RedundancyEncodingElement OPTIONAL,
                                -- Present when redundancyEncoding
                                -- is selected as the dataType
                                -- in an OpenLogicalChannel or
                                -- as part of a MultiplePayloadStream

    secondary                  SEQUENCE OF RedundancyEncodingElement OPTIONAL,
    ...
} OPTIONAL
}

RedundancyEncodingElement      ::=SEQUENCE
{
    dataType                   DataType,
    payloadType                INTEGER(0..127) OPTIONAL,
    ...
}

MultiplePayloadStream          ::=SEQUENCE
{
    elements                   SEQUENCE OF MultiplePayloadStreamElement,
    ...
}

MultiplePayloadStreamElement   ::=SEQUENCE
{
    dataType                   DataType,
    payloadType                INTEGER(0..127) OPTIONAL,
    ...
}

DepFECData                    ::=CHOICE -- Deprecated, do not use
{
    rfc2733                    SEQUENCE
    {
        mode                   CHOICE
        {
            redundancyEncoding  NULL,
            separateStream      CHOICE
            {
                differentPort    SEQUENCE
                {
                    protectedSessionID  INTEGER(1..255),
                    protectedPayloadType  INTEGER(0..127) OPTIONAL,
                    ...
                },
                samePort         SEQUENCE
                {
                    protectedPayloadType  INTEGER(0..127),
                    ...
                },
                ...
            },
            ...
        },
        ...
    }
}

```

```

FECDData ::= CHOICE
{
    rfc2733 SEQUENCE
    {
        protectedPayloadType INTEGER(0..127),
        fecScheme OBJECT IDENTIFIER OPTIONAL,
        pktMode CHOICE
        {
            rfc2198coding NULL,
            rfc2733sameport SEQUENCE
            {
                ...
            },
            rfc2733diffport SEQUENCE
            {
                protectedChannel LogicalChannelNumber,
                ...
            },
            ...
        },
        ...
    },
    ...
}

TransportAddress ::=CHOICE
{
    unicastAddress UnicastAddress,
    multicastAddress MulticastAddress,
    ...
}

UnicastAddress ::=CHOICE
{
    ipAddress SEQUENCE
    {
        network OCTET STRING (SIZE(4)),
        tsapIdentifier INTEGER(0..65535),
        ...
    },
    iPXAddress SEQUENCE
    {
        node OCTET STRING (SIZE(6)),
        netnum OCTET STRING (SIZE(4)),
        tsapIdentifier OCTET STRING (SIZE(2)),
        ...
    },
    iP6Address SEQUENCE
    {
        network OCTET STRING (SIZE(16)),
        tsapIdentifier INTEGER(0..65535),
        ...
    },
    netBios OCTET STRING (SIZE(16)),
    ipSourceRouteAddress SEQUENCE
    {
        routing CHOICE
        {
            strict NULL,
            loose NULL
        },
        network OCTET STRING (SIZE(4)),
        tsapIdentifier INTEGER(0..65535),
    }
}

```

```

        route                SEQUENCE OF OCTET STRING (SIZE(4)),
        ...
    },
    ...,
    nsap                      OCTET STRING (SIZE(1..20)),
    nonStandardAddress       NonStandardParameter
}

MulticastAddress            ::=CHOICE
{
    ipAddress                SEQUENCE
    {
        network              OCTET STRING (SIZE(4)),
        tsapIdentifier       INTEGER(0..65535),
        ...
    },
    iP6Address               SEQUENCE
    {
        network              OCTET STRING (SIZE(16)),
        tsapIdentifier       INTEGER(0..65535),
        ...
    },
    ...,
    nsap                      OCTET STRING (SIZE(1..20)),
    nonStandardAddress       NonStandardParameter
}

EncryptionSync             ::=SEQUENCE
-- used to supply new key and synchronization point
{
    nonStandard               NonStandardParameter OPTIONAL,
    synchFlag                 INTEGER(0..255) , -- may need to be larger
                                -- for H.324, etc.
                                -- shall be the Dynamic
                                -- Payload# for H.323
    h235Key                   OCTET STRING (SIZE(1..65535)), -- H.235
                                -- encoded value
    escrowentry               SEQUENCE SIZE(1..256) OF EscrowData OPTIONAL,
    ...,
    genericParameter          GenericParameter OPTIONAL
}

EscrowData                  ::=SEQUENCE
{
    escrowID                  OBJECT IDENTIFIER,
    escrowValue               BIT STRING (SIZE(1..65535)),
    ...
}

OpenLogicalChannelAck      ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,

    reverseLogicalChannelParameters SEQUENCE
    {
        reverseLogicalChannelNumber LogicalChannelNumber,
        portNumber                 INTEGER (0..65535) OPTIONAL,
        multiplexParameters        CHOICE
        {
            h222LogicalChannelParameters H222LogicalChannelParameters,
            -- H.223 parameters are never present in reverse direction
            ...,
            h2250LogicalChannelParameters H2250LogicalChannelParameters
        }
    }
}

```

```

    } OPTIONAL,                -- not present for H.223
    ...,
    replacementFor             LogicalChannelNumber OPTIONAL
} OPTIONAL,                  -- not present for unidirectional channel
                             -- request
...,
separateStack                 NetworkAccessParameters OPTIONAL,
                             -- for Open requester to establish
                             -- the stack
forwardMultiplexAckParameters CHOICE
{
    -- H.222 parameters are never present in the Ack
    -- H.223 parameters are never present in the Ack
    -- V.76 parameters are never present in the Ack
    h2250LogicalChannelAckParameters H2250LogicalChannelAckParameters,
    ...
} OPTIONAL,
encryptionSync                EncryptionSync OPTIONAL -- used only by Master
}

OpenLogicalChannelReject      ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    cause                       CHOICE
    {
        unspecified             NULL,
        unsuitableReverseParameters NULL,
        dataTypeNotSupported    NULL,
        dataTypeNotAvailable    NULL,
        unknownDataType         NULL,
        dataTypeALCombinationNotSupported NULL,
        ...,
        multicastChannelNotAllowed NULL,
        insufficientBandwidth    NULL,
        separateStackEstablishmentFailed NULL,
        invalidSessionID        NULL,
        masterSlaveConflict      NULL,
        waitForCommunicationMode NULL,
        invalidDependentChannel  NULL,
        replacementForRejected   NULL,
        securityDenied           NULL
    },
    ...
}

OpenLogicalChannelConfirm     ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

H2250LogicalChannelAckParameters ::=SEQUENCE
{
    nonStandard                 SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID                   INTEGER(1..255) OPTIONAL,
    mediaChannel                 TransportAddress OPTIONAL,
    mediaControlChannel          TransportAddress OPTIONAL, -- forward RTCP
                                                -- channel
    dynamicRTTPayloadType       INTEGER(96..127) OPTIONAL, -- used only by
                                                -- the master
                                                -- or MC
    ...,
    flowControlToZero           BOOLEAN,

```

```

    portNumber                INTEGER (0..65535) OPTIONAL
}

CloseLogicalChannel          ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    source                      CHOICE
    {
        user                    NULL,
        lcse                    NULL
    },
    ...,
    reason                      CHOICE
    {
        unknown                NULL,
        reopen                 NULL,
        reservationFailure     NULL,
        ...
    }
}

CloseLogicalChannelAck      ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

RequestChannelClose        ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...,
    qosCapability             QOSCapability OPTIONAL,
    reason                   CHOICE
    {
        unknown                NULL,
        normal                 NULL,
        reopen                 NULL,
        reservationFailure     NULL,
        ...
    }
}

RequestChannelCloseAck     ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

RequestChannelCloseReject  ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    cause                    CHOICE
    {
        unspecified            NULL,
        ...
    },
    ...
}

RequestChannelCloseRelease ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

```



```

-- =====
-- H.223 multiplex table definitions
-- =====

```

```

MultiplexEntrySend                               ::=SEQUENCE
{
    sequenceNumber                               SequenceNumber,
    multiplexEntryDescriptors                   SET SIZE (1..15) OF MultiplexEntryDescriptor,
    ...
}

MultiplexEntryDescriptor                         ::=SEQUENCE
{
    multiplexTableEntryNumber                   MultiplexTableEntryNumber,
    elementList                                 SEQUENCE SIZE (1..256) OF MultiplexElement OPTIONAL
}

MultiplexElement                                ::=SEQUENCE
{
    type                                         CHOICE
    {
        logicalChannelNumber                   INTEGER(0..65535),
        subElementList                         SEQUENCE SIZE (2..255) OF MultiplexElement
    },
    repeatCount                                  CHOICE
    {
        finite                                 INTEGER (1..65535),    -- repeats of type
        untilClosingFlag                       NULL                -- used for last element
    }
}

MultiplexTableEntryNumber                       ::=INTEGER (1..15)

MultiplexEntrySendAck                           ::=SEQUENCE
{
    sequenceNumber                               SequenceNumber,
    multiplexTableEntryNumber                   SET SIZE (1..15) OF
    MultiplexTableEntryNumber,
    ...
}

MultiplexEntrySendReject                       ::=SEQUENCE
{
    sequenceNumber                               SequenceNumber,
    rejectionDescriptions                       SET SIZE (1..15) OF
    MultiplexEntryRejectionDescriptions,
    ...
}

MultiplexEntryRejectionDescriptions            ::=SEQUENCE
{
    multiplexTableEntryNumber                   MultiplexTableEntryNumber,
    cause                                       CHOICE
    {
        unspecifiedCause                       NULL,
        descriptorTooComplex                   NULL,
        ...
    },
    ...
}

```

```

MultiplexEntrySendRelease      ::=SEQUENCE
{
    multiplexTableEntryNumber  SET SIZE (1..15) OF
                                MultiplexTableEntryNumber,
    ...
}

RequestMultiplexEntry          ::=SEQUENCE
{
    entryNumbers               SET SIZE (1..15) OF
                                MultiplexTableEntryNumber,
    ...
}

RequestMultiplexEntryAck      ::=SEQUENCE
{
    entryNumbers               SET SIZE (1..15) OF
                                MultiplexTableEntryNumber,
    ...
}

RequestMultiplexEntryReject   ::=SEQUENCE
{
    entryNumbers               SET SIZE (1..15) OF
                                MultiplexTableEntryNumber,
    rejectionDescriptions     SET SIZE (1..15) OF
                                RequestMultiplexEntryRejectionDescriptions,
    ...
}

RequestMultiplexEntryRejectionDescriptions ::=SEQUENCE
{
    multiplexTableEntryNumber  MultiplexTableEntryNumber,
    cause                      CHOICE
    {
        unspecifiedCause      NULL,
        ...
    },
    ...
}

RequestMultiplexEntryRelease   ::=SEQUENCE
{
    entryNumbers               SET SIZE (1..15) OF
                                MultiplexTableEntryNumber,
    ...
}

-- =====
-- Request mode definitions
-- =====

-- RequestMode is a list, in order or preference, of modes that a terminal would
-- like to have transmitted to it.

RequestMode                    ::=SEQUENCE
{
    sequenceNumber             SequenceNumber,
    requestedModes             SEQUENCE SIZE (1..256) OF ModeDescription,
    ...
}

```

```

RequestModeAck                               ::=SEQUENCE
{
    sequenceNumber                            SequenceNumber,
    response                                   CHOICE
    {
        willTransmitMostPreferredMode        NULL,
        willTransmitLessPreferredMode        NULL,
        ...
    },
    ...
}

RequestModeReject                             ::=SEQUENCE
{
    sequenceNumber                            SequenceNumber,
    cause                                      CHOICE
    {
        modeUnavailable                       NULL,
        multipointConstraint                  NULL,
        requestDenied                         NULL,
        ...
    },
    ...
}

RequestModeRelease                           ::=SEQUENCE
{
    ...
}

-- =====
-- Request mode definitions: Mode description
-- =====

ModeDescription                               ::=SET SIZE (1..256) OF ModeElement

ModeElementType                              ::=CHOICE
{
    nonStandard                               NonStandardParameter,
    videoMode                                 VideoMode,
    audioMode                                 AudioMode,
    dataMode                                  DataMode,
    encryptionMode                           EncryptionMode,
    ...,
    h235Mode                                  H235Mode,
    multiplexedStreamMode                     MultiplexedStreamParameter,
    redundancyEncodingDTMode                  RedundancyEncodingDTMode,
    multiplePayloadStreamMode                 MultiplePayloadStreamMode,
    depFecMode                                DepFECMode, -- deprecated, do not use
    fecMode FECMode
}

ModeElement                                  ::= SEQUENCE
{
    type                                       ModeElementType,

    h223ModeParameters                         H223ModeParameters OPTIONAL,
    ...,
    v76ModeParameters                         V76ModeParameters OPTIONAL,
    h2250ModeParameters                       H2250ModeParameters OPTIONAL,
    genericModeParameters                     GenericCapability OPTIONAL,
    multiplexedStreamModeParameters           MultiplexedStreamModeParameters OPTIONAL,
    logicalChannelNumber                       LogicalChannelNumber OPTIONAL
}

```

```

H235Mode ::=SEQUENCE
{
    encryptionAuthenticationAndIntegrity EncryptionAuthenticationAndIntegrity,
    mediaMode CHOICE
    {
        nonStandard NonStandardParameter,
        videoMode VideoMode,
        audioMode AudioMode,
        dataMode DataMode,
        ...
    },
    ...
}

MultiplexedStreamModeParameters ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    ...
}

RedundancyEncodingDTMode ::=SEQUENCE
{
    redundancyEncodingMethod RedundancyEncodingMethod,
    primary RedundancyEncodingDTModeElement,
    secondary SEQUENCE OF RedundancyEncodingDTModeElement,
    ...
}

RedundancyEncodingDTModeElement ::=SEQUENCE
{
    type CHOICE
    {
        nonStandard NonStandardParameter,
        videoMode VideoMode,
        audioMode AudioMode,
        dataMode DataMode,
        encryptionMode EncryptionMode,
        h235Mode H235Mode,
        ...,
        fecMode FECMode
    },
    ...
}

MultiplePayloadStreamMode ::=SEQUENCE
{
    elements SEQUENCE OF MultiplePayloadStreamElementMode,
    ...
}

MultiplePayloadStreamElementMode ::=SEQUENCE
{
    type ModeElementType,
    ...
}

DepFECMode ::=CHOICE -- deprecated, do not use
{
    rfc2733Mode SEQUENCE
    {
        mode CHOICE
        {

```

```

        redundancyEncoding      NULL,
        separateStream          CHOICE
    {
        differentPort           SEQUENCE
        {
            protectedSessionID  INTEGER(1..255),
            protectedPayloadType INTEGER(0..127) OPTIONAL,
            ...
        },
        samePort                SEQUENCE
        {
            protectedType       ModeElementType,
            ...
        },
        ...
    },
    ...
},
...
}

FECMode ::= SEQUENCE
{
    protectedElement      ModeElementType,
    fecScheme              OBJECT IDENTIFIER OPTIONAL,
                        -- identifies encoding scheme
    rfc2733Format         CHOICE
    {
        rfc2733rfc2198    MaxRedundancy, -- RFC 2198 redundancy
        rfc2733sameport   MaxRedundancy,
                        -- separate packet, same port
        rfc2733diffport   MaxRedundancy
                        -- separate packet and port
    } OPTIONAL,
    ...
}

H223ModeParameters ::=SEQUENCE
{
    adaptationLayerType    CHOICE
    {
        nonStandard        NonStandardParameter,
        allFramed          NULL,
        allNotFramed       NULL,
        al2WithoutSequenceNumbers NULL,
        al2WithSequenceNumbers NULL,
        al3                SEQUENCE
        {
            controlFieldOctets  INTEGER(0..2),
            sendBufferSize      INTEGER(0..16777215) -- units octets
        },
        ...,
        al1M                H223AL1MParameters,
        al2M                H223AL2MParameters,
        al3M                H223AL3MParameters
    },

    segmentableFlag        BOOLEAN,
    ...
}

```

```

V76ModeParameters                               ::=CHOICE
{
    suspendResumewAddress                       NULL,
    suspendResumewoAddress                      NULL,
    ...
}

H2250ModeParameters                             ::=SEQUENCE
{
    redundancyEncodingMode                      RedundancyEncodingMode OPTIONAL,
    ...
}

RedundancyEncodingMode                         ::=SEQUENCE
{
    redundancyEncodingMethod                    RedundancyEncodingMethod,
    secondaryEncoding                           CHOICE
    {
        nonStandard                             NonStandardParameter,
        audioData                               AudioMode,
        ...
    } OPTIONAL,
    ...
}

-- =====
-- Request mode definitions: Video modes
-- =====

VideoMode                                       ::=CHOICE
{
    nonStandard                                 NonStandardParameter,
    h261VideoMode                              H261VideoMode,
    h262VideoMode                              H262VideoMode,
    h263VideoMode                              H263VideoMode,
    is11172VideoMode                           IS11172VideoMode,
    ...,
    genericVideoMode                           GenericCapability
}

H261VideoMode                                  ::=SEQUENCE
{
    resolution                                 CHOICE
    {
        qcif                                    NULL,
        cif                                     NULL
    },
    bitRate                                    INTEGER (1..19200), -- units 100 bit/s
    stillImageTransmission                     BOOLEAN,
    ...
}

H262VideoMode                                  ::=SEQUENCE
{
    profileAndLevel                            CHOICE
    {
        profileAndLevel-SPatML                  NULL,
        profileAndLevel-MPatLL                  NULL,
        profileAndLevel-MPatML                  NULL,
        profileAndLevel-MPatH-14                NULL,
        profileAndLevel-MPatHL                  NULL,
        profileAndLevel-SNRatLL                 NULL,
        profileAndLevel-SNRatML                 NULL,
        profileAndLevel-SpatialatH-14          NULL,

```

```

        profileAndLevel-HPatML      NULL,
        profileAndLevel-HPatH-14    NULL,
        profileAndLevel-HPatHL      NULL,
        ...
    },
    videoBitRate                    INTEGER(0..1073741823) OPTIONAL, -- units 400 bit/s
    vbvBufferSize                    INTEGER(0..262143)  OPTIONAL, -- units 16 384 bits
    samplesPerLine                   INTEGER(0..16383)  OPTIONAL, -- units samples/line
    linesPerFrame                   INTEGER(0..16383)  OPTIONAL, -- units lines/frame
    framesPerSecond                  INTEGER(0..15)    OPTIONAL, -- frame_rate_code
    luminanceSampleRate              INTEGER(0..4294967295) OPTIONAL, -- units samples/s
    ...
}

```

```

H263VideoMode                      ::=SEQUENCE
{
    resolution                       CHOICE
    {
        sqcif                        NULL,
        qcif                         NULL,
        cif                          NULL,
        cif4                         NULL,
        cif16                        NULL,
        ...,
        custom                       NULL
    },
    bitRate                          INTEGER (1..19200), -- units 100 bit/s
    unrestrictedVector               BOOLEAN,
    arithmeticCoding                 BOOLEAN,
    advancedPrediction               BOOLEAN,
    pbFrames                         BOOLEAN,
    ...,

    errorCompensation                BOOLEAN,
    enhancementLayerInfo             EnhancementLayerInfo OPTIONAL,
    h263Options                      H263Options OPTIONAL
}

```

```

IS11172VideoMode                   ::=SEQUENCE
{
    constrainedBitstream             BOOLEAN,
    videoBitRate                    INTEGER(0..1073741823) OPTIONAL, -- units
    -- 400 bit/s
    vbvBufferSize                    INTEGER(0..262143)  OPTIONAL, -- units
    -- 16 384 bits
    samplesPerLine                   INTEGER(0..16383)  OPTIONAL, -- units
    -- samples/line
    linesPerFrame                   INTEGER(0..16383)  OPTIONAL, -- units
    -- lines/frame
    pictureRate                      INTEGER(0..15)    OPTIONAL,
    luminanceSampleRate              INTEGER(0..4294967295) OPTIONAL, -- units
    -- samples/s
    ...
}

```

```

-- =====
-- Request mode definitions: Audio modes
-- =====

```

```

AudioMode                           ::=CHOICE
{
    nonStandard                     NonStandardParameter,
    g711Alaw64k                    NULL,
    g711Alaw56k                    NULL,
}

```

```

g711Ulaw64k          NULL,
g711Ulaw56k          NULL,

g722-64k             NULL,
g722-56k             NULL,
g722-48k             NULL,

g728                 NULL,
g729                 NULL,
g729AnnexA           NULL,

g7231                CHOICE
{
  noSilenceSuppressionLowRate  NULL,
  noSilenceSuppressionHighRate NULL,
  silenceSuppressionLowRate    NULL,
  silenceSuppressionHighRate   NULL
},

is11172AudioMode     IS11172AudioMode,
is13818AudioMode     IS13818AudioMode,

...,
g729wAnnexB          INTEGER(1..256),
g729AnnexAwAnnexB    INTEGER(1..256),
g7231AnnexCMode      G7231AnnexCMode,
gsmFullRate          GSMAudioCapability,
gsmHalfRate          GSMAudioCapability,
gsmEnhancedFullRate  GSMAudioCapability,
genericAudioMode     GenericCapability,
g729Extensions       G729Extensions,
vbd                  VBDMode
}

IS11172AudioMode     ::=SEQUENCE
{
  audioLayer          CHOICE
  {
    audioLayer1       NULL,
    audioLayer2       NULL,
    audioLayer3       NULL
  },

  audioSampling       CHOICE
  {
    audioSampling32k  NULL,
    audioSampling44k1 NULL,
    audioSampling48k  NULL
  },

  multichannelType    CHOICE
  {
    singleChannel     NULL,
    twoChannelStereo  NULL,
    twoChannelDual     NULL
  },

  bitRate             INTEGER (1..448),    -- units kbit/s
  ...
}

```



```

IS13818AudioMode ::=SEQUENCE
{
    audioLayer CHOICE
    {
        audioLayer1 NULL,
        audioLayer2 NULL,
        audioLayer3 NULL
    },
    audioSampling CHOICE
    {
        audioSampling16k NULL,
        audioSampling22k05 NULL,
        audioSampling24k NULL,
        audioSampling32k NULL,
        audioSampling44k1 NULL,
        audioSampling48k NULL
    },
    multichannelType CHOICE
    {
        singleChannel NULL,
        twoChannelStereo NULL,
        twoChannelDual NULL,
        threeChannels2-1 NULL,
        threeChannels3-0 NULL,
        fourChannels2-0-2-0 NULL,
        fourChannels2-2 NULL,
        fourChannels3-1 NULL,
        fiveChannels3-0-2-0 NULL,
        fiveChannels3-2 NULL
    },
    lowFrequencyEnhancement BOOLEAN,
    multilingual BOOLEAN,
    bitRate INTEGER (1..1130), -- units kbit/s
    ...
}

G7231AnnexCMode ::= SEQUENCE
{
    maxAl-sduAudioFrames INTEGER (1..256),
    silenceSuppression BOOLEAN,
    g723AnnexCAudioMode SEQUENCE
    {
        highRateMode0 INTEGER (27..78), -- units octets
        highRateMode1 INTEGER (27..78), -- units octets
        lowRateMode0 INTEGER (23..66), -- units octets
        lowRateMode1 INTEGER (23..66), -- units octets
        sidMode0 INTEGER (6..17), -- units octets
        sidMode1 INTEGER (6..17), -- units octets
        ...
    },
    ...
}

VBDMODE ::=SEQUENCE
{
    type AudioMode, -- shall not be "vbd"
    ...
}

```

```

-- =====
-- Request mode definitions: Data modes
-- =====

```

```

DataMode ::=SEQUENCE
{
  application CHOICE
  {
    nonStandard NonStandardParameter,
    t120 DataProtocolCapability,
    dsm-cc DataProtocolCapability,
    userData DataProtocolCapability,
    t84 DataProtocolCapability,
    t434 DataProtocolCapability,
    h224 DataProtocolCapability,
    nlpid SEQUENCE
    {
      nlpidProtocol DataProtocolCapability,
      nlpidData OCTET STRING
    },
    dsvdControl NULL,
    h222DataPartitioning DataProtocolCapability,
    ...,
    t30fax DataProtocolCapability,
    t140 DataProtocolCapability,
    t38fax SEQUENCE
    {
      t38FaxProtocol DataProtocolCapability,
      t38FaxProfile T38FaxProfile
    },
    genericDataMode GenericCapability
  },
  bitRate INTEGER (0..4294967295), -- units 100 bit/s
  ...
}

```

```

-- =====
-- Request mode definitions: Encryption modes
-- =====

```

```

EncryptionMode ::=CHOICE
{
  nonStandard NonStandardParameter,
  h233Encryption NULL,
  ...
}

```

```

-- =====
-- Round Trip Delay definitions
-- =====

```

```

RoundTripDelayRequest ::=SEQUENCE
{
  sequenceNumber SequenceNumber,
  ...
}

RoundTripDelayResponse ::=SEQUENCE
{
  sequenceNumber SequenceNumber,
  ...
}

```

```

-- =====
-- Maintenance Loop definitions
-- =====

```

```

MaintenanceLoopRequest ::=SEQUENCE
{
    type CHOICE
    {
        systemLoop NULL,
        mediaLoop LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    ...
}

```

```

MaintenanceLoopAck ::=SEQUENCE
{
    type CHOICE
    {
        systemLoop NULL,
        mediaLoop LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    ...
}

```

```

MaintenanceLoopReject ::=SEQUENCE
{
    type CHOICE
    {
        systemLoop NULL,
        mediaLoop LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    cause CHOICE
    {
        canNotPerformLoop NULL,
        ...
    },
    ...
}

```

```

MaintenanceLoopOffCommand ::=SEQUENCE
{
    ...
}

```

```

-- =====
-- Communication Mode definitions
-- =====

```

```

CommunicationModeCommand ::=SEQUENCE
{
    communicationModeTable SET SIZE(1..256) OF CommunicationModeTableEntry,
    ...
}

```

```

CommunicationModeRequest ::=SEQUENCE
{
    ...
}

```

```

CommunicationModeResponse      ::=CHOICE
{
    communicationModeTable     SET SIZE(1..256) OF CommunicationModeTableEntry,
    ...
}

CommunicationModeTableEntry    ::=SEQUENCE
{
    nonStandard                SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID                  INTEGER(1..255),
    associatedSessionID        INTEGER(1..255) OPTIONAL,

    terminalLabel              TerminalLabel OPTIONAL, -- if not present,
                                                                -- it refers to
                                                                -- all
                                                                -- participants in
                                                                -- the conference

    sessionDescription         BMPString (SIZE(1..128)) ,
                                                                -- Basic ISO/IEC 10646-1 (Unicode)

    dataType                   CHOICE
    {
        videoData              VideoCapability,
        audioData              AudioCapability,
        data                    DataApplicationCapability,
        ...
    },
    mediaChannel                TransportAddress OPTIONAL,
    mediaGuaranteedDelivery     BOOLEAN OPTIONAL,
    mediaControlChannel         TransportAddress OPTIONAL,
                                                                -- reverse RTCP channel

    mediaControlGuaranteedDelivery BOOLEAN OPTIONAL,
    ...,
    redundancyEncoding         RedundancyEncoding OPTIONAL,
    sessionDependency           INTEGER (1..255) OPTIONAL,
    destination                 TerminalLabel OPTIONAL
}

```

```

-- =====
-- Conference Request definitions
-- =====

```

```

ConferenceRequest              ::=CHOICE
{
    terminalListRequest         NULL, -- same as H.230 TCU (term->MC)

    makeMeChair                NULL, -- same as H.230 CCA (term->MC)
    cancelMakeMeChair          NULL, -- same as H.230 CIS (term->MC)

    dropTerminal               TerminalLabel, -- same as H.230 CCD (term->MC)

    requestTerminalID          TerminalLabel, -- same as TCP (term->MC)

    enterH243Password          NULL, -- same as H.230 TCS1 (MC->term)
    enterH243TerminalID        NULL, -- same as H.230 TCS2/TCI
                                                                -- (MC->term)
    enterH243ConferenceID      NULL, -- same as H.230 TCS3 (MC->term)
    ...,
    enterExtensionAddress       NULL, -- same as H.230 TCS4 (GW->term)
    requestChairTokenOwner     NULL, -- same as H.230 TCA (term->MC)
    requestTerminalCertificate SEQUENCE
    {

```

```

terminalLabel
certSelectionCriteria
sRandom
...
},
broadcastMyLogicalChannel
makeTerminalBroadcaster
sendThisSource
requestAllTerminalIDs
remoteMCRequest
}

```

```

TerminalLabel OPTIONAL,
CertSelectionCriteria OPTIONAL,
INTEGER (1..4294967295) OPTIONAL,
-- this is the requester's challenge
LogicalChannelNumber, -- similar to H.230 MCV
TerminalLabel, -- similar to H.230 VCB
TerminalLabel, -- similar to H.230 VCS
NULL,
RemoteMCRequest

```

```

CertSelectionCriteria ::=SEQUENCE SIZE (1..16) OF Criteria

```

```

Criteria ::=SEQUENCE
{
    field OBJECT IDENTIFIER, -- may include
    value -- certificate type
    ...
}

```

```

TerminalLabel ::=SEQUENCE
{
    mcuNumber McuNumber,
    terminalNumber TerminalNumber,
    ...
}

```

```

McuNumber ::=INTEGER(0..192)
TerminalNumber ::=INTEGER(0..192)

```

```

-- =====
-- Conference Response definitions
-- =====

```

```

ConferenceResponse ::=CHOICE
{
    mCTerminalIDResponse SEQUENCE -- response to TCP
    { -- (same as TIP)
        terminalLabel TerminalLabel, -- sent by MC only
        terminalID TerminalID,
        ...
    },
    terminalIDResponse SEQUENCE -- response to TCS2 or TCI
    { -- same as IIS
        terminalLabel TerminalLabel, -- (term->MC)
        terminalID TerminalID,
        ...
    },
    conferenceIDResponse SEQUENCE -- response to TCS3
    { -- same as IIS
        terminalLabel TerminalLabel, -- (term->MC)
        conferenceID ConferenceID,
        ...
    },
}

```

```

passwordResponse          SEQUENCE      -- response to TCS1
{
  terminalLabel           TerminalLabel, -- same as IIS
  password                Password,
  ...
},

terminalListResponse      SET SIZE (1..256) OF TerminalLabel,

videoCommandReject       NULL,          -- same as H.230 VCR
terminalDropReject       NULL,          -- same as H.230 CIR

makeMeChairResponse      CHOICE         -- same as H.230 CCR
{
  grantedChairToken      NULL,          -- same as H.230 CIT
  deniedChairToken       NULL,          -- same as H.230 CCR
  ...
},
...,
extensionAddressResponse SEQUENCE      -- response to TCS4
{
  extensionAddress       TerminalID,    -- same as IIS (term->GW)
  ...
},
chairTokenOwnerResponse  SEQUENCE      -- response to TCA (same as TIR)
                               -- sent by MC only
{
  terminalLabel           TerminalLabel,
  terminalID              TerminalID,
  ...
},
terminalCertificateResponse SEQUENCE
{
  terminalLabel           TerminalLabel OPTIONAL,
  certificateResponse     OCTET STRING (SIZE(1..65535)) OPTIONAL,
  ...
},
broadcastMyLogicalChannelResponse CHOICE
{
  grantedBroadcastMyLogicalChannel NULL,      -- similar to H.230 MVA
  deniedBroadcastMyLogicalChannel NULL,      -- similar to H.230 MVR
  ...
},
makeTerminalBroadcasterResponse CHOICE
{
  grantedMakeTerminalBroadcaster NULL,
  deniedMakeTerminalBroadcaster NULL,
  ...
},
sendThisSourceResponse    CHOICE
{
  grantedSendThisSource    NULL,
  deniedSendThisSource    NULL,
  ...
},
requestAllTerminalIDsResponse RequestAllTerminalIDsResponse,
remoteMCResponse          RemoteMCResponse
}

TerminalID                ::=OCTET STRING (SIZE(1..128)) -- as per H.230
ConferenceID              ::=OCTET STRING (SIZE(1..32))
Password                  ::=OCTET STRING (SIZE(1..32))

```

```

RequestAllTerminalIDsResponse      ::=SEQUENCE
{
    terminalInformation             SEQUENCE OF TerminalInformation,
    ...
}

TerminalInformation                ::=SEQUENCE
{
    terminalLabel                  TerminalLabel,
    terminalID                     TerminalID,
    ...
}

-- =====
-- Remote MC Request definitions
-- =====

RemoteMCRequest                    ::=CHOICE
{
    masterActivate                 NULL,
    slaveActivate                  NULL,
    deActivate                     NULL,
    ...
}

RemoteMCResponse                   ::=CHOICE
{
    accept                         NULL,
    reject                         CHOICE
    {
        unspecified                NULL,
        functionNotSupported        NULL,
        ...
    },
    ...
}

-- =====
-- Multilink definitions
-- =====

MultilinkRequest                   ::=CHOICE
{
    nonStandard                    NonStandardMessage,

    callInformation                 SEQUENCE
    {
        maxNumberOfAdditionalConnections  INTEGER (1..65535),
        ...
    },

    addConnection                  SEQUENCE
    {
        sequenceNumber              SequenceNumber, -- Unique ID of request
        dialingInformation           DialingInformation,
        ...
    },

    removeConnection               SEQUENCE
    {
        connectionIdentifier         ConnectionIdentifier,
        ...
    },
}

```

```

maximumHeaderInterval      SEQUENCE
{
  requestType              CHOICE
  {
    currentIntervalInformation  NULL,
    requestedInterval          INTEGER (0..65535), -- Max Header
                                -- Interval,
                                -- milliseconds
    ...
  },
  ...
},
...
}

MultilinkResponse          ::=CHOICE
{
  nonStandard              NonStandardMessage,

  callInformation          SEQUENCE
  {
    dialingInformation      DialingInformation,
    callAssociationNumber   INTEGER (0..4294967295),
    ...
  },

  addConnection            SEQUENCE
  {
    sequenceNumber          SequenceNumber, -- Equal to value in request
    responseCode            CHOICE
    {
      accepted              NULL,
      rejected              CHOICE
      {
        connectionsNotAvailable  NULL, -- due to any technical reason
        userRejected           NULL,
        ...
      },
      ...
    },
    ...
  },

  removeConnection         SEQUENCE
  {
    connectionIdentifier    ConnectionIdentifier,
    ...
  },

  maximumHeaderInterval    SEQUENCE
  {
    currentInterval         INTEGER (0..65535), -- Max Header
                                -- Interval,
                                -- milliseconds
    ...
  },
  ...
}

MultilinkIndication        ::=CHOICE
{
  nonStandard              NonStandardMessage,

```



```

        crcDesired                               SEQUENCE
        {
            ...
        },
        excessiveError                           SEQUENCE
        {
            connectionIdentifier                 ConnectionIdentifier,
            ...
        },
        ...
    }

DialingInformation                             ::= CHOICE
{
    nonStandard                                NonStandardMessage,

    differential                               SET SIZE (1..65535) OF DialingInformationNumber,
                                           -- list of numbers for all additional
                                           -- channels; only least significant digits
                                           -- different from initial channel's number

    infoNotAvailable                           INTEGER (1..65535), -- maximum No. of
                                           -- additional channels
    ...
}

DialingInformationNumber                       ::= SEQUENCE
{
    networkAddress                             NumericString (SIZE (0..40)),
    subAddress                                 IA5String (SIZE (1..40)) OPTIONAL,
    networkType                               SET SIZE (1..255) OF DialingInformationNetworkType,
    ...
}

DialingInformationNetworkType                 ::= CHOICE
{
    nonStandard                                NonStandardMessage,
    n-isdn                                     NULL,
    gstn                                       NULL,
    ...,
    mobile                                     NULL
}

ConnectionIdentifier                           ::= SEQUENCE
{
    channelTag                                INTEGER (0..4294967295), -- from H.226
    sequenceNumber                            INTEGER (0..4294967295), -- from H.226
    ...
}

-- =====
-- Logical channel bit-rate change definitions
-- =====

MaximumBitRate                               ::=INTEGER (0.. 4294967295) -- units of 100 bit/s

LogicalChannelRateRequest                     ::=SEQUENCE
{
    sequenceNumber                             SequenceNumber,
    logicalChannelNumber                       LogicalChannelNumber,
    maximumBitRate                             MaximumBitRate,
    ...
}

```

```

LogicalChannelRateAcknowledge      ::=SEQUENCE
{
    sequenceNumber      SequenceNumber,
    logicalChannelNumber LogicalChannelNumber,
    maximumBitRate     MaximumBitRate,
    ...
}

LogicalChannelRateReject          ::=SEQUENCE
{
    sequenceNumber      SequenceNumber,
    logicalChannelNumber LogicalChannelNumber,
    rejectReason        LogicalChannelRateRejectReason,
    currentMaximumBitRate MaximumBitRate OPTIONAL,
    ...
}

LogicalChannelRateRejectReason    ::=CHOICE
{
    undefinedReason     NULL,
    insufficientResources NULL,
    ...
}

LogicalChannelRateRelease         ::=SEQUENCE
{
    ...
}

-- =====
-- Command Message definitions
-- =====

-- =====
-- Command Message: Send Terminal Capability Set
-- =====

SendTerminalCapabilitySet         ::=CHOICE
{
    specificRequest      SEQUENCE
    {
        multiplexCapability BOOLEAN,

        capabilityTableEntryNumbers SET SIZE (1..65535) OF
            CapabilityTableEntryNumber OPTIONAL,

        capabilityDescriptorNumbers SET SIZE (1..256) OF
            CapabilityDescriptorNumber OPTIONAL,

        ...
    },
    genericRequest        NULL,
    ...
}

-- =====
-- Command Message: Encryption
-- =====

EncryptionCommand                ::=CHOICE
{
    encryptionSE          OCTET STRING,          -- per H.233, but no
                                                    -- error protection
    encryptionIVRequest   NULL,                 -- requests new IV
}

```

```

        encryptionAlgorithmID      SEQUENCE
        {
            h233AlgorithmIdentifier  SequenceNumber,
            associatedAlgorithm       NonStandardParameter
        },
        ...
    }

```

```

-- =====
-- Command Message: Flow Control
-- =====

```

```

FlowControlCommand                ::=SEQUENCE
{
    scope      CHOICE
    {
        logicalChannelNumber      LogicalChannelNumber,
        resourceID                 INTEGER (0..65535),
        wholeMultiplex             NULL
    },
    restriction CHOICE
    {
        maximumBitRate             INTEGER (0..16777215), -- units 100 bit/s
        noRestriction              NULL
    },
    ...
}

```

```

-- =====
-- Command Message: Change or End Session
-- =====

```

```

EndSessionCommand                 ::=CHOICE
{
    nonStandard                    NonStandardParameter,

    disconnect                     NULL,

    gstnOptions                   CHOICE
    {
        telephonyMode             NULL,
        v8bis                     NULL,
        v34DSVD                   NULL,
        v34DuplexFAX              NULL,
        v34H324                   NULL,
        ...
    },

    ...,

    isdnOptions                   CHOICE
    {
        telephonyMode             NULL,
        v140                      NULL,
        terminalOnHold            NULL,
        ...
    }
}

```

```

-- =====
-- Command Message: Conference Commands
-- =====

```

```

ConferenceCommand                 ::=CHOICE
{

```

```

broadcastMyLogicalChannel      LogicalChannelNumber, -- similar to H.230 MCV
cancelBroadcastMyLogicalChannel LogicalChannelNumber, -- similar to
                               -- H.230 Cancel-MCV

makeTerminalBroadcaster      TerminalLabel,      -- same as H.230 VCB
cancelMakeTerminalBroadcaster NULL,              -- same as H.230
                               -- Cancel-VCB

sendThisSource               TerminalLabel,      -- same as H.230 VCS
cancelSendThisSource         NULL,              -- same as H.230
                               -- cancel VCS

dropConference               NULL,              -- same as H.230 CCK
...
substituteConferenceIDCommand SubstituteConferenceIDCommand
}

SubstituteConferenceIDCommand ::=SEQUENCE
{
    conferenceIdentifier      OCTET STRING (SIZE(16)),
    ...
}

-- =====
-- Command Message: Miscellaneous H.230-like commands
-- =====

EncryptionUpdateDirection   ::= CHOICE
{
    masterToSlave            NULL,
    slaveToMaster           NULL,
    ...
}

MiscellaneousCommand         ::=SEQUENCE
{
    logicalChannelNumber     LogicalChannelNumber,
    type                     CHOICE
    {
        equaliseDelay        NULL,              -- same as H.230 ACE
        zeroDelay            NULL,              -- same as H.230 ACZ
        multipointModeCommand NULL,
        cancelMultipointModeCommand NULL,
        videoFreezePicture   NULL,
        videoFastUpdatePicture NULL,

        videoFastUpdateGOB   SEQUENCE
        {
            firstGOB         INTEGER (0..17),
            numberOfGOBs     INTEGER (1..18)
        },

        videoTemporalSpatialTradeOff INTEGER (0..31), -- commands a trade-off value

        videoSendSyncEveryGOB NULL,
        videoSendSyncEveryGOBCancel NULL,

        ...
        videoFastUpdateMB    SEQUENCE
        {
            firstGOB         INTEGER (0..255) OPTIONAL,
            firstMB          INTEGER (1..8192) OPTIONAL,
            numberOfMBs     INTEGER (1..8192),
            ...
        }
    }
}

```

```

},
maxH223MUXPDUsize          INTEGER(1..65535),    -- units octets
encryptionUpdate           EncryptionSync,
encryptionUpdateRequest    EncryptionUpdateRequest,
switchReceiveMediaOff      NULL,
switchReceiveMediaOn       NULL,

progressiveRefinementStart SEQUENCE
{
  repeatCount              CHOICE
  {
    doOneProgression          NULL,
    doContinuousProgressions NULL,
    doOneIndependentProgression NULL,
    doContinuousIndependentProgressions NULL,
    ...
  },
  ...
},
progressiveRefinementAbortOne      NULL,
progressiveRefinementAbortContinuous NULL,

videoBadMBs                    SEQUENCE
{
  firstMB                    INTEGER (1..9216),
  numberOfMBs                INTEGER (1..9216),
  temporalReference          INTEGER (0..1023),
  ...
},
lostPicture                    SEQUENCE OF PictureReference,
lostPartialPicture            SEQUENCE
{
  pictureReference           PictureReference,
  firstMB                    INTEGER (1..9216),
  numberOfMBs                INTEGER (1..9216),
  ...
},
recoveryReferencePicture       SEQUENCE OF PictureReference,
encryptionUpdateCommand       SEQUENCE -- for ack'ed key update in H.235V3
{
  encryptionSync             EncryptionSync,
  multiplePayloadStream      MultiplePayloadStream OPTIONAL,
  ...
},
encryptionUpdateAck           SEQUENCE
{
  synchFlag                  INTEGER (0..255),
  ...
}
},
...,
direction                      EncryptionUpdateDirection OPTIONAL
}

KeyProtectionMethod           ::=SEQUENCE -- specify how the new
                                -- key is to be protected
{
  secureChannel               BOOLEAN,
  sharedSecret                BOOLEAN,
  certProtectedKey            BOOLEAN,
  ...
}

```

```

EncryptionUpdateRequest      ::=SEQUENCE
{
    keyProtectionMethod      KeyProtectionMethod OPTIONAL,
    ...,
    synchFlag                INTEGER (0..255) OPTIONAL
}

PictureReference             ::=CHOICE
{
    pictureNumber            INTEGER (0..1023),
    longTermPictureIndex    INTEGER (0..255),
    ...
}

-----
-- Command Message: H.223 Multiplex Reconfiguration
-----

H223MultiplexReconfiguration ::=CHOICE
{
    h223ModeChange          CHOICE
    {
        toLevel0            NULL,
        toLevel1            NULL,
        toLevel2            NULL,
        toLevel2withOptionalHeader NULL,
        ...
    },
    h223AnnexADoubleFlag    CHOICE
    {
        start                NULL,
        stop                 NULL,
        ...
    },
    ...
}

-----
-- Command Message: New ATM virtual channel command
-----

NewATMVCCCommand           ::=SEQUENCE
{
    resourceID              INTEGER(0..65535),
    bitRate                 INTEGER(1..65535),    -- units 64 kbit/s
    bitRateLockedToPCRClock BOOLEAN,
    bitRateLockedToNetworkClock BOOLEAN,
    aal                     CHOICE
    {
        aall                SEQUENCE
        {
            clockRecovery    CHOICE
            {
                nullClockRecovery NULL,
                srtsClockRecovery NULL,
                adaptiveClockRecovery NULL,
                ...
            },
            errorCorrection    CHOICE
            {
                nullErrorCorrection NULL,
                longInterleaver NULL,
                shortInterleaver NULL,
            }
        }
    }
}

```

```

        errorCorrectionOnly      NULL,
        ...
    },
    structuredDataTransfer      BOOLEAN,
    partiallyFilledCells        BOOLEAN,
    ...
},
aal5                            SEQUENCE
{
    forwardMaximumSDUSize      INTEGER (0..65535),  -- units octets
    backwardMaximumSDUSize     INTEGER (0..65535),  -- units octets
    ...
},
...
},
multiplex                       CHOICE
{
    noMultiplex                NULL,
    transportStream            NULL,
    programStream              NULL,
    ...
},
reverseParameters              SEQUENCE
{
    bitRate                    INTEGER(1..65535),  -- units 64 kbit/s
    bitRateLockedToPCRClock    BOOLEAN,
    bitRateLockedToNetworkClock  BOOLEAN,
    multiplex                   CHOICE
    {
        noMultiplex            NULL,
        transportStream        NULL,
        programStream          NULL,
        ...
    },
    ...
},
...
}

```

```

-- =====
-- Command Message: Mobile Multilink Reconfiguration command
-- =====

```

```

MobileMultilinkReconfigurationCommand ::=SEQUENCE
{
    sampleSize                  INTEGER (1..255),
    samplesPerFrame             INTEGER (1..255),
    status                      CHOICE
    {
        synchronized           NULL,
        reconfiguration        NULL,
        ...
    },
    ...
}

```

```

-- =====
-- Indication Message definitions
-- =====

```

```

-- =====
-- Indication Message: Function not understood
-- =====

```

-- This is used to return a request, response or command that is not understood

```

FunctionNotUnderstood ::=CHOICE
{
    request      RequestMessage,
    response     ResponseMessage,
    command      CommandMessage
}

```

```

-- =====
-- Indication Message: Function not Supported
-- =====

```

-- This is used to return a complete request, response or command that is not recognized

```

FunctionNotSupported ::=SEQUENCE
{
    cause        CHOICE
    {
        syntaxError      NULL,
        semanticError    NULL,
        unknownFunction  NULL,
        ...
    },
    returnedFunction  OCTET STRING OPTIONAL,
    ...
}

```

```

-- =====
-- Indication Message: Conference
-- =====

```

```

ConferenceIndication ::=CHOICE
{
    sbeNumber          INTEGER (0..9), -- same as H.230 SBE Number
    terminalNumberAssign TerminalLabel, -- same as H.230 TIA
    terminalJoinedConference TerminalLabel, -- same as H.230 TIN
    terminalLeftConference TerminalLabel, -- same as H.230 TID
    seenByAtLeastOneOther NULL, -- same as H.230 MIV
    cancelSeenByAtLeastOneOther NULL, -- same as H.230 cancel MIV
    seenByAll          NULL, -- like H.230 MIV
    cancelSeenByAll    NULL, -- like H.230 MIV
    terminalYouAreSeeing TerminalLabel, -- same as H.230 VIN
    requestForFloor    NULL, -- same as H.230 TIF
    ...,
    withdrawChairToken NULL, -- same as H.230 CCR MC-> chair
    floorRequested     TerminalLabel, -- same as H.230 TIF MC-> chair
    terminalYouAreSeeingInSubPictureNumber TerminalYouAreSeeingInSubPictureNumber,
    videoIndicateCompose VideoIndicateCompose
}

```



```

TerminalYouAreSeeingInSubPictureNumber ::= SEQUENCE
{
    terminalNumber      TerminalNumber,
    subPictureNumber   INTEGER (0..255),
    ...
}

VideoIndicateCompose ::= SEQUENCE
{
    compositionNumber  INTEGER (0..255),
    ...
}

-- =====
-- Indication Message: Miscellaneous H.230-like indication
-- =====

MiscellaneousIndication ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    type CHOICE
    {
        logicalChannelActive  NULL,      -- same as H.230 AIA and VIA
        logicalChannelInactive NULL,      -- same as H.230 AIM and VIS

        multipointConference  NULL,
        cancelMultipointConference NULL,

        multipointZeroComm    NULL,      -- same as H.230 MIZ
        cancelMultipointZeroComm NULL,    -- same as H.230 cancel MIZ

        multipointSecondaryStatus NULL,   -- same as H.230 MIS
        cancelMultipointSecondaryStatus NULL, -- same as H.230 cancel MIS

        videoIndicateReadyToActivate NULL, -- same as H.230 VIR

        videoTemporalSpatialTradeOff INTEGER (0..31), -- indicates current
                                                    -- trade-off

        ...,
        videoNotDecodedMBS SEQUENCE
        {
            firstMB          INTEGER (1..8192),
            numberOfMBS      INTEGER (1..8192),
            temporalReference INTEGER (0..255),
            ...
        },
        transportCapability TransportCapability
    },
    ...
}

-- =====
-- Indication Message: Jitter Indication
-- =====

JitterIndication ::=SEQUENCE
{
    scope CHOICE
    {
        logicalChannelNumber LogicalChannelNumber,
        resourceID           INTEGER (0..65535),
        wholeMultiplex       NULL
    },
    ...
}

```

```

    estimatedReceivedJitterMantissa INTEGER (0..3),
    estimatedReceivedJitterExponent INTEGER (0..7),
    skippedFrameCount              INTEGER (0..15) OPTIONAL,
    additionalDecoderBuffer         INTEGER (0..262143) OPTIONAL,
                                   -- 262143 is 2^18 - 1
    ...
}

-- =====
-- Indication Message: H.223 logical channel skew
-- =====

H223SkewIndication                ::=SEQUENCE
{
    logicalChannelNumber1          LogicalChannelNumber,
    logicalChannelNumber2          LogicalChannelNumber,
    skew        INTEGER (0..4095),  -- units milliseconds
    ...
}

-- =====
-- Indication Message: H.225.0 maximum logical channel skew
-- =====

H2250MaximumSkewIndication        ::=SEQUENCE
{
    logicalChannelNumber1          LogicalChannelNumber,
    logicalChannelNumber2          LogicalChannelNumber,
    maximumSkew                    INTEGER (0..4095),  -- units milliseconds
    ...
}

-- =====
-- Indication Message: MC Location Indication
-- =====

MCLocationIndication              ::=SEQUENCE
{
    signalAddress                  TransportAddress, -- this is the
                                   -- H.323 Call Signalling
                                   -- address of the entity
                                   -- which contains the MC
    ...
}

-- =====
-- Indication Message: Vendor Identification
-- =====

VendorIdentification              ::=SEQUENCE
{
    vendor                         NonStandardIdentifier,
    productNumber                  OCTET STRING (SIZE(1..256)) OPTIONAL,
                                   -- per vendor
    versionNumber                  OCTET STRING (SIZE(1..256)) OPTIONAL,
                                   -- per productNumber
    ...
}

```

```

-- =====
-- Indication Message: New ATM virtual channel indication
-- =====

```

```

NewATMVCIndication ::=SEQUENCE
{
  resourceID          INTEGER(0..65535),
  bitRate             INTEGER(1..65535),  -- units 64 kbit/s
  bitRateLockedToPCRClock  BOOLEAN,
  bitRateLockedToNetworkClock  BOOLEAN,
  aal                 CHOICE
  {
    aal1              SEQUENCE
    {
      clockRecovery   CHOICE
      {
        nullClockRecovery  NULL,
        srtsClockRecovery  NULL,
        adaptiveClockRecovery  NULL,
        ...
      },
      errorCorrection  CHOICE
      {
        nullErrorCorrection  NULL,
        longInterleaver      NULL,
        shortInterleaver     NULL,
        errorCorrectionOnly   NULL,
        ...
      },
      structuredDataTransfer  BOOLEAN,
      partiallyFilledCells    BOOLEAN,
      ...
    },
    aal5              SEQUENCE
    {
      forwardMaximumSDUSize  INTEGER (0..65535),  -- units octets
      backwardMaximumSDUSize  INTEGER (0..65535),  -- units octets
      ...
    },
    ...
  },
  multiplex           CHOICE
  {
    noMultiplex        NULL,
    transportStream    NULL,
    programStream      NULL,
    ...
  },
  ...,
  reverseParameters  SEQUENCE
  {
    bitRate            INTEGER(1..65535),  -- units 64 kbit/s
    bitRateLockedToPCRClock  BOOLEAN,
    bitRateLockedToNetworkClock  BOOLEAN,
    multiplex          CHOICE
    {
      noMultiplex      NULL,
      transportStream  NULL,
      programStream    NULL,
      ...
    },
    ...
  },
  ...
}
}

```

```

-- =====
-- Indication Message: User input
-- =====

IV8 ::= OCTET STRING (SIZE(8))
-- initial value for
-- 64-bit block ciphers

IV16 ::= OCTET STRING (SIZE(16))
-- initial value for
-- 128-bit block ciphers

Params ::= SEQUENCE
{
    iv8 IV8 OPTIONAL, -- 8-octet initialization vector
    iv16 IV16 OPTIONAL, -- 16-octet initialization vector
    iv OCTET STRING OPTIONAL, -- arbitrary length
    -- initialization vector
    ...
}

UserInputIndication ::= CHOICE
{
    nonStandard alphanumeric NonStandardParameter,
    ...,
    userInputSupportIndication CHOICE
    {
        nonStandard NonStandardParameter,
        basicString NULL, -- indicates unsecured basic string
        ia5String NULL, -- indicates unsecured IA5 string
        generalString NULL, -- indicates unsecured general string
        ...,
        encryptedBasicString NULL, -- indicates encrypted Basic string
        encryptedIA5String NULL, -- indicates encrypted IA5 string
        encryptedGeneralString NULL -- indicates encrypted general string
    },
    signal SEQUENCE
    {
        signalType IA5String (SIZE (1) ^ FROM ("0123456789#*ABCD!")),
        -- holds dummy "!" if encryptedSignalType
        -- is being used
        duration INTEGER (1..65535) OPTIONAL,
        -- milliseconds
        rtp SEQUENCE
        {
            timestamp INTEGER (0..4294967295) OPTIONAL,
            expirationTime INTEGER (0..4294967295) OPTIONAL,
            logicalChannelNumber LogicalChannelNumber,
            ...
        } OPTIONAL,
        ...,
        rtpPayloadIndication NULL OPTIONAL,
        params Params OPTIONAL, -- any "runtime" parameters
        encryptedSignalType OCTET STRING (SIZE(1)) OPTIONAL,
        -- encrypted signalType
        algorithmOID OBJECT IDENTIFIER OPTIONAL
    },
    signalUpdate SEQUENCE
    {
        duration INTEGER (1..65535), -- milliseconds
        rtp SEQUENCE
        {

```

```

        logicalChannelNumber    LogicalChannelNumber,
        ...
    } OPTIONAL,
    ...
},
extendedAlphanumeric          SEQUENCE
{
    alphanumeric              GeneralString, -- holds empty string if
                                -- encryptedAlphanumeric is
                                -- being used
    rtpPayloadIndication      NULL OPTIONAL,
    ...,
    encryptedAlphanumeric     SEQUENCE
    {
        algorithmOID          OBJECT IDENTIFIER,
        paramS                Params OPTIONAL, -- any "runtime" parameters
        encrypted             OCTET STRING, -- general string encrypted
        ...
    } OPTIONAL
},
encryptedAlphanumeric        SEQUENCE
{
    algorithmOID              OBJECT IDENTIFIER,
    paramS                    Params OPTIONAL, -- any "runtime" parameters
    encrypted                 OCTET STRING, -- basic string encrypted
    ...
}
}

```

```

-- =====
-- Indication Message: Flow Control
-- =====

```

```

FlowControlIndication        ::=SEQUENCE
{
    scope                      CHOICE
    {
        logicalChannelNumber   LogicalChannelNumber,
        resourceID             INTEGER (0..65535),
        wholeMultiplex         NULL
    },
    restriction                 CHOICE
    {
        maximumBitRate         INTEGER (0..16777215), -- units 100 bit/s
        noRestriction          NULL
    },
    ...
}

```

```

-- =====
-- Indication Message: Mobile Multilink Reconfiguration indication
-- =====

```

```

MobileMultilinkReconfigurationIndication ::=SEQUENCE
{
    sampleSize                 INTEGER (1..255),
    samplesPerFrame            INTEGER (1..255),
    ...
}

```

END

## Anexo B

### Mensajes: Definiciones semánticas

**B.0** En este anexo se facilitan definiciones y limitaciones semánticas para los elementos de sintaxis definidos en la cláusula anterior.

**B.0.1 MultimediaSystemControlMessage:** elección de tipos de mensaje. Los mensajes definidos en esta Recomendación se clasifican en mensajes de petición, respuesta, instrucción e indicación.

**B.0.2 RequestMessage:** mensaje de petición que provoca una acción por parte del terminal distante y exige una respuesta inmediata del mismo. Puede utilizarse un mensaje nonStandard para enviar peticiones no normalizadas.

**B.0.3 ResponseMessage:** mensaje de respuesta a un mensaje petición. Puede utilizarse un mensaje nonStandard para enviar respuestas no normalizadas.

**B.0.4 CommandMessage:** mensaje de instrucción que requiere una acción pero no una respuesta explícita. Puede utilizarse un mensaje nonStandard para enviar instrucciones no normalizadas.

**B.0.5 IndicationMessage:** contiene información que no exige ni acción ni respuesta. Puede utilizarse un mensaje nonStandard para enviar indicaciones no normalizadas.

**B.0.6 NonStandardParameter:** puede utilizarse para señalar un parámetro no normalizado. Consta de una identidad y de los parámetros propios codificados en forma de cadena de octetos.

**B.0.7 NonStandardIdentifier:** se utiliza para identificar el tipo de parámetro no normalizado. Es o bien un identificador de objeto, o un tipo H.221 de identificador, que es una cadena de octetos constituida por exactamente cuatro octetos, tal como sigue. El indicativo de país lo componen dos octetos, el primero de los cuales es conforme con el anexo A/T.35. El segundo octeto se asigna a nivel nacional, salvo que el primer octeto sea 1111 1111, en cuyo caso el segundo octeto contendrá el indicativo de país de conformidad con el anexo B/T.35. El código de fabricante del terminal se compone de dos octetos asignados a nivel nacional. Los indicativos de fabricante son los mismos asignados para uso en la Rec. UIT-T H.320 [22]. Los identificadores no normalizados H.245 pueden ser o bien de tipo "object" o de tipo "h221NonStandard", a discreción del fabricante que define el mensaje no normalizado, ya que los OBJECT IDENTIFIER y mensajes h221NonStandard provienen de espacios que no se superponen y no pueden confundirse. Sin embargo, como los mensajes h221NonStandard son también utilizados por la Rec. UIT-T H.320 proceden del mismo espacio que los mensajes H.320, y deberán tener el mismo significado.

#### **B.1 Mensajes de determinación principal-subordinado**

Este conjunto de mensajes es utilizado por un protocolo para determinar el terminal que es el principal y el que es el subordinado.

##### **B.1.1 Determinación principal-subordinado**

Este mensaje lo envía una entidad MSDSE a su entidad par MSDSE.

terminalType es un número que identifica tipos diferentes de terminal, tales como terminales, MCU y pasarelas. La atribución de valores a tipos de terminales está fuera del ámbito de esta Recomendación.

statusDeterminationNumber es un número aleatorio comprendido en la gama de  $0 \dots 2^{24} - 1$ .

## **B.1.2 Acuse de recibo de determinación principal-subordinado**

Se utiliza para confirmar si el terminal es el terminal principal o el terminal subordinado, como se indica por la decisión. Cuando la decisión es de tipo principal, el terminal que recibe este mensaje es el terminal principal y cuando la decisión es de tipo subordinado, es el terminal subordinado.

## **B.1.3 Rechazo de determinación principal-subordinado**

Se utiliza para rechazar el mensaje MasterSlaveDetermination. Cuando la causa es de tipo identicalNumbers, el rechazo se debió a que los números aleatorios eran equivalentes y los tipos de terminal eran iguales.

## **B.1.4 Liberación de determinación principal-subordinado**

Se envía en el caso de expiración de un periodo de temporización.

## **B.2 Mensajes de capacidad de terminal**

Este conjunto de mensajes tiene por objeto el intercambio seguro de capacidades entre dos terminales.

### **B.2.1 Visión general**

El terminal transmisor asigna cada modo individual en el que puede funcionar en varias capabilityTable. Por ejemplo, audio G.723.1, audio G.728 y vídeo CIF H.263, a las que se asignarán números diferentes.

Estos números indicativos de capacidad se agrupan en estructuras AlternativeCapabilitySet. Cada AlternativeCapabilitySet indica que el terminal puede funcionar exactamente en uno de los modos enumerados en el conjunto. Por ejemplo, un listado AlternativeCapabilitySet {G.711, G.723.1, G.728} indica que el terminal puede funcionar en uno cualquiera de esos modos audio, pero solamente en uno.

Estas estructuras AlternativeCapabilitySet se agrupan en estructuras simultaneousCapabilities. Cada estructura simultaneousCapabilities expresa un conjunto de modos que puede utilizar el terminal simultáneamente. Por ejemplo, una estructura simultaneousCapabilities que contenga dos estructuras AlternativeCapabilitySet {H.261, H.263} y {G.711, G.723.1, G.728} indica que el terminal puede funcionar simultáneamente con cualquiera de los códecs de vídeo y con cualquiera de los códecs de audio. El conjunto simultaneousCapabilities {{H.261}, {H.261, H.263}, {G.711, G.723.1, G.728}} significa que el terminal puede funcionar con los dos canales vídeo y un canal de audio simultáneamente: un canal de vídeo según H.261, otro canal de vídeo según H.261 o H.263 y un canal de audio G.711, G.723.1 o G.728.

NOTA – Las capacidades reales almacenadas en capabilityTable son a menudo más complejas que las indicadas anteriormente. Por ejemplo, cada capacidad H.263 indica detalles que comprenden la aptitud para soportar diversos formatos de imagen en intervalos de imagen mínimos dados y la aptitud para utilizar modos de codificación facultativos.

Las capacidades totales de los terminales se describen mediante un conjunto de estructuras CapabilityDescriptor, cada una de las cuales está constituida por una única estructura simultaneousCapabilities y un capabilityDescriptorNumber. Mediante el envío de más de un CapabilityDescriptor, el terminal puede señalar dependencias entre modos de funcionamiento, describiendo los diferentes conjuntos de modos que puede utilizar de forma simultánea. Por ejemplo, un terminal que emita dos estructuras CapabilityDescriptor, {{H.261, H.263}, {G.711, G.723.1, G.728}} como en el ejemplo anterior y otra estructura {{H.262}, {G.711}}, indica que el terminal puede funcionar también con el códec vídeo H.262 pero únicamente con el códec audio G.711 de reducida complejidad.

En el curso de una sesión de comunicación, los terminales pueden, dinámicamente, añadir capacidades emitiendo estructuras CapabilityDescriptor adicionales o eliminar capacidades

mediante la emisión de estructuras CapabilityDescriptor revisadas. Todos los terminales deberán transmitir al menos una estructura CapabilityDescriptor.

## **B.2.2 Conjunto de capacidades de terminal**

Este mensaje contiene información sobre la capacidad del terminal para transmitir y recibir. Indica también la versión de esta Recomendación que se está utilizando. Se envía desde una CESE de salida a la CESE de entrada par.

Se utiliza sequenceNumber para etiquetar ejemplares de TerminalCapabilitySet de forma que pueda identificarse la respuesta correspondiente.

Se utiliza protocolIdentifier para indicar la versión de esta Recomendación que se está utilizando. En el anexo D, se enumeran los identificadores de objeto definidos para su uso por esta Recomendación.

multiplexCapability indica capacidades relacionadas con la multiplexación y la adaptación de red. Todo terminal deberá incluir multiplexCapability en el primer TerminalCapabilitySet que transmita.

V75Capability indica las capacidades de la entidad de control V.75. El audioHeader indica la capacidad del encabezamiento de audio V.75.

### **B.2.2.1 Tabla de capacidades**

Una tabla de capacidades es una lista numerada de capacidades. Un terminal deberá poseer todas las capacidades enumeradas en su tabla de capacidades, aunque no es necesario que tenga la aptitud de ejecutar simultáneamente más de una de ellas.

Un TerminalCapabilitySet puede contener inscripciones CapabilityTableEntry o carecer de ellas. En el arranque no se definen inscripciones de la tabla. Cuando se recibe una CapabilityTableEntry, sustituye a la CapabilityTableEntry recibida anteriormente con el mismo CapabilityTableEntryNumber. Puede utilizarse una CapabilityTableEntry sin Capability para suprimir la CapabilityTableEntry recibida anteriormente con el mismo CapabilityTableEntryNumber.

### **B.2.2.2 Descriptores de capacidad**

Se utilizan los CapabilityDescriptor para indicar la capacidad que tiene un terminal para transmitir y recibir. Cada CapabilityDescriptor proporciona un enunciado independiente acerca de las capacidades del terminal.

Se utiliza el capabilityDescriptorNumber para numerar los CapabilityDescriptor. Si un terminal tiene preferencia sobre el modo que desearía transmitir o recibir y quiere indicar tal preferencia cuando transmite sus capacidades, puede hacerlo proporcionando CapabilityDescriptor que relacionen valores reducidos de capabilityDescriptorNumber con sus modos o modo preferidos.

simultaneousCapabilities es un conjunto de AlternativeCapabilitySet. Se utiliza para listar las capacidades simultáneas del terminal.

Un AlternativeCapabilitySet es una secuencia de CapabilityTableEntryNumbers. En un AlternativeCapabilitySet únicamente estarán presentes aquellas CapabilityTableEntry que se hayan definido, aunque es posible definir CapabilityTableEntry y hacer referencia a ellas en el mismo TerminalCapabilitySet. Si un terminal tiene preferencia sobre el modo que desearía transmitir o recibir, y quiere expresar tal preferencia cuando transmita sus capacidades, puede hacerlo así listando los elementos en AlternativeCapabilitySet en orden de preferencia decreciente.

Un terminal deberá poder ejecutar simultáneamente cualquiera de las capacidades de cada AlternativeCapabilitySet listado en simultaneousCapabilities.

Al menos un descriptor de capacidad deberá tener la siguiente estructura: existirá al menos un AlternativeCapabilitySet que contenga únicamente capacidades de un solo tipo de medio para cada



clase de medio que pueda soportar el terminal, con objeto de que el terminal distante pueda seleccionar un modo de transmisión que incluya, al menos, un caso de cada tipo de medio que pueda sustentar el receptor.

NOTA 1 – Cualquier repetición de una capacidad en un `AlternativeCapabilitySet` es redundante y no proporciona información ulterior, mientras que la repetición de una capacidad en `AlternativeCapabilitySet` diferentes del mismo `CapabilityDescriptor`, indica la posibilidad de un caso adicional y simultáneo de esa capacidad concreta.

NOTA 2 – Los terminales que no puedan variar la atribución de recursos pueden indicar completamente su capacidad empleando un único `CapabilityDescriptor`.

### B.2.2.3 Capacidad

Las opciones `receiveVideoCapability`, `receiveAudioCapability`, `receiveDataApplicationCapability`, `receiveUserInputCapability` y `receiveMultiplexedStreamCapability` indican la capacidad para recibir conforme a la `VideoCapability`, `AudioCapability`, `DataApplicationCapability`, `UserInputCapability` y `MultiplexedStreamCapability` respectiva.

Las opciones `transmitVideoCapability`, `transmitAudioCapability`, `transmitDataApplicationCapability`, `transmitUserInputCapability` y `transmitMultiplexedStreamCapability`, indican la capacidad para transmitir conforme a la `VideoCapability`, `AudioCapability`, `DataApplicationCapability`, `UserInputCapability` y `MultiplexedStreamCapability` respectiva.

Las opciones `receiveAndTransmitVideoCapability`, `receiveAndTransmitAudioCapability`, `receiveAndTransmitDataApplicationCapability`, `receiveAndTransmitUserInputCapability` y `receiveAndTransmitMultiplexedStreamCapability` indican la capacidad de recibir y transmitir conforme a la `VideoCapability`, `AudioCapability`, `DataApplicationCapability`, `UserInputCapability` y `MultiplexedStreamCapability`, respectiva. Estos puntos de código pueden ser útiles para indicar que las capacidades de recepción y transmisión no son independientes.

A modo de aclaración, citamos un ejemplo: un terminal que declara `{{Rx-G.723.1, Rx-G.729}, {Tx-G.723.1, Tx-G.729}}` no indica una limitación simétrica y, por lo tanto, es capaz de recibir la señal del códec G.723.1 mientras transmite la señal del códec G.729, en tanto que un terminal que declara `{{RxAndTx-G.723.1, RxAndTx-G.729}}` indica una limitación simétrica y, por lo tanto, no es capaz de recibir la señal del códec G.723.1 mientras transmite la señal del códec G.729.

La `h233EncryptionTransmitCapability` booleana, cuando es verdadera, indica que el terminal soporta la criptación conforme a las Recs. UIT-T H.233 [14] y H.234 [15].

La variable booleana `h233IVResponseTime` se mide en milisegundos e indica el tiempo mínimo que el receptor solicita del transmisor que espere tras completar la transmisión de un mensaje IV antes de comenzar a utilizar el nuevo IV. En esta Recomendación no se define la forma de transmitir el IV.

`ConferenceCapability` indica diversas capacidades de conferencia.

`multipointVisualizationCapability` (similar a MVC H.230) está incluido en el conjunto de capacidades de una unidad MCU o de un terminal para indicar que generará o tratará correctamente las señales `conferenceResponse.broadcastMyLogicalChannel.grantedBroadcastMyLogicalChannel` (similar a MVA H.230) y `conferenceResponse.broadcastMyLogicalChannel.deniedBroadcastMyLogicalChannel` (similar a MVR H.230) en respuesta a `conferenceRequest.BroadcastMyLogicalChannel` (similar a MCV H.230).

La variable booleana `h235SecurityCapability` indica las capacidades que soporta el terminal de conformidad con la Rec. UIT-T H.235 [16]. El campo `mediaCapability` se referirá a las entradas de la tabla de capacidades que contienen una capacidad de transmisión o recepción o una `receiveAndTransmit` `AudioCapability`, `VideoCapability`, `DataApplicationCapability` o capacidad similar indicada por un `NonStandardParameter` solamente. `genericH235SecurityCapability` indica la

capacidad de seguridad según el anexo G/H.235 con la que el punto extremo soporta el protocolo MIKEY indicado [79]. Cuando se ejecuta MIKEY a nivel medio, `genericParameters` dentro de `genericH235securityCapability` retienen también los mensajes MIKEY.

`EncryptionAuthenticationAndIntegrity` indica las capacidades de criptación, autenticación e integridad que se soportan para la `mediaCapability` señalada. La `mediaCapability` define el audio, el vídeo o los algoritmos de datos soportados así como los métodos de distribución que se soportan (por ejemplo, recepción, transmisión o recepción y transmisión). El parámetro `maxPendingReplacementFor` indica el número máximo de operaciones de apertura de canal lógico que se encuentren simultáneamente en el estado `REPLACEMENT PENDING`. El estado `REPLACEMENT PENDING` se produce cuando se ha establecido un canal lógico utilizando el parámetro `replacementFor`, pero el canal lógico reemplazado todavía no ha sido cerrado.

`genericControlCapability` indica capacidades de control genéricas.

### **B.2.2.3.1 Ejemplo (informativo)**

Supóngase que el punto extremo puede soportar diversos códecs de audio, codificaciones alternativas de un canal de audio tal como DTMF y datos en banda vocal (VBD, *voice band data*), uno o más esquemas de criptación y redundancia RFC 2198, pero no desea soportar todas las combinaciones posibles de dichas capacidades. En particular, puede preferir soportar un canal de cabida útil múltiple con uno de sus códecs de audio soportados, DTMF o VBD. Es capaz de criptar cualesquiera de las cabidas útiles en virtud de algún algoritmo (o conjunto de algoritmos), y puede transmitir VBD con redundancia RFC 2198.

Se podrían anunciar entonces sus capacidades de la siguiente manera:

Primero, las capacidades de audio básicas:

Capacidad 1 = `g711Ulaw64k`

Capacidad 2 = `g729wAnnexB`

Capacidad 3 = `vbd`

Capacidad 4 = `audioTone`

Existe una capacidad alternativa para G.711 o bien G.729 anexo B (pero no para ambas):

Capacidad 5 = `oneOfCapabilities (1, 2)`

Luego, VBD con codificación redundante y transporte DTMF 'fiable':

Capacidad 6 = `RedundancyEncodingCapability` (primario = 3, secundario = 3)

Capacidad 7 = `RedundancyEncodingCapability` (primario = 4, secundario = 4, 4)

A continuación el tren de cabida útil múltiple, con redundancia para VBD:

Capacidad 8 = `MultiplePayloadStreamCapability (5, 6, 7)`

Por último, el tren de cabida útil múltiple con todas las cabidas útiles criptadas:

Capacidad 9 = `H235SecurityCapability (8) [MPS criptada]`

En estas circunstancias, un `alternativeCapabilitySet` podría constar de capacidades (9, 8, 7, 2, 1), que se combinan con capacidades vídeo alternativas y/o capacidades de datos alternativas para formar un `CapabilityDescriptor` multimedia.

### **B.2.2.4 Capacidades de múltiple**

`MultiplexCapability` expresa capacidades relacionadas con la multiplexación y la adaptación de la red. Todo terminal deberá enviar `MultiplexCapability` en el primer `TerminalCapabilitySet` que transmita. A menos que se indique otra cosa se trata de capacidades de recepción.

**H222Capability:** indica capacidades de multiplexación y de adaptación de red propias del múltiplex definido en la Rec. UIT-T H.222.1 [9].

numberOfVCs indica el número de canales virtuales (VC, *virtual channels*) ATM simultáneos que puede soportar el terminal. Se incluyen todos los VC que transportan datos H.245, T.120, DSM-CC o cualesquiera otros, así como todos los VC que transportan información audiovisual. No se incluye el VC utilizado para la señalización Q.2931 [26].

vcCapability es un conjunto de tamaño igual al valor de numberOfVCs, que expresa las capacidades presentes en cada VC disponible.

La secuencia aal1, cuando está presente, indica las opciones de la capa 1 de adaptación ATM, especificadas en las Recs. UIT-T I.363.x [25], soportadas. En el cuadro B.1 se definen los puntos de código.

**Cuadro B.1/H.245 – Puntos de código de la capa 1 de adaptación ATM**

Puntos de código ASN.1	Significado semántico del punto de código
nullClockRecovery	Método de recuperación de frecuencia de reloj de fuente nulo: transporte por circuito síncrono.
srtsClockRecovery	Método de recuperación de frecuencia de reloj de fuente mediante indicación de tiempo residual síncrono.
adaptiveClockRecovery	Método de recuperación de frecuencia de reloj de fuente mediante reloj autoadaptable.
nullErrorCorrection	No se soporta la corrección de errores.
longInterleaver	Se soporta el método de corrección intrínseca de errores para el transporte de señales sensibles a las pérdidas.
shortInterleaver	Se soporta el método de corrección intrínseca de errores para el transporte de señales sensibles al retardo.
errorCorrectionOnly	Se soporta el método de corrección intrínseca de errores sin entrelazado de células.
structuredDataTransfer	Se soporta la transferencia de datos estructurados.
partiallyFilledCells	Se soportan células rellenas parcialmente.

La secuencia aal5, cuando está presente, indica las opciones soportadas de la capa 5 de adaptación ATM, especificada en las Recs. UIT-T I.363.x [25]. forwardMaximumSDUSize y backwardMaximumSDUSize indican el tamaño CPCS-SDU máximo en los sentidos hacia adelante y hacia atrás, medido en octetos. Cualquier aal1 o aal5 o ambos pueden estar presentes.

Las variables booleanas transportStream y programStream, cuando toman el valor verdadero, expresan la capacidad para soportar los múltiplex de tren de transporte y tren de programa, respectivamente [8].

availableBitRates indica las capacidades de velocidad binaria para el canal virtual (VC). Es una secuencia de las diferentes velocidades binarias que pueden ser soportadas, medidas en unidades de 64 kbit/s. Las velocidades binarias se indican en orden decreciente, es decir, la velocidad binaria más alta se indica primero. Las velocidades binarias soportadas pueden indicarse como valores individuales utilizando el campo singleBitRate, o como una gama rangeOfBitRates entre los valores extremos lowerBitRate y higherBitRate, con lo que se indica que todos los valores comprendidos entre este límite menor y el límite mayor, incluidos estos límites, están soportados. Las velocidades binarias indicadas se miden en el punto AAL-SAP.

La secuencia aal1 ViaGateway, cuando está presente, indica la capacidad de la capa 1 de adaptación ATM soportada por pasarelas de conversión AAL1/5. Los puntos de código son los mismos que los

de la secuencia aal1. La secuencia Q2931Address indica uno o más conjuntos de números de parte o subdirecciones de parte Q.2931.

**H223Capability:** indica capacidades específicas del multiplex H.223 [10].

La variable booleana transportWithI-frames, cuando toma el valor verdadero, indica que el terminal es capaz de enviar y recibir mensajes de canal de control, utilizando tramas I LAPM como las definidas en la Rec. UIT-T V.42 [38].

Las variables booleanas videoWithAL1, videoWithAL2, videoWithAL3, audioWithAL1, audioWithAL2, audioWithAL3, dataWithAL1, dataWithAL2 y dataWithAL3, cuando toman el valor verdadero, indican la capacidad de recibir el tipo de medio indicado (vídeo, audio o datos) utilizando la capa de adaptación indicada (AL1, AL2 o AL3).

Los enteros maximumAl2SDUSize y maximumAl3SDUSize indican el número máximo de octetos de cada SDU que puede recibir el terminal, cuando utilice los tipos de capa de adaptación 2 y 3, respectivamente.

maximumDelayJitter indica la máxima fluctuación de multiplexación de cresta a cresta que producirá el transmisor. Se mide en milisegundos. Se define la fluctuación de multiplexación como la diferencia en los tiempos de entrega del primer octeto de una trama de audio cuando se entrega en el tren multiplexado y cuando se entregase con una velocidad de bits constante sin multiplexación.

**h223MultiplexTableCapability:** indica la aptitud del terminal para recibir y procesar inscripciones de la tabla multiplex.

La indicación básico expresa que el multiplex únicamente puede recibir los MultiplexEntryDescriptors básicos definidos en la Rec. UIT-T H.223 [10].

La indicación intensificado expresa que el multiplex puede recibir MultiplexEntryDescriptor intensificados con los parámetros adicionales definidos seguidamente.

La profundidad maximumNestingDepth indica la profundidad máxima de anidamiento de los campos subElementList invocados recursivamente. Los MultiplexEntryDescriptor que no utilizan el campo subElementList se considerará que tienen una profundidad de anidamiento de nula.

El elemento maximumElementListSize indica el número máximo de campos en la ASN.1 SEQUENCE.

El elemento maximumSubElementListSize indica el número máximo de subelementos de SubElementList.

La variable booleana maxMUXPDUSizeCapability, cuando toma el valor verdadero, indica que el transmisor es capaz de restringir el tamaño de las unidades MUX-PDU de H.223 que transmite. No tiene ningún significado cuando forma parte de una capacidad de recepción.

La variable booleana nsrpSupport, cuando toma el valor verdadero, indica soporte del modo NSRP del anexo A/H.324.

**MobileOperationTransmitCapability:** indica la capacidad de transmitir las capas de multiplex descritas en el anexo A/H.223 y en el anexo B/H.223.

La variable booleana h223AnnexA, cuando toma el valor verdadero, indica que el terminal puede transmitir las MUX-PDU definidas en el anexo A/H.223.

La variable booleana h223AnnexADoubleFlag, cuando toma el valor verdadero, indica que el terminal puede transmitir las MUX-PDU definidas en el anexo A/H.223 con su modo doble bandera opcional.

La variable booleana h223AnnexB, cuando toma el valor verdadero, indica que el terminal puede transmitir las MUX-PDU definidas en el anexo B/H.223.

La variable booleana `h223AnnexBwithOptionalHeaderField`, cuando toma el valor verdadero, indica que el terminal puede transmitir la MUX-PDU definida en el anexo B/H.223 con su campo de encabezamiento opcional.

**h223AnnexCCapability**: indica la capacidad para recibir y procesar las AL-PDU como se describe en el anexo C/H.223, con las siguientes condiciones.

Las variables booleanas `videoWithAL1M`, `videoWithAL2M`, `videoWithAL3M`, `audioWithAL1M`, `audioWithAL2M`, `audioWithAL3M`, `dataWithAL1M`, `dataWithAL2M` y `dataWithAL3M`, cuando son verdaderas, indican la capacidad para recibir el tipo medio fijado (vídeo, audio o datos) utilizando la capa de adaptación establecida (AL1M, AL2M o AL3M).

`alpduInterleaving`, si es verdadero, indica la capacidad de recibir y procesar las AL-PDU para las que se aplica entrelazado.

El entero `maximumA11MPDUSize` indica el número máximo de octetos en cada PDU que el terminal puede recibir cuando se utiliza la capa de adaptación AL1M.

Los enteros `maximumA12MSDUSize` y `maximumAL3MSDUSize` indican el número máximo de octetos en cada SDU que el terminal puede recibir cuando se utilizan las capas de adaptación AL2M y AL3M, respectivamente.

`rsCodeCapability`, si es verdadero, indica la capacidad para recibir las AL-PDU para las que se indica codificación Reed-Solomon.

`bitRate`, si está presente, indica la velocidad binaria para transmitir el tren de bits de salida desde el multiplexor H.223.

`mobileMultilinkFrameCapability`, si está presente, indica la capacidad para recibir y procesar tramas de multienlace móviles con `maximumSampleSize` y `maximumPayloadLength` especificados. `maximumSampleSize` indica el número máximo de octetos en cada muestra que el terminal puede procesar. `maximumPayloadLength` indica la longitud máxima de tramas en octetos que el terminal puede procesar.

**V76Capability**: indica capacidades específicas del multiplex V.76.

La dirección `suspendResumeCapabilitywAddress` indica la capacidad para soportar suspensión/reanudación V.76 con un campo de dirección. La dirección `suspend/ResumeCapabilitywoAddress` indica la capacidad para soportar la suspensión/reanudación V.76 sin un campo de dirección.

`rejCapability` indica la capacidad de la función de control de error del multiplex V.76 para efectuar un rechazo.

`sREJCapability` indica la capacidad de la función de control de error del multiplex para efectuar un rechazo selectivo.

`mREJCapability` indica la capacidad de la función de control de error del multiplex para realizar un rechazo selectivo múltiple.

`crc8bitCapability` es la capacidad del multiplex para utilizar CRC de 8 bits.

`crc16bitCapability` es la capacidad del multiplex para utilizar CRC de 16 bits.

`crc32bitCapability` es una capacidad del multiplex para utilizar CRC de 32 bits.

`uihCapability` indica soporte de tramas UIH V.76.

`numOfDLCS` indica el número de DLC que el multiplex V.76 puede soportar.

`twoOctectAddressFieldCapability` indica la aptitud del multiplex V.76 para soportar un campo de dirección de dos octetos.

loopBackTestCapability indica el soporte de conexiones en bucle según la Rec. UIT-T V.76.  
n401Capability indica el valor máximo de N401 descrito en la Rec. UIT-T V.76.  
maxWindowSizeCapability indica el tamaño máximo de ventana que el multiplex V.76 puede soportar.

**H2250Capability:** indica capacidades específicas a la capa de paquetización de medios H.225.0.

maximumAudioDelayJitter indica la fluctuación máxima de cresta a cresta en la entrega de paquetes de audio a la capa de transporte que introducirá el transmisor. Se mide en milisegundos.

receiveMultipointCapability indica las capacidades de recepción de un terminal en una conferencia multipunto.

transmitMultipointCapability indica las capacidades de transmisión de un terminal en una conferencia multipunto.

receiveAndTransmitMultipointCapability indica las capacidades de recepción y transmisión de un terminal en una conferencia multipunto.

mcCapability indica la aptitud de un terminal para actuar como un MC en una conferencia centralizada o distribuida.

rtcpVideoControlCapability indica la aptitud de un terminal para procesar mensajes de petición interna completa (FIR, *full intra request*) y acuse de recibo negativo (NACK, *negative acknowledgement*) del RTCP.

MediaPacketizationCapability indica qué esquema facultativo de paquetización de medios soportan los puntos extremos.

h261aVideoPacketization indica que se está utilizando el formato de cabida útil RTP alternativo de H261, descrito en la Rec. UIT-T H.225.0.

rtpPayloadType indica los esquemas de paquetización de la cabida útil RTP soportados por el punto extremo, como sigue.

payloadDescriptor identifica las semánticas asociadas con el payloadType: si se elige el rfc-number, indica el documento oficial del Grupo de tareas especiales de ingeniería en Internet (IETF, *Internet engineering task force*) en que se define el formato de la cabida útil; no se deberá hacer referencia aquí a los RFC obsoletos. Si se elige el componente identificador de objeto (oid, *object identifier*), se identifica un formato de cabida útil especificado como parte de una Recomendación definida por la UIT o una Norma Internacional definida por la ISO y registrada en el documento respectivo bajo ese identificador de objeto. Lo anterior es aplicable igualmente al intercambio de capacidades y a la apertura de canales lógicos. El payloadDescriptor deberá completarse del modo siguiente:

- 1) si la Rec. UIT-T H.225.0 especifica que para el códec se ha de utilizar un identificador de objeto o rfc-number, se deberán seguir las indicaciones de dicha Recomendación. De lo contrario,
- 2) si en la Recomendación sobre códec del UIT-T se describe un identificador de objeto para el códec, deberá utilizarse ese identificador de objeto. De lo contrario,
- 3) si el códec es definido por una Recomendación del UIT-T (sin un identificador de objeto explícito), deberá utilizarse el componente identificador de objeto, que será el del número de la Rec. UIT-T, como sigue: {itu-t(0) recommendation (0) <letter> (<letter>) <number>}. Por ejemplo, la Rec. UIT-T G.711 podría utilizar el identificador de objeto {itu-t (0) recommendation (0) g (7) 711}. De lo contrario,
- 4) si existe un RFC que define la paquetización de códec, deberá utilizarse el componente rfc-number. De lo contrario,
- 5) deberá utilizarse el componente nonStandardIdentifier.

Se hallarán más identificaciones del tipo de cabida útil (modos opcionales, versiones, velocidades binarias, etc., si existen) en las estructuras `DataType` de `OpenLogicalChannel`. Los decodificadores H.245 deberán reconocer el identificador o los identificadores de objeto mencionados *supra* además de cualquier rfc-number definido para el códec.

`payloadType` se puede incluir para identificar el tipo de cabida útil asociado con este formato. Si se utiliza en el intercambio de capacidades, el `payloadType` deberá fijarse a un tipo de cabida útil asignado estáticamente, si es que existe alguno para este formato de cabida útil. De no ser así, deberá omitirse el `payloadType`. Si se utiliza junto con `OpenLogicalChannel`, el `payloadType` deberá indicar el valor de tipo de cabida útil RTP que se ha de utilizar (estático o dinámico), independientemente de cualquier tipo de cabida útil asignado estáticamente. Cabe señalar que si el valor de tipo de cabida útil se halla en la gama 96..127, el valor idéntico también deberá colocarse en `h2250logicalChannelParameters.dynamicRTPPayloadType`.

`TransportCapability` indica capacidades de transporte facultativas, tales como las de calidad de servicio y de tipo de canal de medio.

`redundancyEncodingCapability` indica los modos de codificación por redundancia que son soportados (si lo es alguno). Para cada entrada de capacidad, el `redundancyEncodingMethod` especifica el tipo de codificación que se ha de utilizar, la codificación primaria y las codificaciones secundarias que se soportan para esa codificación primaria. La elección de los esquemas de codificación depende del modo seleccionado. `rtpAudioRedundancyEncoding` se refiere a la codificación por redundancia de audio; si este modo es el `redundancyEncodingMethod` seleccionado, sólo son válidos los `CapabilityEntryNumbers` que se refieran a codificaciones audio. `rtpH263VideoRedundancyEncoding` indica que es posible la codificación por redundancia de vídeo de acuerdo con H.263 + anexo N o que deberá abrirse un canal lógico utilizando la codificación por redundancia de vídeo. Se proporcionan los parámetros adicionales que se indican a continuación:

`numberOfThreads`, que indica el número máximo de hilos que soporta el emisor/receptor cuando se utilizan durante el intercambio de capacidad; contiene el número efectivo de hilos de un tren específico cuando se abre un canal lógico.

`framesBetweenSyncPoints`, que define el número máximo de tramas de vídeo que se pueden transmitir (sumadas las de todos los hilos) entre dos puntos de sincronización de todos los hilos durante el intercambio de capacidad; define el número efectivo de tramas de un tren específico para `OpenLogicalChannel`.

`frameToThreadMapping`, que define los modos soportados por un emisor/receptor durante el intercambio de capacidad y el nodo que se ha de utilizar cuando se abre un canal lógico: "round-robin" indica que las tramas se asignan de manera consecutiva a los hilos, asignándose la primera trama después de un punto de sincronización al hilo 0, la segunda al hilo 1 y así sucesivamente. El formato "personalizado" permite especificar correspondencias arbitrarias entre tramas e hilos; durante el intercambio de capacidad, el soporte del formato personalizado se indica eligiendo este componente y codificando una SEQUENCE cualquiera (posiblemente vacía). El soporte de formatos personalizados implica el soporte de correspondencias "round-robin" (consecutivas).

`containedThreads`, que se aplica solamente a instrucciones con las que se abren canales lógicos: este parámetro indica a continuación cuál de los hilos se transmite por el canal lógico que se va a abrir. Un canal lógico puede contener cualquier número de hilos entre 1 y 15, no obstante, no deberá especificarse que dos canales lógicos contengan el mismo hilo.

En el caso de `rtpH263VideoRedundancyEncoding`, el parámetro `secondaryEncoding` no deberá estar presente, lo que también es aplicable a los `H2250ModeParameters` y a las estructuras ASN.1 de `RedundancyEncoding` de la Rec. UIT-T H.245.

Cuando se abra un canal lógico para codificación por redundancia de vídeo, deberá abrirse primero el canal lógico que contenga el hilo 0 y a este canal lógico deberán hacer referencia todos los demás canales lógicos por medio del parámetro `forwardLogicalChannelDependency` en la instrucción `OpenLogicalChannel`.

`LogicalChannelSwitchingCapability` indica la capacidad de un receptor de conmutar al tren (por ejemplo el canal lógico) que esté siendo entregado en base a las instrucciones `switchReceiveMedia` de activación y desactivación.

`t120DynamicPortCapability` indica que el punto extremo puede efectuar una llamada T.120 [32] a una dirección de transporte dinámica en vez de a la dirección normalizada de puerto bien conocida definida en la Rec. UIT-T T.123 [33].

**MultipointCapability:** indica capacidades de un terminal específicas de la transmisión multipunto.

`multicastCapability` indica la aptitud de un terminal para difundir tráfico de audio o vídeo.

`multiUnitCastConference` indica la aptitud de un terminal para participar en una conferencia `multiUnitCast`.

**MediaDistributionCapability:** indica las capacidades de un terminal para transmisión y recepción de medios en una conferencia multipunto. Control y audio centralizados serán TRUE para terminales H.323. Si se soporta vídeo, vídeo centralizado será TRUE. Si se soporta T.120, la capacidad de aplicación de datos T.120 datos centralizados estará presente.

`Control`, `audio` y `vídeo centralizados` y `distribuidos` indican la aptitud de un terminal para participar en una conferencia con esos tipos de distribución de medios. `Datos centralizados` y `distribuidos` indica la aptitud de un terminal para participar en una conferencia con esos tipos de distribución de medios para el protocolo de aplicación de datos específico. `MediaDistributionCapability` es una secuencia que permite la definición de capacidades simultáneas (por ejemplo `audio centralizado` con `vídeo distribuido` o `vídeo centralizado` con `audio distribuido` o capacidades de datos específicas según un protocolo de aplicación de datos).

`QOSCapabilities` indica capacidades de calidad de servicio tales como los parámetros `RSVPPParameters` y `ATMPParameters`.

`mediaChannelCapabilities` indica los transportes por los que se puede llevar el canal de medio. `IP-UDP` indica que el punto extremo soporta el transporte del canal de medio por una capa de red IP y una capa de transporte UDP. `IP-TCP` indica que el punto extremo soporta el canal de medio por una capa de red IP y una capa de transporte TCP. `atm-AAL5-UNIDIR` indica que el punto extremo soporta el transporte del canal de medio por un circuito virtual unidireccional AAL5 del ATM. `atm-AAL5-BIDIR` indica que el punto extremo soporta el transporte del canal de medio por un circuito virtual bidireccional AAL5 del ATM.

`RSVPPParameters` indica información específica de parámetros sobre el protocolo RSVP.

`ATMPParameters` indica información específica de parámetros sobre un circuito virtual ATM.

`QosMode` indica si el modo es de calidad de servicio garantizada o de carga controlada en el que no se impone ningún límite superior al retardo de extremo a extremo.

**genericMultiplexCapability:** indica capacidades de múltiplex genéricas.

### **B.2.2.5 Capacidades vídeo**

Indica las capacidades de vídeo. La indicación de más de una capacidad en una única `VideoCapability` no implica la capacidad de procesamiento simultáneo. La capacidad de procesamiento simultáneo puede expresarse mediante ejemplares de `VideoCapability` en `AlternativeCapabilitySet` diferentes dentro de un único `CapabilityDescriptor`.



**ExtendedVideoCapability:** indica capacidades vídeo con una secuencia de estructuras GenericCapability asociadas.

videoCapability indica una secuencia de capacidades vídeo; cualesquiera de las capacidades VideoCapability se pueden utilizar con la videoCapabilityExtension indicada.

videoCapabilityExtension, si está presente, indica una secuencia de estructuras GenericCapability asociadas con videoCapability.

La secuencia de las estructuras VideoCapability no debe contener una ExtendedVideoCapability.

Cuando se utiliza en un mensaje OpenLogicalChannel, ExtendedVideoCapability.videoCapability contendrá exactamente una VideoCapability.

**H261VideoCapability:** indica capacidades H.261 [18].

Si está presente, qcifMPI indica el intervalo de imagen mínimo en unidades 1/29,97 para la codificación y/o decodificación de imágenes QCIF y, si no está presente, no se indica ninguna capacidad de QCIF.

Si está presente, cifMPI indica el intervalo de imagen mínimo en unidades 1/29,97 para la codificación y/o decodificación de imágenes CIF y, si no está presente, no se indica ninguna capacidad de CIF.

La variable booleana temporalSpatialTradeOffCapability, cuando toma el valor verdadero, indica la aptitud del codificador para variar sus posibilidades de compromiso entre resolución temporal y espacial según lo ordene el terminal distante. Carece de significado cuando forme parte de una capacidad de recepción.

maxBitRate indica la velocidad binaria máxima en unidades de 100 bit/s a que un transmisor puede transmitir vídeo o un receptor puede recibir vídeo.

stillImageTransmission indica la capacidad para imágenes fijas como se especifica en el anexo D/H.261.

videoBadMBsCap, cuando es verdadera, indica la capacidad de un codificador para recibir o de un decodificador para transmitir la instrucción videoBadMBs. Cuando es parte de la capacidad de transmisión, indica la aptitud del codificador para procesar instrucciones videoBadMBs y efectuar las medidas correctivas apropiadas para obtener la recuperación de la calidad vídeo. Cuando es parte de la capacidad de recepción, indica la aptitud del decodificador para enviar las indicaciones videoBadMBs apropiadas.

**H262VideoCapability:** expresa capacidades H.262 [19].

La lista de variables booleanas indica la capacidad de procesamiento de niveles y perfiles determinados: el valor verdadero expresa que esa operación es posible, en tanto que el valor falso indica la imposibilidad de tal operación. Todo codificador deberá producir trenes de bits conformes con las especificaciones de nivel y perfil para los que haya expresado capacidad y con las limitaciones impuestas por los campos facultativos (véase más adelante). Todo decodificador deberá ser capaz de aceptar todos los trenes de bits conformes con el perfil y nivel para los que haya expresado su capacidad, siempre que cumplan las limitaciones indicadas por los campos facultativos. Tales campos facultativos contienen números enteros en las unidades definidas en el cuadro B.2.

videoBadMBsCap se utiliza en H262VideoCapability de la misma manera como se utiliza en H261VideoCapability.

**Cuadro B.2/H.245 – Unidades para los puntos de código H.262**

<b>Punto de código ASN.1</b>	<b>Unidades del parámetro referenciado</b>
videoBitRate	400 bit/s
vbvBufferSize	16 384 bits
samplesPerLine	muestras por línea
linesPerFrame	líneas por trama
framesPerSecond	índice, código de velocidad de trama en el cuadro 6-4/H.262
luminanceSampleRate	muestras por segundo

**H263VideoCapability:** indica capacidades H.263 [20].

Si está presente, sqcifMPI indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o decodificación de imágenes SQCIF y, si no está presente, indica que no hay capacidad de imágenes SQCIF.

Si está presente, qcifMPI indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o decodificación de imágenes QCIF y, si no está presente, indica que no hay capacidad de imágenes QCIF.

Si está presente, cifMPI indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o decodificación de imágenes CIF y, si no está presente, indica que no hay capacidad de imágenes CIF.

Si está presente, cif4MPI indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o decodificación de imágenes 4 veces CIF y, si no está presente, indica que no hay capacidad de imágenes 4 veces CIF.

Si está presente, cif16MPI indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o decodificación de imágenes 16 veces CIF y, si no está presente, indica que no hay capacidad de imágenes 16 veces CIF.

maxBitRate indica la velocidad binaria máxima, en unidades de 100 bit/s, a la que el transmisor puede transmitir vídeo o el receptor recibir vídeo.

Las variables booleanas unrestrictedVector (anexo D/H.263), arithmeticCoding (anexo E/H.263), advancedPrediction (anexo F/H.263) y pbFrames (anexo G/H.263), cuando toman el valor verdadero, indican la capacidad de transmitir y/o recibir esos modos facultativos definidos en los anexos de la Rec. UIT-T H.263.

La variable booleana temporalSpatialTradeOffCapability, cuando toma el valor verdadero, indica que el codificador puede variar sus posibilidades de compromiso entre resolución espacial y temporal según lo indique el terminal distante. Carece de significado cuando forme parte de una capacidad de recepción.

El número entero hrd-B, cuando está presente, indica el parámetro B de HRD del anexo B/H.263 y se mide en unidades de 128 bits. Cuando no está presente, se aplica el valor supletorio definido en el anexo B/H.263. Es una capacidad del receptor y no tiene significado alguno en conjuntos de capacidades de transmisión.

El número entero bppMaxKb, cuando está presente, indica el número máximo de bits para una imagen codificada que el receptor puede recibir y decodificar correctamente, y se mide en unidades de 1024 bits. Cuando no está presente, se aplica el valor supletorio definido en H.263. Es una capacidad del receptor y no tiene significado alguno en conjuntos de capacidades de transmisión.

Las siguientes capacidades están previstas para ser utilizadas en ciertas aplicaciones de muy bajas velocidades de trama, como las aplicaciones para vigilancia:

Si está presente, `slowSqcifMPI` indica el intervalo mínimo de imagen, en unidades de segundos por trama, para la codificación y/o decodificación de imágenes SQCIF. Si no está presente y tampoco lo está `sqcifMPI`, no se indica ninguna capacidad de imágenes SQCIF. Si `sqcifMPI` está presente, `slowSqcifMPI` no podrá estar presente.

Si está presente, `slowQcifMPI` indica el intervalo mínimo de imagen, en unidades de segundos por trama, para la codificación y/o decodificación de imágenes QCIF. Si no está presente y `qcifMPI` tampoco lo está, no se indica capacidad de imágenes QCIF. Si `qcifMPI` está presente, `slowQcifMPI` no podrá estar presente.

Si está presente, `slowCifMPI` indica el intervalo mínimo de imagen, en unidades de segundos por trama, para la codificación y/o decodificación de imágenes CIF. Si no está presente y tampoco lo está `cifMPI`, no se indica capacidad para imágenes CIF. Si `cifMPI` está presente, `slowCifMPI` no podrá estar presente.

Si está presente, `slowCif4MPI` indica el intervalo mínimo de imagen, en unidades de segundos por trama, para la codificación y/o decodificación en imágenes 4 veces CIF. Si no está presente y tampoco lo está `cif4MPI`, no se indica capacidades para imágenes 4 veces CIF. Si `cif4MPI` está presente, `slowCif4MPI` no podrá estar presente.

Si está presente, `slowCif16MPI` indica el intervalo mínimo de imagen, en unidades por segundos por trama, para la codificación y/o decodificación de imágenes 16 veces CIF. Si no está presente y tampoco lo está `cif16MPI`, no se indica capacidad para imágenes 16 veces CIF. Si `cif16MPI` está presente, `slowCif16MPI` no podrá estar presente.

Son aplicables los valores de MPI cuando se utilizan todos los modos facultativos para los que se haya formulado capacidad, así como cuando se emplee cualquier combinación de los mismos. Cuando no se utilicen algunas opciones, todo terminal podrá indicar la capacidad de un MPI menor transmitiendo otro `VideoCapability` que incluya este MPI menor e indicando el conjunto reducido de opciones.

La variable booleana `errorCompensation`, cuando toma el valor verdadero, indica la capacidad para transmitir y/o recibir información de retroalimentación para la compensación de error como se ilustra en el apéndice I/H.263. Cuando forma parte de una capacidad de transmisión, indica la aptitud del codificador para procesar indicaciones `videoNotDecodedMB` y compensar errores. Cuando forma parte de una capacidad de recepción, indica la aptitud del decodificador para identificar MB erróneos, tratarlos como no codificados y enviar indicaciones `videoNotDecodedMB` apropiadas.

Si está presente, `enhancementLayerInfo` indica la capacidad del codificador de transmitir, o del decodificador de recibir, trenes de bits con el modo escalabilidad opcional (anexo O/H.263). `enhancementLayerInfo` es una secuencia que indica los parámetros de la configuración del modo escalabilidad.

Si está presente, `H263Options` indica la capacidad para modos opcionales de la Rec. UIT-T H.263.

**EnhancementLayerInfo:** indica capacidad para el modo escalabilidad de la Rec. UIT-T H.263.

`baseBitRateConstrained` indica si la capa básica está constreñida a no exceder la velocidad binaria máxima de la capacidad de vídeo menos la suma de la velocidad binaria máxima de cada una de las opciones de mejora.

Cuando está presente, `snrEnhancement` indica la presencia de una capacidad de capa de mejora snr. El tamaño fijado indica el número de capas `snrEnhancement` que el terminal es capaz de soportar dentro de un solo canal lógico.

Cuando está presente, `spatialEnhancement` indica la presencia de una capacidad de capa de mejora espacial. Un tren de bits de capa de mejora contiene un tamaño de imagen con el doble de anchura o el doble de altura, o el doble de anchura y altura que la imagen en la capa a la que hace referencia.

Para que un terminal tenga capacidad de mejora espacial en una dimensión (anchura o altura), el terminal debe indicar también la capacidad de soportar el formato de imagen personalizado asociado que se requiere en la capa de mejora. El tamaño fijado indica el número de capas spatialEnhancement que el terminal puede soportar dentro de un solo canal lógico.

Cuando está presente, bPictureEnhancement indica la presencia de una capacidad de capa de mejora de imágenes B. El tamaño fijado indica el número de capas bPictureEnhancement que el terminal puede soportar dentro de un solo canal lógico.

EnhancementOptions dentro de la secuencia bPictureEnhancement indica las opciones adicionales que puede transmitir un codificador o recibir un decodificador en las imágenes B.

numberOfBPictures indica el número máximo de imágenes B que el terminal puede soportar entre pares sucesivos de imágenes de referencia de anclaje utilizadas en la predicción de las imágenes B. Por ejemplo, si es igual a 2, se pueden enviar dos imágenes B entre cada par de imágenes P o de otras imágenes de anclaje.

**EnhancementOptions:** indica capacidades de capa de mejora de escalabilidad.

Los parámetros de EnhancementOptions tienen las mismas definiciones semánticas que los parámetros del mismo nombre de H263VideoCapability.

**H263Options:** indica capacidad de modos opcionales adicionales de la Rec. UIT-T H.263.

advancedIntraCodingMode, cuando toma el valor verdadero, indica la capacidad de transmitir o recibir el modo de codificación INTRA avanzada del anexo I/H.263.

deblockingFilterMode, cuando toma el valor verdadero, indica la capacidad de transmitir o recibir el modo filtro de desbloqueo del anexo J/H.263.

improvedPBFramesMode, cuando toma el valor verdadero, indica la capacidad de transmitir o recibir el modo tramas PB mejoradas del anexo M/H.263.

unlimitedMotionVectors, cuando toma el valor verdadero, indica la capacidad del codificador o del decodificador de soportar una gama de vectores de movimiento sin límite si se indica también el modo vector de movimiento sin restricción (anexo D/H.263). unlimitedMotionVectors será FALSE si unrestrictedVector es FALSE en la misma H263VideoCapability o el mismo H263VideoMode.

fullPictureFreeze, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir las instrucciones de congelación de toda la imagen que se describen en el anexo L/H.263.

partialPictureFreezeAndRelease, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir las instrucciones de congelación y liberación de toda la imagen que se describen en el anexo L/H.263.

resizingPartPicFreezeAndRelease, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir las instrucciones de congelación y redimensionamiento de parte de la imagen y liberación descritas en el anexo L/H.263.

fullPictureSnapshot, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir instantáneas de toda la imagen del contenido de vídeo como se describe en el anexo L/H.263.

partialPictureSnapshot, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir instantáneas de parte de la imagen del contenido de vídeo como se describe en el anexo L/H.263.

videoSegmentTagging, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir rotulación de segmento de vídeo del contenido de vídeo como se describe en el anexo L/H.263.

progressiveRefinement, cuando toma el valor verdadero, indica la capacidad del codificador de enviar o del decodificador de recibir rotulación de refinamiento progresivo como se describe en el anexo L/H.263. Además, cuando sea verdadero, el codificador responderá a las instrucciones diversas de refinamiento progresivo doOneProgression doContinuousProgressions, doOneIndependentProgression, doContinuousIndependentProgressions, progressiveRefinementAbortOne y progressiveRefinementAbortContinuous. Además, el codificador insertará los rótulos de comienzo de segmento de refinamiento progresivo y los rótulos de final de segmento de refinamiento progresivo como se define en la especificación de la información sobre mejoras suplementarias del anexo L/H.263.

NOTA – La rotulación del refinamiento progresivo puede ser enviada por un codificador y recibida por un decodificador incluso cuando no se indique en una instrucción diversos.

dynamicPictureResizingByFour, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el submodo redimensionamiento por cuatro (con recorte) del modo repetición del muestreo de imagen de referencia del anexo P/H.263.

dynamicPictureResizingSixteenthPel, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el redimensionamiento de una imagen de referencia a cualquier anchura y altura utilizando el modo repetición del muestreo de imagen de referencia implícito del anexo P/H.263 (con recorte).

dynamicWarpingHalfPel, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el funcionamiento alabeo de imagen arbitrario en el modo repetición del muestreo de imagen de referencia del anexo P/H.263 (con cualquier modo de relleno) utilizando alabeo con exactitud de medio píxel.

dynamicWarpingSixteenthPel, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el funcionamiento alabeo de imagen arbitrario en el modo repetición del muestreo de imagen de referencia anexo P/H.263 (con cualquier modo de relleno) utilizando alabeo con una exactitud de medio píxel o de un dieciseisavo de píxel.

Si DynamicPictureResizingSixteenthPel toma el valor verdadero, también tomará el valor verdadero DynamicPictureResizingByFour. Si DynamicWarpingSixteenthPel toma el valor verdadero, también tomarán el valor verdadero DynamicWarpingHalfPel, DynamicPictureResizingByFour y DynamicPictureResizingSixteenthPel.

La declaración de la capacidad dynamicPictureResizingByFour con un tamaño de imagen dado, denominado tamaño de imagen originaria, conlleva la capacidad de soportar otros dos tamaños de imagen, denominados tamaños de imagen derivada. Si el tamaño de imagen originaria tiene anchura de imagen  $W$  y altura de imagen  $H$ , los tamaños de imagen derivada soportados deberán tener anchura de imagen  $W/2$  y altura de imagen  $H/2$ , y anchura de imagen  $W/4$  y altura de imagen  $H/4$ , y estar sujetos a la siguiente restricción: cada tamaño de imagen derivada será soportado siempre que su anchura de imagen no sea inferior a 128 y su altura de imagen inferior a 96 (128 y 96 son, respectivamente, la anchura y altura de imagen del formato SQCIF). Los tamaños de imagen derivada deberán ser soportados con los mismos modos opcionales, el mismo MPI (Intervalo de Imagen Mínimo) y la misma frecuencia de reloj admitidos con el tamaño de imagen originaria.

independentSegmentDecoding, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el modo codificación de segmento independiente del anexo R/H.263.

slicesInOrder-NonRect, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el submodo del modo estructura en rebanada del anexo K/H.263, en el que las rebanadas transmiten en orden y contienen macrobloques en orden de barrido de la imagen.

slicesInOrder-Rect, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el submodo del modo estructura en rebanada del anexo K/H.263, en el que las rebanadas se transmiten en orden y cada rebanada ocupa una parte rectangular de la imagen.

slicesNoOrder-NonRect, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el submodo del modo estructura en rebanada del anexo K/H.263, en el que las rebanadas contienen macrobloques en orden de barrido de la imagen y no es preciso que se transmitan en orden.

slicesNoOrder-Rect, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el submodo del modo estructura en rebanada del anexo K/H.263, en el que las rebanadas ocupan una parte rectangular de la imagen y no es preciso que se transmitan en orden.

alternateInterVLCMode, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el modo códigos de longitud variable (VLC, *variable length code*) Inter alternativos del anexo S/H.263.

modifiedQuantizationMode, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el modo cuantificación modificada del anexo T/H.263.

reducedResolutionUpdate, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de soportar el modo actualización de resolución reducida definido en el anexo Q/H.263.

videoBadMBsCap se utiliza en H263VideoCapability de la misma manera como se utiliza en H261VideoCapability.

dataPartitionedSlices, cuando es verdadero, indica la capacidad de un codificador o decodificador para soportar el modo rebanada partida de datos definido en el anexo V/H.263. dataPartitionedSlices será falso si slicesInOrder-NonRect y slicesInOrder-Rect, slicesNoOrder-NonRect y slicesNoOrder-Rect son todos falsos en el mismo mensaje H263Options.

fixedPointIDCT0, cuando es verdadero, indica la capacidad de un codificador o decodificador para soportar referencia IDCT 0 definida en el anexo W/H.263.

interlacedFields, cuando es verdadero, indica la capacidad de un codificador o decodificador para soportar codificación de campo entrelazado como se define en el anexo W/H.263.

currentPictureHeaderRepetition, cuando es verdadero, indica la capacidad de un codificador o de un decodificador para soportar la repetición del encabezamiento de imagen actual como se define en el anexo W/H.263.

previousPictureHeaderRepetition, cuando es verdadero, indica la capacidad de un codificador o decodificador para soportar la repetición del encabezamiento de imagen previa como se define en el anexo W/H.263.

nextPictureHeaderRepetition cuando es verdadero, indica la capacidad de un codificador o decodificador para soportar la repetición del encabezamiento de imagen siguiente (con o sin indicación de referencia temporal fiable) como se define en al anexo W/H.263.

currentPictureHeaderRepetition, previousPictureHeaderRepetition y nextPictureHeaderRepetition, cuando son verdaderos y cuando son parte de las capacidades del receptor, indican que un decodificador se puede recuperar de una corrupción de encabezamiento de imagen o pérdida reemplazando el encabezamiento de imagen corrupto o perdido por un encabezamiento de imagen transmitido conforme al anexo W/H.263.

pictureNumber, cuando es verdadero, indica la capacidad de un codificador para transmitir números de imagen conforme al anexo W/H.263 o la capacidad de un decodificador para detectar pérdidas de imagen de referencia a partir de los números de imagen transmitidos.

spareReferencePictures, cuando es verdadero, indica la capacidad de un codificador para soportar la generación de indicaciones de imagen de referencia de reserva como se define en el anexo W/H.263 o la capacidad de un decodificador para utilizar una imagen de referencia de reserva si no dispone de la imagen de referencia real.

**TransparencyParameters:** indica parámetros que especifican una capa de vídeo transparente.

presentationOrder indica la estratificación de las capas vídeo transparentes. Durante el intercambio de capacidades, presentationOrder tomará el valor de 0, 1 y 2: si es 0, indica que se soporta el tipo de soporte de transparencia segundo plano de la imagen de referencia (RPB, *reference picture background*) definido en el anexo L/H.263; si es 1, indica que se puede utilizar una imagen de segundo plano controlada externamente; y si es 2, indica que el tren de bits puede especificar la utilización de transparencia de segundo plano de imagen de referencia o de un tipo de transparencia de imagen de segundo plano controlada externamente. Mientras el canal lógico está abierto, el valor INTEGER especifica el orden de presentación. Una capa con un orden de presentación más alto se situará por encima de una capa con un orden de presentación más bajo. El presentationOrder puede considerarse como un eje perpendicular a la pantalla en el sentido de parámetros crecientes hacia el observador.

offset-x y offset-y indican el desplazamiento de los píxeles, en octavos de píxel, de la capa transparente señalada con respecto a la capa básica, en unidades relativas a la capa básica. Cuando se utilizan en una capacidad, representan la capacidad de desplazar la ubicación de la capa de vídeo transparente y sus valores se deben limitar a 1, 2, 4 u 8 en unidades de 1/8 de píxel; por ejemplo, si el valor es 4, se indica la capacidad de desplazar la capa transparente en incrementos de 1/2 píxel.

scale-x y scale-y indican el factor de ajuste de escala que se ha de aplicar en las coordenadas correspondientes x e y a la capa transparente señalada antes de la estructuración en capas del vídeo en unidades relativas a la capa básica. En un mensaje de capacidad, indican el factor de escala máximo que se puede aplicar: 1 indica que no se soporta el reescalamiento, 2 indica que se puede duplicar el tamaño de la capa o mantenerlo sin escalamiento, 3 indica que se puede duplicarlo, triplicarlo o mantenerlo sin escalamiento y así sucesivamente.

La variable booleana **separateVideoBackChannel** indica, cuando toma el valor verdadero, que el terminal puede soportar el modo canal lógico separado: ninguna otra capacidad de vídeo se indicará en la misma H263VideoCapability, no habrá valores MPI presentes y ninguna otra bandera de modo o contenido tendrá significado y serán todos ellos falsos o estarán ausentes. Cuando se envíe en un mensaje de petición de modo, separateVideoBackChannel = verdadero, el envío se hará como si fuese la única capacidad de vídeo en ese H263VideoMode e indicará que el receptor desea recibir un canal que contenga solamente datos de canal de retorno H.263. Si está presente en el mensaje OpenLogicalChannel, indica que el canal lógico está destinado solamente a mensajes de canal de retorno de vídeo y que por ese canal lógico no se deberá entregar ningún otro tren de bits de vídeo H.263.

**refPictureSelection:** indica la capacidad del modo selección de la imagen de referencia del anexo N/H.263 y, facultativamente, la capacidad del modo selección de imagen de referencia mejorada (anexo U/H.263).

Si está presente, additionalPictureMemory indica la presencia de un volumen adicional de memoria, además del que puede ser utilizado por un decodificador normal que no soporte el modo selección de la imagen de referencia. Si no está presente, ello indica que la información relativa al volumen adicional de memoria que el decodificador puede utilizar está a disposición de un codificador en el otro terminal. Si se señala en H263VideoMode, indica la presencia de un volumen adicional de memoria de imagen utilizado para decodificar.

sqcifAdditionalPictureMemory indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el decodificador tenga memoria adicional para almacenar el número indicado de imágenes de tamaño subcuarto de CIF, o de un tamaño menor

tanto en su dimensión horizontal como en su dimensión vertical, si se indica soporte de formato de imagen personalizado para esas imágenes en `customPictureFormat`.

`qcifAdditionalPictureMemory` indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el decodificador tenga memoria adicional para almacenar el número indicado de imágenes de tamaño cuarto de CIF, o de un tamaño menor tanto en su dimensión horizontal como en su dimensión vertical, si se indica soporte de formato de imagen personalizado para esas imágenes en `customPictureFormat`. El número de memorias de imagen indicado en `qcifAdditionalPictureMemory` no deberá ser mayor que el número de imágenes indicado en `sqcifAdditionalPictureMemory` (si está presente).

`cifAdditionalPictureMemory` indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el decodificador tenga memoria adicional para almacenar el número indicado de imágenes de tamaño CIF, o de un tamaño menor tanto en su dimensión horizontal como en su dimensión vertical, si se indica soporte de formato de imagen personalizado para esas imágenes en `customPictureFormat`. El número de memorias de imagen indicado en `cifAdditionalPictureMemory` no deberá ser mayor que el número de imágenes indicado en `sqcifAdditionalPictureMemory` o `qcifAdditionalPictureMemory` (si está presente).

`cif4AdditionalPictureMemory` indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el codificador tenga memoria adicional para almacenar el número indicado de imágenes de tamaño 4 veces CIF, o de un tamaño menor tanto en su dimensión horizontal como en su dimensión vertical, si se indica soporte de formato de imagen personalizado para esas imágenes en `customPictureFormat`. El número de memorias de imagen indicado en `cif4AdditionalPictureMemory` no deberá ser mayor que el número de imágenes indicado en `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory` o `cifAdditionalPictureMemory` (si está presente).

`cif16AdditionalPictureMemory` indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el decodificador tenga memoria adicional para almacenar el número indicado de imágenes de tamaño 16 veces CIF, o de un tamaño menor tanto en su dimensión horizontal como en su dimensión vertical, si se indica soporte de formato de imagen personalizado para esas imágenes en `customPictureFormat`. El número de memorias de imagen indicado en `cif16AdditionalPictureMemory` no deberá ser mayor que el número de imágenes indicado en `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory`, `cifAdditionalPictureMemory` o `cif4AdditionalPictureMemory` (si está presente).

`bigCpfAdditionalPictureMemory` indica que el codificador puede enviar o el decodificador puede recibir un tren de bits H.263 que requiere que el decodificador tenga memoria adicional para almacenar el número indicado de imágenes que tienen un formato de imagen personalizado cuyo tamaño se indica en `customPictureFormat` y es mayor que 16 veces CIF en su dimensión horizontal o en su dimensión vertical. El número de memorias de imagen indicado en `bigCpfAdditionalPictureMemory` no deberá ser mayor que el número de imágenes indicado en `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory` o `cifAdditionalPictureMemory`, `cif4AdditionalPictureMemory`, o `cif16AdditionalPictureMemory` (si está presente).

`videoMux` indica, durante el procedimiento de intercambio de capacidad, que el terminal puede soportar el modo múltiplex de vídeo ilustrado en el anexo N/H.263. Cuando toma el valor verdadero, el codificador o el decodificador pueden utilizar un tren de bits que contenga mensajes de canal de retorno de vídeo. Si se señala en `H263VideoMode`, indica que se prefiere la recepción de mensajes de canal de retorno de vídeo en modo múltiplex de vídeo. Cuando se utiliza en `H263VideoMode`, `videoMux` (múltiplex de vídeo) y `separateVideoBackChannel` no tomarán el valor verdadero.

`videoBackChannelSend` indica el tipo de mensaje de canal de retorno de vídeo que soporta el terminal. Si se señala en `H263VideoMode`, indica el tipo preferido de mensaje de canal de retorno.



none (ninguno) indica que el codificador no es capaz de enviar o el decodificador no es capaz de recibir un tren de bits H.263 que contiene peticiones de devolución de mensajes de canal de retorno.

ackMessageOnly indica que el codificador es capaz de enviar o el decodificador es capaz de recibir un tren de bits H.263 que contiene peticiones de devolución de mensajes de canal de retorno de acuse de recibo solamente.

nackMessageOnly indica que el codificador es capaz de enviar o el decodificador es capaz de recibir un tren de bits que contiene peticiones de devolución de mensajes de canal de retorno de acuse de recibo negativo solamente.

ackOrNackMessageOnly indica que el codificador es capaz de enviar o el decodificador es capaz de recibir un tren de bits H.263 que contiene peticiones de devolución de mensajes de canal de retorno de acuse de recibo o de acuse de recibo negativo, pero solamente uno para un determinado tren de bits de vídeo.

ackAndNackMessage indica que el codificador es capaz de enviar o el decodificador es capaz de recibir un tren de bits H.263 que contiene peticiones de devolución de mensajes de canal de retorno de acuse de recibo y de acuse de recibo negativo.

enhancedReferencePicSelect, cuando está presente, indica la capacidad del codificador o del decodificador de utilizar el modo selección de imagen de referencia mejorada del anexo U/H.263. Si el codificador es capaz de utilizar dicho modo, también deberá ser capaz de recibir los tres mensajes de instrucción diversos siguientes, a saber lostPicture, lostPartialPicture y recoveryReferencePicture, y de llevar a cabo las acciones necesarias para recuperar la calidad de las imágenes decodificadas en el extremo distante.

subPictureRemovalParameters, cuando está presente, indica la capacidad de eliminar la subimagen de la imagen de referencia utilizando el procedimiento del anexo U/H.263.

mpuHorizMBs indica la dimensión horizontal en macrobloques de la unidad de imagen mínima para la eliminación de la subimagen de la imagen de referencia utilizando el procedimiento del anexo U/H.263.

mpuVertMBs indica la dimensión vertical en macrobloques de la unidad de imagen mínima para la eliminación de la subimagen de la imagen de referencia utilizando el procedimiento del anexo U/H.263.

mpuTotalNumber indica la capacidad total de la memoria tampón multiimagen cuando funciona con eliminación de subimagen utilizando el procedimiento del anexo U/H.263 en unidades de imagen mínima.

**CustomPictureClockFrequency:** indica la capacidad de soportar frecuencia de reloj de imagen personalizada cuando esté presente como una capacidad y parámetros para la frecuencia de reloj de imagen personalizada cuando esté presente en OpenLogicalChannel y RequestMode.

Cuando se utiliza en OpenLogicalChannel, si customPictureClockFrequency tiene más de un miembro en su conjunto, se permite que el tren de bits de vídeo conmute entre las diversas frecuencias de reloj de imagen (PCF, *picture clock frequencies*) de ese conjunto dentro del mismo tren de bits de vídeo. Incluso cuando el conjunto sólo consta de una PCF, si se envían cualesquiera valores MPI para la PCF normalizada a niveles superiores en el mismo mensaje (por ejemplo, en la misma H263VideoCapability), puede producirse conmutación dentro del mismo tren de bits entre la PCF normalizada y la personalizada. Si se desea indicar que la PCF no debe cambiar dentro del tren de bits, deberán enviarse datos relacionados solamente con una PCF (ya sea valores MPI únicamente para la PCF normalizada o sólo la customPictureClockFrequency).

clockConversionCode indica el código de conversión de reloj cuando se utiliza frecuencia de reloj de imagen personalizada en los procedimientos de la Rec. UIT-T H.263.

clockDivisor indica la representación binaria natural del valor del divisor de reloj. La frecuencia de reloj de imagen personalizada viene dada por  $1\ 800\ 000/(\text{divisor de reloj} * \text{factor de conversión de reloj})$  Hz.

Si está presente, sqcifMPI indica el intervalo de imagen mínimo en unidades de  $1/(\text{frecuencia de reloj de imagen personalizada})$  para la codificación y/o decodificación de imágenes de subcuarto de CIF y, si no está presente, indica que no hay capacidad de imágenes de subcuarto de CIF.

Si está presente, qcifMPI indica el intervalo de imagen mínimo en unidades de  $1/(\text{frecuencia de reloj de imagen personalizada})$  para la codificación y/o decodificación de imágenes de cuarto de CIF y, si no está presente, indica que no hay capacidad de imágenes de cuarto de CIF.

Si está presente, cifMPI indica el intervalo de imagen mínimo en unidades de  $1/(\text{frecuencia de imagen de reloj personalizada})$  para la codificación y/o decodificación de CIF y, si no está presente, indica que no hay capacidad de imágenes de CIF.

Si está presente, cif4MPI indica el intervalo de imagen mínimo en unidades de  $1/(\text{frecuencia de reloj de imagen})$  para la codificación y/o decodificación de imágenes de 4 veces CIF y, si no está presente, indica que no hay capacidad de imágenes de 4 veces CIF.

Si está presente, cif16MPI indica el intervalo de imagen mínimo en unidades de  $1/(\text{frecuencia de reloj de imagen personalizada})$  para la codificación y/o decodificación de imágenes de 16 veces CIF y, si no está presente, indica que no hay capacidad de imágenes de 16 veces CIF.

**CustomPictureFormat:** indica la capacidad de soportar formato de imagen personalizado cuando está presente como una capacidad y parámetros para el formato de imagen personalizado cuando esté presente en OpenLogicalChannel y RequestMode.

Los parámetros maxCustomPictureWidth, maxCustomPictureHeight, minCustomPictureWidth y minCustomPictureHeight indican la gama de tamaños de imagen en unidades de 4 píxels que un codificador o un decodificador pueden soportar y el tamaño de imagen pedido en el caso en que se utilicen con RequestMode.

standardMPI expresa el intervalo de imagen mínimo en unidades de  $1/29,97$  cuando no se utiliza frecuencia de reloj de imagen personalizada.

customPCF indica los parámetros para la frecuencia de reloj de imagen personalizada cuando se utiliza junto con el formato de imagen personalizado.

clockConversionCode indica el código de conversión de reloj cuando se utiliza frecuencia de reloj de imagen personalizada en los procedimientos de la Rec. UIT-T H.263.

clockDivisor indica la representación binaria natural del valor del divisor de reloj. La frecuencia de reloj de imagen personalizada viene dada por  $1\ 800\ 000/(\text{divisor de reloj} * \text{factor de conversión de reloj})$  Hz.

customMPI indica el intervalo de imagen mínimo en unidades  $1/(\text{frecuencia de reloj de imagen personalizada})$  para la codificación y/o decodificación de imágenes con tamaño de formato de imagen personalizado.

pixelAspectInformation indica la capacidad de un codificador o de un decodificador de soportar diversas relaciones de aspecto de píxel y la relación de aspecto de píxel pedida en el caso en que se utilice con RequestMode.

pixelAspectCode indica la capacidad de soportar la relación de aspecto de píxel indicada por el código PAR de la Rec. UIT-T H.263.

extendedPAR: anchura y altura que indican la capacidad de soportar la relación de aspecto de píxel indicada por el código de relación de aspecto de píxel ampliada (EPAR, *extended pixel aspect ratio*) de la Rec. UIT-T H.263.

## H263VideoModeCombos

Cuando está presente, se utiliza `h263VideoModeCombos` para indicar dependencias entre modos opcionales de la Rec. UIT-T H.263. Las combinaciones de modos para las que se indican capacidades en `h263VideoModeCombos` quizás no puedan ser utilizadas con otros modos opcionales señalados a niveles superiores dentro del mismo mensaje `H263Options` o `H263VideoCapability` o `H263VideoMode`, salvo lo que se indica en el cuarto párrafo de esta cláusula y en el tercero de la siguiente. En otras palabras, si se indica soporte de algunas variables booleanas de modo para los mismos modos que contienen booleanas en `H263VideoModeCombos` a niveles superiores de la sintaxis en `h263Mode` o `H263Capability`, no por ello se supone que esos modos se aplican también en combinaciones no acopladas con los modos declarados en `H263VideoModeCombos`.

`h263VideoUncoupledModes` indica cuáles son los modos opcionales de funcionamiento H.263 que se pueden activar o desactivar, con independencia de unos con respecto a otros, de cualquier manera correcta sintácticamente para una imagen y cuáles se pueden activar o desactivar con independencia de los modos indicados en el mensaje `h263VideoCoupledModes` enviado en la misma secuencia `H263VideoModeCombos`.

`h263VideoCoupledModes` indica uno o más conjuntos de los modos opcionales de funcionamiento H.263 que se pueden activar o desactivar juntos para una imagen dentro de un tren de bits H.263, pero que no necesariamente tienen la posibilidad de activar o desactivar independientemente cualquier subconjunto de esos modos. Cualquier conjunto de modos que se indiquen acoplados en un mensaje `h263VideoCoupledModes` puede ser utilizado junto con el conjunto completo o cualquier subconjunto de los modos indicados como no acoplados en el mensaje `h263VideoUncoupledModes` acompañante dentro del mismo mensaje `H263VideoModeCombos`. En el contenido de cada mensaje `H263ModeComboFlags` de un mensaje `h263VideoCoupledModes` deberá haber por lo menos dos banderas booleanas fijadas a verdadero y no deberá haber ningún conjunto de banderas de modo fijadas a verdadero que indique una combinación acoplada de modos no permitida sintácticamente dentro de la misma imagen de un tren de bits H.263.

Algunas características facultativas de la Rec. UIT-T H.263 no están incluidas en `H263ModeComboFlags` porque se piensa que probablemente no requieran acoplamiento en la implementación. En concreto, se trata de las características especificadas en el anexo L/H.263 (por ejemplo, `fullPictureFreeze`, `partialPictureFreezeAndRelease` y `resizingPartPicFreezeAndRelease`) y los formatos de imagen facultativos y frecuencias de reloj de imagen facultativas. Si se señala soporte de cualquiera de estas características en un nivel superior dentro del mismo mensaje `H263Options` o `H263VideoCapability` o `H263VideoMode`, dichas características funcionarán de manera no acoplada con las combinaciones de modo señaladas en `H263VideoModeCombos`. A continuación se da un ejemplo bastante complejo de la utilización de `H263VideoModeCombos`.

El ejemplo se refiere al caso en que `H263VideoCapability` indica que se soportan `advancedPrediction` y `unrestrictedVector`, y (en el mismo mensaje `H263VideoCapability`) dentro de un mensaje `H263Options` se indica que se soporta `dynamicPictureResizingByFour` y (en el mismo mensaje `H263VideoCapability`) dentro de un mensaje `H263VideoModeCombos` hay un mensaje `h263VideoUncoupledModes` que indica que se soporta `advancedIntraCodingMode` de manera no acoplada junto con un mensaje `h263VideoCoupledModes` que indica que se soportan `modifiedQuantizationMode` y `slicesInOrder-NonRect` de manera acoplada. Esto significa que el tren de bits de vídeo puede contener (solamente) imágenes con las siguientes combinaciones de modo: ninguno, `advancedPrediction`, `unrestrictedVector`, `dynamicPictureResizingByFour`, `advancedPrediction` con `unrestrictedVector`, `advancedPrediction` con `dynamicPictureResizingByFour`, `unrestrictedVector` con `dynamicPictureResizingByFour`, `advancedPrediction` con `unrestrictedVector` con `dynamicPictureResizingByFour`,

advancedIntraCodingMode, modifiedQuantizationMode con slicesInOrder-NonRect y advancedIntraCodingMode con modifiedQuantizationMode con slicesInOrder-NonRect.

### H263ModeComboFlags

Cada uno de los parámetros de H263ModeComboFlags tiene el mismo significado que el parámetro del mismo nombre de H263VideoCapability y H263Options.

unlimitedMotionVectors deberá ser FALSE si unrestrictedVector es FALSE en el mismo mensaje H263VideoUncoupledModes. unlimitedMotionVectors deberá ser FALSE si unrestrictedVector es FALSE en el mismo mensaje H263VideoCoupledModes y en el mensaje H263VideoUncoupledModes del mismo mensaje H263VideoModeCombos.

referencePicSelect, cuando toma el valor verdadero, indica la capacidad del codificador o del decodificador de utilizar el modo selección de la imagen de referencia de la Rec. UIT-T H.263. Cuando es verdadero, los parámetros que en concreto especifican cómo se puede utilizar el modo selección de la imagen de referencia deberán ser tal como se envían en el campo refPictureSelection del mismo mensaje H263Options. referencePicSelect no deberá tomar el valor verdadero si el campo refPictureSelection está presente en el mismo mensaje H263Options.

enhancedReferencePicSelect deberá ser FALSE si referencePicSelect es FALSE en el mismo mensaje H263VideoUncoupledModes. El campo enhancedReferencePicSelect deberá ser FALSE si referencePicSelect es FALSE en el mismo mensaje H263VideoCoupledModes y en el mensaje H263VideoUncoupledModes en el mismo mensaje H263VideoModeCombos.

dataPartitionedSlices deberá ser FALSE si slicesInOrder-NonRect y slicesInOrder-Rect y slicesNoOrder-NonRect y slicesNoOrder-Rect son todos FALSE en el mismo mensaje H263VideoUncoupledModes. El campo dataPartitionedSlices deberá ser FALSE si slicesInOrder-NonRect y slicesInOrder-Rect y slicesNoOrder-NonRect y slicesNoOrder-Rect son todos FALSE en el mismo mensaje H263VideoUncoupledModes y en el mensaje H.263VideoUncoupledModes en el mismo mensaje H263VideoModeCombos.

**IS11172 VideoCapability:** indica capacidades IS11172 [44].

constrainedBitstream indica la capacidad para trenes de bits en los cuales la bandera constrained\_parameters está fijada a "1": el valor verdadero indica que tal operación es posible y el valor falso indica que tal operación no es posible. Un codificador deberá producir trenes de bits dentro de las limitaciones impuestas por los campos facultativos (véase más adelante). Un decodificador deberá poder aceptar todos los trenes de bits dentro de las limitaciones indicadas por los campos facultativos. Los campos facultativos contienen números enteros en las unidades definidas en el cuadro B.3.

videoBadMBsCap se utiliza en IS11172VideoCapability de la misma manera como se utiliza en H261VideoCapability.

**Cuadro B.3/H.245 – Unidades para puntos de código IS11172-2**

Punto de código ASN.1	Unidades para los parámetros referenciados
videoBitRate	400 bit/s
vbvBufferSize	16 384 bits
samplesPerLine	muestras por línea
linesPerFrame	líneas por trama
pictureRate	véase la sección 2.4.3.2 de IS11172-2
luminanceSampleRate	muestras por segundo

**genericVideoCapability**: indica capacidades vídeo genéricas.

### B.2.2.6 Capacidades audio

Este parámetro expresa las capacidades audio. La indicación de más de una capacidad dentro de un AudioCapability único no implica una capacidad de procesamiento simultánea. Tal capacidad puede expresarse mediante ejemplares de AudioCapability en AlternativeCapabilitySet diferentes en un único CapabilityDescriptor.

La capacidad de transmitir y/o recibir audio según la serie G se expresa mediante una selección de enteros. Cuando se utilice un múltiplex H.222.1, esos números se refieren al tamaño de la memoria tampón STD disponible, en unidades de 256 octetos. Cuando se emplee un múltiplex H.223, tales números hacen referencia al número máximo de tramas de audio por cada AL-SDU. Cuando se utiliza el múltiplex H.225.0 estos números indican el número máximo de tramas de audio por paquete: un punto extremo soportará la recepción de cualquier número de tramas por paquete hasta el número máximo inclusive indicado en la AudioCapability; además, el punto extremo no transmitirá más tramas por paquete que la que indica en su AudioCapability de transmisión. El significado exacto de los puntos de código se indica en el cuadro B.4.

**Cuadro B.4/H.245 – Puntos de código de audio de la serie G**

Punto de código ASN.1	Significación semántica del punto de código
g711Alaw64k	Audio G.711 a 64 kbit/s, ley A
g711Alaw56k	Audio G.711 a 56 kbit/s, ley A, truncado a 7 bits
g711Ulaw64k	Audio G.711 a 64 kbit/s, ley $\mu$
g711Ulaw56k	Audio G.711 a 56 kbit/s, ley $\mu$ , truncado a 7 bits
g722-64k	Audio G.722 7 kHz a 64 kbit/s
g722-56k	Audio G.722 7 kHz a 56 kbit/s
g722-48k	Audio G.722 7 kHz a 48 kbit/s
g7231	Audio G.723.1 a 5,3 ó 6,3 kbit/s
g728	Audio G.728 a 16 kbit/s
g729	Audio G.729 a 8 kbit/s
g729AnnexA	Audio anexo A/G.729 a 8 kbit/s
g729wAnnexB	Audio G.729 a 8 kbit/s con supresión de los periodos de silencio como en el anexo B
g729AnnexAwAnnexB	Audio anexo A/G.729 a 8 kbit/s con supresión de periodos de silencio como en el anexo B
g7231AnnexCCapability	G.723.1 con anexo C/G.723.1
gsmFullRate	Transcodificación de señales vocales a velocidad plena (GSM 06.10)
gsmHalfRate	Transcodificación de señales vocales a velocidad mitad (GSM 06.20)
gsmEnhancedFullRate	Transcodificación de señales vocales a velocidad plena intensificada (GSM 06.60)
g729Extensions	Ampliaciones de G.729

**G7231**: indica la aptitud para procesar códec con audio G.723.1. maxA1-sduAudioFrames indica el número máximo de tramas de audio por AL-SDU. La variable booleana silenceSuppression, cuando es verdadera, indica la capacidad para utilizar la compresión de periodos de silencio definida en el anexo A/G.723.1.

**G7231AnnexCCapability:** indica la aptitud para procesar códec con audio G.723.1, con su anexo C. maxAl-sduAudioFrames indica el número máximo de tramas de audio por AL-SDU. La variable booleana silenceSupression, cuando es verdadera, indica la capacidad para utilizar la compresión de periodos de silencio definida en el anexo A/G.723.1. g723AnnexCAudioMode no estará presente cuando G7231AnnexCCapability está incluida en un mensaje TerminalCapabilitySet, pero estará presente cuando G7231AnnexCCapability esté incluida en un mensaje OpenLogicalChannel. Los campos highRateMode0, highRateMode1, lowRateMode0, lowRateMode1, sidMode0 y sidMode1 indican el número de octetos por trama para cada uno de los modos de audio y protección contra errores de G.723.1 y anexo C/G.723.1 que se utilizarán en el canal lógico.

**IS11172AudioCapability:** indica la aptitud para el procesamiento de audio codificado según ISO/CEI 11172-3 [45].

Las variables booleanas que tengan el valor de verdadero indican que es posible ese modo concreto de funcionamiento. El valor falso indica que no es posible ese modo de funcionamiento. Las variables booleanas audioLayer1, audioLayer2 y audioLayer3, expresan las capas de codificación de audio que pueden procesarse. Las variables booleanas audioSampling32k, audioSampling44k1 y audioSampling48k, indican las velocidades de muestreo de audio, 32 kHz, 44,1 kHz y 48 kHz que pueden procesarse, respectivamente. Las variables booleanas singleChannel y twoChannels indican la capacidad de funcionamiento con un único canal y dos canales estéreo, respectivamente. El entero bitRate indica la máxima capacidad de velocidad de bits de audio y se mide en unidades de kbit/s.

**IS13818AudioCapability:** expresa la aptitud para el procesamiento de audio codificado según ISO/CEI 13818-3 [46].

Las variables booleanas que tengan el valor de verdadero indican que es posible este modo concreto de funcionamiento. El valor falso expresa que no es posible este modo de funcionamiento. Las variables booleanas audioLayer1, audioLayer2 y audioLayer3, indican las capas de codificación de audio que pueden procesarse. Las variables booleanas audioSampling16k, audioSampling22k05, audioSampling24k, audioSampling32k, audioSampling44k1 y audioSampling48k indican las velocidades de muestreo de audio, 16 kHz, 22,05 kHz, 24 kHz, 32 kHz, 44,1 kHz y 48 kHz respectivamente, que pueden procesarse.

Las variables booleanas relativas al funcionamiento multicanal indican la capacidad de funcionamiento en modos concretos, como se indica en el cuadro B.5.

**Cuadro B.5/H.245 – Puntos de código multicanal según ISO/CEI 13818-3**

<b>Punto de código ASN.1</b>	<b>Significación semántica del punto de código</b>
singleChannel	Un canal, utilizando la configuración 1/0. Modo canal único (como en ISO/CEI 11172-3)
twoChannels	Dos canales, utilizando la configuración 2/0. Modo canal estéreo o dual (como en ISO/CEI 11172-3)
threeChannels2-1	Tres canales, utilizando la configuración 2/1. Canal izquierdo, derecho y canal ambiente único
threeChannels3-0	Tres canales, utilizando la configuración 3/0. Canales izquierdo, central y derecho, sin canal ambiente
fourChannels2-0-2-0	Cuatro canales, utilizando la configuración 2/0 + 2/0. Canales izquierdo y derecho del primer programa e izquierdo y derecho del segundo programa
fourChannels2-2	Cuatro canales, utilizando la configuración 2/2. Canales izquierdo, derecho, izquierdo ambiente y derecho ambiente
fourChannels3-1	Cuatro canales, utilizando la configuración 3/1. Canales izquierdo, central, derecho y un único canal ambiente
fiveChannels3-0-2-0	Cinco canales, utilizando la configuración 3/0 + 2/0. Canales izquierdo, central y derecho del primer programa e izquierdo y derecho del segundo programa
fiveChannels3-2	Cinco canales, utilizando la configuración 3/2. Canales izquierdo, central, derecho, izquierdo ambiente y derecho ambiente

La variable booleana `lowFrequencyEnhancement` indica la capacidad de un canal de intensificación de baja frecuencia.

La variable booleana `multilingual`, cuando toma el valor verdadero, expresa la capacidad de soportar hasta siete canales multilingües. Cuando toma el valor falso expresa que no se soporta ningún canal multilingüe.

El entero `bitRate` indica la máxima capacidad de velocidad de bits de audio medida en unidades de kbit/s.

**GSMAudioCapability:** indica capacidades para los códecs de audio de transcodificación de señales vocales a velocidad plena, velocidad mitad y velocidad plena intensificada del sistema global para comunicaciones móviles (GSM, *global system for mobile communication*). `audioUnitSize` indica el número máximo de bytes que se han de enviar en cada paquete. `comfortNoise`, cuando toma el valor verdadero, indica la capacidad de soportar el procesamiento de ruido de confort para el canal de tráfico de señales vocales a velocidad plena, velocidad mitad o velocidad plena intensificada (GSM 06.12, GSM 06.22 y GSM 06.62) y aleatorizado, cuando toma el valor verdadero, indica la capacidad de soportar la aleatorización de bits para los canales de tráfico de señales vocales a velocidad plena, velocidad mitad o velocidad plena intensificada (GSM 06.10, GSM 06.20 y GSM 06.60).

**genericAudioCapability:** indica capacidades audio genéricas.

**g729Extensions:** indica la capacidad de modos facultativos adicionales de G.729. Este punto de código no deberá utilizarse para indicar `g729AnnexA`, `g729wAnnexB` y `g729AnnexAwAnnexB`, para lo cual deberán utilizarse `g729AnnexA`, `g729AnnexB` y `g729AnnexAwAnnexB`.

**audioUnit:**

- para un multiplex H.222, indica el tamaño de la memoria tampón STD en unidades de 256 octetos;
- para un multiplex H.223, indica el número máximo de tramas audio por AL-SDU; y

– para un múltiplex H.225.0, indica el número máximo de tramas audio por paquete.

**audioUnit** deberá estar presente para el intercambio de capacidades. Puede estar presente para la petición de modo.

**annexA**, cuando es verdadero, indica la capacidad para transmitir o recibir la señal de audio anexo A/G.729 en 8 kbit/s, en vez del texto principal G.729.

**annexB**, cuando es verdadero, indica la capacidad de supresión de periodos de silencio como en el anexo B/G.729.

**annexD**, cuando es verdadero, indica la capacidad de transmitir o recibir la señal de audio del anexo D/G.729 en 6,4 kbit/s.

**annexE**, cuando es verdadero, indica la capacidad de transmitir o recibir la señal de audio del anexo E/G.729 en 11,8 kbit/s.

**annexF**, cuando es verdadero, indica la capacidad de supresión de periodos de silencio como en el anexo F/G.729.

**annexG**, cuando es verdadero, indica la capacidad de supresión de periodos de silencio como en el anexo G/G.729.

**annexH**, cuando es verdadero, indica la capacidad de conmutación de operación entre 6,4 kbit/s (anexo D/G.729) y 11,8 kbit/s (anexo E/G.729).

**audioTelephonyEvent** puede estar incluido para indicar soporte de eventos de telefonía audio dentro de banda conforme a RFC 2833. Los eventos soportados se describirán en `audioTelephoneEvent` como se indica en la <list of value> de la sección 3.9 de RFC 2833. Los eventos 0-15 (correspondientes a los dígitos DTMF 0-9, \*, #, A, B, C, D) son los únicos eventos obligatorios.

**audioTone** se puede incluir para indicar soporte de tonos de audio dentro de banda según RFC 2833.

### **B.2.2.7 Capacidades de aplicación de datos**

Este parámetro expresa capacidades de datos. La indicación de más de una capacidad dentro de un solo `DataApplicationCapability` no implica una capacidad de procesamiento simultáneo. Tal capacidad puede expresarse mediante ejemplares de `DataApplicationCapability` en `AlternativeCapabilitySet` diferentes dentro de un único `CapabilityDescriptor`.

Las Recomendaciones que empleen esta Recomendación pueden imponer restricciones acerca de cuáles de estos modos pueden señalarse.

Algunas de las capacidades de datos requieren canales lógicos bidireccionales, por ejemplo para ejecutar un protocolo de retransmisión. Este requisito está incluido implícitamente en los puntos de código de capacidad apropiados.

**DataApplicationCapability**: lista de aplicaciones de datos. Cada aplicación de datos incluida en la lista deberá soportarse mediante uno o más `DataProtocolCapability`.

`maxBitRate` indica la velocidad binaria máxima, en unidades de 100 bit/s, a que un transmisor puede transmitir vídeo o un receptor puede recibir la aplicación de datos en cuestión.

`t120` indica la capacidad de soportar el protocolo T.120 [32].

`dsm-cc` indica la capacidad de soportar el protocolo DSM-CC [47].

`userData` indica la capacidad de soportar datos de usuario no especificados procedentes de puertos de datos externos.



t84 indica la capacidad de soportar la transferencia de imágenes de tipo T.84 [31] (JPEG, JBIG, Facsímil Gr.3/4).

t434 indica la capacidad de soportar la transferencia de ficheros binarios telemáticos T.434 [35].

h224 indica la capacidad de soportar el protocolo H.224 [11] de control de dispositivos simples en tiempo real.

nlpid indica la capacidad de soportar el protocolo de capa de red especificado mediante nlpidData, definido en ISO/CEI TR 9577 [52]. Estos protocolos incluyen, entre otros, el protocolo Internet (IP, *Internet protocol*) y el protocolo de punto a punto (PPP) del IETF.

NOTA – El empleo del NLPID se describe con detalle en RFC 1490 "Multiprotocol Interconnect over Frame Relay".

dsvdControl indica la capacidad del terminal DSVD para soportar un canal de control fuera de banda.

h222DataPartitioning indica la capacidad para soportar la utilización modificada y restringida de las particiones de datos H.262, tal como se especifican en la Rec. UIT-T H.222.1, en las cuales los datos de intensificación se transmiten como un canal de datos soportado por la DataProtocolCapability indicada.

t30fax: Este punto de código indica la capacidad de utilizar el modo analógico del anexo C/T.30 (G3V), especificado en la Rec. UIT-T T.39 para los modos DSVF/MSVF.

t140: Este punto de código indica la capacidad de utilizar el protocolo de conversación de texto T.140, especificado en la Rec. UIT-T T.140.

t38fax: Este punto de código indica un protocolo de datos conforme a la Rec. UIT-T T.38 [29].

Los campos **version**, **t38FaxRateManagement**, **t38FaxUdpOptions** y **t38FaxTcpOptions** se definen en la Rec. UIT-T T.38.

fillBitRemoval, cuando toma el valor verdadero, indica que la pasarela/el terminal puede eliminar e insertar bits de relleno.

transcodingJBIG, cuando toma el valor verdadero, indica que la pasarela puede transcodificar en tiempo real entre la comprensión de líneas y JBIG para transferencia por la red IP.

transcodingMMG, cuando toma el valor verdadero, indica que la pasarela puede transcodificar en tiempo real entre la compresión de líneas y MMG para transferir por la red IP.

genericDataCapability indica capacidades de datos genéricas. Cuando maxBitRate esté incluida en la genericDataCapability, su valor será el mismo que el de maxBitRate en DataApplicationCapability.

**DataProtocolCapability**: contiene una lista de protocolos de datos.

v14buffered indica la capacidad de soportar una aplicación de datos especificada empleando la Rec. UIT-T V.14 [36] con memoria tampón.

v42lapm indica la capacidad de soportar una aplicación de datos especificada que utilice el protocolo LAPM definido en la Rec. UIT-T V.42 [38].

hdlcFrameTunnelling indica la capacidad de soportar una aplicación de datos especificada que utilice el efecto túnel de tramas HDLC. Véase la sección 4.5.2 de ISO/CEI 13239 [43].

h310SeparateVCStack indica la capacidad para soportar una aplicación de datos especificada utilizando la pila de protocolos definida en la Rec. UIT-T H.310 para el transporte de mensajes H.245 a través de un canal virtual ATM VC distinto del utilizado para comunicación audiovisual.

h310SingleVCStack indica la capacidad para soportar una aplicación de datos especificada utilizando la pila de protocolos definida en la Rec. UIT-T H.310 para el transporte de

mensajes H.245 en el mismo canal virtual ATM VC que el utilizado para comunicación audiovisual.

transparent indica la capacidad para soportar una aplicación de datos especificada utilizando una transferencia de datos transparente.

v120: la utilización de v120 queda en estudio en la Rec. UIT-T H.323.

separateLANStack indica que se utilizará una pila de transporte distinta para transportar los datos. La intención de utilizar una conexión de red distinta para datos se indica por dataType en OpenLogicalChannel que tomará el valor h310SeparateVCStack o separateLANStack de DataProtocolCapability. Cuando la DataApplicationCapability seleccionada es t120, la elección de estos valores puede implicar la utilización del perfil básico T.123 para la red digital de servicios integrados de banda ancha y la LAN, respectivamente. Una nonStandardDataProtocolCapability puede seleccionar otros perfiles LAN diferentes.

Si se selecciona separateLANStack y separateStack está presente en la petición de OpenLogicalChannel, el receptor debe tratar de establecer la pila indicada. Responderá con OpenLogicalChannelAck si tiene éxito, de lo contrario responderá con OpenLogicalChannelReject señalando una causa adecuada.

Si se selecciona separateLANStack y separateStack no está presente en la petición de OpenLogicalChannel, el receptor deberá suministrar una separateStack apropiada en su respuesta de OpenLogicalChannelAck. El receptor de esta respuesta (el solicitante inicial) deberá tratar de establecer la pila indicada. Si fracasa, emitirá CloseLogicalChannel.

Si se selecciona separateLANStack y separateStack está presente en la petición OpenLogicalChannel, podrá ser contraordenada por separateStack en la respuesta OpenLogicalChannelAck. Si el solicitante inicial no soporta la contraorden, emitirá CloseLogicalChannel.

Si se selecciona separateLANStack y separateStack no está presente en la petición OpenLogicalChannel y tampoco está presente en la respuesta OpenLogicalChannelAck, el solicitante inicial puede llegar a la conclusión de que el respondedor no comprende estas extensiones ASN.1 y deberá emitir CloseLogicalChannel para despejar la situación.

v76wCompression indica la capacidad para soportar la compresión de datos en un canal de datos V.76.

tcp indica la capacidad de soportar TCP/IP para esta aplicación.

udp indica la capacidad de soportar UDP para esta aplicación.

**T84Profile:** indica el tipo de perfil de imagen fija que puede soportar el terminal.

t84Unrestricted no proporciona ninguna indicación del tipo de imagen fija T.84 que puede soportar el terminal: deberá utilizarse la información de capa T.84 para determinar si puede recibirse una imagen concreta.

t84Restricted indica el tipo de imagen fija T.84 que puede soportar el terminal.

qcif indica el soporte de una imagen de tipo YCrCb de color secuencial con resolución QCIF.

cif indica el soporte de una imagen de tipo YCrCb de color secuencial con resolución CIF.

ccir601Seq indica el soporte de una imagen de tipo YCrCb de color secuencial con resolución CCIR601.

ccir601Prog indica el soporte de una imagen de tipo YCrCb de color progresivo con resolución CCIR601.

hdtvSeq indica el soporte de una imagen de tipo YCrCb de color secuencial con resolución HDTV.

hdtvProg indica el soporte de una imagen de tipo YCrCb de color progresivo con resolución HDTV.

g3FacsMH200x100 indica el soporte de una imagen facsímil secuencial de grupo 3 MH (Modified Huffman) codificada en dos niveles con resolución normal ( $200 \times 100$  ppi).

g3FacsMH200x200 indica el soporte de una imagen facsímil secuencial de grupo 3 MH (Modified Huffman) codificada en dos niveles con alta resolución ( $200 \times 200$  ppi).

g4FacsMMR200x100 indica el soporte de una imagen facsímil secuencial de grupo 4 codificada con dos niveles MMR (Modified Modified Reed) con una resolución normal ( $200 \times 100$  ppi).

g4FacsMMR200x200 indica el soporte de una imagen facsímil secuencial de grupo 4 codificada con dos niveles MMR (Modified Modified Reed) con alta resolución ( $200 \times 200$  ppi).

jbig200x200Seq indica el soporte de una imagen secuencial de tipo JBIG de dos niveles, codificada en dos niveles con una resolución de  $200 \times 200$  ppi.

jbig200x200Prog indica el soporte de una imagen progresiva tipo JBIG de dos niveles, codificada en dos niveles con una resolución de  $200 \times 200$  ppi.

jbig300x300Seq indica el soporte de una imagen secuencial de tipo JBIG de dos niveles, codificada con dos niveles con una resolución de  $300 \times 300$  ppi.

jbig300x300Prog indica el soporte de una imagen progresiva de tipo JBIG de dos niveles, codificada con dos niveles con una resolución de  $300 \times 300$  ppi.

digPhotoLow indica el soporte de una imagen secuencial de color codificada según JPEG con un tamaño de imagen de hasta  $720 \times 576$ .

digPhotoMedSeq indica el soporte de una imagen de color secuencial codificada según JPEG con un tamaño de imagen de hasta  $1440 \times 1152$ .

digPhotoMedProg indica el soporte de una imagen de color progresiva codificada según JPEG con un tamaño de imagen de hasta  $1440 \times 1152$ .

digPhotoHighSeq indica el soporte de una imagen de color secuencial codificada según JPEG con un tamaño de imagen de hasta  $2880 \times 2304$ .

digPhotoHighProg indica el soporte de una imagen de color progresiva codificada según JPEG con un tamaño de imagen de hasta  $2880 \times 2304$ .

#### **B.2.2.8 Capacidades de criptación, autenticación e integridad**

EncryptionCapability, si está presente, indica las capacidades de criptación de un terminal para cada tipo de medio en el que están presentes las capacidades. El alcance de la criptación indica si la misma se aplica al tren de bits en su totalidad, a una porción del tren de bits de manera normalizada o a una porción del tren de manera no normalizada. El algoritmo selecciona el algoritmo de criptación.

AuthenticationCapability, si está presente, indica que el terminal soporta los componentes de autenticación de la Rec. UIT-T H.235 [16]. antiSpamAlgorithm indica el método y el algoritmo que se utiliza para proporcionar contramedidas frente a ataques de desbordamiento y de negación de servicio.

IntegrityCapability, si está presente, indica que el terminal soporta los componentes de integridad de la Rec. UIT-T H.235 [16].

#### **B.2.2.9 Capacidades de conferencia**

ConferenceCapability indica capacidades de conferencia tales como la aptitud para soportar el control de la presidencia que se describe en la Rec. UIT-T H.243.

videoIndicateMixingCapability será definido como VIM H.230.

### B.2.2.10 Capacidades de entrada de usuario

UserInputCapabilities indica los parámetros del mensaje UserInputIndication que soporta el terminal. BasicString indica que el terminal soporta la opción basicString de userInputSupportIndication, iA5String indica que el terminal soporta la opción iA5String de userInputSupportIndication y generalString indica que el terminal soporta la opción generalString de userInputSupportIndication. Dtmf indica que el terminal soporta dtmf utilizando la señal y los componentes de actualización de señal del mensaje userInputIndication. La señal de enganchado indica que el terminal soporta señal de enganchado utilizando la señal y los componentes de actualización de la señal del mensaje userInputIndication.

Para DTMF segura UserInputCapabilities indica qué parámetros son criptados en el mensaje UserInputIndication.

encryptedBasicString indica que el terminal soporta la opción encryptedalphanumeric de UserInputIndication.

encryptedA5String indica que el terminal soporta la opción encryptedSignalType de UserInputIndication.

encryptedGeneralString indica que el terminal soporta la opción encryptedalphanumeric de extendedAlphanumeric de UserInputIndication.

secureDTMF indica que el terminal soporta la opción encryptedSignalType dentro de la señal para DTMF seguro.

### B.2.2.11 Capacidades genéricas

El tipo **GenericCapability** permite especificar nuevas capacidades de tal modo que no sea necesario publicar una nueva versión de las sintaxis H.245. Esta manera genérica de especificar las capacidades permite que dispositivos basados en red, tales como las MC, determinen un modo de funcionamiento común muy elevado sin tener un conocimiento detallado de la capacidad utilizada. Permite definir descripciones de capacidades basadas en normas del UIT-T y en otras normas (incluidas las descripciones de capacidades patentadas). Las descripciones de capacidades basadas en normas del UIT-T deberán incluirse a modo de anexos a esta Recomendación. Las descripciones de capacidades ajenas a las normas del UIT-T se pueden publicar de la forma que mejor convenga.

El campo **capabilityIdentifier** indica el tipo de capacidad que se define. Las descripciones de capacidades basadas en las normas del UIT-T deberán utilizar el **OBJECT IDENTIFIER normalizado**, mientras que las descripciones basadas en otras normas y las descripciones de capacidades patentadas deberán utilizar la forma **standard**, **h221NonStandard**, **uuid** y **domainBased** según proceda.

El campo **subIdentifier** indica un tipo o conjunto de parámetros asociados con el **capabilityIdentifier**.

**maxBitRate** indica la velocidad máxima a la que la capacidad puede funcionar cuando se intercambian capacidades y la velocidad binaria que se va a utilizar realmente cuando se produzca la señalización de apertura de canal lógico. Deberá estar presente siempre que pueda ser indicado un valor significativo y cuando así lo imponga la especificación de capacidades de que se trate. Se define separadamente para que los intermediarios del trayecto de señalización puedan ver la anchura de banda utilizada sin tener un conocimiento detallado de cada capacidad.

Los parámetros de la capacidad se pueden describir como una combinación de **collapsing**, **nonCollapsing** y **nonCollapsingRaw**, junto con **transport**, especificada en la descripción de capacidades.

El campo **collapsing** indica capacidades descritas de tal manera que una MC puede combinar las de varios puntos extremos y elaborar un conjunto de capacidades comunes utilizando un juego sencillo de reglas sin tener conocimiento detallado del códec particular.

El campo **nonCollapsing** indica capacidades que utilizan la misma sintaxis que **collapsing** pero que pueden ser procesadas por una MC. En este caso, la semántica de **ParameterValue** cambia para indicar solamente valores y no reglas que desaparecen. Por ejemplo, **unsignedMin** y **unsignedMax** tienen la misma semántica e indican simplemente un parámetro entero de 16 bits.

El campo **nonCollapsingRaw** indica capacidades que utilizan una OCTET STRING (cadena de octetos). Ésta consta normalmente de una estructura de datos ASN.1 con codificación PER. Se señala que una MC ha de tener un conocimiento específico de las capacidades descritas de este modo para poder utilizarlas.

El campo **transport** indica parámetros de transporte específicos de la capacidad que se describe.

Se recomienda que, cuando se especifiquen descripciones de capacidades, se definan tantos parámetros **collapsing** como sea posible, ya que sólo los parámetros definidos de este modo tienen la seguridad de que serán procesados en vez de ser simplemente reenviados por los elementos de red.

Los tipos **GenericCapabilities** que incluyen secuencias **collapsing** y **nonCollapsing** no deberían incluir estructuras **GenericParameter** de tipos diferentes (**collapsing**, **nonCollapsing**) que utilizan el mismo **parameterIdentifier**.

NOTA 1 – Esta reutilización del mismo **parameterIdentifier** podría causar una colisión de valores **parameterIdentifier** si el parámetro se traslada automáticamente a un sistema, por ejemplo un sistema H.320, que no tiene distinción entre los parámetros **collapsing** y **nonCollapsing**.

El campo **parameterIdentifier** estándar de una estructura **GenericParameter** no debería tener asignado el valor 0.

NOTA 2 – La asignación al valor 0 podría interferir con la traducción automática a señalización H.320, como por ejemplo se efectúa en el anexo A/H.239 y en la Rec. UIT-T H.241.

**GenericParameter** indica un parámetro de capacidad o un grupo de parámetros de capacidad.

El **parameterIdentifier** permite indicar los valores de parámetros normalizados (es decir, definidos en la descripción de capacidades) y parámetros patentados. Los parámetros definidos en la descripción de capacidades utilizan la forma **standard** que identifica el parámetro con un entero. Los parámetros que son ampliaciones patentadas utilizan la forma **h221NonStandard**, **uuid** o **domainBased**.

El campo **parameterValue** indica el valor del parámetro. La presencia de un parámetro **logical** indica que el punto extremo soporta la opción que representa el parámetro. El campo **booleanArray** contiene hasta ocho variables booleanas independientes. Los campos **unsignedMin** y **unsignedMax** indican un parámetro utilizando un entero de 16 bits sin signo. Los campos **unsigned32Min** y **unsigned32Max** indican un parámetro utilizando un entero de 32 bits sin signo. El campo **octetString** indica un parámetro como una OCTET STRING. El campo **genericParameter** indica una secuencia de parámetros que han sido agrupados conjuntamente en esta capa de la jerarquía de capacidades.

Para combinar descripciones de capacidades procedentes de múltiples puntos extremos en una descripción de capacidades comunes para una capacidad de la que una MC no tiene conocimiento incorporado, la MC deberá ignorar primero cualquier parámetro no soportado por todos los puntos extremos que, según ella, pueden utilizar una capacidad determinada. A continuación, para cada uno de los parámetros de esos puntos extremos con el mismo **parameterIdentifier**, la MC deberá:

- efectuar un AND lógico en caso de **booleanArray**;
- elegir el valor mínimo en caso de **unsignedMin** o **unsigned32Min**; y
- elegir el valor máximo en caso de **unsignedMax** o **unsigned32Max**.

El campo **supersedes** permite que una descripción de capacidades contenga un grupo de parámetros del que sólo se deberá seleccionar uno cuando se determine una descripción de capacidades comunes. Éste podría ser el caso de un códec de vídeo que soportara resoluciones SQCIF, QCIF y CIF con intervalos mínimos de imagen diferentes. El valor del `parameterIdentifier` se refiere a un parámetro situado en el mismo nivel de anidamiento. Se incluyen múltiples campos **supersedes** con un parámetro de manera que se pueda expresar un árbol de dependencias de parámetros tal como figura en la descripción de capacidades H.262. De la descripción de capacidades comunes deberá descartarse cada uno de los parámetros identificados en el campo **supersedes**. Los parámetros sustituidos por los parámetros descartados deberán ser descartados a su vez y el proceso deberá repetirse hasta que se hayan descartado todos los parámetros sustituidos.

El resultado de esta operación es la descripción de capacidades comunes.

NOTA 3 – Una MC que tenga conocimiento incorporado de una determinada descripción de capacidades podrá utilizar su propio juego de reglas para producir una descripción de capacidades comunes.

#### **B.2.2.12 Capacidades del tren multiplexado**

`multiplexedStreamCapability` indica la capacidad de soportar un tren multiplexado en un único canal lógico.

`multiplexFormat` indica el protocolo de multiplexación soportado.

`controlOnMuxStream`, si es verdadero, indica que la señalización de canal lógico para el tren multiplexado se soporta utilizando el canal de control transportado a través del tren multiplexado. Si es falso, la señalización de canal lógico para el tren multiplexado se soporta utilizando este canal de control H.245. Cuando `controlOnMuxStream` es falso y `multiplexFormat` es `H223Capability`, deberá abrirse a lo sumo un canal lógico para el tren multiplexado H.223. `controlOnMuxStream` será falso si `MultiplexFormat` se fija en `h222Capability`.

`capabilityOnMuxStream`, si está presente, indica el conjunto de capacidades para el tren multiplexado. Esas capacidades se indican con `AlternativeCapabilitySet`. Este `AlternativeCapabilitySet` no deberá incluir la capacidad `multiplexedStreamTransmission`. Si está ausente, el conjunto de capacidades para el tren multiplexado deberá intercambiarse utilizando el canal de control que se transporta a través del tren multiplexado, una vez abierto el canal lógico del tren multiplexado.

#### **B.2.2.13 Cabida útil RTP para la capacidad de evento de telefonía audio y la capacidad de tono audio**

El parámetro `receiveRTPAudioTelephonyEventCapability` se puede incluir para indicar el soporte de eventos telefónicos audio en la banda según RFC 2833. El parámetro `dynamicRTPPayloadType` indica el tipo de cabida útil dinámica RTP que debe utilizarse para transportar dichos eventos. Los eventos soportados se describirán en el parámetro `audioTelephoneEvent` descrito en la <lista de valores> que figura en la sección 3.9 de RFC 2833. Los eventos 0 a 15 (correspondientes a los valores numéricos DTMF 0 a 9, \*, #, A, B, C y D) son los únicos eventos obligatorios.

El parámetro `receiveRTPAudioToneCapability` puede incluirse para indicar el soporte de los tonos en la banda según RFC 2833. El parámetro `dynamicRTPPayloadType` indica el tipo de cabida útil dinámica RTP que debe utilizarse para transportar dichos tonos.

#### **B.2.2.14 Tren de cabida útil múltiple**

Un tren de cabida útil múltiple (MPS, *multiple payload stream*) contiene paquetes que representan un tren de medios lógicos simple, es decir todos los paquetes representan codificaciones del mismo tren para intervalos de tiempo especificados. Para permitir la identificación y correlación de las diversas codificaciones utilizadas, todos los paquetes en un MPS simple DEBERÁN transportar identificadores de cabida útil en la misma ubicación en el paquete y DEBERÍAN utilizar indicadores de tiempo en el mismo formato y derivados de una fuente de reloj simple (por ejemplo,

las cabidas útiles RTP deberían utilizar la misma fuente de sincronización). En la mayoría de los casos estos paquetes representarán intervalos de tiempo secuenciales no superpuestos y simplemente eligen diferentes codificaciones para intervalos distintos, pero existen casos en los que las codificaciones alternadas representan intervalos de superposición, de modo tal que, cuando un evento se produce en el medio de un intervalo de codificación, se debe codificar de forma distinta en la codificación alternada. Esto puede ocurrir, por ejemplo, cuando se detecta un tono DTMF en el medio de un intervalo de codificación vocal y se debería enviar utilizando un evento telefónico RFC 2833. En este caso, la indicación de tiempo en el paquete de evento telefónico corresponderá a un tiempo en el medio del intervalo de codificación local. Se pueden utilizar paquetes con duración cero en el que el tren de eventos representado no tiene duración mensurable. Es también posible utilizar RFC 2198 para enviar muchas veces un paquete, empaquetado con otros tipos de cabida útil e intervalos de tiempo.

Cuando se abre un canal lógico que contiene un MPS, cada tren dentro del MPS tiene su propia velocidad binaria, que es independiente de los otros valores de velocidad binaria para otros trenes. Suponiendo que los medios dentro de un MPS estén efectivamente intercalados (es decir, sólo un tren dentro de un canal MPS se transmitiría en algún punto del tiempo), la velocidad binaria total para el canal MPS es el valor máximo de las velocidades binarias de todos los trenes MPS.

La velocidad binaria para un canal MPS se puede controlar a través de diversas instrucciones H.245, como con canales no MPS. En el caso en que la velocidad binaria fijada para el canal sea menor que la velocidad binaria de un determinado tren, no podrá utilizarse este tren para la transmisión de medios. Por ejemplo, si un canal MPS se abre con trenes G.729 y G.711, y para ajustar la velocidad binaria del canal a 32 kbit/s se utiliza instrucción de control de flujo, el punto extremo sólo puede transmitir utilizando G.729.

La velocidad binaria para un determinado tren de un canal MPS también se puede controlar a través de diversas instrucciones H.245. En ese caso, la velocidad binaria sólo afectaría a dicho tren en particular. Aquí también, la velocidad binaria para el canal sería el valor máximo de las velocidades binarias de todos los trenes MPS, salvo en el caso que se haya disminuido la velocidad binaria para todo el canal.

De esta manera, cuando se utiliza un tren de cabida útil múltiple, se deben considerar dos valores de velocidad binaria. El primero es la velocidad binaria implícita para el canal, que es el valor máximo de las velocidades binarias de todos los trenes MPS. El segundo es la velocidad binaria máxima para el canal en su totalidad, como se señala en las diversas instrucciones H.245 (por ejemplo, instrucción de control de flujo). Cuando se utiliza una instrucción de control de flujo u otra instrucción H.245 para eliminar restricciones de velocidad binaria del canal, se considera, una vez más, que la velocidad binaria para el canal es el valor máximo de las velocidades binarias de todos los trenes MPS.

NOTA – En razón de que todos los paquetes deben presentar codificaciones de trenes de una sola fuente (destino) no es apropiado incluir tipos de medios distintos, tal como audio y vídeo, aunque los paquetes de tipo de datos que representan datos derivados del tren de medios, tal como dígitos DTMF detectados en un tren de audio, pueden constituir una representación o codificación alternativa y son apropiados.

#### **B.2.2.15 Corrección de errores en recepción**

Un punto extremo puede indicar la posibilidad de efectuar corrección de errores en recepción (FEC, *forward error correction*). Cuando el punto extremo anuncia RFC 2733, significa que posee la capacidad de señalar que los datos FEC se pueden enviar en un tren separado o en el mismo tren (utilizando codificación redundante), conforme a RFC 2198. Esta capacidad permite al punto extremo indicar (por número de entrada a la tabla de capacidad) cuáles son los códecs que se pueden utilizar en un tren FEC.

Si el punto extremo que envía **OpenLogicalChannel** desea utilizar RFC 2198 (y que la capacidad sea soportada por el recipiente) para transportar los datos FEC, utilizará **DataType redundancyEncoding**, que incluye la codificación VBD, por ejemplo, como codificación **primaria** y **DataType fec** como codificación secundaria. El tipo de cabida útil para los paquetes RFC 2198 se especificarán en el campo **dynamicPayloadType** de **OpenLogicalChannel**. El tipo de cabida útil para la codificación primaria y los datos FEC se deben señalar en el campo **payloadType** de los campos **RedundancyEncodingElement primario** y **secundario**.

Si un punto extremo desea transmitir los datos FEC en un tren separado, tiene dos elecciones: transmitir al mismo puerto como datos protegidos FEC o a un puerto diferente. Cuando se transmite sobre un puerto diferente, se utilizará un **OpenLogicalChannel** separado explícitamente para el tren FEC. El **dataType** seleccionado será **fec** y no estará contenido dentro de un campo **redundancyEncoding**. Seleccionará **mode.separateStream.differentPort** e incluirá el ID de sesión del tren protegido y, facultativamente, el tipo de cabida útil del medio protegido, en el caso de que el canal utilizado transporte tipos de cabida útil múltiples, tal como un tren MPS. Cuando se transmite en un tren separado, pero al mismo puerto que a los medios protegidos, los datos FEC han de ser señalados como parte de un tren MPS. En ese caso, un elemento del tren MPS sería la señal de audio protegida y el otro elemento sería **fec**. De esta manera, se seleccionaría **mode.separateStream.samePort** y se anunciaría el tipo de carga útil del tren protegido.

### B.2.3 Acuse de recibo del conjunto de capacidades de terminal

Se utiliza para confirmar la recepción de un TerminalCapabilitySet desde la CESE par.

El sequenceNumber será el mismo que el sequenceNumber del TerminalCapabilitySet para el cual es la confirmación.

### B.2.4 Rechazo del conjunto de capacidades de terminal

Se utiliza para rechazar un TerminalCapabilitySet procedente de la CESE par.

El sequenceNumber será el mismo que el sequenceNumber del TerminalCapabilitySet para el que éste es el acuse de recibo negativo.

En el cuadro B.6 se indican los motivos de envío de este mensaje.

**Cuadro B.6/H.245 – Motivos para rechazar un TerminalCapabilitySet**

Punto de código ASN.1	Causa
unspecified	No se ha especificado la causa del rechazo.
undefinedTableEntryUsed	Un descriptor de capacidad hace referencia a una inscripción de capabilityTable que no está definida.
descriptorCapacityExceeded	El terminal no puede almacenar toda la información de TerminalCapabilitySet.
tableEntryCapacityExceeded	El terminal era incapaz de almacenar más inscripciones que las indicadas en highestEntryNumberProcessed o no pudo almacenar ninguna.

### B.2.5 Liberación del conjunto de capacidades de terminal

Se envía en el caso de expiración de un periodo de temporización.

## B.3 Mensajes de señalización de canal lógico

Este conjunto de mensajes se utiliza para señalización de canal lógico local. El mismo conjunto de mensajes se utiliza para señalización de canal lógico unidireccional y bidireccional; sin embargo, algunos parámetros sólo están presentes en el caso de la señalización de canal lógico bidireccional.



Se utiliza "forward" (hacia adelante o de ida) para hacer referencia a la transmisión en el sentido del terminal que hace la petición inicial de un canal lógico hacia el otro terminal y "reverse" (inverso o de vuelta) se utiliza para hacer referencia al sentido opuesto de transmisión, en el caso de una petición de canal lógico bidireccional.

### **B.3.1 Apertura de canal lógico**

Se utiliza este mensaje para intentar abrir una conexión de canal lógico unidireccional entre una LCSE de salida y una LCSE de entrada par y para abrir una conexión de canal lógico bidireccional entre una B-LCSE de salida y una B-LCSE de entrada par.

**forwardLogicalChannelNumber:** indica el número de canal lógico del canal lógico de ida que va a ser abierto.

**forwardLogicalChannelParameters:** incluye parámetros asociados con el canal lógico en el caso de intento de abrir un canal unidireccional y parámetros asociados con el canal lógico de ida en el caso de intento de abrir un canal bidireccional.

**reverseLogicalChannelParameters:** incluye parámetros asociados con el canal lógico inverso en el caso de intento de apertura de un canal bidireccional. Su presencia indica que la petición es de un canal lógico bidireccional con los parámetros expresados y su ausencia indica que la petición es de un canal lógico unidireccional.

NOTA – No se incluyen parámetros H.222 en reverseLogicalChannelParameters, ya que sus valores no son conocidos por el terminal que inicia la petición.

portNumber es un parámetro de usuario a usuario que puede utilizarse para fines tales como la asociación de un puerto de entrada o salida, o un número de canal de capa superior, con el canal lógico.

dataType indica los datos que deben transportarse por el canal lógico.

Si se trata de nullData, no se empleará el canal lógico para el transporte de trenes de datos elementales sino únicamente para información de la capa de adaptación – si debe transmitirse una señal vídeo en un sentido solamente y ha de utilizarse un protocolo de retransmisión como el AL3 definido en H.223, se necesita un canal de retorno para transportar las peticiones de retransmisión – puede también utilizarse para describir un canal lógico que únicamente contiene valores PCR en el caso de trenes de transporte H.222.1 [9].

Un dataType de h235Media se utiliza para especificar la criptación del canal lógico; el tipo de datos reales es indicado dentro de H235Media, junto con la especificación de criptación.

Los terminales que sólo son capaces de funcionamiento unidireccional (transmisión o recepción) en tipos de medios que utilizan canales bidireccionales enviarán capacidades solamente para el sentido de funcionamiento soportado. Para el sentido inverso se utilizará el tipo nullData, para el cual no se necesita ninguna capacidad. Los terminales de transmisión solamente deben enviar capacidades de transmisión, pero los terminales no deben suponer que la ausencia de capacidades de transmisión implica que no es posible el funcionamiento de transmisión solamente.

separateStack indica que se utilizará una pila de transporte por separado para transportar datos y proporciona una dirección que se utilizará para establecer la pila; ésta es una dirección de transporte Q.2931, E.164 o de red de área local.

networkAccessParameters define la información de distribución, dirección de red, y creación y asociación que se utilizará para la separateStack.

distribution deberá estar presente cuando networkAddress se fija a localAreaNetwork e indicará si la networkAddress es una dirección de transporte de unidifusión o de multidifusión.

networkAddress indica la dirección de la pila que se está utilizando efectivamente: dirección de transporte de Q.2931, E.164 o de red de área local.

associateConference indica si la conferencia de datos es nueva (associateConference=FALSE) o es una conferencia de datos existente que debe asociarse con la llamada audio/vídeo (associateConference=TRUE).

externalReference indica la información que puede utilizarse para proporcionar posteriormente asociación o información relativa a la separateStack.

Si es del tipo VideoCapability o AudioCapability indicadas para cada capacidad individual, puede utilizarse el canal lógico para cualquiera de las variaciones descritas, debiendo ser posible la conmutación entre esas variaciones utilizando únicamente señalización en banda del canal lógico – por ejemplo, en el caso de vídeo H.261, si se indican QCIF y CIF, deberá ser posible la conmutación entre ellas imagen por imagen. En el caso de DataApplicationCapability sólo puede indicarse un ejemplar de una capacidad porque no hay señalización en la banda que permita una conmutación entre variaciones.

Si se trata de encryptionData, se utilizará el canal lógico para el transporte de información de criptación según se haya especificado.

Si se trata de multiplexedStream, se utilizará el canal lógico para el transporte de audio/vídeo/datos como un tren multiplexado, según se haya especificado. Los campos de MultiplexedStreamParameter tienen el mismo significado que los campos del mismo nombre en MultiplexedStreamCapability.

forwardLogicalChannelDependency indica el número del canal lógico del que depende el canal directo que se va a abrir.

reverseLogicalChannelDependency indica el número del canal lógico del que depende el canal inverso que se va a abrir.

El parámetro replacementFor indica que el canal lógico que se va a abrir *reemplazará* al canal lógico existente especificado, que ya está abierto. Este parámetro se utilizará solamente para hacer referencia a canales lógicos que ya se encuentran en el estado ESTABLISHED. Los canales lógicos abiertos utilizando este parámetro no darán curso a ningún tráfico de datos mientras no cese todo el tráfico del canal lógico establecido al que se ha hecho referencia. En este caso, nunca se requerirá de los decodificadores de medios que decodifiquen tráfico de datos procedente de ambos canales lógicos simultáneamente. Una vez que comience el tráfico por el canal lógico recién establecido, se cerrará inmediatamente el canal lógico antiguo. Los receptores pueden acusar la apertura de canales lógicos utilizando el mecanismo replacementFor en el entendimiento de que los canales antiguo y nuevo no se utilizarán simultáneamente y, por consiguiente, no se rebasará la capacidad de decodificación del receptor.

El campo encryptionSync será utilizado por el principal para proporcionar el valor de la clave de criptación y el punto de sincronización en el que deberá utilizarse la clave. En el caso de H.323, la syncFlag se fijará en el número de cabida útil dinámica RTP que concuerde con la clave.

**H222LogicalChannelParameters:** se utiliza para indicar parámetros específicos de la utilización de la Rec. UIT-T H.222.1 [9]. Deberá estar presente en forwardLogicalChannelParameters y no estarlo en reverseLogicalChannelParameters.

resourceID indica en qué canal virtual ATM deberá transportarse el canal lógico. La forma de asociación de este parámetro con un canal virtual ATM no se especifica en esta Recomendación. Cuando la Rec. UIT-T H.222.0 se utiliza en la Rec. UIT-T H.323 como formato del tren multiplexado, este parámetro contiene el número de canal lógico del tren multiplexado en el que se efectuará la multiplexación de dicho canal lógico.

subChannelID indica el subcanal H.222.1 utilizado para el canal lógico. Será igual al PID de un tren de transporte y al stream\_id en un tren de programa.

pcr-pid indica el PID utilizado para el transporte de las referencias de reloj de programa cuando se utilice el tren de transporte. Deberá estar presente cuando el canal virtual ATM transmita un tren de transporte y no estarlo cuando el canal virtual ATM transporte un tren de programa.

programDescriptors es una cadena de octetos facultativa la cual, si está presente, contiene uno o más descriptores, como se especifica en las Recs. UIT-T H.222.0 y H.222.1 que describen el programa del que forma parte la información que debe transportarse por el canal lógico.

streamDescriptors es una cadena de octetos facultativa, la cual, si está presente, contiene uno o más descriptores, según se especifica en las Recs. UIT-T H.222.0 y H.222.1 que describen la información que debe transportarse por el canal lógico.

**H223LogicalChannelParameters:** se utiliza para expresar parámetros específicos de la utilización de la Rec. UIT-T H.223 [10]. Deberá estar presente en forwardLogicalChannelParameters y reverseLogicalChannelParameters.

adaptationLayerType indica la capa de adaptación y opciones que se utilizarán en el canal lógico. Los puntos de código son los siguientes: nonStandard, al1Framed (modo tramado AL1), al1NotFramed (modo no tramado AL1), al2WithoutSequenceNumbers (AL2 sin números de secuencia presentes), al2WithSequenceNumbers (AL2 con números de secuencia presentes) y al3 (AL3, indicando el número de octetos del campo de control que estarán presentes y el tamaño de la memoria tampón de transmisión,  $B_S$ , que se utilizará, midiéndose el tamaño en octetos), al1M (AL1M se define en el anexo C/H.223 con los parámetros especificados), al2M (AL2M se define en el anexo C/H.223 con los parámetros especificados) o al3M (AL3M se define en el anexo C/H.223 con los parámetros especificados).

segmentableFlag, cuando toma el valor verdadero, indica que se ha marcado el canal como segmentable y, cuando toma el valor falso, indica que se ha marcado el canal como no segmentable.

**H223AL1MParameters:** se utiliza para indicar parámetros específicos de la utilización de la capa de adaptación AL1M.

transferMode indica si se utiliza el modo tramado o el modo no tramado.

headerFEC indica si la corrección de errores en recepción (FEC) es SEBCH(16,7) o Golay(24,12).

La longitud en bits de la CRC de la cabida útil es indicada por crcLength como de 4, 8, 12, 16, 20, 28 ó 32 bits o por crcNotUsed.

rcpcCodeRate indica la proporción de código RCPC como 8/8, 8/9, ..., 8/32.

arqType indica el modo ARQ de funcionamiento: noARQ indica no retransmisión, typeIArq indica ARQ tipo I y typeIIArq indica ARQ tipo II.

alpduInterleaving, si toma el valor verdadero, indica utilización de entrelazado de las AL-PDU.

alsduSplitting, si toma el valor verdadero, indica utilización del modo división de las AL-SDU.

rsCodeCorrection indica la capacidad de corrección del código RS como 0, 1, ..., 127 octetos. A cada campo AL-SDU y CRC de longitud variable se le añade un número fijo de símbolos (octetos) de paridad del código RS correspondiente a rsCodeCorrection. Cuando se utiliza la codificación RS, no se soporta typeIIArq ni alpduInterleaving.

**H223AL2MParameters:** se utiliza para indicar parámetros específicos de la utilización de la capa de adaptación AL2M.

headerFEC indica si FEC es SEBCH(16,5) o Golay(24,12).

alpduInterleaving, si toma el valor verdadero, indica utilización de entrelazado de las AL-PDU.

**H223AL3MParameters:** se utiliza para indicar parámetros específicos de la utilización de la capa de adaptación AL3M.

Tiene los mismos parámetros que AL1MParameters, con la salvedad de que transferMode y alduSplitting no están presentes.

### **H223AnnexCArqParameters**

numberOfRetransmissions indica el número máximo de retransmisiones que se puede utilizar: si es un número finito, indica un límite finito al número de retransmisiones que se puede utilizar en la gama de 0 a 16; si es infinito, indica que no hay límite al número de retransmisiones que se puede utilizar. Si numberOfRetransmissions tiene el valor finito de 0, ello indica que el campo de control se utiliza para el modo división pero que no se utilizan retransmisiones.

sendBufferSize indica el tamaño de la memoria tampón de envío que se utilizará, midiéndose el tamaño en octetos.

**V76LogicalChannelParameters:** se utiliza para indicar parámetros específicos de la utilización de la Rec. UIT-T V.76.

audioHeader se utiliza para indicar la utilización de un encabezamiento de audio en el canal lógico. Es un parámetro válido para canales del tipo de datos audio.

suspendResume se utiliza para indicar que el canal puede utilizarse para los procedimientos de suspensión/reanudación con el fin de suspender otros canales lógicos. Puede seleccionarse una de estas tres opciones relativas al canal: no efectuar suspensión o reanudación en el canal, efectuar suspensión o reanudación utilizando una dirección o efectuar suspensión o reanudación sin una dirección como se define en la Rec. UIT-T V.76. suspendResumewAddress indica que el canal de suspensión/reanudación utilizará el campo de dirección definido en la Rec. UIT-T V.76. suspendResumewoAddress indica que el canal de suspensión/reanudación no utilizará el campo de dirección.

eRM indica que el canal lógico efectuará los procedimientos de recuperación tras error definidos en la Rec. UIT-T V.76.

uNERM indica que el canal lógico funcionará en el modo de ausencia de recuperación tras error definido en la Rec. UIT-T V.76.

Para la descripción de n401, windowSize y loopbackTestProcedure véase 12.2.1/V.42 y sus cláusulas. Para los fines de la Rec. UIT-T V.70, n401 se codificará en octetos.

crcLength es un parámetro facultativo que indica la longitud CRC utilizada en el modo de recuperación tras error. Si este parámetro no está presente, se utilizará la longitud CRC supletoria. crc8bit indica que se utilizará una CRC de 8 bits, crc16bit indica que se utilizará una CRC de 16 bits y crc32bit indica que se utilizará una CRC de 32 bits definida en la Rec. UIT-T V.76.

recovery es un parámetro facultativo que indica los procedimientos de recuperación tras error definidos en la Rec. UIT-T V.76. Si este parámetro no está presente, se utilizará el procedimiento supletorio de recuperación tras error. sREJ indica que se habrá de utilizar el procedimiento de rechazo de trama selectivo y mSREJ indica que se utilizará el procedimiento de rechazo de trama selectivo múltiple definido en la Rec. UIT-T V.76.

uIH indica la utilización de tramas UIH V.76.

rej indica la utilización del procedimiento de rechazo de la Rec. UIT-T V.76.

V75Parameters se utiliza para indicar parámetros específicos de la utilización de la Rec. UIT-T V.75. audioHeaderPresent indica la presencia del encabezamiento de audio V.75.

**H2250LogicalChannelParameters:** se utiliza para indicar parámetros específicos de la utilización de la Rec. UIT-T H.225.0. Estará presente en forwardLogicalChannelParameters y reverseLogicalChannelParameters.

La `sessionID` es un identificador de sesión RTP o T.120 único en la conferencia. Lo utiliza el transmisor para hacer referencia a la sesión a que se aplica el canal lógico. Sólo el terminal principal puede crear la identificación de sesión. Por convenio, hay tres sesiones primarias. La primera sesión primaria, con una identificación de sesión de 1, es la sesión de audio, la segunda sesión primaria, con una identificación de 2, es la sesión de vídeo y la tercera sesión primaria, con una identificación de 3, es la sesión de datos. Una entidad subordinada puede abrir una sesión adicional proporcionando una identificación de sesión de 0 en el mensaje `openLogicalChannel`. El principal creará una identificación de sesión única y la proporcionará en el mensaje `openLogicalChannelAck`.

La `associatedSessionID` se utiliza para asociar una sesión con otra. Una utilización típica será para asociar una sesión de audio con una sesión de vídeo, con el fin de indicar las sesiones que habrán de procesarse para la sincronización de labios.

El `mediaChannel` indica una `transportAddress` que se utilizará para el canal lógico. No está presente en el mensaje `OpenLogicalChannel forwardLogicalChannelParameters` cuando el transporte es de unidifusión, pero estará presente en `reverseLogicChannelParameters`. Si la `transportAddress` es de multidifusión, el principal es responsable de la creación de la dirección de transporte de multidifusión e incluirá la dirección en el mensaje `OpenLogicalChannel`. Una entidad subordinada que desea abrir un nuevo canal multidifusión insertará ceros en el campo `transportAddress` de multidifusión. El principal creará y proporcionará la `transportAddress` de multidifusión en el mensaje `OpenLogicalChannelAck` para la entidad subordinada. Obsérvese que el MC utilizará la `communicationModeCommand` para especificar los detalles sobre todas las sesiones RTP en la conferencia.

El `mediaChannel` se utiliza para describir la dirección de transporte para el canal lógico. Las direcciones IPv4 e IPv6 se codificarán comenzando por el octeto más significativo de la dirección, en la OCTET STRING respectiva, por ejemplo, la dirección IPv4 de la clase B 130.1.2.97 tendrá el "130" codificado en el primer octeto de la OCTET STRING, seguido por el "1", y así sucesivamente. La dirección IPv6 al48:2:3:4:a:b:c:d tendrá el "al" codificado en el primer octeto, "48" en el segundo, "00" en el tercero, "02" en el cuarto y así sucesivamente. Las direcciones IPX, node, netnum y port se codificarán comenzando por el octeto más significativo de cada campo, en la OCTET STRING respectiva.

`mediaGuaranteedDelivery` indica si debe o no seleccionarse el transporte del medio subyacente para proporcionar o no proporcionar la entrega garantizada de datos.

`mediaControlChannel` indica el canal de control de medio en el que el emisor del canal lógico abierto estará en escucha de mensajes de control de medios para esta sesión. Este campo sólo está presente cuando se requiere un canal de control de medio.

`mediaControlGuaranteedDelivery` indica si se debe seleccionar o no el transporte de control de medio subyacente para proporcionar o no una entrega garantizada de datos. Este campo está presente solamente cuando se requiere un canal de control de medio.

La `silenceSupression` se utiliza para indicar si el transmisor detiene el envío de paquetes durante los periodos de silencio. Se incluirá en el mensaje `openLogicalChannel` en el caso de un canal de audio y se omitirá en el caso de un canal de otro tipo.

`destination` indica la `terminalLabel` del destino si se ha asignado alguna.

`dynamicRTPPayloadType` indica un valor de cabida útil dinámica. Cuando se utiliza este campo, `RTPPayloadType.payloadType` y el valor de este campo deberán concordar.

`mediaPacketization` indica el esquema de paquetización de medio facultativo que se está utilizando.

`redundancyEncoding` indica que se ha de utilizar el método de codificación redundante indicado en este parámetro para el canal lógico que se va a abrir. La codificación primaria se define mediante el `dataType` de los `forwardLogicalChannelParameters` o de los `reverseLogicalChannelParameters`, respectivamente. El tipo de codificación por redundancia que se ha de aplicar para este canal lógico

se identifica mediante el parámetro *redundancyEncodingMethod* y la codificación secundaria se especifica mediante el parámetro *secondaryEncoding*. El *DataType* (audio, vídeo, etc.) seleccionado para ambas codificaciones, primaria y secundaria, deberá concordar con el *redundancyEncodingMethod* seleccionado. El parámetro *source* se utiliza para identificar el número de terminal del emisor del mensaje *OpenLogicalChannel*.

La apertura de un canal protegido por redundancia, como se especifica en RFC 2198, se efectúa utilizando **dataType.redundancyEncoding**. Este campo permite la señalización de un tipo de datos primario y una serie de tipos de datos **secundarios**. También es posible utilizar RFC 2198 con "tren de cabida útil múltiple" y con corrección de errores en recepción.

Cuando se abre un canal lógico, el tipo de cabida útil RTP para el paquete RFC 2198 viene especificado por el campo **dynamicPayloadType** en el **OpenLogicalChannel** o por el campo **payloadType** dentro de la estructura **multiplePayloadStreamElement**. Para los tipos de cabida útil primario y secundario se especifican en la estructura **RedundancyEncodingElement**, conjuntamente con el **DataType** de los datos primarios o secundarios.

Cuando se utiliza codificación redundante RFC 2198, **redundancyEncodingMethod** se pone en **rtpRedundancyEncoding**. Asimismo, cuando se utiliza RFC 2198 y se aplica **RedundancyEncoding SEQUENCE**, sólo se utilizará **rtpRedundancyEncoding SEQUENCE**. Los campos **RedundancyEncoding.secondaryEncoding** y **RedundancyEncoding.rtpRedundancyEncoding** no se utilizarán al mismo tiempo.

Cuando se especifica criptación para un canal que transporta múltiples cabidas útiles, se utiliza codificación redundante conforme a RFC 2198 para preservar los tipos de cabida útil reales transmitidos. El tipo de cabida útil *Encapsulating* se fija al valor especificado en el campo *syncFlag* del elemento *encryptionSync*.

**h235 Key:** se utiliza para incluir y especificar el método mediante el cual se protegen las claves de sesión específicas de los medios cuando se transfieren entre dos puntos extremos. La codificación de este campo es un valor ASN.1 anidado descrito en la Rec. UIT-T H.235.

*EscrowData* se utiliza para especificar el tipo y contenido de cualquier mecanismo de custodia de claves que se utilice. Las implementaciones pueden requerir tipos y contenidos específicos cuando se habilite la criptación de medios.

*T120SetupProcedure* indica cómo se ha de establecer la conferencia T.120. Para *originateCall* y *waitForCall*, el llamante deberá obtener el nombre numérico de la conferencia T.120 a partir del identificador de conferencia (*CID*, *conference identifier*) H.323 (descrito en la Rec. UIT-T H.323) y emitir la PDU apropiada (si el punto extremo es directo, deberá existir una petición de invitación, mientras que si es subordinado deberá emitir una petición de incorporación). En el caso de *issueQuery*, el llamante deberá emitir primero una petición de indagación y a continuación establecer la conferencia T.120 de acuerdo con el contenido de la respuesta a la indagación (como se describe en la Rec. UIT-T T.124).

### **B.3.2 Acuse de recibo de apertura de canal lógico**

Se utiliza para confirmar la aceptación de la petición de conexión de canal lógico desde la LCSE o B-LCSE par. En el caso de petición de un canal lógico unidireccional, indica la aceptación de ese canal lógico unidireccional. En el caso de petición de un canal lógico bidireccional, indica la aceptación de ese canal lógico bidireccional, así como los parámetros apropiados del canal inverso.

*forwardLogicalChannelNumber* indica el número de canal lógico del canal de ida que se está abriendo.

*reverseLogicalChannelParameters* está presente únicamente si se está respondiendo a una petición de canal bidireccional.

*reverseLogicalChannelNumber* indica el número de canal lógico del canal inverso.

portNumber es un parámetro de usuario a usuario que puede ser utilizado por un usuario para fines tales como la asociación de un puerto de entrada o salida, o un número de canal de capa superior, con el canal lógico inverso.

multiplexParameters indica parámetros específicos del múltiplex, H.222, H.223 o H.225.0, que se utilizan para transportar el canal lógico inverso.

FlowControlToZero indica si se permite al transmisor comenzar la transmisión por el canal lógico. Si se fija a verdadero, indica que el transmisor no deberá transmitir por el canal lógico hasta que reciba un mensaje FlowControl subsiguiente, aplicable al canal lógico, con el que se le autorice a hacerlo. Si se fija a falso, o está ausente, el transmisor está autorizado a comenzar la transmisión inmediatamente después de que se establezca el canal.

El parámetro replacementFor indica que el canal lógico que se va a abrir *reemplazará* al canal lógico existente especificado, que está ya abierto. Este parámetro se utiliza solamente para hacer referencia a canales lógicos que ya se encuentran en el estado ESTABLISHED. Los canales lógicos abiertos utilizando este parámetro no darán curso a ningún tráfico de datos mientras no cese todo el tráfico del canal lógico establecido al que se ha hecho referencia. En este caso, nunca se requerirá de los decodificadores de medios que decodifiquen tráfico de datos procedentes de ambos canales lógicos simultáneamente. Una vez que comience el tráfico por el canal lógico recién establecido, se cerrará inmediatamente el canal lógico antiguo. Los receptores pueden acusar apertura de canales lógicos utilizando el mecanismo replacementFor en el entendimiento de que los canales lógicos antiguo y nuevo no serán utilizados simultáneamente y, por consiguiente, no se rebasará la capacidad de decodificación del receptor.

separateStack indica que se utilizará una pila de transporte separada para transportar los datos y proporciona una dirección que se utilizará para establecer la pila; ésta es una dirección Q.2931, E.164 o una dirección de transporte de red de área local.

forwardMultiplexAckParameters indican parámetros específicos del múltiplex, H.222, H.223 o H.225.0 que se utilizarán para transportar el canal lógico de ida.

El campo encryptionSync será utilizado por el principal para proporcionar el valor de la clave de criptación y el punto de sincronización en el que deberá utilizarse la clave. En el caso de la Rec. UIT-T H.323, la syncFlag se fijará en el número de cabida útil dinámica RTP que concuerde con la clave.

H2250LogicalChannelAckParameters se utilizan para indicar parámetros específicos de la utilización de la Rec. UIT-T H.225.0.

sessionID es un identificador de sesión RTP único en la conferencia, que sólo puede ser creado por el principal. Éste lo crea y lo proporciona si el subordinado desea crear una nueva sesión especificando una indicación de sesión no válida de 0 en el mensaje OpenLogicalChannelAck.

El mediaChannel indica una transportAddress que se utilizará para el canal lógico. Deberá estar presente en el mensaje OpenLogicalChannelAck cuando el transporte es de unidifusión excepto donde la petición OpenLogicalChannel especifica un retorno unidifusión mediaChannel. Si la transportAddress es de multidifusión, el principal es responsable de la creación de la dirección de transporte de multidifusión e incluirá la dirección en el mensaje OpenLogicalChannel. Una entidad subordinada que desea abrir un nuevo canal multidifusión insertará ceros en el campo transportAddress de multidifusión. El principal creará y proporcionará la transportAddress de multidifusión en el mensaje OpenLogicalChannelAck para la entidad subordinada. Obsérvese que el MC utilizará la communicationModeCommand para especificar los detalles sobre todas las sesiones RTP en la conferencia.

El mediaChannel se utiliza para describir la dirección de transporte para el canal lógico. Las direcciones IPv4 e IPv6 se codificarán comenzando por el octeto más significativo de la dirección, en la OCTET STRING respectiva, por ejemplo, la dirección IPv4 de la clase B 130.1.2.97 tendrá

el "130" codificado en el primer octeto de la OCTET STRING, seguido por el "1" y así sucesivamente. La dirección IPv6 al48:2:3:4:a:b:c:d tendrá el "al" codificado en el primer octeto, "48" en el segundo, "00" en el tercero, "02" en el cuarto y así sucesivamente. Las direcciones IPX, node, netnum y port se codificarán comenzando por el octeto más significativo de cada campo, en la OCTET STRING respectiva.

mediaControlChannel indica el canal de control de medio en el que el emisor de OpenLogicalChannelAck estará en escucha de mensajes de control de medio para esta sesión. Este campo sólo está presente cuando se requiere un canal de control de medio.

dynamicRTTPayloadType indica un valor de cabida útil dinámica que se utiliza en H.323 para el esquema de paquetización de vídeo H.261, alternativa H.225.0. Este campo está presente solamente cuando se está utilizando una cabida útil RTP dinámica.

El campo portNumber se utiliza en el anexo C/H.323 cuando el punto extremo encuentra que el B-HLI dado en el campo portNumber del mensaje OpenLogicalChannel es inapropiado e indica el valor alternativo que se ha de utilizar.

NOTA – Los parámetros H.223 no se han incluido en reverseLogicalChannelParameters, pues sus valores se especificaron en el mensaje de petición OpenLogicalChannel.

### B.3.3 Rechazo de apertura de canal lógico

Se utiliza para rechazar la petición de conexión de canal lógico proveniente de la LCSE o B-LCSE par.

NOTA – En el caso de una petición de canal bidireccional, el rechazo se aplica al canal de ida y al de retorno. No es posible aceptar uno y rechazar el otro.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida especificado en la petición que se rechaza.

El campo cause (causa) indica el motivo del rechazo del establecimiento del canal lógico. Los valores de este campo se indican en el cuadro B.7.

**Cuadro B.7/H.245 – Motivos para rechazar un OpenLogicalChannel**

Punto de código ASN.1	Causa
unspecified	No se especifica una causa para el rechazo.
unsuitableReverseParameters	Se utilizará solamente para rechazar una petición de canal lógico bidireccional cuando el único motivo del rechazo es que los reverseLogicalChannelParameters solicitados no son apropiados. Tal rechazo deberá ir seguido inmediatamente por la iniciación de procedimientos para abrir un canal lógico bidireccional similar, pero aceptable.
dataTypeNotSupported	El terminal no era capaz de soportar el tipo de datos indicado en OpenLogicalChannel.
dataTypeNotAvailable	El terminal no era capaz de soportar el tipo de datos indicado en OpenLogicalChannel simultáneamente con los tipos de datos de canales lógicos que ya estaban abiertos.
unknownDataType	El terminal no comprendió el tipo de datos indicado en OpenLogicalChannel.
dataTypeALCombinationNotSupported	El terminal no era capaz de soportar el tipo de datos indicado en OpenLogicalChannel simultáneamente con el tipo de capa de adaptación indicado en H223LogicalChannelParameters.
multicastChannelNotAllowed	No pudo abrirse el canal de multidifusión.



**Cuadro B.7/H.245 – Motivos para rechazar un OpenLogicalChannel**

<b>Punto de código ASN.1</b>	<b>Causa</b>
insufficientBandwidth	El canal no pudo abrirse porque se denegó el permiso para utilizar la anchura de banda solicitada para el canal lógico.
separateStackEstablishmentFailed	Una petición de efectuar la parte datos de una llamada en una pila separada ha fracasado.
invalidSessionID	El subordinado intentó fijar SessionID al abrir un canal lógico hacia el principal.
masterSlaveConflict	El subordinado ha tratado de abrir un canal lógico en el que el director ha determinado que puede producirse un conflicto (véanse C.4.1.3 y C.5.1.3).
waitForCommunicationMode	Intento de abrir un canal lógico antes de que el MC haya transmitido la CommunicationModeCommand.
invalidDependentChannel	Intento de abrir un canal lógico con un canal dependiente especificado que no está presente.
replacementForRejected	Un canal lógico del tipo del canal cuya apertura se ha intentado no puede abrirse utilizando el parámetro replacementFor. El transmisor quizás desee intentarlo de nuevo cerrando primero el canal lógico que se había de reemplazar y abriendo a continuación el reemplazante.

#### **B.3.4 Confirmación de apertura de canal lógico**

Se utiliza en señalización bidireccional para indicar a la B-LCSE de entrada que el canal inverso está abierto y puede utilizarse para transmisión.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida que se abrió.

#### **B.3.5 Cierre de canal lógico**

Lo utiliza la LCSE o B-LCSE de salida para cerrar una conexión de canal lógico entre dos LCSE o B-LCSE pares.

NOTA – En el caso de un canal lógico bidireccional, este mensaje cierra el canal de ida y el canal inverso. No es posible cerrar uno y no el otro.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico que se cierra.

La fuente de la liberación del canal lógico se indica en el cuadro B.8.

**Cuadro B.8/H.245 – Fuentes de liberación del canal lógico**

<b>Punto de código ASN.1</b>	<b>Causa</b>
user	La fuente de la liberación es la LCSE o B-LCSE usuaria.
lcse	La LCSE o B-LCSE es la fuente de la liberación. Esto puede ocurrir como consecuencia de un error de protocolo.

reason (motivo) indica por qué se cierra el canal. reservationFailure indica que no se pudo efectuar una reserva de QOS para el canal y que, por consiguiente, se procede a su cierre. reopen (reapertura) indica que el punto extremo debería cerrar el canal y reabrir a continuación un canal utilizando los procedimientos OpenLogicalChannel. Esto es lo que puede ocurrir, por ejemplo, si

una llamada multipunto se reduce a una llamada punto a punto porque se reparan de una conferencia puntos extremos.

### **B.3.6 Acuse de recibo de cierre de canal lógico**

Se utiliza para confirmar el cierre de una conexión de canal lógico.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico que se está cerrando.

### **B.3.7 Petición de cierre de canal lógico**

La utiliza la CLCSE de salida para solicitar el cierre de una conexión de canal lógico de dos LCSE pares.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico cuyo cierre se pide.

qosCapability se utiliza para indicar los parámetros de QOS que se utilizaron en el canal.

reason indica por qué se produce la petición de cierre del canal. reservationFailure indica que no se pudo efectuar una reserva de QOS para el canal y que, por consiguiente, se procede a su cierre.

reopen indica que el punto extremo debería cerrar el canal y reabrir a continuación un canal utilizando los procedimientos OpenLogicalChannel. Esto es lo que puede ocurrir, por ejemplo, si una llamada multipunto se reduce a una llamada punto a punto porque se separan de una conferencia puntos extremos.

### **B.3.8 Acuse de recibo de petición de cierre de canal lógico**

Lo utiliza la CLCSE de entrada para indicar que se cerrará la conexión de canal lógico.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico cuyo cierre se ha solicitado.

### **B.3.9 Rechazo de petición de cierre de canal**

Lo utiliza la CLCSE de entrada para indicar que no se cerrará la conexión de canal lógico.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico cuyo cierre se ha solicitado.

El campo cause indica el motivo del rechazo de la petición de cierre del canal lógico. El único valor de causa válido es no especificado.

### **B.3.10 Liberación de petición de cierre de canal**

Lo envía la CLCSE de salida en caso de expiración de un periodo de temporización.

forwardLogicalChannelNumber indica el número de canal lógico del canal de ida del canal lógico cuyo cierre se ha solicitado.

## **B.4 Mensajes de señalización de tabla múltiplex**

Este conjunto de mensajes está destinado a la transmisión segura de entrada de la tabla múltiplex H.223 del transmisor al receptor.

### **B.4.1 Envío de entradas múltiplex**

Se utiliza para el envío de entradas de tabla múltiplex H.223 del transmisor al receptor. Se envía desde una MTSE de salida a la MTSE de entrada par.

Se utiliza sequenceNumber para etiquetar casos de MultiplexEntrySend de forma que pueda identificarse la respuesta correspondiente.

MultiplexEntryDescriptors es un conjunto de 1 a 15 MultiplexEntryDescriptor.

**MultiplexEntryDescriptor:** describe una única entrada de tabla múltiplex. Comprende el MultiplexTableEntryNumber y una lista de MultiplexElement. La ausencia de la lista de elementos indica que está desactivada la inscripción.

**MultiplexElement:** es una estructura recursiva que describe un único elemento y una cuenta repetitiva. Si el elemento es del tipo logicalChannelNumber, expresa un único intervalo de tiempo del canal lógico dado y la cuenta repetitiva indica la longitud del intervalo de tiempo en octetos. Si el elemento es del tipo subElementList, expresa una secuencia de MultiplexElement anidados y la cuenta repetitiva indica el número de veces que debe repetirse la secuencia. En cualquier caso, si el campo repeatCount es untilClosingFlag, ello significa que debe repetirse el elemento indefinidamente hasta la bandera de cierre de la MUX-PDU.

En cada MultiplexEntryDescriptor, la repeatCount del MultiplexElement final de elementList se pondrá a "untilClosingFlag" y las repeatCount de los demás MultiplexElement de elementList se pondrán en "finite". Esto asegura que todas las entradas de la tabla múltiplex definen un patrón de secuencia múltiplex de longitud indefinida que se repite hasta la bandera de cierre de la MUX-PDU. Un MultiplexEntryDescriptor al que falte el campo elementList indicará una inscripción desactivada.

Cada petición MultiplexEntrySend puede contener hasta 15 MultiplexEntryDescriptor, cada uno de los cuales describe una sola entrada de tabla múltiplex. Las entradas múltiplex pueden enviarse en cualquier orden.

#### **B.4.2 Acuse de recibo de envío de entradas múltiplex**

Se utiliza para confirmar la recepción de uno o más multiplexEntryDescriptor desde una MultiplexEntrySend de la MTSE par.

El sequenceNumber será el mismo que el sequenceNumber del MultiplexEntrySend del cual es la confirmación.

multiplexTableEntryNumber expresa las entradas de la tabla múltiplex que se confirman.

#### **B.4.3 Rechazo de envío de entradas múltiplex**

Se utiliza para rechazar uno o más multiplexEntryDescriptor de una MultiplexEntrySend desde la MTSE par.

El sequenceNumber será el mismo que el sequenceNumber del MultiplexEntrySend del cual es el rechazo.

MultiplexEntryRejectionDescriptions especifica qué entradas de la tabla se rechazan y por qué. En el cuadro B.9 se indican los motivos de rechazo.

**Cuadro B.9/H.245 – Motivos de rechazo de un MultiplexEntrySend**

<b>Punto de código ASN.1</b>	<b>Causa</b>
unspecified	No se han especificado causas del rechazo.
descriptorTooComplex	El MultiplexEntryDescriptor rebasaba la capacidad del terminal de recepción.

#### **B.4.4 Liberación de envío de entradas múltiplex**

La envía la MTSE de salida en el caso de expiración de un periodo de temporización.

multiplexTableEntryNumber indica las entradas de la tabla múltiplex a las que se ha aplicado la temporización.

## B.5 Mensajes de señalización de petición de tabla múltiplex

Este conjunto de mensajes se utiliza para la petición segura de retransmisión de uno o más MultiplexEntryDescriptor del transmisor al receptor.

### B.5.1 Petición de entrada múltiplex

Se utiliza para solicitar la retransmisión de uno o más MultiplexEntryDescriptor.

entryNumbers es una lista de los MultiplexTableEntryNumber de los MultiplexEntryDescriptor cuya retransmisión se ha pedido.

### B.5.2 Acuse de recibo de petición de entrada múltiplex

Lo utiliza la RMESE de entrada para indicar que se transmitirá la entrada múltiplex.

entryNumbers es una lista de los MultiplexTableEntryNumber de los MultiplexEntryDescriptor que se transmitirán.

### B.5.3 Rechazo de petición de entrada múltiplex

Lo utiliza la RMESE de entrada para indicar que no se transmitirá la entrada múltiplex.

entryNumbers es una lista de los MultiplexTableEntryNumber de los MultiplexEntryDescriptor que no se transmitirán. Los valores de MultiplexTableEntryNumber en entryNumbers deben concordar con los valores de MultiplexTableEntryNumber en rejectionDescriptions; de lo contrario pueden producirse errores durante la operación.

RequestMultiplexEntryRejectionDescriptions especifica las entradas de la tabla que se rechazan y el motivo por el que se rechazan. Las causas de rechazo se indican en el cuadro B.10.

**Cuadro B.10/H.245 – Motivos para el rechazo de un MultiplexEntrySend**

Punto de código ASN.1	Causa
unspecified	No se especifica causa de rechazo.

### B.5.4 Liberación de petición de entrada múltiplex

Lo envía la RMESE de salida cuando ha expirado un periodo de temporización.

entryNumbers es una lista de los MultiplexTableEntryNumber de los MultiplexEntryDescriptor para los que ha expirado el periodo de temporización.

## B.6 Mensajes de petición de modo

Un terminal receptor utiliza este conjunto de mensajes para solicitar del terminal transmisor modos de transmisión determinados.

### B.6.1 Petición de modo

Se utiliza para solicitar modos concretos de transmisión del terminal transmisor. Se trata de una lista, en orden de preferencia (comenzando por el que se prefiere en primer término), de los modos que el terminal desearía recibir. Se describe cada modo mediante un ModeDescription.

Se emplea sequenceNumber para etiquetar ejemplares de RequestMode de forma que pueda identificarse la respuesta correspondiente.

**ModeDescription:** conjunto de uno o más ModeElement.

**ModeElement:** se utiliza para describir un elemento de modo, es decir una de las partes constituyentes de una descripción de modo completa. Indica el tipo de tren elemental solicitado y, facultativamente, cómo se solicita su multiplexación.

Se utiliza *type* para indicar el tipo de tren elemental solicitado. Es una elección entre *VideoMode*, *AudioMode*, *DataMode*, *EncryptionMode* y *H235Mode*. *H235Mode* indica que se piden medios criptados.

*multiplexedStreamMode* indica el modo de transmisión del tren multiplexado requerido. Los campos de *MultiplexedStream* tienen el mismo significado que los campos del mismo nombre en *MultiplexedStreamCapability*.

**h223ModeParameters**: se emplea para expresar los parámetros propios de la utilización de la Rec. UIT-T H.223 [10].

*adaptationLayerType* indica la capa de adaptación y opciones solicitadas para el tipo solicitado. Los puntos de código son: *nonStandard*, *al1Framed* (modo AL1 tramado), *al1NotFramed* (modo AL1 no tramado), *al2WithoutSequenceNumbers* (AL2 sin números de secuencia), *al2WithSequenceNumbers* (AL2 con números de secuencia), *al3* (AL3, indicando el número de octetos del campo de control que estarán presentes y el tamaño de la memoria tampón de transmisión,  $B_S$ , que se utilizará, midiéndose dicho tamaño en octetos), *al1M* (A11M se define en el anexo C/H.223 con parámetros especificados), *al2M* (A12M se define en el anexo C/H.223 con parámetros especificados), y *al3M* (A13M se define en el anexo C/H.223 con parámetros especificados).

*segmentableFlag*, cuando toma el valor verdadero, indica que se ha solicitado la multiplexación segmentable y, cuando toma el valor falso, indica que se ha solicitado la multiplexación no segmentable.

**h2250ModeParameters** contiene información específica para la utilización junto con las Recs. UIT-T H.225.0 y H.323.

*redundancyEncodingMode* (si está presente) especifica qué *redundancyEncodingMethod* deberá utilizarse y qué *secondaryEncoding* deberá utilizarse como codificación por redundancia. La codificación primaria se especifica mediante el elemento *type* contenido en el *ModeElement*.

**genericModeParameters** indica parámetros de modo genérico.

*multiplexedStreamModeParameters* indica el canal lógico del tren multiplexado en el que se aplica esta petición de modo: el canal lógico se identifica por el campo *logicalChannelNumber*.

**logicalChannelNumber**: si está presente, indica el canal lógico para el cual se solicita el modo especificado. *logicalChannelNumber* sólo se debe utilizar para especificar un canal lógico abierto.

### B.6.1.1 Modo vídeo

Se trata de una elección de *VideoMode*.

**H261VideoMode**: indica la resolución de imagen (QCIF o CIF), velocidad binaria, en unidades de 100 bit/s, y transmisión de imágenes fijas solicitadas.

**H262VideoMode**: indica el perfil y nivel solicitados y los campos facultativos que, si están presentes, indican los valores solicitados de los parámetros proporcionados. Los campos facultativos son números enteros con las unidades definidas en el cuadro B.2.

**H263VideoMode**: indica la resolución de imagen (SQCIF, QCIF, CIF, 4 veces CIF y 16 veces CIF o algún formato de imagen personalizado), y velocidad binaria, en unidades de 100 bit/s, solicitadas. Cuando se comunica con un punto extremo que soporta la Rec. UIT-T H.245 versión 8 o anterior, no es posible solicitar sólo un formato de imagen personalizado. Por tanto, cuando se recibe *RequestMode* de un punto extremo que soporta la Rec. UIT-T H.245 versión 8 o anterior, si el *RequestMode* contiene un formato de imagen personalizado se debe considerar la resolución solicitada en lugar de la resolución indicada en el campo resolución de *H263VideoMode*.

Las variables booleanas `unrestrictedVector`, `arithmeticCoding`, `advancedPrediction` y `pbFrames`, cuando toman el valor verdadero, indican que se ha solicitado el empleo de estos modos facultativos que se definen en los anexos de la Rec. UIT-T H.263.

La variable booleana `errorCompensation`, cuando toma el valor verdadero, indica que el codificador es capaz de procesar indicaciones `videoNotDecodedMB` y compensar errores como se ilustra en el apéndice I/H.263. No se requiere que el codificador responda a indicaciones `videoNotDecoded`. En una unidad de control multipunto (MCU), puede que, en la práctica, no convenga que la MCU responda a todas las indicaciones.

**EnhancementOptions:** indica los parámetros de capa de mejora de escalabilidad pedidos.

**H263Options:** indica los modos opcionales de la Rec. UIT-T H.263 pedidos.

**IS11172VideoMode:** indica la petición de `constrainedBitstream`, y los campos facultativos, si están presentes, indican los valores solicitados de los parámetros dados. Los campos facultativos contienen números enteros con las unidades definidas en el cuadro B.3.

**genericVideoMode** indica parámetros de modo vídeo genérico.

### B.6.1.2 Modo audio

Se trata de una elección de `AudioMode`.

En el cuadro B.4 se facilita el significado exacto de los puntos de código de audio de la serie G. Hay cuatro posibilidades para el audio G.723.1, que permiten la elección de velocidades de bits (velocidad de bits reducida de 5,3 kbit/s o velocidad de bits elevada de 6,3 kbit/s) con el empleo o no de la supresión de los periodos de silencio.

**G7231AnnexCMode:** se utiliza para solicitar audio, codificado de acuerdo con el anexo C/G.723.1. `maxAl-sduAudioFrames` indica el número máximo solicitado de tramas audio por cada AL-SDU. La variable booleana `silenceSupression`, cuando toma el valor verdadero, solicita la utilización de la compresión de periodos de silencio definida en el anexo A/G.723.1. Los campos de `g723AnnexCAudioMode`, `highRateMode0`, `highRateMode1`, `lowRateMode0`, `lowRateMode1`, `sidMode0` y `sidMode1` indican el número requerido de octetos por trama para cada uno de los modos de audio y de protección contra errores de la Rec. UIT-T G.723.1 y anexo C/G.723.1.

**IS11172AudioMode:** se utiliza para solicitar audio codificado según ISO/CEI 11172-3 [45].

`audioLayer` indica la capa de codificación solicitada: `audioLayer1`, `audioLayer2` o `audioLayer3`.

`audioSampling` indica la velocidad de muestreo solicitada: `audioSampling32k`, `audioSampling44k1` y `audioSampling48k` indican velocidades de muestreo de audio de 32 kHz, 44,1 kHz y 48 kHz, respectivamente.

`multichannelType` indica el modo multicanal solicitado: `singleChannel`, `twoChannelStereo` y `twoChannelDual` que corresponden a un funcionamiento con un solo canal, con dos canales en estéreo o con dos canales en doble, respectivamente.

`bitRate` indica la velocidad de bits de audio solicitada, medida en unidades de kbit/s.

**IS13818AudioMode:** se utiliza para solicitar audio codificado según ISO/CEI 13818-3 [46].

`audioLayer` indica la capa de codificación solicitada: `audioLayer1`, `audioLayer2` o `audioLayer3`.

`audioSampling` indica la velocidad de muestreo solicitada: `audioSampling16k`, `audioSampling22k05`, `audioSampling24k`, `audioSampling32k`, `audioSampling44k1` y `audioSampling48k` expresan velocidades de muestreo de audio de 16 kHz, 22,05 kHz, 24 kHz, 32 kHz, 44,1 kHz y 48 kHz, respectivamente.

`multichannelType` indica el modo multicanal solicitado según se especifica en el cuadro B.11.

**Cuadro B.11/H.245 – Códigos de punto multicanal de ISO/CEI 13818-3**

<b>Punto de código ASN.1</b>	<b>Significado semántico de punto de código</b>
singleChannel	Un canal, con la configuración 1/0. Modo de canal único (como en ISO/CEI 11172-3)
twoChannelStereo	Dos canales, con la configuración 2/0, modo canal estéreo (como en ISO/CEI 11172-3)
twoChannelDual	Dos canales, con la configuración 2/0, modo de dos canales (como en ISO/CEI 11172-3)
threeChannels2-1	Tres canales, con la configuración 2/1, canal izquierdo, derecho y canal ambiente único
threeChannels3-0	Tres canales, con la configuración 3/0. Canal izquierdo, central y derecho sin canal ambiente
fourChannels2-0-2-0	Cuatro canales, con la configuración 2/0 + 2/0. Canales izquierdo y derecho del primer programa e izquierdo y derecho del segundo programa
fourChannels2-2	Cuatro canales, con la configuración 2/2. Canales izquierdo, derecho, ambiente izquierdo y ambiente derecho
fourChannels3-1	Cuatro canales, con la configuración 3/1. Canales izquierdo, central, derecho y canal ambiente único
fiveChannels3-0-2-0	Cinco canales, con la configuración 3/0 + 2/0. Canales izquierdo, central y derecho del primer programa, e izquierdo y derecho del segundo programa
fiveChannels3-2	Cinco canales, con la configuración 3/2. Canales izquierdo, central, derecho, ambiente izquierdo y ambiente derecho

La variable booleana `lowFrequencyEnhancement`, cuando toma el valor verdadero, solicita un canal de intensificación de baja frecuencia.

La variable booleana `multilingual`, cuando toma el valor verdadero, solicita hasta siete canales multilingües.

`bitRate` expresa la velocidad de bits de audio solicitada, medida en unidades de kbit/s.

`genericAudioMode` indica parámetros de modo audiógenérico.

### **B.6.1.3 Modo datos**

Se trata de una elección de aplicaciones de datos y velocidades binarias.

`bitRate` indica la velocidad binaria solicitada en unidades de 100 bit/s.

`t120` solicita la utilización del protocolo T.120 [32].

`dsm-cc` solicita la utilización del protocolo DSM-CC [47].

`userData` solicita la utilización de datos de usuario no especificados procedentes de puertos de datos externos.

`t84` solicita la utilización de la Rec. UIT-T T.84 [31] para la transferencia de esas imágenes (JPEG, JBIG, facsímil Gr.3/4).

`t434` solicita el uso de la Rec. UIT-T T.434 [35] para la transferencia de ficheros binarios telemáticos.

`h224` solicita la utilización del protocolo H.224 [11] de control de dispositivos símplex en tiempo real.

`nlpid` solicita la utilización de la aplicación de datos de capa de enlace de red especificada.

dsvdControl solicita la utilización del terminal DSVD para el soporte de un canal de control fuera de banda.

h222DataPartitioning solicita la utilización modificada y restringida de las particiones de datos de H.262, como se especifica en la Rec. UIT-T H.222.1, en las cuales los datos intensificados se transmiten como un canal de datos soportado por la DataProtocolCapability listada.

t30fax solicita la utilización del modo analógico del anexo C/T.30 (G3V), especificado en la Rec. UIT-T T.39 para los modos DSVF/MSVF.

t140 solicita la utilización del protocolo de conversación de texto, especificado en la Rec. UIT-T T.140.

t38fax solicita la utilización de la Rec. UIT-T T.38 [29].

**genericDataMode:** indica parámetros de modo datos genérico. Cuando maxBitRate esté incluida en el genericDataMode, su valor será el mismo que el de maxBitRate en el DataMode.

#### **B.6.1.4 Modo criptación**

Se trata de una elección de modos de criptado.

h233Encryption solicita el uso de criptación de conformidad con las Recs. UIT-T H.233 [14] y H.234 [15].

#### **B.6.2 Acuse de recibo de petición de modo**

Se envía para confirmar que el terminal transmisor intenta transmitir en uno de los modos solicitados por el terminal receptor.

El sequenceNumber será el mismo que el sequenceNumber del RequestMode del cual es la confirmación.

El campo de respuesta indica la acción del terminal distante. En el cuadro B.12 se indican los posibles valores de respuesta.

**Cuadro B.12/H.245 – Respuestas de confirmación a petición de modo**

<b>Punto de código ASN.1</b>	<b>Respuesta</b>
willTransmitMostPreferredMode	El terminal transmisor pasará al modo preferido en primer término por el receptor.
willTransmitLessPreferredMode	El terminal transmisor pasará a uno de los modos preferidos del receptor, pero no al preferido en primer término.

#### **B.6.3 Rechazo de petición de modo**

Se envía para rechazar la petición hecha por el terminal receptor.

El sequenceNumber será el mismo que el sequenceNumber en el RequestMode para el cual éste es la respuesta.

El campo de causa indica el motivo del rechazo del modo solicitado. En el cuadro B.13 se indican los valores de causa.



**Cuadro B.13/H.245 – Respuestas de rechazo a petición de modo**

<b>Punto de código ASN.1</b>	<b>Respuesta</b>
modeUnavailable	El terminal transmisor no modificará su modo de transmisión ya que los modos solicitados no están disponibles.
multipointConstraint	El terminal transmisor no modificará su modo de transmisión debido a una limitación multipunto.
requestDenied	El terminal transmisor no modificará su modo de transmisión.

#### **B.6.4 Liberación de petición de modo**

Se utiliza la MRSE de salida en caso de expiración de un periodo de temporización.

#### **B.7 Mensajes de retardo de ida y vuelta**

Un terminal utiliza este conjunto de mensajes para determinar el retardo de ida y vuelta entre dos terminales en comunicación. Permite también a un usuario H.245 determinar si existe la entidad de protocolo H.245 par.

##### **B.7.1 Petición de retardo de ida y vuelta**

Se envía desde la RTDSE de salida a la RTDSE de entrada.

Se utiliza sequenceNumber para etiquetar ejemplares de RoundTripDelayRequest de forma que pueda identificarse la respuesta correspondiente.

##### **B.7.2 Respuesta de retardo de ida y vuelta**

Se envía desde la RTDSE de entrada a la RTDSE de salida.

El sequenceNumber será el mismo que el sequenceNumber de RoundTripDelayRequest para el cual es la respuesta.

#### **B.8 Mensajes de bucle de mantenimiento**

Un terminal utiliza este conjunto de mensajes para ejecutar funciones de bucle de mantenimiento.

##### **B.8.1 Petición de bucle de mantenimiento**

Se envía para solicitar un determinado tipo de conexión en bucle. Los tipos mediaLoop y logicalChannelLoop solicitan la conexión en bucle de un único canal lógico indicado por logicalChannelNumber, en tanto que el tipo systemLoop se refiere a la totalidad de los canales lógicos. La definición exacta de estos tipos es característica del sistema y queda fuera del alcance de esta Recomendación.

##### **B.8.2 Acuse de recibo de bucle de mantenimiento**

Se utiliza para confirmar que el terminal establecerá el bucle en la forma pedida.

##### **B.8.3 Rechazo de bucle de mantenimiento**

Se utiliza para indicar que el terminal no establecerá el bucle solicitado.

Un terminal puede utilizar la causa canNotPerfomLoop para indicar que no tiene la capacidad para establecer el bucle solicitado.

##### **B.8.4 Instrucción de desconexión de bucle de mantenimiento**

Al recibir esta instrucción el terminal desconectará todos los bucles y restituirá los trayectos de audio, vídeo y datos a su condición normal.

## B.9 Mensajes de modo de comunicación

Este conjunto de mensajes lo utiliza una MC H.323 para indicar el modo de comunicación de una conferencia H.323.

### B.9.1 Instrucción de modo de comunicación

La **CommunicationModeCommand** la envía una MC H.323 para especificar el modo de comunicación para cada tipo de medio, a saber, unidifusión o multidifusión. Esta instrucción puede provocar la conmutación entre conferencia centralizada y conferencia descentralizada e implicar, por ello, el cierre de todos los canales lógicos existentes y la apertura de nuevos canales lógicos.

La **CommunicationModeCommand** especifica todas las sesiones de la conferencia. Para cada sesión se especifican los siguientes datos: el identificador de sesión RTP, el ID de sesión RTP asociado si procede, una etiqueta de terminal si procede, una descripción de la sesión, el tipo de datos de la sesión (por ejemplo, G.711) y una dirección unidifusión o multidifusión para los canales de medios y de control de medios, según convenga, de acuerdo con la configuración y el tipo de conferencia. Si se utiliza la codificación por redundancia, la **communicationModeTableEntry** especifica también el **redundancyEncodingMethod** así como el formato de codificación secundaria.

La **CommunicationModeCommand** lleva los modos de transmisión que han de utilizar los puntos extremos de conferencia de una conferencia. La instrucción no lleva modos de recepción, ya que son especificados por las instrucciones **OpenLogicalChannel** enviadas desde la MC de los puntos extremos.

Se supone que la **CommunicationModeCommand** define los modos de una conferencia y, por tanto, se envía después de la indicación **multipointConference** con la que se notifica un punto extremo que debe cumplir las instrucciones de la MC. Los puntos extremos deben esperar una **CommunicationModeCommand** antes de abrir canales lógicos cuando hayan recibido una indicación **multipointConference**.

Los puntos extremos que reciben una **CommunicationModeCommand**, utilizan el campo **terminalLabel** de cada entrada de la tabla para determinar si la entrada es aplicable a su propio procesamiento. Las entradas que no contienen una **terminalLabel** se aplican a todos los puntos extremos de la conferencia. Las entradas que contienen **terminalLabel** son instrucciones a puntos extremos específicos y que concuerdan con la **terminalLabel** de la entrada. Por ejemplo, cuando se sitúen trenes de audio procedentes de todos los puntos extremos en una dirección (o sesión) multidifusión, la entrada de la tabla para el modo audio, la dirección de los medios y la dirección de control de los medios no contendrá una **terminalLabel**. Cuando la entrada de la tabla ordene a un punto extremo que envíe su vídeo a una dirección multidifusión, la MC incluirá la **terminalLabel** de ese punto extremo.

La MC fija **sessionDependency** para indicar cuándo una sesión depende de otra sesión a efectos de una decodificación significativa de los datos.

El campo **destination** en la **CommunicationModeTableEntry** indica el punto extremo al que el punto extremo transmisor debe abrir un canal lógico. Si el campo **destination** existe en la **CommunicationModeTableEntry**, el punto extremo utilizará el destino como campo **destination** en **H2250LogicalChannelParameters** del mensaje **OpenLogicalChannel**.

La **CommunicationModeCommand** se puede utilizar para ordenar a los puntos extremos de una conferencia (o a una llamada punto a punto) que cambien de modos (indicando un nuevo modo con el **mediaChannel** ya utilizado) o que transmitan a una nueva dirección (indicando el modo utilizado en esos momentos, pero con un nuevo **mediaChannel**). De manera similar, un punto extremo que reciba una **CommunicationModeCommand** en la que se indique el modo que se está utilizando y no un **mediaChannel** deberá cerrar el canal que proceda y tratar a continuación de reabrir un nuevo canal utilizando los procedimientos **OpenLogicalChannel**, en donde el **OpenLogicalChannelAck** contiene la dirección a la que el punto extremo enviará el medio.

## **B.9.2 Petición de modo de comunicación**

La envía el MC para solicitar el modo de comunicación de la conferencia actual.

## **B.9.3 Respuesta a modo de comunicación**

La envía el MC en respuesta a una CommunicationModeRequest, para especificar el modo de comunicación de una conferencia.

## **B.10 Mensajes de petición y respuesta de conferencia**

TerminalID, que se utiliza en los mensajes de petición y respuesta de conferencia, tiene una longitud de 128 octetos. En el caso de la comunicación entre un terminal H.323 y un terminal H.320 a través de una pasarela H.323, este campo será truncado a 32 octetos.

### **B.10.1 Petición de lista de terminales**

Esta petición es equivalente a la TCU de la H.230, descrita en la Rec. UIT-T H.243.

### **B.10.2 Respuesta a lista de terminales**

Esta respuesta es equivalente a una secuencia de terminalNumber, descrita en la Rec. UIT-T H.230.

### **B.10.3 Concédame la presidencia**

Esta petición es equivalente a CCA, descrita en la Rec. UIT-T H.230.

### **B.10.4 Anulación de concédame la presidencia**

Esta petición es equivalente a CIS, descrita en la Rec. UIT-T H.230.

### **B.10.5 Respuesta a concédame la presidencia**

Esta petición es equivalente a la CIT de H.230 si se concede el testigo del control de la dirección de la conferencia o la CCR de H.230 si se deniega el testigo del control de la dirección de la conferencia.

### **B.10.6 Separación de terminal**

Esta petición es equivalente a CCD, descrita en la Rec. UIT-T H.230.

### **B.10.7 Rechazo de separación de terminal**

Esta respuesta es equivalente a CIR, descrita en la Rec. UIT-T H.230.

### **B.10.8 Petición de identificación de terminal**

Esta petición es equivalente a TCP, descrita en la Rec. UIT-T H.230.

### **B.10.9 Respuesta a identificación de terminal MC**

Esta respuesta es equivalente a TIP, descrita en la Rec. UIT-T H.230.

### **B.10.10 Petición de introducción de contraseña H.243**

Esta petición es equivalente a TCS1, descrita en la Rec. UIT-T H.230.

### **B.10.11 Respuesta a contraseña**

Esta respuesta es equivalente a IIS, descrita en la Rec. UIT-T H.230.

### **B.10.12 Petición de introducción de identificación de terminal H.243**

Esta petición es equivalente a TCS2/TCI, descrita en la Rec. UIT-T H.230.

### **B.10.13 Respuesta a identificación de terminal**

Esta respuesta es equivalente a IIS, descrita en la Rec. UIT-T H.230.

### **B.10.14 Petición de introducción de identificación de conferencia H.243**

Esta petición es equivalente a TCS3, descrita en la Rec. UIT-T H.230.

### **B.10.15 Respuesta a identificación de conferencia**

Esta respuesta es equivalente a IIS, descrita en la Rec. UIT-T H.230.

### **B.10.16 Rechazo de instrucción de vídeo**

Esta petición es equivalente a VCR descrita en la Rec. UIT-T H.230.

### **B.10.17 Petición de introducción de dirección de extensión**

Esta petición es equivalente a TCS4, descrita en la Rec. UIT-T H.230.

### **B.10.18 Respuesta a dirección de extensión**

Esta respuesta es equivalente a IIS, descrita en la Rec. UIT-T H.230.

### **B.10.19 Petición al propietario del testigo de control por la presidencia**

Esta petición es equivalente a TCA, descrita en la Rec. UIT-T H.230 para el testigo de control por la presidencia.

### **B.10.20 Respuesta del propietario del testigo de la presidencia**

Esta respuesta es equivalente a TIR, descrita en la Rec. UIT-T H.230 para el testigo de control por la presidencia.

### **B.10.21 Petición de certificado de terminal**

Esta petición es emitida por cualquier punto extremo de una conferencia a su MC. Permite a un punto extremo obtener el certificado digital para el usuario en un determinado terminal. El terminal peticionario puede incluir, facultativamente, su propio terminalCertificate y una challengeString que es criptada con la clave privada.

CertSelectionCriteria define un conjunto de certificados que son aceptables para el peticionario. El respondedor (la MC) deberá tratar de satisfacer esos criterios. CertSelectionCriteria puede estar presente junto con terminalLabel. En este caso, el MC puede utilizar los criterios para seleccionar un certificado apropiado de entre los presentados por el terminal especificado o pedir al terminal especificado un certificado que concuerde con los criterios, que a continuación devuelve al terminal original solicitante.

Esta respuesta puede devolver el certificado original y, facultativamente, una signatura asociada con el certificado de acuerdo con lo que sigue:

- Si la fuente de la terminalCertificateResponse no tiene un certificado adecuado, este mensaje puede ser devuelto sin certificado (y, por consiguiente, sin estructura certificateResponse).
- Si un punto extremo pide el certificado de otro punto extremo en una conferencia multipunto (indicado por la terminalLabel), la MC respondedora deberá devolver un certificado asociado al punto extremo solicitado (contenido en la estructura certificateResponse).
- La estructura certificateResponse deberá estar presente. En el caso de que el MC presente el certificado al peticionario en nombre de otro punto extremo, deberá haber un enlace criptográfico entre las signaturas y la del MC. Esto se hará de una de las dos maneras siguientes:

- Se utilizará la clave privada empleada para proteger el material clave de la sesión distribuido en el intercambio más reciente.
- Si no se ha intercambiado material clave, o la clave no es la adecuada para la firma, el certificado utilizado durante la autenticación punto extremo MC más reciente será la fuente de la clave privada.

#### **B.10.22 Respuesta de certificado de terminal**

Esta respuesta devuelve el certificado digital y una responseString que está criptada con la clave privada de un terminal específico.

#### **B.10.23 Difunde mi canal lógico**

Esta petición es similar a MCV H.230 utilizada conforme al procedimiento indicado en 6.3.2.2/H.243, pero se refiere a un solo canal lógico y tiene un mensaje de respuesta broadcastMyLogicalChannelResponse que acusa recibo de la petición. Se observa que cuando se aplica el procedimiento MCV en 6.3.2.1/H.243 (es decir, cuando el extremo de un terminal MCU o el enlace entre MCU no tiene multipointVisualizationCapability), se utiliza entonces la forma conferenceCommand de BroadcastMyLogicalChannel.

#### **B.10.24 Respuesta a difunde mi canal lógico**

Da una respuesta de concedido o denegado a la petición BroadcastMyLogicalChannel (difundir mi canal lógico).

#### **B.10.25 Haz al terminal difundir**

Esta petición es similar a VCB de H.230 y tiene como mensaje de respuesta makeTerminalBroadcasterResponse que acusa recibo de la petición.

#### **B.10.26 Respuesta a haz al terminal difundir**

Da una respuesta de concedido o denegado a la petición MakeTerminalBroadcaster (haz al terminal difundir).

#### **B.10.27 Envía esta fuente**

Esta petición es similar a VCS H.230 y tiene como mensaje de respuesta SendThisSourceResponse que acusa de recibo de la petición.

#### **B.10.28 Respuesta a envía esta fuente**

Da una respuesta de concedido o denegado a la petición SendThisSource (envía esta fuente).

#### **B.10.29 Petición de las identificaciones de todos los terminales**

Enviado por un punto extremo a la MC de una conferencia para obtener las etiquetas y las identificaciones de todos los terminales de los participantes en la conferencia.

#### **B.10.30 Respuesta a petición de identificaciones de todos los terminales**

Respuesta a RequestAllTerminalID (petición de las identificaciones de todos los terminales) que contiene una lista de todos los puntos extremos de la conferencia por terminalLabel y terminalID.

#### **B.10.31 Petición a MC distante**

Esta petición es enviada por una MC activa a otra MC para activarla/desactivarla. La petición de la MC distante con elección de masterActivate o slaveActivate puede ser enviada por una MC activa a una MC inactiva para activarla como principal o subordinada, respectivamente, de una conexión en cascada. La petición de la MC distante con elección de deactivate puede ser enviada por una MC principal a una MC subordinada ya activa para desactivarla.

### **B.10.32 Respuesta de la MC distante**

La respuesta de la MC distante se envía para indicar aceptación o rechazo de la petición de la MC distante. La aceptación de esta petición viene determinada por los siguientes criterios:

Elección = activateSlave

El receptor está inactivo y el emisor de la petición inició esta llamada con un conferenceGoal (objeto de la conferencia) de INVITE (invitación) en el mensaje Setup (establecimiento) H.225 o el receptor de la petición inicia esta llamada con un conferenceGoal de JOIN (incorporación) en el mensaje Setup (establecimiento) H.225.

Elección = activateMaster

El receptor está inactivo y el emisor de la petición inició esta llamada con un conferenceGoal (objeto de conferencia) de CREATE (creación) en el mensaje Setup (establecimiento) H.225.

Elección = deActivate

El receptor es una MC activa.

Si las condiciones indicadas más arriba no se cumplen, la petición deberá ser rechazada con la elección de invalidConfiguration.

La elección rechazo de functionNotSupported es utilizada por los puntos extremos que no soportan la configuración en cascada.

### **B.11 Mensajes multienlace**

Los mensajes multilinkRequest, multilinkResponse y multilinkIndication se utilizan para soportar la utilización de la agregación de canales de acuerdo con la Rec. UIT-T H.226, según se especifica en el anexo F/H.324. Con estos mensajes se pueden establecer y eliminar conexiones físicas, intercambiar de manera automática direcciones de red (números telefónicos) y controlar la operación H.226.

#### **B.11.1 Petición y respuesta de callInformation**

MultilinkRequest.callInformation es utilizado por el iniciador, según se define en el anexo F/H.324, para pedir la información necesaria para establecer y asociar conexiones físicas adicionales. El número máximo de conexiones adicionales que el emisor es capaz de establecer se envía en el parámetro maxNumberOfAdditionalConnections.

El mensaje MultilinkResponse.callInformation incluye el parámetro DialingInformation, con el contenido que se describe más adelante, así como un callAssociationNumber. El callAssociationNumber deberá contener un número aleatorio de 32 bits distribuidos uniformemente. Cualesquiera intercambios subsiguientes de callInformation dentro de la misma sesión deberán utilizar el mismo callAssociationNumber.

#### **B.11.2 Petición y respuesta de addConnection**

El mensaje MultilinkRequest.addConnection puede ser utilizado por el respondedor, según se define en el anexo F/H.324, para pedir al iniciador que añada conexiones físicas. La estructura DialingInformation indicará las conexiones que se han de añadir. El parámetro sequenceNumber deberá aumentarse en 1, módulo 256, por cada mensaje MultilinkRequest.addConnection enviado.

Al recibir este mensaje, el iniciador responderá con un mensaje MultilinkResponse.addConnection indicando si va a añadir las conexiones solicitadas, o si no va a añadirlas, junto con el código de motivo apropiado. El parámetro sequenceNumber deberá ser igual al parámetro sequenceNumber del mensaje MultilinkRequest.addConnection correspondiente.

### **B.11.3 Petición y respuesta de removeConnection**

El mensaje MultilinkRequest.removeConnection puede ser utilizado por el iniciador o el respondedor, según se define en el anexo F/H.324, para pedir al extremo distante que elimine un canal del conjunto de canales H.226. Se utiliza como parte del procedimiento del anexo F/H.324 de eliminación de conexiones físicas. El parámetro connectionIdentifier identificará el canal que ha de ser eliminado, utilizando la numeración de canales recibida vía H.226 del terminal que recibe el mensaje MultilinkRequest.removeConnection.

En respuesta se enviará el mensaje MultilinkResponse.removeConnection, una vez eliminado el canal del conjunto de canales H.226, indicando que este canal ya no se utiliza (o nunca se utilizó). El parámetro connectionIdentifier deberá ser idéntico al valor que figure en el mensaje MultilinkRequest.removeConnection correspondiente.

### **B.11.4 Petición y respuesta de maximumHeaderInterval**

El mensaje MultilinkRequest.maximumHeaderInterval puede ser utilizado para pedir el intervalo de encabezamiento máximo H.226 que está siendo utilizado realmente por el transmisor distante sin alterarlo (la opción currentIntervalInformation) o para pedir un determinado valor a utilizar en su lugar (la opción requestedInterval, con unidades en milisegundos).

En respuesta se enviará el mensaje MultilinkResponse.maximumHeaderInterval. Si la petición correspondiente fue una petición de información sobre la velocidad mínima corriente, el terminal proporcionará el valor que el transmisor utiliza en esos momentos como intervalo de encabezamiento máximo en la respuesta. Si la petición correspondiente especificaba la utilización de una velocidad mínima particular, el terminal deberá tratar de cumplir esa petición modificando el intervalo de encabezamiento máximo utilizado por su transmisor. Tanto si introduce como si no un cambio en el intervalo de encabezamiento máximo, la respuesta deberá indicar el valor nuevo que se utiliza (que puede ser diferente del valor pedido).

### **B.11.5 Indicaciones multienlace**

El mensaje MultilinkIndication.crcDesired puede ser enviado por un terminal para indicar su deseo de que el terminal distante envíe la verificación por redundancia cíclica (CRC, *cyclic redundancy check*) opcional de los datos H.226 en todos los conjuntos de datos subsiguientes. El terminal receptor puede cumplir ese deseo facultativamente: no se requiere una respuesta o un acuse de recibo explícito.

El mensaje MultilinkIndication.excessiveError puede ser enviado para indicar al terminal distante que se está recibiendo un número excesivo de errores por una conexión determinada. La manera que tiene el terminal de determinar la tasa de errores o el criterio para establecer lo que se entiende por excesivo se define localmente en ese terminal. La conexión se indica utilizando el parámetro connectionIdentifier. Al recibir este mensaje, el terminal puede optar por tomar medidas correctivas. La acción correctora que en concreto debería realizar no se especifica.

### **B.11.6 DialingInformation**

El tipo DialingInformation se utiliza para proporcionar información de marcación explícita (números telefónicos) de modo que las conexiones físicas se establezcan de manera automática. La opción diferencial proporciona una lista de parámetros DialingInformationNumber, uno para cada posible conexión adicional. La longitud de esta lista indica el número máximo de conexiones adicionales disponibles. Si no se dispone de esa información, no se utiliza la opción infoNotAvailable y se indica solamente el número de conexiones adicionales de que se dispone.

### **B.11.7 DialingInformationNumber**

El tipo DialingInformationNumber incluye hasta tres subparámetros que indican la información de marcación para una conexión física de manera diferencial con respecto a la información correspondiente a una conexión inicial ya establecida.

El parámetro `networkAddress` deberá incluir el tramo menos significativo (el situado más a la derecha) del número telefónico de esa conexión, hasta, e incluyendo, la cifra más significativa que difiera del número de la conexión establecida inicialmente y no deberá incluir ninguna cifra que sea más significativa que ésta. Si el número de la conexión adicional es idéntico al de la conexión inicial, el parámetro `networkAddress` consistirá en una cadena de longitud cero (puesto que no hay cifras que difieran en el número telefónico).

NOTA – Se utiliza el método de las cifras diferentes en vez de la cadena de cifras E.164 completa porque las primeras cifras del número que se ha de marcar pueden variar dependiendo de la ubicación geográfica de los dos terminales, por ejemplo, si están situados o no en la misma ciudad.

Si se utiliza una subdirección para marcación, y la subdirección de una conexión dada difiere de la de la conexión inicial, el respondedor incluirá la subdirección, completa, en el parámetro `subAddress` opcional.

Los tipos de red que se soportan para la conexión (RTGC, RDSI o ambos tipos) deberán indicarse utilizando el parámetro `networkType`.

#### **B.11.8 DialingInformationNetworkType**

El tipo `DialingInformationNetworkType` indica un tipo de red por conmutación de circuitos, ya sea `n-isdn` (RDSI-BE), `gstn` (RTGC) o `mobile` (móvil).

#### **B.11.9 ConnectionIdentifier**

El tipo `ConnectionIdentifier` se utiliza para identificar de manera exclusiva una sola conexión física en H.226, empleando una combinación de `channelTag` y `sequenceNumber` de un encabezamiento H.226. Si en el encabezamiento no se hubiera especificado ningún rótulo de canal, deberá utilizarse un valor de cero para el parámetro `channelTag`.

### **B.12 Mensajes de cambio de velocidad binaria de canal lógico**

Los mensajes `LogicalChannelRateRequest`, `LogicalChannelRateAcknowledge`, `LogicalChannelRateReject` y `LogicalChannelRateRelease` se utilizan para cambiar la velocidad binaria de un canal lógico. El procedimiento de utilización de estos mensajes consiste en que un terminal pide una velocidad binaria objetivo para un canal lógico específico y el terminal distante puede acusar recibo o rechazar la petición.

Con estos mensajes se obtiene un nivel mejorado de interacción, ya que permiten que se formulen peticiones de velocidad binaria objetivo en contraste con la aplicación forzada de una velocidad binaria máxima impuesta por la `FlowControlCommand` y proporcionan información de retorno sobre si la petición se satisface o se deniega.

#### **B.12.1 Petición de velocidad de canal lógico**

Este mensaje es utilizado por un terminal para pedir un cambio de la velocidad binaria del canal lógico preciso que se le está transmitiendo.

`sequenceNumber` se utiliza para etiquetar ejemplares de `LogicalChannelRateRequest` de manera que la respuesta correspondiente pueda ser identificada.

`logicalChannelNumber` indica el canal lógico al que se aplica la petición de cambio de la velocidad binaria.

`maximumBitRate` indica, en unidades de 100 bit/s, la velocidad binaria máxima pedida para el canal lógico.

#### **B.12.2 Acuse de recibo de petición de canal lógico**

Este mensaje se envía para acusar recibo de una petición de cambio de velocidad binaria de un canal lógico.



El `sequenceNumber` deberá ser el mismo `sequenceNumber` del mensaje `LogicalChannelRateRequest` al que éste responde.

`logicalChannelNumber` indica el canal lógico al que se aplica la petición de cambio de velocidad binaria.

`maximumBitRate` indica, en unidades de 100 bit/s, la velocidad binaria máxima del canal lógico de la que el terminal está acusando recibo.

### **B.12.3 Rechazo de velocidad de canal lógico**

Este mensaje se envía para rechazar una petición de cambio de la velocidad binaria de un canal lógico.

El `sequenceNumber` deberá ser el mismo `sequenceNumber` del mensaje `LogicalChannelRateRequest` al que éste responde.

`logicalChannelNumber` indica el canal lógico al que se aplica la petición de cambio de velocidad binaria.

`rejectReason` indica el motivo por el que se ha denegado la petición. Motivos definidos corrientes son por causa no especificada o por insuficiencia de recursos.

`currentMaximumBitRate` indica, en unidades de 100 bit/s, la velocidad binaria máxima a la que el terminal va a transmitir por el canal lógico.

### **B.12.4 Liberación de velocidad de canal lógico**

Este mensaje se envía en el caso de que expire un periodo de temporización.

## **B.13 Instrucciones**

Un mensaje de instrucción requiere una acción pero no una respuesta explícita.

### **B.13.1 Envío del conjunto de capacidades de terminal**

`specificRequest` ordena al terminal del extremo distante que indique sus capacidades de transmisión y recepción enviando uno o más `TerminalCapabilitySet` que contienen la información solicitada, como se indica seguidamente. Esta instrucción puede enviarse en cualquier momento para expresar las capacidades del terminal distante, por ejemplo, tras una interrupción o cualquier otra causa de incertidumbre. Sin embargo, no deberán enviarse repetidamente esos mensajes sin una causa justificada.

Un terminal sólo pedirá la transmisión de `capabilityTableEntryNumbers` y `capabilityDescriptorNumbers` que haya recibido previamente. Un terminal ignorará toda petición de transmisión de `capabilityTableEntryNumbers` y `capabilityDescriptorNumbers` que no haya transmitido previamente y no se considerará que se haya producido ningún fallo.

La variable booleana `multiplexCapability`, cuando toma el valor verdadero, solicita la transmisión de la `MultiplexCapability`.

`capabilityTableEntryNumbers` es un conjunto de `CapabilityTableEntryNumbers` que indica las `CapabilityTableEntry` que el terminal solicita que se transmitan.

`capabilityDescriptorNumbers` es un conjunto de los `capabilityDescriptorNumbers` que indican los `CapabilityDescriptor` cuya transmisión solicita el terminal.

`genericRequest` ordena al terminal del extremo distante que envíe su conjunto completo de capacidades de terminal.

### B.13.2 Criptación

Se utiliza esta instrucción para intercambiar capacidades de criptación y ordenar la transmisión de un vector de inicialización (IV, *initialization vector*); véanse las Recs. UIT-T H.233 [14] y H.234 [15].

encryptionSE es un mensaje de intercambio de sesión (SE, *session exchange*) H.233 con la salvedad de que no se aplicarán los bits de protección contra errores descritos en la Rec. UIT-T H.233.

encryptionIVRequest ordena al criptador del extremo distante que transmita un nuevo IV en un canal lógico abierto para encryptionData.

encryptionAlgorithmIdentifier indica al receptor que el terminal transmisor asociará el valor h233AlgorithmIdentifier dado con el algoritmo de criptación no normalizado associatedAlgorithm.

### B.13.3 Control de flujo

Se utiliza esta instrucción para especificar el límite superior de la velocidad binaria en un solo canal lógico o en la totalidad del múltiplex. Un terminal puede enviar esta instrucción para limitar la velocidad binaria a que transmite el terminal distante. Todo terminal que reciba esta instrucción deberá cumplimentarla.

Cuando el alcance es del tipo logicalChannelNumber, el límite se aplica al canal lógico dado cuando el alcance es del tipo resourceID, el límite se aplica al canal virtual ATM dado y cuando el alcance es del tipo wholeMultiplex, el límite se aplica a la totalidad del múltiplex.

maximumBitRate se mide en unidades de 100 bit/s promediadas sobre periodos consecutivos no superpuestos de un segundo. Cuando está presente, el límite especificado sustituye cualquier límite anterior, ya sea superior o inferior. Cuando no está presente, dejará de ser aplicable toda restricción previa de la velocidad binaria del canal.

En esta Recomendación no se especifican ni el punto en el que se aplica el límite de velocidad binaria ni qué bits se incluyen en el cálculo de la velocidad binaria, pero estos aspectos deberán especificarse en Recomendaciones que hagan uso de la presente Recomendación.

Cada transmisión de esta instrucción afecta a un canal lógico concreto o a todo el múltiplex. Para la limitación múltiplex global, pueden estar activas al mismo tiempo más de una de estas instrucciones, hasta un número igual al de canales lógicos abiertos más uno.

NOTA – Cuando la velocidad binaria que pueda transmitirse por un canal lógico esté limitada a valores específicos, por ejemplo audio G.723.1, y la petición sea transmitir a una velocidad inferior a la velocidad mínima a la cual funcionaría normalmente, responderá deteniendo la transmisión por el canal lógico.

### B.13.4 Finalización de sesión

Esta instrucción indica el final de la sesión H.245. Tras la transmisión de EndSessionCommand, el terminal no enviará más ninguno de los mensajes definidos en esta Recomendación.

disconnect indica que se eliminará la conexión.

**gstnOptions:** es una elección de alternativas que pueden producirse tras la finalización de la sesión H.245, cuando se utiliza un módem de la serie V por la red telefónica general conmutada (RTGC).

En el cuadro B.14 se indican las posibles opciones.

**Cuadro B.14/H.245 – Opciones posteriores a EndSessionCommand cuando se utilice el módem V.34**

Punto de código ASN.1	Opción
telephonyMode	El terminal iniciará los procedimientos de liberación definidos en las Recomendaciones de la serie V sobre módems, con la salvedad de que no desconectará físicamente la conexión RTGC.
v8bis	El terminal iniciará los procedimientos de liberación definidos en la Recomendación sobre los módems de la serie V y dará comienzo a una sesión V.8 <i>bis</i> .
v34DSVD	El terminal conservará la conexión del módem V.34, pero la utilizará para soportar V.70.
v34DuplexFAX	El terminal conservará la conexión del módem V.34, pero la utilizará para soportar T.30 FAX [27].
v34H324	El terminal conservará la conexión del módem V.34, pero la utilizará para soportar la Rec. UIT-T H.324 [24].

**isdnOptions:** es una elección de las alternativas que pueden producirse tras la finalización de la sesión H.245, cuando se utiliza un terminal de comunicaciones digitales por una red digital.

En el cuadro B.15 se indican las posibles opciones.

**Cuadro B.15/H.245 – Opciones después de EndSessionCommand cuando se utiliza un terminal de comunicaciones digitales por una red digital**

Punto de código ASN.1	Opción
telephonyMode	El terminal iniciará los procedimientos de liberación definidos en la Recomendación aplicable a la comunicación por el canal digital al que, en concreto, está conectado el terminal, con la salvedad de que no desconectará físicamente la conexión digital.
v140	El terminal iniciará los procedimientos de liberación definidos en la Recomendación aplicable a la comunicación por el canal digital al que, en concreto, está conectado el terminal y pasará a una sesión V.140 [39].
terminalOnHold	El terminal iniciará los procedimientos de terminal retenido definidos en la Recomendación aplicable a la comunicación por el canal digital al que en concreto está conectado el terminal.

### **B.13.5 Instrucción diversos**

Se usa para diversas instrucciones, algunas de las cuales se indican en las Recs. UIT-T H.221 [7] y H.230 [13].

logicalChannelNumber indica el número de canal lógico al que se aplica la instrucción. Indicará un canal lógico abierto para datos de vídeo cuando el tipo sea uno de los siguientes videoFreezePicture, videoFastUpdatePicture, videoFastUpdateGOB, videoTemporalSpatialTradeOff, videoSendSyncEveryGOB, videoFastUpdateMB, videoSendSyncEveryGOBCancel, lostPicture, lostPartialPicture y recoveryReferencePicture. Cuando el tipo es uno de los siguientes: equaliseDelay, zeroDelay, multipointModeCommand o cancelMultipointModeCommand en que intervienen canales lógicos, el logicalChannelNumber será un número arbitrario, pero tendrá que ser un LogicalChannelNumber válido (es decir, en la gama 1-65535) y el receptor no tendrá en cuenta el valor.

equaliseDelay y zeroDelay tendrán el mismo significado que en las instrucciones ACE y ACZ definidas en la Rec. UIT-T H.230 [13].

multipointModeCommand ordena que un terminal que se encuentre en recepción cumpla con todas las peticiones requestMode emitidas por la MCU. Un ejemplo de cambio de modo es un cambio de la codificación de audio de G.711 a G.728.

cancelMultipointModeCommand anula una instrucción multipointModeCommand anteriormente enviada.

videoFreezePicture ordena al decodificador vídeo que concluya la actualización de la trama vídeo actual y presente subsiguientemente la imagen estática hasta la recepción de la señal de control de liberación de imagen estática apropiada.

videoFastUpdatePicture ordena al codificador vídeo que pase al modo de actualización rápida lo antes que pueda.

videoFastUpdateGOB ordena al codificador de vídeo del extremo distante que ejecute una actualización rápida de uno o más GOB. firstGOB indica el número del primer GOB que debe actualizarse y numberOfGOB indica el número de los GOB que han de actualizarse. Únicamente se utilizará con algoritmos de comprensión de vídeo que definan GOB, por ejemplo H.261 y H.263. La numeración de GOB se efectúa como en H.263, aun cuando se utiliza H.261. El primer GOB de la imagen es el GOB número 0, el segundo GOB es el GOB número 1, etc. La exploración de los GOB para la interpretación del parámetro numberOfGOB se efectuará conforme a la norma de codificación de vídeo pertinente, de modo que el segundo GOB en una imagen CIF H.261 está a la derecha del primero, mientras que las imágenes QCIF H.261 y H.263 están debajo del primer GOB.

videoTemporalSpatialTradeOff ordena al codificador vídeo del extremo distante que modifique su solución de compromiso entre resolución espacial y temporal. El valor 0 ordena una alta resolución espacial y el valor 31 indica una alta velocidad de trama. Los valores comprendidos entre 0 y 31 indican, sucesivamente, el deseo de una velocidad de trama más alta. Los valores reales no corresponden a valores precisos de resolución espacial o velocidad de trama.

videoSendSyncEveryGOB ordena al codificador vídeo del extremo distante que utilice la sincronización para cada GOB como se define en la Rec. UIT-T H.263 [20], hasta que se reciba la instrucción videoSendSyncEveryGOBCancel, a partir de la cual el codificador vídeo del extremo distante puede decidir la frecuencia de las sincronizaciones de los GOB. Estas instrucciones únicamente se utilizarán con señales vídeo codificadas según la Rec. UIT-T H.263.

videoFastUpdateMB ordena al codificador vídeo del extremo distante que efectúe una actualización rápida de uno o más MB, firstGOB indica el número del primer GOB que habrá de actualizarse, firstMB indica el número del primer MB que habrá de actualizarse y numberOfMB indica el número de MB que habrá que actualizarse. Sólo se utilizará con algoritmos de compresión vídeo que definen MB, por ejemplo, H.261 y H.263. Los terminales pueden responder a esta instrucción con una actualización de GOB que incluya los MB solicitados. La numeración de GOB se efectúa como en H.263, aun cuando se utiliza H.261. El primer GOB de la imagen es el GOB número 0, el segundo GOB es el GOB número 1, etc. Estará presente firstGOB o firstMB, o ambos. Cuando firstGOB está presente y firstMB está ausente, el primer macrobloque que se debe actualizar es el primer macrobloque del GOB indicado. Cuando firstGOB y firstMB están presentes, firstMB viene señalado con relación al comienzo del firstGOB indicado, de modo tal que el primer macrobloque del GOB indicado se considera macrobloque número 1. Cuando firstGOB está ausente y firstMB está presente, firstMB pertenece a la parte superior izquierda de la imagen, con el macrobloque superior izquierdo considerado como macrobloque número 1. El orden de exploración de los macrobloques en el resto del GOB y subsiguientes a ese punto se define como el orden de exploración de la norma de codificación de vídeo pertinente. Así, el orden de exploración que se inicia en el tercer GOB de una imagen CIF H.261 comienza a partir del macrobloque número 1, que es el macrobloque de la columna izquierda de la cuarta fila de la imagen, y continúa con una exploración descendente a través de tres filas de GOB, hasta llegar al macrobloque número 33 en la

undécima columna de la sexta fila y luego continúa verticalmente para comenzar la exploración del GOB siguiente en la duodécima columna de la cuarta fila.

maxH223MUXPDUsize ordena al transmisor que limite el tamaño de las MUX-PDU H.223 que está transmitiendo a un número máximo especificado de octetos.

encryptionUpdate y EncryptionUpdateRequest se utilizan para iniciar y distribuir el nuevo material de claves que se ha de utilizar en la criptación de los canales de medios indicados.

La conmutación de los medios de recepción a activado y desactivado puede ser utilizada por un MC para ordenar a un punto extremo que conmute entre un canal unidifusión y un canal multidifusión cuando el MC+MP está mezclando audio. En este caso, si el tren MC incluye el audio del terminal, el MC+MP puede conmutar el punto extremo a un tren unidifusión que contendrá una mezcla especial para el terminal con su audio eliminado.

switchReceiveMediaOff es utilizado por un MC para indicar a un punto extremo que no debería utilizarse un determinado canal lógico para medios de recepción.

switchReceiveMediaOn es utilizado por un MC para indicar a un punto extremo que no debería utilizarse un determinado canal lógico para medios de recepción.

doOneProgression ordena al codificador de vídeo que empiece a generar una secuencia de refinamiento progresivo. En este modo, el codificador produce datos de vídeo que constan de una imagen seguida de una secuencia de cero o más tramas de refinamiento de la calidad de la misma imagen. El codificador permanece en este modo hasta que constata que se ha alcanzado un nivel de fidelidad aceptable o bien se recibe la instrucción progressiveRefinementAbortOne. Además, el codificador insertará el rótulo de comienzo de segmento de refinamiento progresivo y el rótulo de final de segmento de refinamiento progresivo para marcar el comienzo y el final del refinamiento progresivo, según lo definido en la especificación de información sobre mejora suplementaria del anexo L/H.263.

doContinuousProgressions ordena al codificador de vídeo que empiece a generar secuencias de refinamiento progresivo. En este modo, el codificador produce datos de vídeo que constan de una imagen seguida de una secuencia de cero o más tramas de refinamiento de la calidad de la misma imagen. Cuando el codificador constata que se ha alcanzado un nivel de fidelidad aceptable o bien se recibe la instrucción progressiveRefinementAbortOne, deja de refinar la progresión en curso y empieza otro refinamiento progresivo de una imagen diferente. La secuencia de refinamientos progresivos continúa hasta que se recibe la instrucción progressiveRefinementAbortContinuous. Además, el codificador insertará rótulos de comienzo de segmento de refinamiento progresivo y rótulos de final de segmento de refinamiento progresivo para marcar el comienzo y el fin de cada refinamiento progresivo, según lo definido en la especificación de información sobre mejora suplementaria del anexo L/H.263.

doOneIndependentProgression ordena al codificador de vídeo que empiece una secuencia de refinamiento progresivo independiente. En este modo, el codificador produce datos de vídeo que constan de una imagen Intra seguida de una secuencia de cero o más tramas de refinamiento de la calidad de la misma imagen. El codificador permanece en este modo hasta que constata que se ha alcanzado un nivel de fidelidad aceptable o bien se recibe la instrucción progressiveRefinementAbortOne. Además, el codificador insertará el rótulo de comienzo de segmento de refinamiento progresivo y el rótulo de final de segmento de refinamiento progresivo para marcar el comienzo y el final del refinamiento progresivo, según lo definido en la especificación de información sobre mejora suplementaria del anexo L/H.263.

doContinuousIndependentProgressions ordena al codificador de vídeo que empiece a generar secuencias de refinamiento progresivo independientes. En este modo, el codificador produce datos de vídeo que constan de una imagen Intra seguida de una secuencia de cero o más tramas de refinamiento de la calidad de la misma imagen. Cuando el codificador constata que se ha alcanzado

un nivel de fidelidad aceptable o bien se recibe la instrucción `progressiveRefinementAbortOne`, deja de refinar la progresión en curso y empieza otro refinamiento progresivo independiente de una imagen diferente. La secuencia de refinamientos progresivos independientes continúa hasta que se recibe la instrucción `progressiveRefinementAbortContinuous`. Además, el terminal insertará rótulos de comienzo de segmento de refinamiento progresivo y rótulos de final de segmento de refinamiento progresivo para marcar el comienzo y el final de cada refinamiento progresivo independiente, según lo definido en la especificación de información sobre mejora suplementaria del anexo L/H.263.

`progressiveRefinementAbortOne` ordena al codificador de vídeo que termine `doOneProgression`, `doOneIndependentProgression` o el refinamiento progresivo en curso de la secuencia de refinamientos progresivos ya sea en `doContinuousProgressions` o en `doContinuousIndependentProgressions`.

`progressiveRefinementAbortContinuous` ordena al codificador de vídeo que termine `doContinuousProgressions` o `doContinuousIndependentProgressions`.

`videoBadMBs` ordena al codificador de vídeo del extremo distante que lleve a cabo medidas correctivas cuando un conjunto de macrobloques no ha sido recibido apropiadamente. El codificador utilizará esta información a fin de tomar las medidas pertinentes para recuperar la calidad vídeo. A diferencia de los macrobloques `videoNotDecodedMBs`, la instrucción `videoBadMBs` carece de una definición específica de cómo el decodificador ha tratado el conjunto especificado de macrobloques. El codificador debería responder a esta instrucción asegurando que el conjunto especificado de macrobloques no se utiliza para la predicción de imágenes vídeo subsiguientes a la recepción de la instrucción del codificador. La acción concreta que debe efectuar el codificador no está definida, pero puede incluir cualquier acción correctiva apropiada, tal como el envío de una trama INTRA. Esta instrucción no será transmitida por un decodificador vídeo si el codificador del extremo distante correspondiente no ha indicado la capacidad `videoBadMBsCap`. Esta instrucción sólo se utilizará con algoritmos de codificación vídeo que definen macrobloques como, por ejemplo, H.261, H.262, IS11172 y H.263. La numeración de los macrobloques se efectúa conforme al orden de exploración del barrido dentro de la imagen, con el macrobloque superior izquierdo de la imagen definido como macrobloque número 1, y el número MB que aumenta primero de izquierda a derecha y luego de arriba hacia abajo.

`lostPicture` ordena al codificador de vídeo del extremo distante que lleve a cabo acciones correctoras debido a la pérdida o corrupción de las imágenes indicadas por `pictureNumber`, un número de imagen de corta duración, o bien por `longTermPictureIndex`, un índice de imagen de corta duración. Si un codificador tiene la capacidad de aplicar los procedimientos del anexo U/H.263 (selección de imagen de referencia mejorada, con o sin eliminación de subimagen) y/o W.6.3.12/H.263 (número de imagen), será capaz de comprender este mensaje y llevar a cabo las acciones correctoras correspondientes.

`lostPartialPicture` ordena al codificador de vídeo del extremo distante que lleve a cabo acciones correctoras cuando un conjunto de macrobloques no ha sido recibido adecuadamente. Es la misma instrucción que `videoBadMBs` excepto que en este caso la imagen es indicada por `pictureNumber`, un número de imagen de corta duración, o bien por `longTermPictureIndex`, un índice de imagen de larga duración. Si un codificador tiene la capacidad de aplicar los procedimientos del anexo U/H.263 (selección de imagen de referencia mejorada, con o sin eliminación de subimagen) y/o W.6.3.12/H.263 (número de imagen), será capaz de comprender este mensaje y llevar a cabo las acciones correctoras correspondientes.

`recoveryReferencePicture` ordena al codificador del extremo distante que utilice únicamente las imágenes indicadas para la predicción por `pictureNumber`, un número de imagen de corta duración, o bien por `longTermPictureIndex`, un índice de imagen de larga duración. Si un codificador tiene la capacidad de aplicar los procedimientos del anexo U/H.263 (selección de imagen de referencia mejorada, con o sin eliminación de subimagen) y/o W.6.3.12/H.263 (número de imagen), será capaz

de comprender este mensaje y llevar a cabo las acciones correctoras correspondientes. Puede enviarse desde un decodificador que considera que las imágenes indicadas se han recibido y decodificado correctamente y que existen otras imágenes (no especificadas) corruptas a causa de la transmisión.

encryptionUpdateCommand se utilizará en la Rec. UIT-T H.235 en el procedimiento de actualización de clave mejorada para distribuir nuevo material de clave de sesión (véase B.2.6.2/H.235). multiplePayloadStream sólo se utiliza cuando se debe establecer una nueva clave en un tren de cabida útil múltiple, en cuyo caso se ignorará el tipo de cabida útil dinámico en EncryptionSync.

encryptionUpdateAck se utilizará en la Rec. UIT-T H.235 para el procedimiento de actualización de clave mejorada para permitir la recepción de acuse de recibo de subordinado de nuevo material de clave de sesión en un canal lógico de propiedad del principal (véase B.2.6.2/H.235).

direction indicará el sentido (masterToSlave o slaveToMaster) del canal lógico en el cual se distribuye el material clave (véase B.2.6.2/H.235).

### **B.13.6 Instrucción de conferencia**

BroadcastMyLogicalChannel será similar al MCV H.230 utilizado conforme con el procedimiento indicado en 6.3.2.1/H.243, pero sólo se referirá a un canal lógico único. Cabe señalar que cuando se utiliza el procedimiento MCV preferido indicado en 6.3.2.2/H.243 (es decir, cuando ambos extremos de un terminal MCU o entre enlaces MCU poseen multipointVisualizationCapability), se emplea entonces la forma conferenceRequest de BroadcastMyLogicalChannel.

CancelBroadcastMyLogicalChannel será similar a Cancel-MCV H.230 pero se referirá a un canal lógico único.

MakeTerminalBroadcaster se definirá como VCB H.230.

CancelMakeTerminalBroadcaster se definirá como Cancel-VCB.

SendThisSource se definirá como VCS H.230.

CancelSendThisSource se definirá como Cancel-VCS H.230.

DropConference se definirá como CCK H.230.

La instrucción sustitución de CID permite a un MC activo cambiar el identificador de conferencia (CID, *conference identifier*), trasladando efectivamente al recipiente de esta instrucción a otra conferencia. El recipiente de la instrucción utilizará el CID recién asignado en todos los mensajes futuros de señalización de llamada.

### **B.13.7 Reconfiguración de múltiplex H.223**

h223ModeChange ordena al transmisor que cambie el nivel del modo múltiplex, descrito en el anexo C/H.324, al nivel 0, nivel 1, nivel 2 o nivel 2 con encabezamiento opcional del anexo B/H.223.

h223AnnexADoubleFlag ordena al transmisor que comience a, o deje de, utilizar el modo doble bandera del anexo A/H.223.

### **B.13.8 Instrucción de nuevo canal virtual ATM**

Esta instrucción se utiliza para indicar al terminal distante que abra un canal virtual ATM con los parámetros dados.

resourceID se utiliza para identificar el canal virtual ATM. El medio utilizado para asociar este parámetro con un canal virtual ATM no se especifica en la presente Recomendación.

bitRate indica la velocidad binaria, medida en el punto AAL-SAP, del canal virtual, y se expresa en unidades de 64 kbit/s.

bitRateLockedToPCRClock indica que la velocidad binaria del canal virtual está cadenciada por el reloj utilizado para producir valores de referencia de reloj H.222.0 (referencia de reloj de programa o referencia de reloj de sistema).

bitRateLockedToNetworkClock indica que la velocidad binaria del canal virtual está cadenciada por el reloj de red local. Esto no garantiza que el reloj de velocidad binaria estará sincronizado con la red local en el receptor, ya que es posible que no se disponga de relojes de red comunes.

aal indica la capa de adaptación ATM que se utilizará y sus parámetros.

La secuencia aal1 indica cuáles de las opciones para la capa de adaptación ATM 1, entre las especificadas en las Recs. UIT-T I.363.x [25], están soportadas. Los puntos de código se definen en el cuadro B.1.

La secuencia aal5 indica cuáles de las opciones para la capa de adaptación ATM 5, entre las especificadas en las Recs. UIT-T I.363.x [25], están soportadas. forwardMaximumSDUSize y backwardMaximumSDUSize indican el tamaño máximo de la CPCS-SDU en los sentidos de transmisión de ida y retorno, medido en octetos.

multiplex indica el tipo de múltiplex que se utilizará en el canal virtual ATM. Las opciones son noMultiplex (no hay múltiplex H.222.0), tren de transporte H.222.0 y tren de programa H.222.0.

### **B.13.9 Instrucción de reconfiguración multienlace móvil**

Esta instrucción se utiliza para que el transmisor cambie la configuración de trama multienlace como se describe en el anexo H/H.324.

sampleSize indica el tamaño de una muestra en octetos. Una muestra es el número de octetos que será distribuido en los canales físicos disponibles.

samplesPerFrame indica la longitud de cabida multienlace en muestras.

status indica la situación del receptor cuando envía este mensaje de instrucción. Si está sincronizado, indica que el receptor ha establecido la sincronización de trama y ordena al transmisor que comience a enviar la trama de encabezamiento comprimida. Si está reconfigurado, significa que se impulsa al transmisor a modificar el tamaño de la muestra y/o la longitud de trama y a comenzar a enviar la trama de encabezamiento completa.

## **B.14 Indicaciones**

Una indicación contiene informaciones que no requieren acción ni respuesta.

### **B.14.1 Función no comprendida**

Se utiliza para devolver al transmisor peticiones, respuestas e instrucciones no comprendidas.

Si el terminal recibe una petición, respuesta o instrucción que no comprende ya sea porque no esta normalizada o se ha definido en una revisión ulterior de esta Recomendación, responderá con el envío de FunctionNotSupported o FunctionNotUnderstood.

NOTA – FunctionNotUnderstood se denominó FunctionNotSupported en la versión 1 de esta Recomendación. El nombre de esta función se cambió para permitir la adición de una instrucción FunctionNotSupported más potente sin quebrantar la retrocompatibilidad con la sintaxis de la versión 1.

### **B.14.2 Indicación diversos**

Se utiliza para una diversidad de indicaciones, algunas de las cuales están presentes en las Recs. UIT-T H.221 [7] y H.230 [13].

logicalChannelNumber indica el número de canal lógico al que se aplica la indicación. Expresará un canal lógico abierto para datos vídeo cuando el tipo sea videoIndicateReadyToActivate y videoTemporalSpatialTradeOff. Cuando el tipo es uno de los siguientes: multipointConference, cancelMultipointConference, multipointZeroComm, cancelMultipointZeroComm,



multipointSecondaryStatus o cancelMultipointSecondaryStatus, donde intervienen múltiples canales lógicos, el logicalChannelNumber será un número arbitrario, pero deberá ser un LogicalChannelNumber válido (es decir, estará comprendido en la gama 1-65535) y el receptor no tendrá en cuenta el valor.

Se utiliza logicalChannelInactive para indicar que el contenido del canal lógico no representa una señal normal. Es análogo a AIM y VIS definidas en la Rec. UIT-T H.230.

logicalChannelActive es complementario de logicalChannelInactive. Es análogo a AIA y VIA definidas en la Rec. UIT-T H.230. MultipointZeroComm, cancelMultipointZeroComm, multipointSecondaryStatus y cancelMultipointSecondaryStatus tendrán el mismo significado que MIZ, cancelMIZ, MIS y cancelMIS respectivamente, definidos en la Rec. UIT-T H.230.

multipointConference indica que el terminal se ha incorporado a una conferencia multipunto H.243, y se espera que el terminal obedezca a la simetrización de la velocidad binaria. Sin embargo, la simetrización de la velocidad binaria se hará cumplir mediante mensajes FlowControlCommand. Obsérvese que multipointConference tiene exactamente el mismo significado que MCC en la Rec. UIT-T H.230. Obsérvese asimismo que multipointConference, al igual que MCC, no requiere la simetría de modo.

videoIndicateReadyToActivate tendrá el mismo significado que VIR definido en la Rec. UIT-T H.230, es decir, lo transmite el terminal cuyo usuario ha decidido no enviar vídeo a menos que también reciba vídeo desde el otro extremo.

videoTemporalSpatialTradeOff indica al decodificador vídeo del extremo distante su solución actual de compromiso entre resolución temporal y espacial. El valor 0 indica una alta resolución espacial y el valor 31 indica una alta velocidad de trama. Los valores comprendidos entre 0 y 31 indican, sucesivamente, una velocidad de trama más alta. Los valores reales no corresponden a valores precisos de la resolución espacial o velocidad de trama. Un terminal que haya indicado temporalSpatialTradeOffCapability deberá transmitir esta indicación siempre que cambie su solución de compromiso y cuando se abra inicialmente un canal lógico de vídeo.

videoNotDecodedMBs indica al codificador vídeo del extremo distante que se ha recibido erróneamente un conjunto de MB y que cualquier MB perteneciente al conjunto especificado se ha tratado como no codificado. El codificador puede utilizar esta información para compensar errores de transmisión como se ilustra en el apéndice I/H.263. firstMB indica que el número del primer MB se trató como no codificado y numberOfMB indica el número de MB tratados como no codificados. La numeración de MB se efectúa de forma tal que el macrobloque en el ángulo superior izquierdo de la imagen se considera el macrobloque número 1 y el número de cada macrobloque aumenta de izquierda a derecha y luego de arriba hacia abajo en el orden de exploración por barrido (de modo que si en una imagen hay un total de N macrobloques, el macrobloque inferior derecho se considera como macrobloque número N). La referencia temporal de la imagen que contiene MB no decodificados se indica en temporalReference. Esta indicación sólo se utilizará con el algoritmo de compresión vídeo de H.263.

### **B.14.3 Indicación de fluctuación**

Se utiliza para indicar la cuantía de la fluctuación de un canal lógico estimada por el terminal receptor. Puede ser útil para la elección de la velocidad binaria y el control de la memoria tampón en canales vídeo o para determinar una velocidad apropiada de transmisión de la información de temporización, etc. El codificador vídeo tendrá entonces la opción de emplear esta información para limitar la velocidad binaria de vídeo o las fluctuaciones de la memoria tampón del decodificador vídeo para ayudar a evitar el rebasamiento por debajo o por encima de la memoria tampón del decodificador para la fluctuación existente. Si el decodificador hace uso de esta opción, esto permitirá el funcionamiento correcto de los diseños existentes de memoria tampón de decodificación de vídeo, independientemente de la amplitud de la fluctuación recibida, así como el funcionamiento correcto con un retardo mínimo.

Cuando el alcance sea del tipo `logicalChannelNumber`, la información se aplica al canal lógico dado; cuando el alcance sea del tipo `resourceID`, la información se aplica al canal virtual ATM dado; y cuando la información sea del tipo `wholeMultiplex`, la información se aplica a la totalidad del múltiplex.

`estimatedReceivedJitterMantissa` y `estimatedReceivedJitterExponent` proporcionan una estimación de la fluctuación recibida por el terminal que ha enviado el mensaje.

`estimatedReceivedJitterMantissa` expresa la mantisa de la estimación de fluctuación como se indica en el cuadro B.16.

**Cuadro B.16/H.245 – Mantisa de `estimatedReceivedJitterMantissa` en `JitterIndication`**

<code>estimatedReceivedJitterMantissa</code>	Mantisa
0	1
1	2,5
2	5
3	7,5

`estimatedReceivedJitterExponent` expresa el exponente de la estimación de la fluctuación como se indica en el cuadro B.17.

**Cuadro B.17/H.245 – Exponente de `estimatedReceivedJitterExponent` en `JitterIndication`**

<code>estimatedReceivedJitterExponent</code>	Exponente
0	Fuera de gama
1	1 $\mu$ s
2	10 $\mu$ s
3	100 $\mu$ s
4	1 ms
5	10 ms
6	100 ms
7	1 s

Se obtiene la estimación de la fluctuación multiplicando la mantisa por el exponente, a menos que `estimatedReceivedJitterExponent` sea igual a cero, en cuyo caso se conoce que la estimación será superior a 7,5 segundos.

`skippedFrameCount` indica cuántas tramas ha saltado el decodificador desde que se recibió el último mensaje de `JitterIndication`. Como el valor máximo que puede codificarse es 15, si se implementa esta opción, debe transmitirse la información antes de que se hayan saltado más de 15 tramas.

NOTA – Como se saltan las tramas cuando la memoria tampón del decodificador rebasa por debajo, la fluctuación adicional puede provocar el rebasamiento por debajo de la memoria tampón del decodificador con una frecuencia mayor o menor que la frecuencia con la que se espera que se produzcan los saltos de trama del codificador.

`additionalDecoderBuffer` indica el tamaño adicional de la memoria tampón del decodificador vídeo por encima de los valores requeridos por el perfil y nivel indicados. Se define de la misma forma que `vbv_buffer_size` en la Rec. UIT-T H.262 [19].

#### **B.14.4 Indicación de asimetría H.223**

Se utiliza para indicar al terminal del extremo distante la cuantía media de la asimetría temporal entre dos canales lógicos.

logicalChannelNumber1 y logicalChannelNumber2 son los números de canal lógico de los canales lógicos abiertos.

La asimetría se mide en milisegundos e indica el retardo que debe aplicarse a los datos pertenecientes a logicalChannelNumber2, medido a la salida del múltiplex, para conseguir la sincronización con logicalChannelNumber1 medido a la salida del múltiplex. La asimetría incluye diferencias en: tiempo de muestreo, retardo del codificador y retardo de la memoria tampón del transmisor, y se mide con relación al tiempo de transmisión del primer bit de datos que representa un determinado punto de muestreo. El retardo real necesario para la sincronización depende de la implementación del decodificador y es un asunto local del receptor.

#### **B.14.5 Indicación de nuevo canal virtual ATM**

Se utiliza para indicar los parámetros de un canal virtual ATM que el terminal tiene el propósito de abrir.

Se utiliza resourceID para identificar el canal virtual ATM. El medio utilizado para asociar este parámetro con un canal virtual no se especifica en esta Recomendación.

bitRate indica la velocidad binaria, medida en el punto AAL-SAP, del canal virtual y se expresa en unidades de 64 kbit/s.

bitRateLockedToPCRClock indica que la velocidad binaria del canal virtual está cadenciada por el reloj utilizado para producir valores de referencia de reloj H.222.0 (referencia de reloj de programa o referencia de reloj de sistema).

bitRateLockedToNetworkClock indica que la velocidad binaria del canal virtual está cadenciada por el reloj de la red local. Esto no garantiza que el reloj de velocidad binaria estará sincronizado con la red local en el receptor, pues puede no haber relojes comunes disponibles.

aal indica la capa de adaptación de ATM que se utilizará y sus parámetros.

La secuencia aal1 indica cuáles de las opciones para la capa de adaptación ATM 1, entre las especificadas en la Rec. UIT-T I.363 [25], están soportadas. Los puntos de código se definen en el cuadro B.1.

La secuencia aal5 indica cuáles de las opciones para la capa de adaptación ATM 5, entre las especificadas en la Rec. UIT-T I.363 [25], están soportadas. forwardMaximumSDUSize y backwardMaximumSDUSize indican el tamaño máximo de la CPCS-SDU en los sentidos de transmisión de ida y de retorno, medido en octetos.

multiplex indica el tipo de múltiplex que se utilizará en el canal virtual ATM. Las opciones son noMultiplex (no hay múltiplex H.222.0), tren de transporte H.222.0 y tren de programa H.222.0.

#### **B.14.6 Entrada de usuario**

Se utiliza para los mensajes de entrada de usuario.

alphanumeric es una cadena de caracteres codificada según la Rec. UIT-T T.51 [30]. Podría utilizarse para entrada por teclado secundario, que equivale a DTMF.

**userInputSupportIndication:** indica al terminal distante los tipos GENERALSTRING que soporta el terminal.

NOTA 1 – Cabe esperar que la mayor parte de las implementaciones de los decodificadores PER no serán capaces de decodificar otras cadenas que no sean las del IA5. Esta indicación debe utilizarse para "advertir" al terminal distante que no debe tratar de utilizar esquemas de codificación poco usuales, de longitud variable.

Si la DTMF se envía a través del RTP y en `UserInputIndication` en forma alfanumérica, se codificará en la secuencia `extendedAlphanumeric` y se incluirá la bandera `rtpPayloadIndication`.

`nonStandard` es un `NonStandardParameter` que indica una utilización no normalizada del mensaje de indicación `UserInput`.

La variable booleana `basicString`, cuando toma el valor verdadero, indica que los caracteres 0-9, \* y # están soportados.

La variable booleana `iA5String`, cuando toma el valor verdadero, indica que el conjunto completo de caracteres `IA5String` está soportado.

La variable booleana `generalString`, cuando toma el valor verdadero, indica que el conjunto completo de caracteres `GeneralString` está soportado.

La variable booleana `encryptedBasicString`, cuando toma el valor verdadero, indica una cadena básica criptada.

La variable booleana `encryptedIA5String`, cuando toma el valor verdadero, indica una cadena `IA5` criptada.

La variable booleana `encryptedGeneralString`, cuando toma el valor verdadero, indica una cadena general criptada.

En la cláusula 8.7/H.235 se describen los procedimientos para DTMF H.245 criptado y cómo aplicar los campos `encryptedAlphanumeric` dentro de `UserInputIndication` (= cadena básica criptada), `encryptedSignalType` dentro de la señal (= cadena `IA5` criptada) y `encryptedAlphanumeric` dentro de `extendedAlphanumeric` (= cadena general criptada).

Las indicaciones **signal** y **signalUpdate** pueden utilizarse cuando se desee un control preciso de la alineación de la DTMF o de una señal de enganchado con audio en el canal lógico asociado y cuando se necesite el control o la indicación de la duración de la multifrecuencia bitono (DTMF, *dual tone multi-frequency*).

**signal** indica el elemento de señalización que se ha de producir cuando se envía a una pasarela de línea de red telefónica pública conmutada (RTPC), que se detectó en el tren de audio cuando se envió desde una pasarela de línea de RTPC, o que se ha de señalar entre otras combinaciones de puntos extremos. Cuando es recibida por una pasarela a la RTPC, **signal** hace que la pasarela inyecte el elemento de señalización especificado en el canal RTPC; cuando sea recibida por una pasarela de línea a otro terminal de la serie H, **signal** será traducida al mensaje apropiado del protocolo del terminal conectado. Las pasarelas de línea generan mensajes **signal** (y **signalUpdate**) para indicar la detección de elementos de señalización en el audio recibido procedente de un punto extremo RTPC o por la traducción de mensajes correspondientes de otros protocolos.

**signalType** se fija a "!" (signo de exclamación) para indicar una señal de enganchado o a uno de los signos "0123456789\*#ABCD" para indicar un tono DTMF.

NOTA 2 – La señal de enganchado es una condición de enganchado momentánea (con una duración, por lo general, de medio segundo), utilizada normalmente para controlar características del equipo de conmutación adjunto. Quizá no sea posible que una pasarela produzca o detecte una señal de enganchado debido a las particularidades del canal RTPC o por la configuración local (para evitar la activación no deseada de características del equipo adjunto). Por ello, la aptitud para transmitir o recibir indicaciones de señal de enganchado se declaran por separado en **UserInputCapability**.

**duration** indica la duración total del tono si es que se conoce o una estimación inicial de dicha duración si el tono se sigue emitiendo en el momento en que se transmite **signal**. Si se omite **duration**, el receptor utilizará un valor por defecto apropiado basado en la configuración local y en los requisitos de la red. **duration** deberá ser ignorada cuando se produzca una indicación de señal de enganchado ("!").

**signalUpdate** revisa la estimación de la duración total o declara la duración medida efectiva del tono detectado o que se ha de generar. Deberá ser transmitida de manera que llegue antes de que concluya la estimación que se envió anteriormente en **signal** o en **signalUpdate**; de otro modo, la duración revisada será ignorada porque el receptor ya habrá hecho que cese el tono. Se señala que no es necesario enviar **signalUpdate** si la duración total se hubiera indicado en **signal**.

**rtp** contiene los parámetros que se necesitan para alinear el tono o la señal de enganchado con un tren RTP/UDP (H.323). En **signalUpdate**, este elemento sólo se ha de incluir si se han emitido mensajes de señales múltiples especificando **LogicalChannelNumber** diferentes y es necesario indicar cuál es la señal que ha de ser actualizada.

**timestamp** especifica, en términos de la indicación de tiempo del RTP del codificador primario en el canal de audio asociado, el momento en el cual deberían ser generados la señal de enganchado o el tono (entregado o inyectado en el tren de audio). La señal de enganchado o el tono no deberán generarse antes de que se reproduzca audio con la misma indicación de tiempo; deberán generarse tan pronto como sea posible después de ese momento, pero no más tarde que el tiempo indicado en **expirationTime**. El emisor de una indicación no deberá fijar **timestamp** en un momento "del futuro"; **timestamp** se fija normalmente igual a la indicación de tiempo del audio que se envía en esos momentos o que se ha enviado más recientemente por el canal de audio asociado. Si no se especifica **timestamp**, la señal será entregada o inyectada tras su recepción.

**expirationTime** especifica, en términos de la indicación de tiempo del RTP del codificador primario en el canal de audio asociado, el momento después del cual deberán ser considerados "caducos" y descartados por el receptor la señal de enganchado o el tono. Los puntos extremos que reciban **signal** y no sean capaces de actuar en ella antes de la indicación de tiempo **expirationTime** en el canal asociado deberán descartar el mensaje. Si el emisor no especifica un tiempo **expirationTime**, el mensaje podrá ser descartado de todos modos como resultado de la configuración local del recipiente.

**logicalChannelNumber** deberá especificar el **LogicalChannelNumber** del canal de audio asociado, el contexto en el que tienen significado **timestamp** y **expirationTime**.

Un MC deberá convertir las indicaciones de tiempo y el número de canal lógico de la indicación recibida en las indicaciones de tiempo y el número de canal lógico correctos para cada canal de salida cuando reenvíe la indicación a cada punto extremo receptor (las indicaciones de tiempo podrían cambiar si el audio se transcodifica o mezcla en un MP). Si un MC recibe una indicación después del tiempo **expirationTime** puede descartar el mensaje inmediatamente sin reenviarlo; de no ser así, reenviará todas las peticiones inmediatamente sin esperar que llegue el momento **timestamp**.

Los puntos extremos utilizarán la indicación **alphanumeric** para llevar una entrada de usuario DTMF si el otro punto extremo no ha indicado aptitud para recibir DTMF utilizando la **UserInputCapability**.

Si un punto extremo tiene la capacidad de recibir indicaciones DTMF utilizando **signal**, también podrá aceptar indicaciones **alphanumeric** a efectos de compatibilidad con terminales más antiguos. Una indicación **alphanumeric** puede ser tratada como una secuencia de una o más indicaciones **signal** en las que se han omitido los elementos **duration**, **timestamp** y **expirationTime** y donde se descartan los caracteres no válidos de **signalType**.

Si la DTMF se envía a través del RTP según las indicaciones de 10.5/H.323 y en **UserInputIndication** en forma de señal, se incluirá la bandera **rtpPayloadIndication**.

En una utilización típica, la pasarela que detecte DTMF en el tren de audio procedente de un canal RTPC enviará **signal** inmediatamente después de la detección de un tono, utilizando una estimación relativamente alta de **duration**, y comenzará a medir la duración del tono. Cuando finaliza el tono, se envía **signalUpdate** para indicar la duración medida total. Si el tono no ha terminado pero la

duración medida se aproxima a la estimación anterior (de tal manera que la estimación podría ser rebasada por la duración medida antes de que se pudiera recibir una **signalUpdate**), se envía una **signalUpdate** con la que se incrementa la estimación. La frecuencia con que se envía **signalUpdate**, la estimación de duración inicial enviada en **signal** y la cuantía en la que se incrementan las estimaciones subsiguientes se dejan a criterio del implementador, pero deberá actuarse con prudencia para no cargar la red con un gran número de mensajes **signalUpdate** y para evitar la expiración prematura de estimaciones previas.

En una utilización típica desde un punto extremo no pasarela, el elemento **signal** contendrá la duración total del tono que ha de ser producido por la pasarela. En algunas aplicaciones, sin embargo, quizá convenga proporcionar al usuario control interactivo en tiempo real de la duración del tono. En este caso, se utilizarían **signal** y **signalUpdate** de manera similar a la descrita para las pasarelas en el párrafo precedente, enviándose **signal** tras la activación de la entrada del usuario (por ejemplo, la depresión de una tecla o un control en pantalla) con una duración de tono estimada, y utilizando **signalUpdate** para enviar estimaciones actualizadas en tanto la entrada continúe estando activada y para indicar la duración total cuando se desactive la entrada.

#### **B.14.7 Indicaciones de conferencia**

sbeNumber se definirá como número SBE H.230.

terminalNumberAssign se definirá como TIA H.230.

terminalJoinedConference se definirá como TIN H.230.

terminalLeftConference se definirá como TID H.230.

seenByAtLeastOneOther se definirá como MIV H.230.

cancelSeenByAtLeastOneOther se definirá como cancel-MIV H.230.

seenByAll se definirá como MIV H.230.

cancelSeenByAll se definirá como MIV H.230.

terminalYouAreSeeing se definirá como VIN H.230.

requestForFloor se definirá como TIF H.230 y se enviará desde un terminal al MC.

WithdrawChairToken se definirá como CCR H.230 y se enviará del MC al poseedor del testigo para la presidencia.

FloorRequested se definirá como TIF H.230 cuando se envíe del MC al poseedor del testigo para la presidencia. Esta petición incluye la TerminalLabel del terminal solicitante.

terminalYouAreSeeingInSubPictureNumber se definirá como VIN2 H.230. subPictureNumber se define como N tal y como muestran las figuras 2-4/H.243.

videoIndicateCompose se definirá como VIC H.230. compositionNumber se define como M tal y como muestra el cuadro 4/H.243.

#### **B.14.8 Máxima asimetría de canal lógico H.2250**

H2250MaximumSkewIndication indica la asimetría máxima entre canales lógicos.

La asimetría se mide en milisegundos e indica el número máximo de milisegundos que los datos en logicalChannelNumber2 están retardados con respecto a los datos en logicalChannelNumber1, cuando son entregados a la red de transporte. La asimetría se mide con relación a la hora (o tiempo) de entrega a la red de transporte del primer bit de datos que representa un determinado punto de muestra. La sincronización de labios, si se desea, es un asunto local del receptor y se conseguirá mediante la utilización de sellos de tiempo.

### **B.14.9 Indicación de la ubicación del MC**

Esta indicación se envía al MC para indicar a otros terminales la dirección de señalización que debe utilizarse para alcanzar al MC.

### **B.14.10 Indicación identificación del vendedor**

La indicación vendorIdentification debe enviarse al comienzo de cada llamada para indicar el fabricante, producto y número de la versión del producto.

### **B.14.11 Función no soportada**

Se utiliza para devolver al transmisor peticiones, respuestas e instrucciones que no son comprendidas.

Se devuelve la totalidad del RequestMessage, ResponseMessage o CommandMessage.

Si un terminal recibe una petición, respuesta o instrucción que no comprende, sea porque no están normalizadas o porque están definidas en una revisión ulterior de esta Recomendación, responderá enviando FunctionNotSupported.

Si un terminal recibe una petición, respuesta o instrucción cuya codificación es incorrecta, fijará el campo cause al valor syntaxError. Si la codificación es correcta pero los valores codificados son semánticamente incorrectos, fijará el campo cause al valor semanticError. Si el mensaje es una extensión no reconocida de MultimediaSystemControlMessage, RequestMessage, ResponseMessage o CommandMessage, fijará el campo cause al valor unknownFunction.

En cada uno de estos casos se devolverá la totalidad del mensaje MultimediaSystemControlMessage como una cadena de octetos en returnedFunction.

FunctionNotSupported no se utilizará en ninguna otra ocasión. En particular, cuando una extensión no reconocida está presente en otros puntos de la sintaxis, FunctionNotSupported no se utilizará: el terminal responderá al mensaje de manera normal, como si no hubiese presente ninguna extensión. FunctionNotSupported nunca se enviará en respuesta a una indicación recibida.

### **B.14.12 Indicación de control de flujo**

Se utiliza para señalar al terminal distante que el terminal ha ajustado su velocidad binaria máxima de salida ya sea en respuesta a una FlowControlCommand entrante o porque el terminal desea ajustar su velocidad de salida. Esto permite a un terminal señalar cualquier cambio de la velocidad binaria máxima de salida dentro de la restricción de los límites superiores fijados por el canal lógico abierto y las capacidades del terminal.

Cualquier terminal que reciba una FlowControlCommand deberá responder con una indicación de FlowControl para indicar la nueva velocidad binaria máxima a la que ha sido fijado.

Los campos de FlowControlIndication tienen el mismo significado que los campos del mismo nombre de FlowControlCommand.

### **B.14.13 Indicación de reconfiguración multienlace móvil**

Esta indicación se utiliza para señalar al receptor que el transmisor modificará el valor del tamaño de la muestra y/o de las muestras por trama en el encabezamiento de trama de información como se describe en el anexo H/H.324. Esta indicación puede enviarse cuando se encuentra en el modo encabezamiento completo y no deberá enviarse cuando se encuentra en el modo encabezamiento comprimido.

sampleSize indica el tamaño de una muestra en octetos. Una muestra es el número de octetos que será distribuido en los canales físicos disponibles.

samplesPerFrame indica la longitud de cabida multienlace en muestras.

## B.15 Mensajes genéricos

El tipo **GenericMessage** permite que los nuevos elementos RequestMessage, CommandMessage, ResponseMessage e IndicationMessage se especifiquen de modo tal que no sea necesario emitir una nueva versión de la sintaxis H.245. Este método permite definir mensajes normalizados y no normalizados.

NOTA 1 – Las estructuras GenericMessage definidas en esta Recomendación deberán incluirse en la misma a modo de anexos. Las estructuras GenericMessage definidas en otras Recs. UIT-T serán referidas en un apéndice a esta Recomendación. Las estructuras GenericMessage ajenas a las normas del UIT-T se pueden publicar de la forma que mejor convenga.

El campo **messageIdentifier** indica el tipo de mensaje único. Los identificadores de mensajes basados en las normas del UIT-T deberán utilizar el OBJECT IDENTIFIER normalizado mientras que los identificadores de mensajes basados en otras normas y los identificadores patentados deberán utilizar la forma normalizada h221NonStandard, uuid y domainBased, según proceda.

El campo opcional **subMessageIdentifier** indica un submensaje asociado con el messageIdentifier.

El campo **messageContents** indica parámetros del mensaje.

Para evitar problemas de ambigüedad e interoperabilidad, no se debe definir un **ParameterIdentifier** normalizado con el valor 0 para su utilización en el campo messageContents.

NOTA 2 – Algunas Recomendaciones definen procesos automáticos para la traducción de GenericParameters del sistema de señalización H.245 al sistema de señalización códec BAS utilizado en la Rec. UIT-T H.320. Estos procedimientos utilizan el valor 0 en lugar de un ParameterIdentifier normalizado como un signo especial que demarca el fin de una lista de elementos GenericParameter.

El campo genericRequest es un GenericMessage utilizado para enviar un RequestMessage genérico.

El campo genericResponse es un GenericMessage utilizado para enviar un ResponseMessage genérico.

El campo genericCommand es un GenericMessage utilizado para enviar un CommandMessage genérico.

El campo genericIndication es un GenericMessage utilizado para enviar un IndicationMessage genérico.

## Anexo C

### Procedimientos

#### C.1 Introducción

En este anexo se definen procedimientos genéricos de control de sistemas multimedia que emplean los mensajes definidos en esta Recomendación. Las Recomendaciones que hagan uso de la presente Recomendación indicarán cuáles de entre estos procedimientos son aplicables y definirán también cualesquiera requisitos específicos.

En este anexo se describen procedimientos para la ejecución de las siguientes funciones:

- determinación principal-subordinado;
- intercambio de capacidad de terminal;
- señalización de canal lógico unidireccional;
- señalización de canal lógico bidireccional;
- petición de cierre de canal lógico por el terminal receptor;



- modificación de entrada en la tabla múltiplex H.223;
- petición de inscripción múltiplex;
- petición de modo transmisión de receptor a transmisor;
- determinación del retardo de ida y vuelta;
- bucle de mantenimiento.

### **C.1.1 Método de especificación**

En esta cláusula se especifican de forma general los procedimientos utilizando el lenguaje SDL. El SDL proporciona una especificación gráfica de los procedimientos e incluye la especificación de acciones en el caso de condiciones de excepción.

### **C.1.2 Comunicación entre una entidad de protocolo y un usuario de protocolo**

La interacción con el usuario de una función particular se especifica mediante primitivas transferidas en la interfaz situada entre la entidad de protocolo y el usuario de protocolo. Las primitivas tienen como finalidad definir procedimientos de protocolo y no están previstas para especificar o limitar la implementación. Cada primitiva puede tener asociados cierto número de parámetros.

Para apoyar la especificación se definen estados de protocolo. Estos estados son conceptuales y reflejan condiciones generales de la entidad de protocolo en las secuencias de primitivas intercambiadas entre la entidad de protocolo y el usuario, así como el intercambio de mensajes entre la entidad de protocolo y su par.

Para cada entidad de protocolo se define la secuencia permitida de primitivas entre el usuario y la entidad de protocolo. La secuencia permitida limita las acciones del usuario y define las posibles respuestas de la entidad de protocolo.

Un parámetro de primitiva descrito como nulo es equivalente a la ausencia del parámetro.

### **C.1.3 Comunicación entre entidades pares**

La información de protocolo se transfiere a la entidad de protocolo par mediante mensajes apropiados definidos en el anexo A. Algunas entidades de protocolo descritas tienen asociadas variables de estado. Algunas de las entidades de protocolo descritas tienen también asociados temporizadores.

Un temporizador se identifica por la notación  $T_n$ , siendo  $n$  un número. En los diagramas SDL, por iniciar un temporizador ha de entenderse cargar un temporizador con un valor especificado y arrancar dicho temporizador. Por reiniciar un temporizador ha de entenderse detener un temporizador y retener el valor que tenía cuando fue reiniciado. Por expiración del temporizador ha de entenderse que un temporizador ha funcionado durante el tiempo especificado y ha llegado al valor de cero.

Una entidad de protocolo puede también tener asociados parámetros. Un parámetro se identifica por la notación  $N_n$ , siendo  $n$  un número.

Los temporizadores y contadores se indican en el apéndice III.

Algunas entidades de protocolo definen una primitiva de error para informar sobre condiciones de error de protocolo a la entidad de gestión.

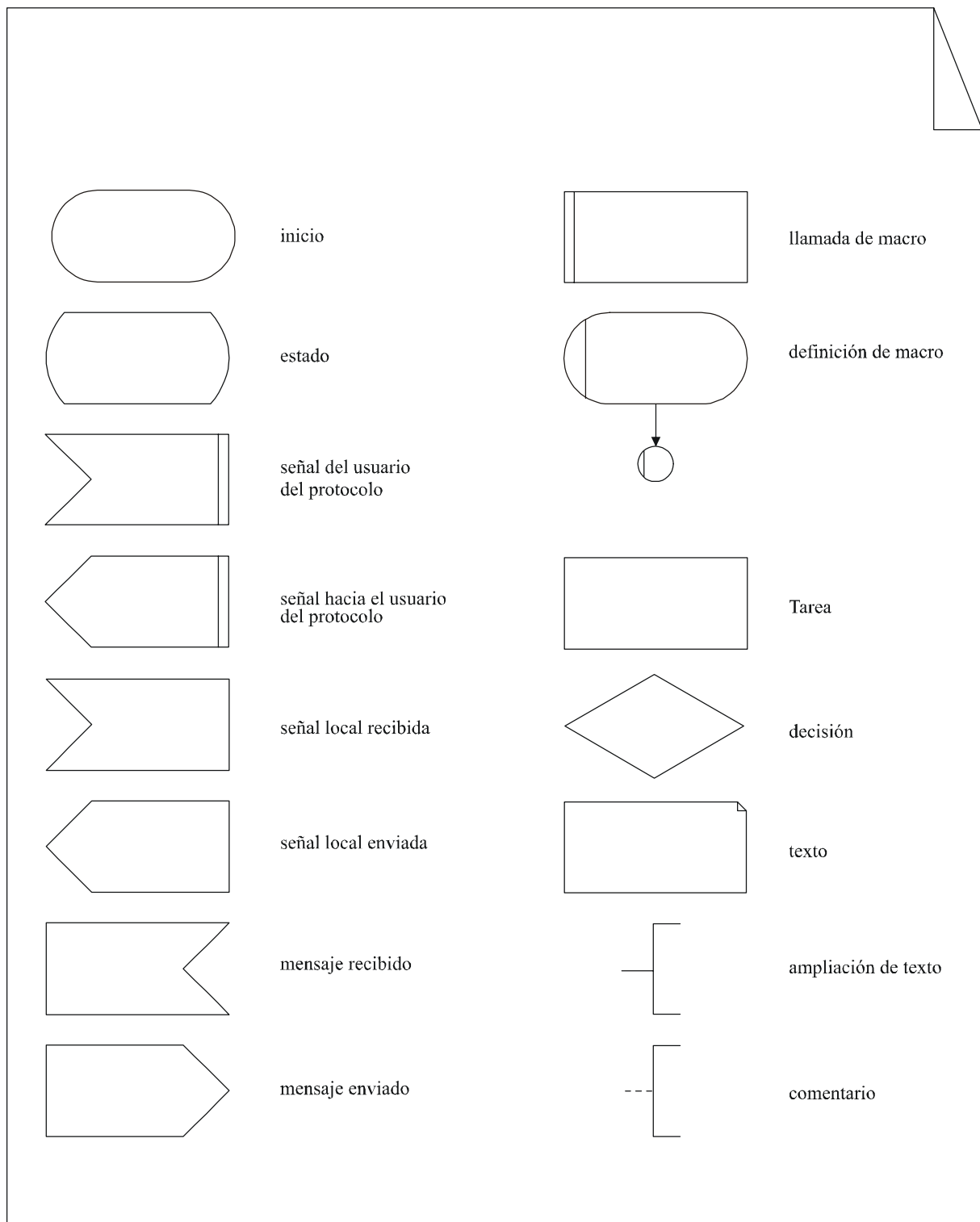
### **C.1.4 Diagramas SDL**

Los diagramas SDL muestran acciones relativas a las interacciones permitidas con el usuario del protocolo y a la recepción de mensajes desde la entidad de protocolo par. Las primitivas que no están permitidas para un estado determinado, especificado por los diagramas de transición de

estados, no se muestran en los diagramas SDL. Sin embargo, las respuestas a la recepción de mensajes inapropiados se describen en los diagramas de SDL.

### C.1.5 Símbolos gráficos utilizados en los diagramas SDL

En la figura C.1 se muestran los símbolos gráficos utilizados en los diagramas SDL.



H.245\_FC.1

**Figura C.1/H.245 – Símbolos gráficos utilizados en los diagramas SDL**

## **C.2 Procedimientos de determinación principal-subordinado**

### **C.2.1 Introducción**

Pueden presentarse situaciones de conflicto cuando dos o más terminales que participan en una llamada inician simultáneamente eventos similares, habiendo solamente recursos disponibles para una sola aparición del evento, por ejemplo, apertura de canales lógicos. Para resolver esos conflictos, un terminal puede actuar como el principal y el otro, o los otros terminales, pueden actuar como los subordinados. Los procedimientos aquí descritos permiten a los terminales que participan en la llamada determinar cuál es el terminal principal y cuál es o cuáles son los subordinados.

El protocolo descrito se designa por entidad de señalización de determinación de principal-subordinado (MSDSE, *master-slave determination signalling entity*). Hay un ejemplar de la MSDSE en cada terminal que participa en una llamada.

Cualquiera de los terminales puede iniciar el procedimiento de la determinación del principal y subordinado enviando la primitiva DETERMINACIÓN.petición a su MSDSE. El resultado del procedimiento se devuelve mediante las primitivas DETERMINACIÓN.indicación y DETERMINACIÓN.confirmación. La primitiva DETERMINACIÓN.indicación indica el resultado, pero no que el resultado es conocido en el terminal distante. La primitiva DETERMINACIÓN.confirmación indica el resultado y confirma que también se conoce en el terminal distante. Un terminal sólo puede iniciar el proceso de determinación principal-subordinado si no está localmente activo ningún procedimiento que dependa de su resultado.

Un terminal responderá a procedimientos que se basan en el conocimiento del resultado y se inician por el terminal distante en cualquier momento después de que el resultado de la determinación de la categoría del terminal se conoce en el terminal local. Esto puede suceder antes de que el terminal local haya recibido la confirmación de que el terminal distante también conoce el resultado. Un terminal no deberá iniciar procedimientos que se basan en el conocimiento del resultado hasta que haya recibido la confirmación de que el terminal distante también conoce el resultado del ejemplar vigente del procedimiento de determinación.

El siguiente texto proporciona una visión general del funcionamiento del protocolo. En caso de cualquier discrepancia con la especificación formal del protocolo que sigue, prevalecerá la especificación formal.

#### **C.2.1.1 Visión general del protocolo – Iniciación por el usuario local**

Se inicia un procedimiento principal-subordinado cuando la primitiva DETERMINACIÓN.petición es emitida por el usuario de la MSDSE. Se envía un mensaje MasterSlaveDetermination a la MSDSE par y se arranca el temporizador T106. Si se recibe un mensaje MasterSlaveDeterminationAck en respuesta al mensaje MasterSlaveDetermination, se detiene el temporizador T106 y se informa al usuario con la primitiva DETERMINACIÓN.confirmación de que el procedimiento de determinación principal-subordinado tuvo éxito y se envía un mensaje MasterSlaveDeterminationAck a la MSDSE par. Sin embargo, si se recibe un mensaje MasterSlaveDeterminationReject en respuesta al mensaje MasterSlaveDetermination, se genera un nuevo número de determinación de categoría, se rearranca el temporizador T106 y se envía otro mensaje MasterSlaveDetermination. Si tras el envío de un mensaje MasterSlaveDetermination N100 veces, no se ha recibido todavía un MasterSlaveDeterminationAck, se detiene el temporizador y se informa al usuario con la primitiva RECHAZO.indicación de que el procedimiento de determinación principal-subordinado no ha conseguido producir un resultado.

Si expira el temporizador T106, se informa entonces al usuario de la MSDSE con la primitiva RECHAZO.indicación y se envía un mensaje MasterSlaveDeterminationRelease a la MSDSE par.

### **C.2.1.2 Visión general del protocolo – Iniciación por el usuario distante**

Cuando se recibe un mensaje MasterSlaveDetermination en la MSDSE, se inicia un procedimiento de determinación de categoría. Si el procedimiento de determinación de categoría devuelve un resultado determinado, se informa entonces al usuario del resultado de determinación principal-subordinado con la primitiva DETERMINACIÓN.indicación, se envía un mensaje MasterSlaveDeterminationAck a la MSDSE par y se arranca el temporizador T106. Si se recibe un mensaje MasterSlaveDeterminationAck en respuesta al mensaje MasterSlaveDeterminationAck, se detiene entonces el temporizador T106 y se informa al usuario con la primitiva DETERMINACIÓN.confirmación de que el procedimiento de determinación principal-subordinado tuvo éxito.

Si expira el temporizador T106, se informa entonces al usuario de la MSDSE con la primitiva RECHAZO.indicación.

Sin embargo, si el procedimiento de determinación de categoría devuelve un resultado indeterminado, se envía entonces el mensaje MasterSlaveDeterminationReject a la MSDSE par.

### **C.2.1.3 Visión general del protocolo – Iniciación simultánea**

Cuando se recibe un mensaje MasterSlaveDetermination en la MSDSE de que ya ha iniciado ella misma un procedimiento de determinación de categoría, y está esperando un mensaje MasterSlaveDeterminationAck o MasterSlaveDeterminationReject, se inicia entonces un procedimiento de determinación de categoría. Si el procedimiento de determinación de categoría devuelve un resultado determinado, la MSDSE responde como si el procedimiento lo hubiera iniciado el usuario distante y se aplican los procedimientos arriba descritos para esta condición.

Sin embargo, si el procedimiento de determinación de categoría devuelve un resultado indeterminado, se genera un nuevo número de determinación de categoría, y la MSDSE responde como si el procedimiento lo hubiese de nuevo iniciado el usuario MSDSE local como se ha descrito más arriba.

### **C.2.1.4 Procedimiento de determinación de categoría**

Se utiliza el siguiente procedimiento para determinar qué terminal es el principal a partir de los valores de terminalType y statusDeterminationNumber. En primer lugar, se comparan los valores de terminalType y se determina como principal el terminal con el número de tipo de terminal más grande. Si los números de tipo de terminal son los mismos, se comparan los statusDeterminationNumber utilizando aritmética de módulo para determinar cuál es el principal.

Si ambos terminales tienen iguales valores del campo terminalType y la diferencia entre los valores del campo statusDeterminationNumber en módulo  $2^{24}$  es 0 ó  $2^{23}$ , se obtiene un resultado indeterminado.

## **C.2.2 Comunicación entre la MSDSE y el usuario de MSDSE**

### **C.2.2.1 Primitivas entre la MSDSE y el usuario de MSDSE**

La comunicación entre la MSDSE y el usuario de la MSDSE se realiza mediante las primitivas indicadas en el cuadro C.1.

**Cuadro C.1/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
DETERMINACIÓN	– (nota 1)	TIPO	no definida (nota 2)	TIPO
RECHAZO	no definida	–	no definida	no definida
ERROR	no definida	ERRCODE	no definida	no definida
NOTA 1 – "-" indica que no hay parámetros.				
NOTA 2 – "no definido" indica que esta primitiva no está definida.				

### C.2.2.2 Definiciones de las primitivas

La definiciones de estas primitivas son:

- a) La primitiva DETERMINACIÓN se utiliza para iniciar el procedimiento de determinación de principal-subordinado y para devolver el resultado de dicho procedimiento.

La primitiva DETERMINACIÓN.petición se utiliza para iniciar el procedimiento de determinación de principal-subordinado.

La primitiva DETERMINACIÓN.indicación se utiliza para indicar el resultado del procedimiento de determinación de principal-subordinado. Puesto que el resultado del procedimiento puede no conocerse en el terminal distante, el terminal no iniciará ningún procedimiento basado en el conocimiento del resultado aunque deberá responder a cualquier procedimiento que se base en el conocimiento del resultado.

La primitiva DETERMINACIÓN.confirmación se utiliza para indicar el resultado del procedimiento de determinación de principal-subordinado y que el resultado del procedimiento se conoce en ambos terminales. El terminal puede iniciar cualquier procedimiento basado en el conocimiento del resultado y deberá responder a dicho procedimiento.

- b) La primitiva RECHAZO indica que el procedimiento de determinación de principal-subordinado ha fracasado.
- c) La primitiva ERROR informa errores de la MSDSE a una entidad de gestión.

### C.2.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas se indican en el cuadro C.1 y son las siguientes:

- a) El parámetro TYPE indica la categoría del terminal. Toma el valor de "MASTER" o "SLAVE".
- b) El valor ERRCODE indica el tipo del error de la MSDSE. El cuadro C.5 indica los valores que puede tomar el parámetro ERRCODE.

### C.2.2.4 Estados de la MSDSE

Se utilizan los siguientes estados para especificar la secuencia permitida de primitivas entre la MSDSE y el usuario de MSDSE.

Estado 0: REPOSO

No se ha iniciado ningún procedimiento de determinación de principal-subordinado.

Estado 1: ESPERA DE RESPUESTA EN SALIDA

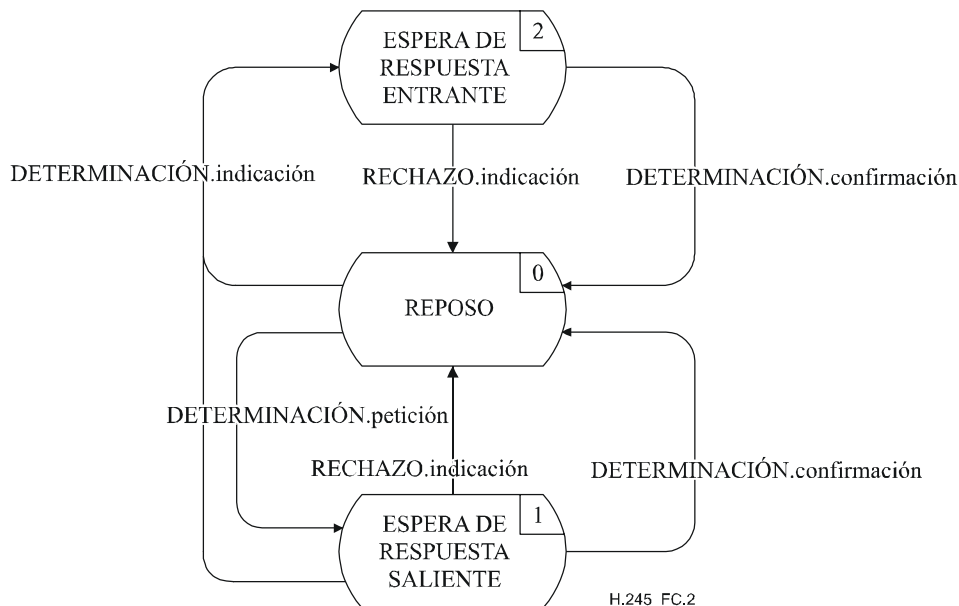
El usuario de la MSDSE local ha pedido un procedimiento de determinación principal-subordinado. Se espera una respuesta de la MSDSE distante.

## Estado 2: ESPERA DE RESPUESTA EN ENTRADA

La MSDSE distante ha iniciado un procedimiento de determinación principal-subordinado en la MSDSE local. Se envió un acuse de recibo a la MSDSE distante y se espera una respuesta de la MSDSE distante.

### C.2.2.5 Diagrama de transición de estados

La secuencia permitida de primitivas entre la MSDSE y el usuario de MSDSE se define a continuación. Las secuencias permitidas se muestran en la figura C.2.



**Figura C.2/H.245 – Diagrama de transición de estados para la secuencia de las primitivas en la MSDSE**

## C.2.3 Comunicación entre MSDSE pares

### C.2.3.1 Mensajes de MSDSE

El cuadro C.2 indica los mensajes y campos de la MSDSE, definidos en el anexo A, que son apropiados para el protocolo de la MSDSE.

**Cuadro C.2/H.245 – Nombres y campos de los mensajes de la MSDSE**

Función	Mensaje	Campo
determinación	MasterSlaveDetermination	terminalType statusDeterminationNumber
	MasterSlaveDeterminationAck	decision
	MasterSlaveDeterminationReject	cause
recuperación tras error	MasterSlaveDeterminationRelease	–

### C.2.3.2 Variables de estado de la MSDSE

Se definen las siguientes variables de estado de la MSDSE:

sv\_TT

Esta variable de estado contiene el número del tipo de terminal para este terminal.

sv\_SDNUM

Esta variable de estado contiene el número de la determinación de categoría para este terminal.

sv\_STATUS

Esta variable de estado se utiliza para almacenar el resultado del procedimiento de determinación principal-subordinado. Puede tomar los valores de "master", "slave" e "indeterminate".

sv\_NCOUNT

Esta variable de estado se utiliza para contar el número de mensajes MasterSlaveDetermination que se han enviado durante el estado ESPERA DE RESPUESTA EN SALIDA.

### C.2.3.3 Temporizadores de la MSDSE

Se especifica el siguiente temporizador para la MSDSE de salida:

T106

Se utiliza este temporizador durante el estado ESPERA DE RESPUESTA EN SALIDA y durante el estado ESPERA DE RESPUESTA EN ENTRADA. Especifica el tiempo máximo permitido durante el cual no pueden recibirse mensajes de acuse de recibo.

### C.2.3.4 Contadores de la MSDSE

Se especifica el siguiente parámetro para la MSDSE:

N100

Este parámetro especifica el valor máximo de sv\_NCOUNT.

## C.2.4 Procedimientos de la MSDSE

### C.2.4.1 Introducción

En la figura C.3 se resumen las primitivas de la MSDSE y sus parámetros, y los mensajes.

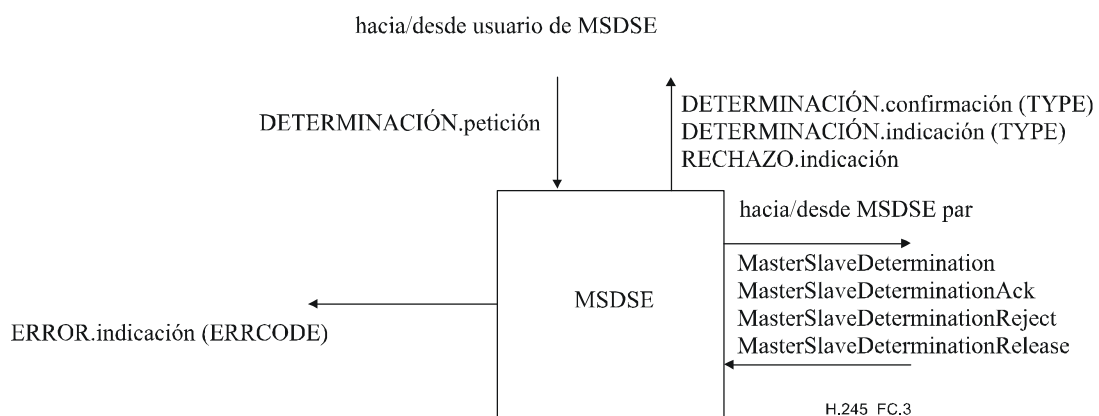


Figura C.3/H.245 – Primitivas y mensajes en la MSDSE

### C.2.4.2 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL, los parámetros de las primitivas indicación y confirmación tomarán los valores indicados en el cuadro C.3.

**Cuadro C.3/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio
DETERMINACIÓN.confirmación	TIPO	MasterSlaveDeterminationAck.decision
DETERMINACIÓN.indicación	TIPO	sv_STATUS

### C.2.4.3 Valores supletorios de los campos de mensajes

Cuando en los diagramas SDL no se indique explícitamente ningún valor, los valores de los campos de mensajes serán los indicados en el cuadro C.4.

**Cuadro C.4/H.245 – Valores supletorios de los campos de mensajes**

Mensaje	Campo	Valor supletorio
MasterSlaveDetermination	terminalType	sv_TT
	statusDeterminationNumber	sv_SDNUM
MasterSlaveDeterminationAck	decision	Opuesto a sv_STATUS, es decir, si(sv_STATUS == master) decision = slave si(sv_STATUS == slave) decision = master
MasterSlaveDeterminationReject	cause	identicalNumbers

### C.2.4.4 Valores del parámetro ERRCODE

El cuadro C.5 muestra los valores que el parámetro ERRCODE de la primitiva ERROR.indicación puede tomar para la MSDSE.

**Cuadro C.5/H.245 – Valores del parámetro ERRCODE en la MSDSE**

Tipo de error	Código de error	Condición de error	Estado
no hay respuesta de la MSDSE distante	A	expiración del temporizador T106 local	ESPERA DE RESPUESTA EN SALIDA ESPERA DE RESPUESTA EN ENTRADA
la MSDSE distante no percibe ninguna respuesta de la MSDSE local	B	expiración del temporizador T106 distante	ESPERA DE RESPUESTA EN SALIDA ESPERA DE RESPUESTA EN ENTRADA
mensaje no apropiado	C	MasterSlaveDetermination	ESPERA DE RESPUESTA EN ENTRADA
	D	MasterSlaveDeterminationReject	ESPERA DE RESPUESTA EN ENTRADA
valor de campo inconsecuente	E	MasterSlaveDeterminationAck.decision != sv_STATUS	ESPERA DE RESPUESTA EN ENTRADA
número máximo de intentos de repetición	F	sv_NCOUNT == N100	ESPERA DE RESPUESTA EN SALIDA

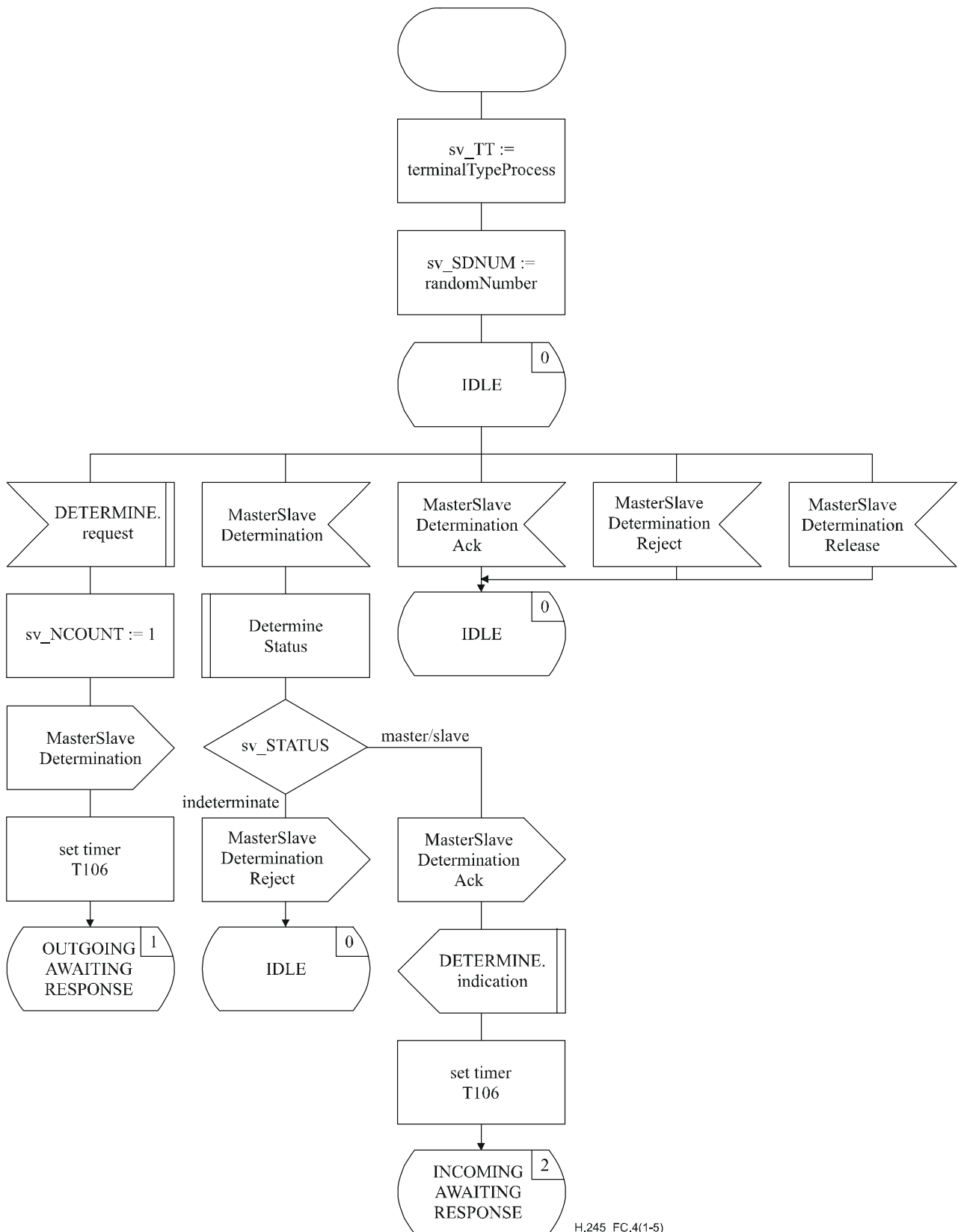
### C.2.4.5 Diagramas SDL

Los procedimientos de la MSDSE se expresan en forma de diagramas SDL en la figura C.4.



terminalTypeProcess es un proceso que devuelve un número que identifica los diferentes tipos de terminal, tales como terminales, MCU y pasarelas.

randomNumber es un proceso que devuelve un número aleatorio comprendido en la gama  $0 \dots 2^{24} - 1$ .



H.245\_FC.4(1-5)

Figura C.4/H.245 – Diagrama SDL de la MSDSE (hoja 1 de 5)

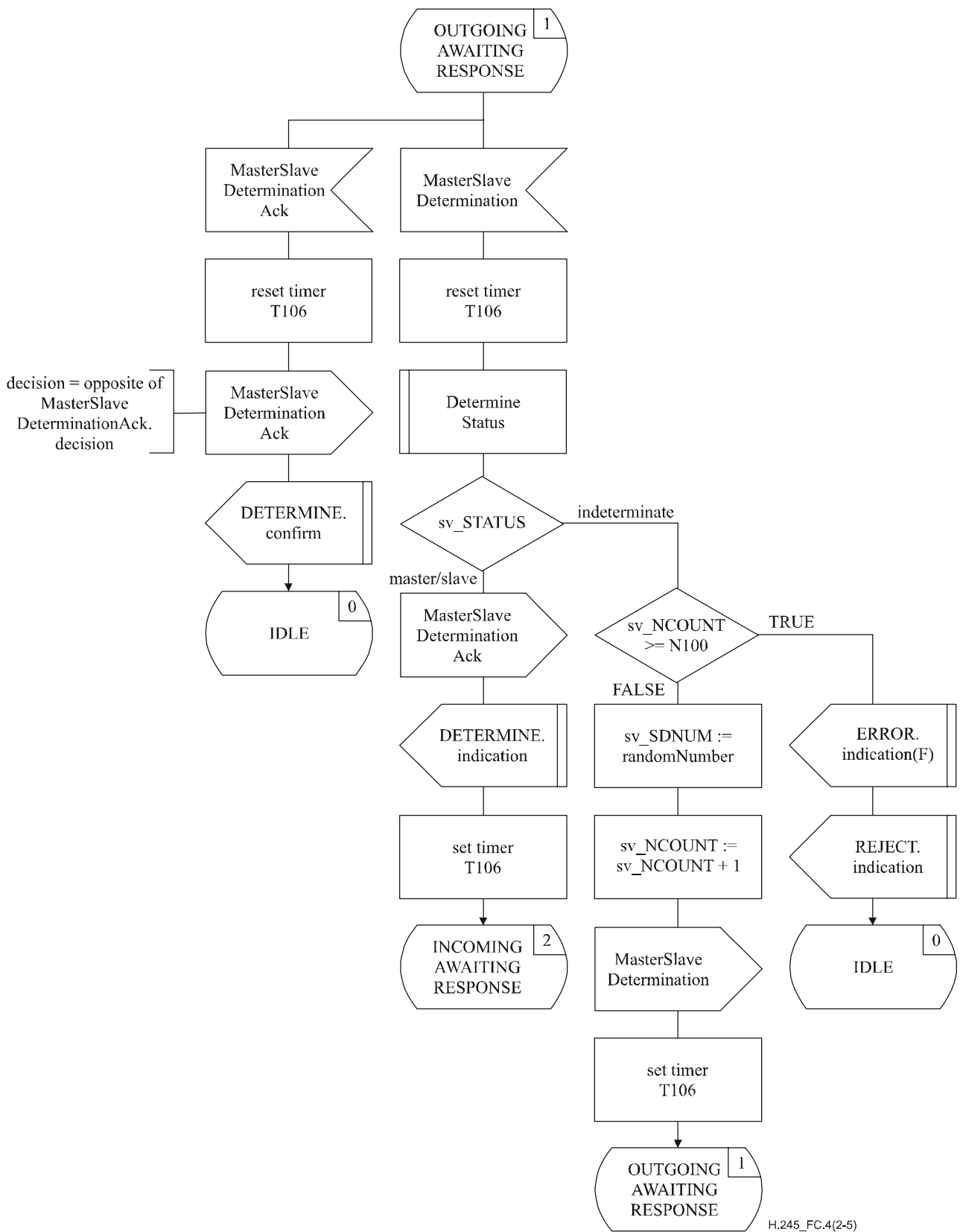
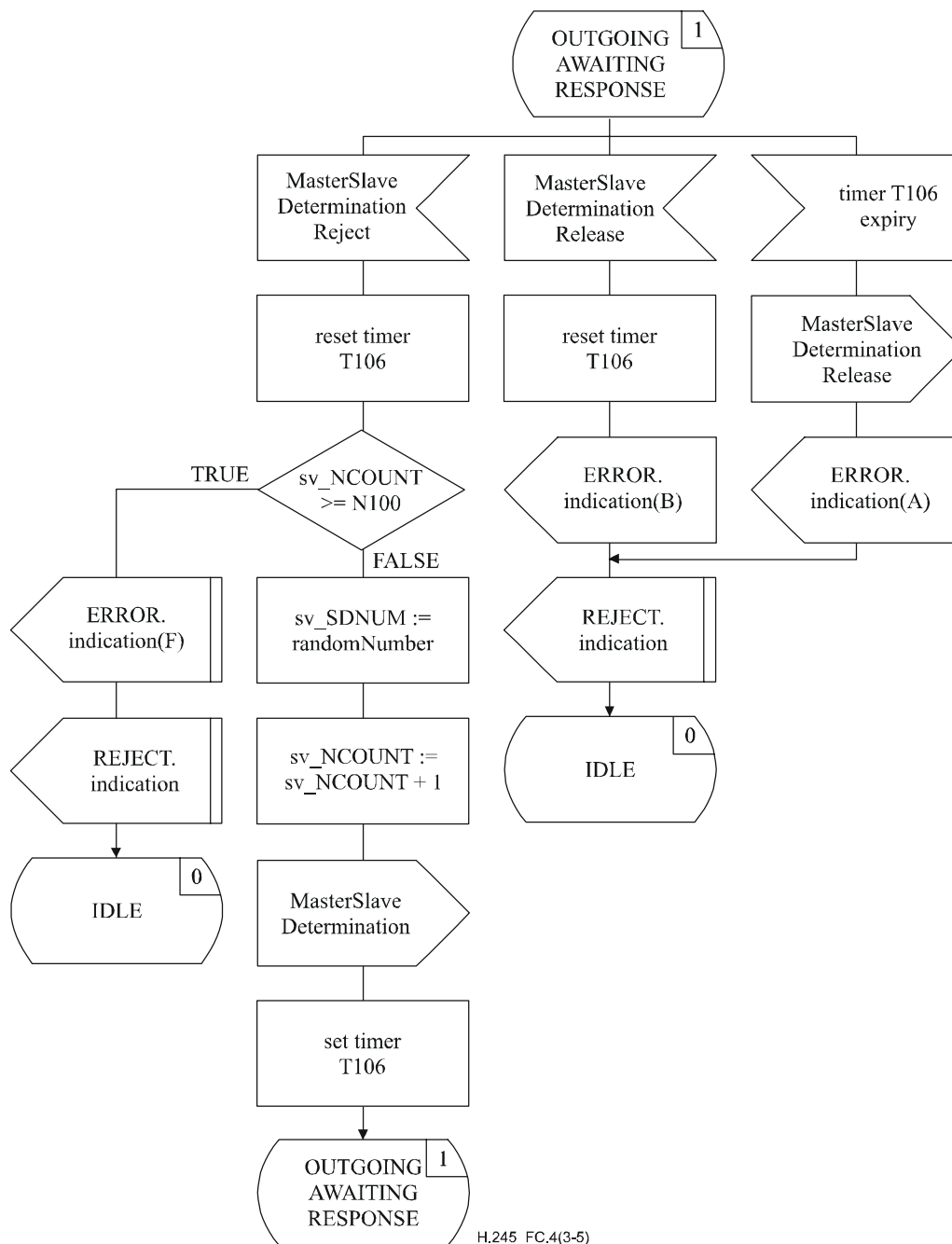


Figura C.4/H.245 – Diagrama SDL de la MSDSE (hoja 2 de 5)



**Figura C.4/H.245 – Diagrama SDL de la MSDSE (hoja 3 de 5)**

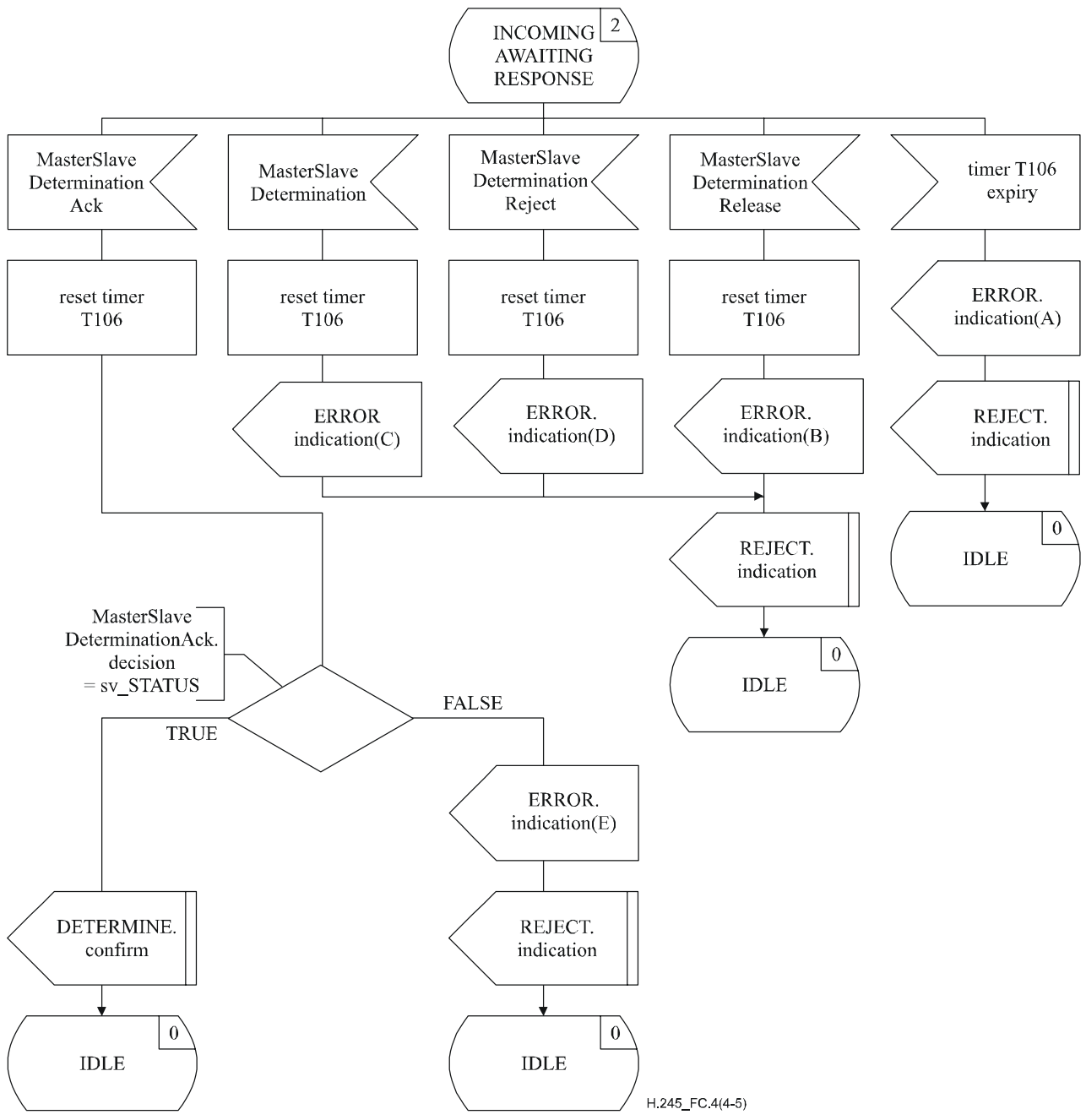
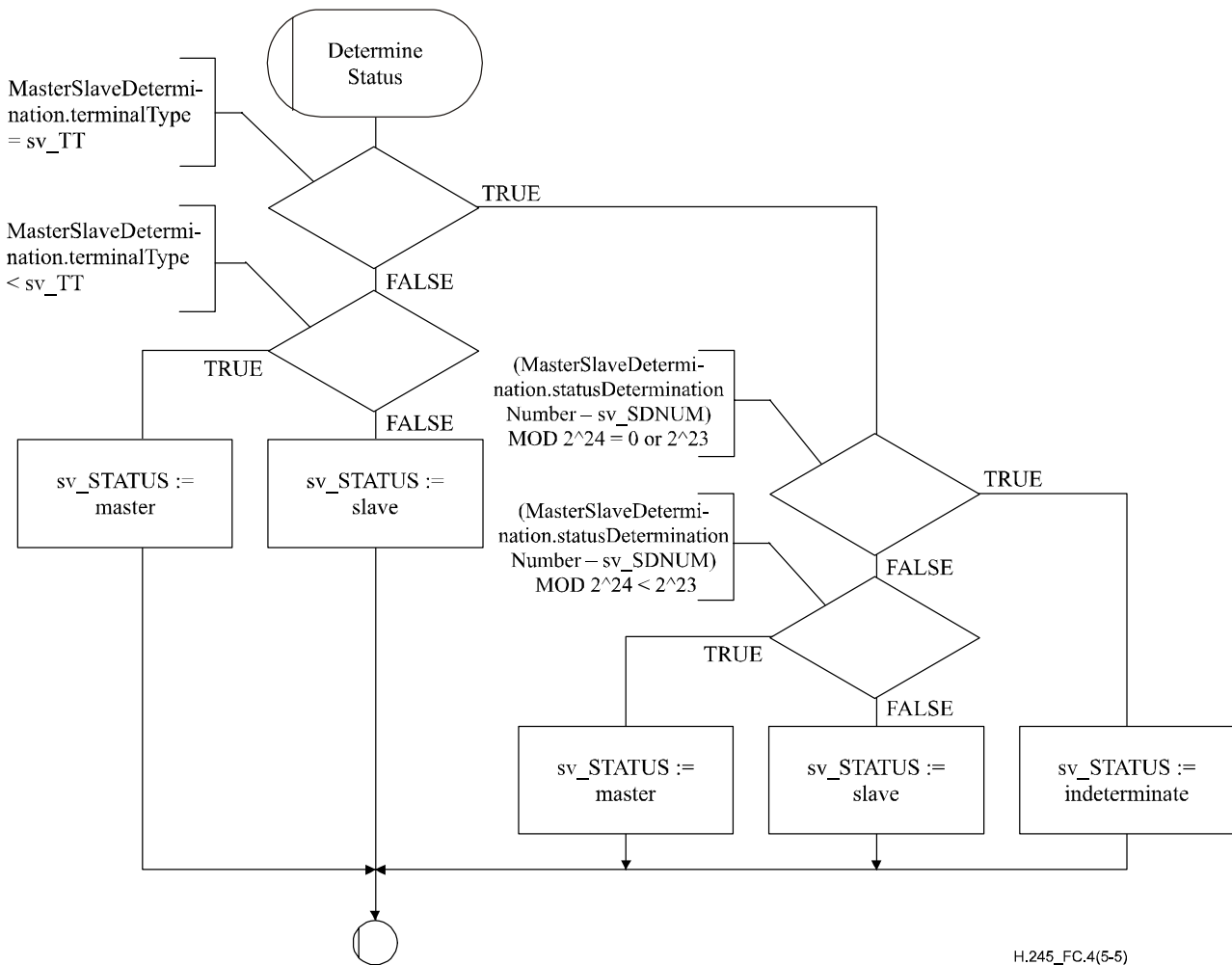


Figura C.4/H.245 – Diagrama SDL de la MSDSE (hoja 4 de 5)



H.245\_FC.4(5-5)

Figura C.4/H.245 – Diagrama SDL de la MSDSE (hoja 5 de 5)

### C.3 Procedimientos de intercambio de capacidad

#### C.3.1 Introducción

Para comunicar sus capacidades, los terminales utilizan estos procedimientos que constituyen la entidad de señalización de intercambio de capacidad (CESE, *capability exchange signalling entity*). Estos procedimientos se especifican mediante primitivas y estados en la interfaz entre la CESE y el usuario de la CESE. Se transfiere información de protocolo a la CESE por por conducto de los mensajes apropiados definidos en el anexo A. En cada extremo, de entrada y de salida, hay un ejemplar de CESE para cada llamada.

Todos los terminales que hayan de utilizarse en aplicaciones punto a punto o los conectados a una MCU deberán poder identificar un terminal TerminalCapabilitySet y su estructura, así como los valores de capacidad contenidos que serán obligatorios en esas aplicaciones. Todo valor de capacidad no reconocido deberá descartarse, sin que ello implique una situación de fallo.

El intercambio de capacidad puede realizarse en cualquier momento y podrá señalar capacidades modificadas y no modificadas. No deberán enviarse de forma repetida las capacidades no modificadas, salvo por causa justificada.

El texto que sigue proporciona una visión general del funcionamiento del protocolo. En el caso de cualquier discrepancia con la especificación formal del protocolo que sigue, prevalecerá la especificación formal.

### C.3.1.1 Visión general del protocolo – CESE de salida

Se inicia un intercambio de capacidad cuando la primitiva TRANSFERENCIA.petición es emitida por el usuario en la CESE de salida. Se envía un mensaje TerminalCapabilitySet a la CESE entrante par y se arranca el temporizador T101. Si se recibe un mensaje TerminalCapabilitySetAck en respuesta al mensaje TerminalCapabilitySet, se detiene el temporizador T101 y se informa al usuario con la primitiva TRANSFERENCIA.confirmación que el intercambio de capacidad tuvo éxito. Sin embargo, si se recibe un mensaje TerminalCapabilitySetReject en respuesta al mensaje TerminalCapabilitySet, se detiene el temporizador T101 y se informa al usuario con la primitiva RECHAZO.indicación de que el usuario CESE par ha rechazado el intercambio de capacidad.

Si expira el temporizador T101, se informa al usuario CESE saliente con la primitiva RECHAZO.indicación y se envía un mensaje TerminalCapabilitySetRelease.

### C.3.1.2 Visión general del protocolo – CESE de entrada

Cuando se recibe un mensaje TerminalCapabilitySet en la CESE de entrada, se informa al usuario de la petición de intercambio de capacidad con la primitiva TRANSFERENCIA.indicación. El usuario de la CESE de entrada señala la aceptación de la petición de intercambio de capacidad emitiendo la primitiva TRANSFERENCIA.respuesta y se envía un mensaje TerminalCapabilitySetAck a la CESE de salida par. El usuario de la CESE de entrada señala el rechazo de la petición de intercambio de capacidad emitiendo la primitiva RECHAZO.petición y se envía un mensaje TerminalCapabilitySetReject a la CESE de salida par.

## C.3.2 Comunicación entre la CESE y el usuario de CESE

### C.3.2.1 Primitivas entre la CESE y el usuario de CESE

La comunicación entre la CESE y el usuario de la CESE se realiza mediante las primitivas del cuadro C.6.

**Cuadro C.6/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
TRANSFERENCIA	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	– (nota 1)	–
RECHAZO	CAUSE	SOURCE CAUSE	no definida (nota 2)	no definida

NOTA 1 – "-" indica que no hay parámetros.  
NOTA 2 – "no definida" indica que esta primitiva no está definida.

### C.3.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- Se utilizan las primitivas TRANSFERENCIA para la transferencia del intercambio de capacidad.
- Se utilizan las primitivas RECHAZO para rechazar una inscripción de descriptor de capacidad y para terminar una transferencia de una capacidad en curso.

### **C.3.2.3 Definición de los parámetros**

La definición de los parámetros de las primitivas del cuadro C.6 es la siguiente:

- a) El parámetro PROTOID es el parámetro identificador de protocolo. Este parámetro se hace corresponder al campo protocolIdentifier del mensaje TerminalCapabilitySet y se transporta transparentemente al usuario de la CESE par. Este parámetro es obligatorio.
- b) El parámetro MUXCAP es el parámetro de capacidad múltiplex. Este parámetro se hace corresponder con el campo multiplexCapability del mensaje TerminalCapabilitySet y se transmite transparentemente al usuario de la CESE par. Este parámetro es facultativo.
- c) El parámetro CAPTABLE es el parámetro de la tabla de capacidades. Puede haber una o más entradas en la tabla de capacidades descritas en este parámetro. Este parámetro se hace corresponder con el campo capabilityTable del mensaje TerminalCapabilitySet y se transmite transparentemente al usuario de la CESE par. Este parámetro es facultativo.
- d) El parámetro CAPDESCRIPTORS es el parámetro de descriptores de capacidades. Puede haber uno o más descriptores de capacidades descritos en este parámetro. Este parámetro se hace corresponder con el campo capabilityDescriptors del mensaje TerminalCapabilitySet y se transmite transparentemente al usuario de la CESE par. Este parámetro es facultativo.
- e) El parámetro SOURCE indica la fuente de la primitiva RECHAZO.indicación. El parámetro SOURCE toma el valor de "USER" o "PROTOCOL". Este último caso puede producirse como resultado de la expiración de un temporizador.
- f) El parámetro CAUSE indica el motivo del rechazo de un parámetro CAPTABLE o CAPDESCRIPTORS. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "PROTOCOL".

### **C.3.2.4 Estados de CESE**

Para especificar la secuencia permitida de primitivas entre la CESE y el usuario de la CESE, se emplean los siguientes estados:

En el caso de una CESE de salida, los estados son:

Estado 0: REPOSO

La CESE está en reposo.

Estado 1: ESPERA DE RESPUESTA

La CESE está a la espera de una respuesta procedente de la CESE distante.

En el caso de una CESE de entrada, los estados son:

Estado 0: REPOSO

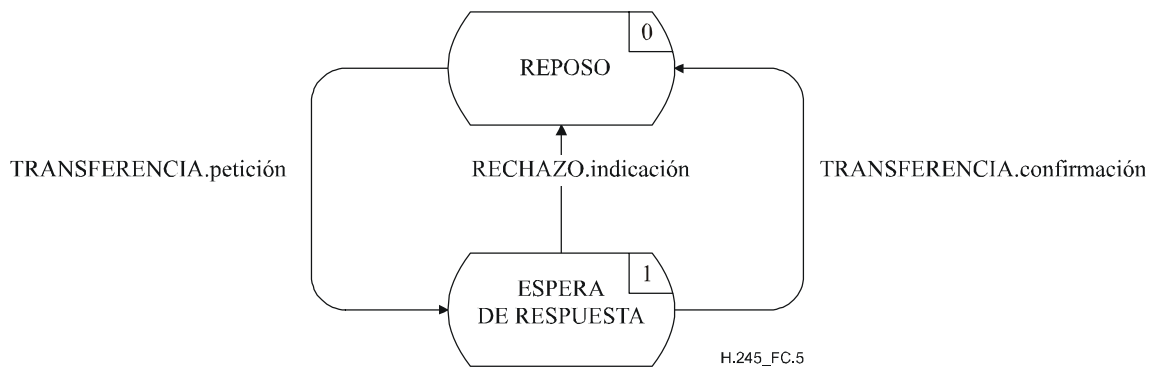
La CESE está en reposo.

Estado 1: ESPERA DE RESPUESTA

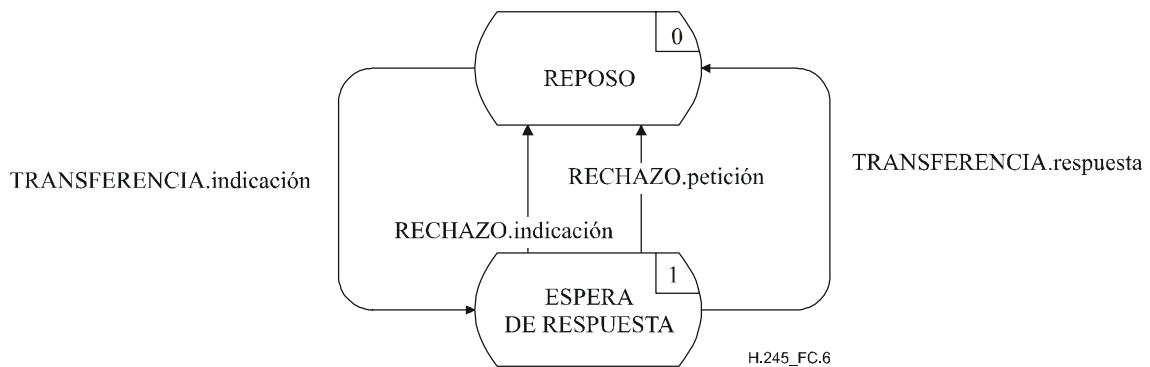
La CESE está a la espera de una respuesta procedente del usuario de la CESE.

### **C.3.2.5 Diagrama de transición de estados**

A continuación se define la secuencia permitida de primitivas entre la CESE y el usuario de la CESE. Esta secuencia se refiere a los estados de la CESE percibidos por el usuario de CESE. Las secuencias permitidas se especifican de forma separada para cada CESE de salida y cada CESE de entrada, como se representa en las figuras C.5 y C.6, respectivamente.



**Figura C.5/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CESE de salida**



**Figura C.6/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CESE de entrada**

### C.3.3 Comunicación entre CESE pares

#### C.3.3.1 Mensajes

En cuadro C.7 muestra los mensajes y campos de la CESE, definidos en el anexo A, apropiados para el protocolo de la CESE.

**Cuadro C.7/H.245 – Nombres de mensajes y campos de la CESE**

Función	Mensaje	Sentido	Campo
transferencia	TerminalCapabilitySet	O → I (nota)	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors
	TerminalCapabilitySetAck	O ← I	sequenceNumber
rechazo	TerminalCapabilitySetReject	O ← I	sequenceNumber cause
reinicio	TerminalCapabilitySetRelease	O → I	–

NOTA – Sentido: O – de salida, I – de entrada.



### C.3.3.2 Variables de estado de la CESE

En la CESE de salida se define la siguiente variable de estado:

out\_SQ

Se utiliza esta variable de estado para indicar el mensaje TerminalCapabilitySet más reciente. Antes de la transmisión del mensaje TerminalCapabilitySet, se incrementa en uno y se pone en correspondencia con el campo sequenceNumber del mensaje TerminalCapabilitySet. En out\_SQ se utiliza aritmética módulo 256.

En la CESE de entrada se define la siguiente variable de estado:

in\_SQ

Se utiliza esta variable de estado para almacenar el valor del campo sequenceNumber del mensaje TerminalCapabilitySet recibido más recientemente. Los mensajes TerminalCapabilitySetAck y TerminalCapabilitySetReject tienen sus campos sequenceNumber fijados en el valor in\_SQ antes de su envío a la CESE par.

### C.3.3.3 Temporizadores de la CESE

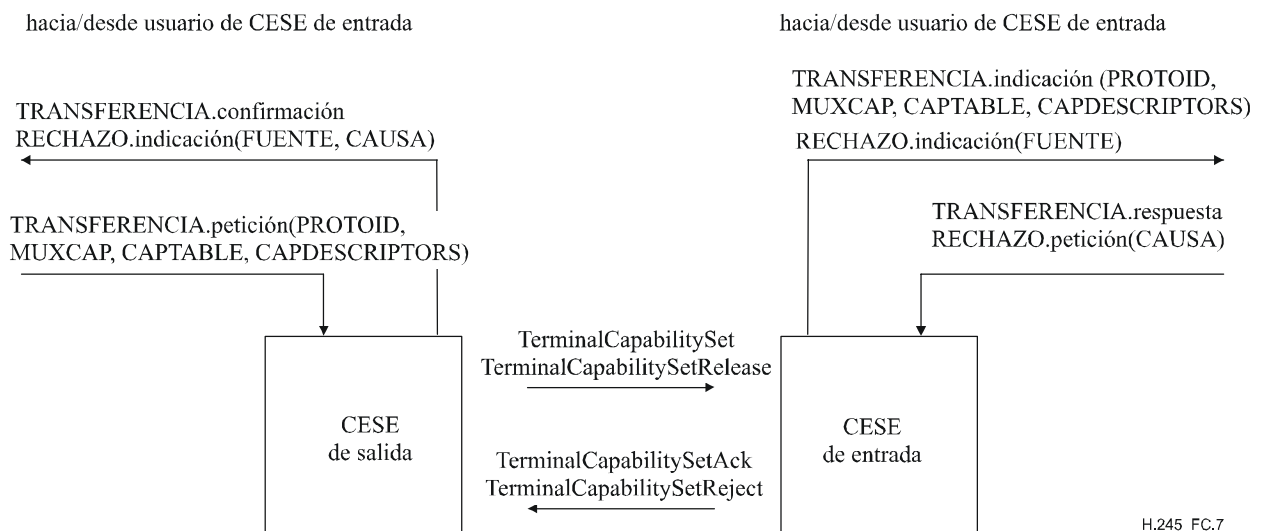
Para la CESE de salida se especifica el siguiente temporizador:

T.101

Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo intervalo de tiempo que puede transcurrir sin que se reciba un mensaje TerminalCapabilitySetAck o TerminalCapabilitySetReject.

### C.3.4 Procedimientos de CESE

En la figura C.7 se resumen las primitivas de la CESE, sus parámetros y mensajes para las CESE de salida y de entrada.



**Figura C.7/H.245 – Primitivas y mensajes en la entidad de señalización de intercambio de capacidad**

#### C.3.4.1 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL, los parámetros de las primitivas de indicación y confirmación tomarán los valores indicados en el cuadro C.8.

**Cuadro C.8/H.245 – Valores supletorios de los parámetros de las primitivas**

<b>Primitiva</b>	<b>Parámetro</b>	<b>Valor supletorio</b>
TRANSFERENCIA.indicación	PROTOID	TerminalCapabilitySet.protocolIdentifier
	MUXCAP	TerminalCapabilitySet.multiplexCapability
	CAPTABLE CAPDESCRIPTORS	TerminalCapabilitySet.capabilityTable TerminalCapabilitySet.capabilityDescriptors
RECHAZO.indicación	SOURCE	USER

**C.3.4.2 Valores supletorios de los campos de mensajes**

Cuando no se indiquen explícitamente en los diagramas SDL, los campos de mensaje tomarán los valores indicados en el cuadro C.9.

**Cuadro C.9/H.245 – Valores supletorios de los campos de mensajes**

<b>Mensaje</b>	<b>Campo</b>	<b>Valor supletorio (nota)</b>
TerminalCapabilitySet	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors	out_SQ TRANSFER.request(PROTOID) TRANSFER.request(MUXCAP) TRANSFER.request(CAPTABLE) TRANSFER.request(CAPDESCRIPTORS)
TerminalCapabilitySetAck	sequenceNumber	in_SQ
TerminalCapabilitySetReject	sequenceNumber cause	in_SQ REJECT.request(CAUSE)
TerminalCapabilitySetRelease	–	–
NOTA – No deberá codificarse ningún campo de mensaje si el parámetro de la primitiva correspondiente es nulo, es decir no está presente.		

### C.3.4.3 Diagramas de SDL

Los procedimientos en la CESE de salida y en la CESE de entrada se expresan mediante diagramas SDL en las figuras C.8 y C.9, respectivamente.

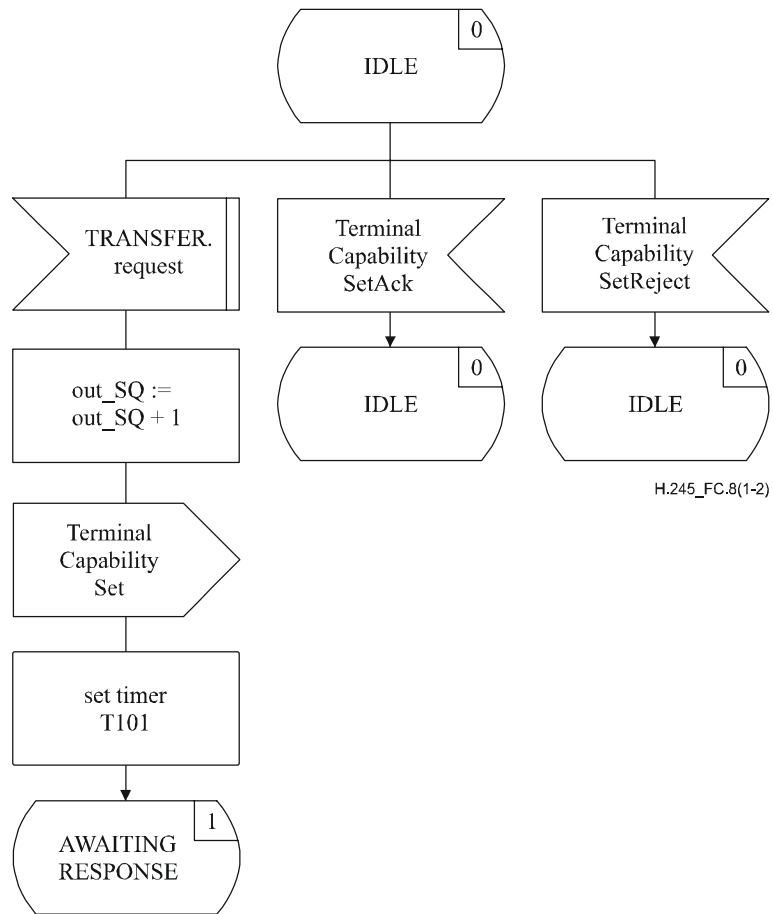
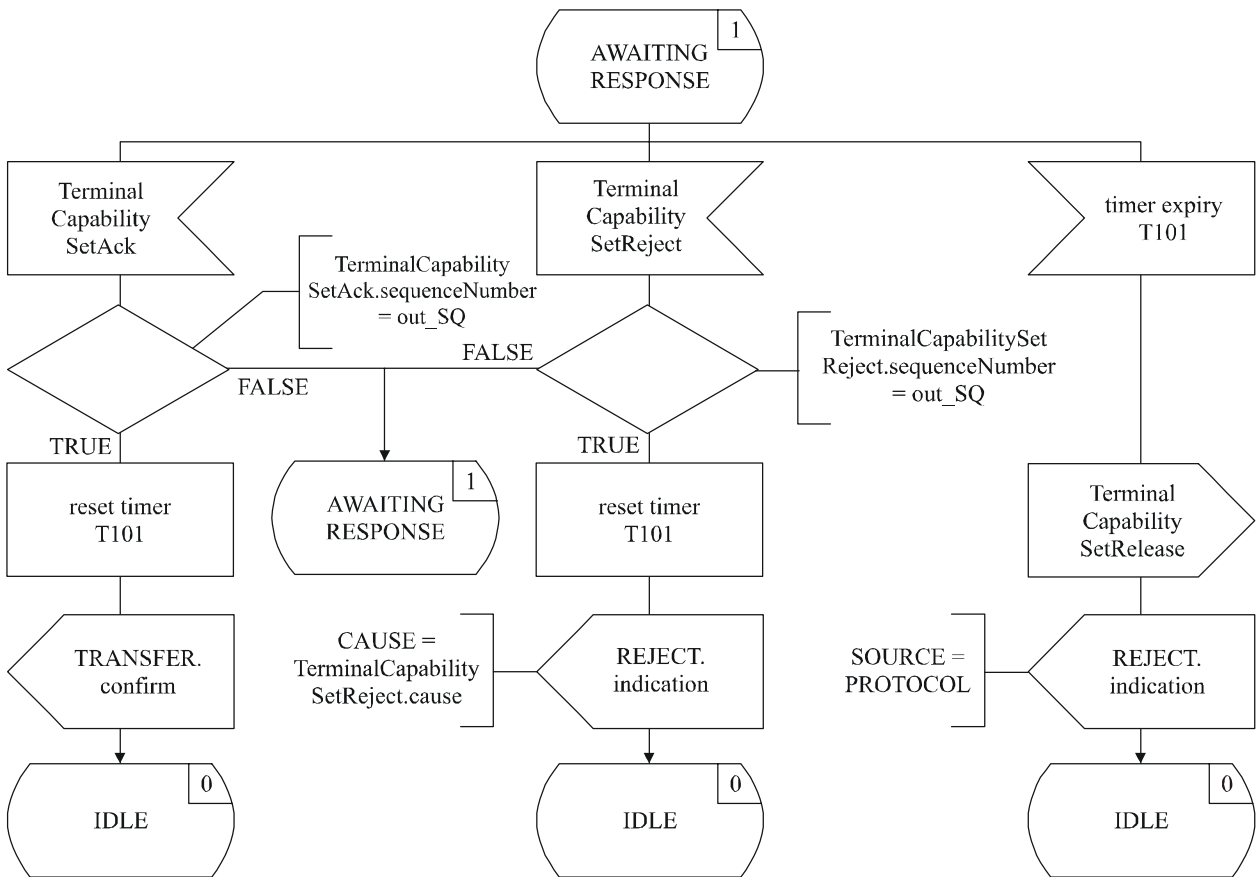
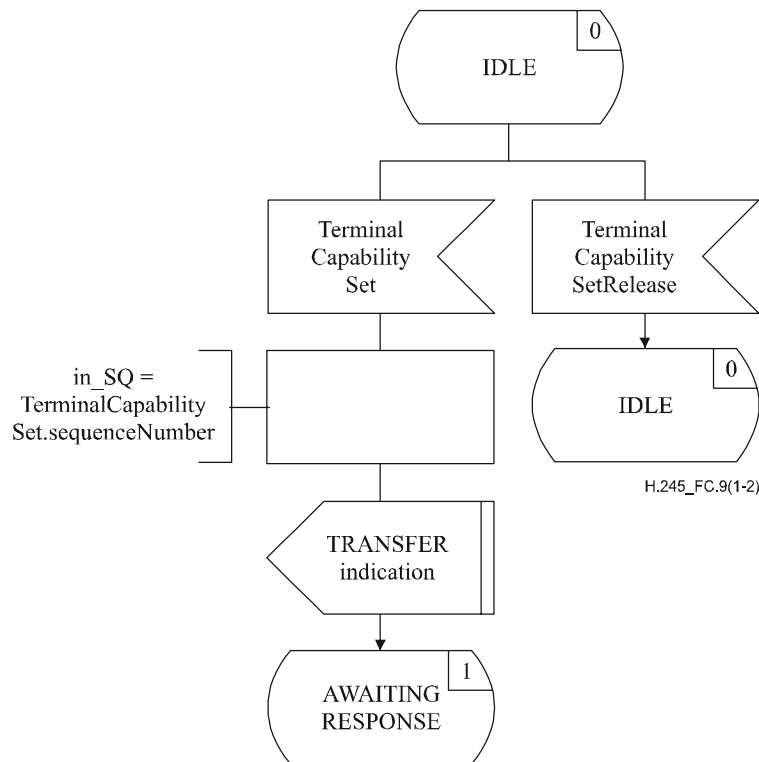


Figura C.8/H.245 – Diagrama SDL de la CESE de salida (hoja 1 de 2)



H.245\_FC.8(2-2)

**Figura C.8/H.245 – Diagrama SDL de la CESE de salida (hoja 2 de 2)**



H.245\_FC.9(1-2)

**Figura C.9/H.245 – Diagrama SDL de la CESE de entrada (hoja 1 de 2)**

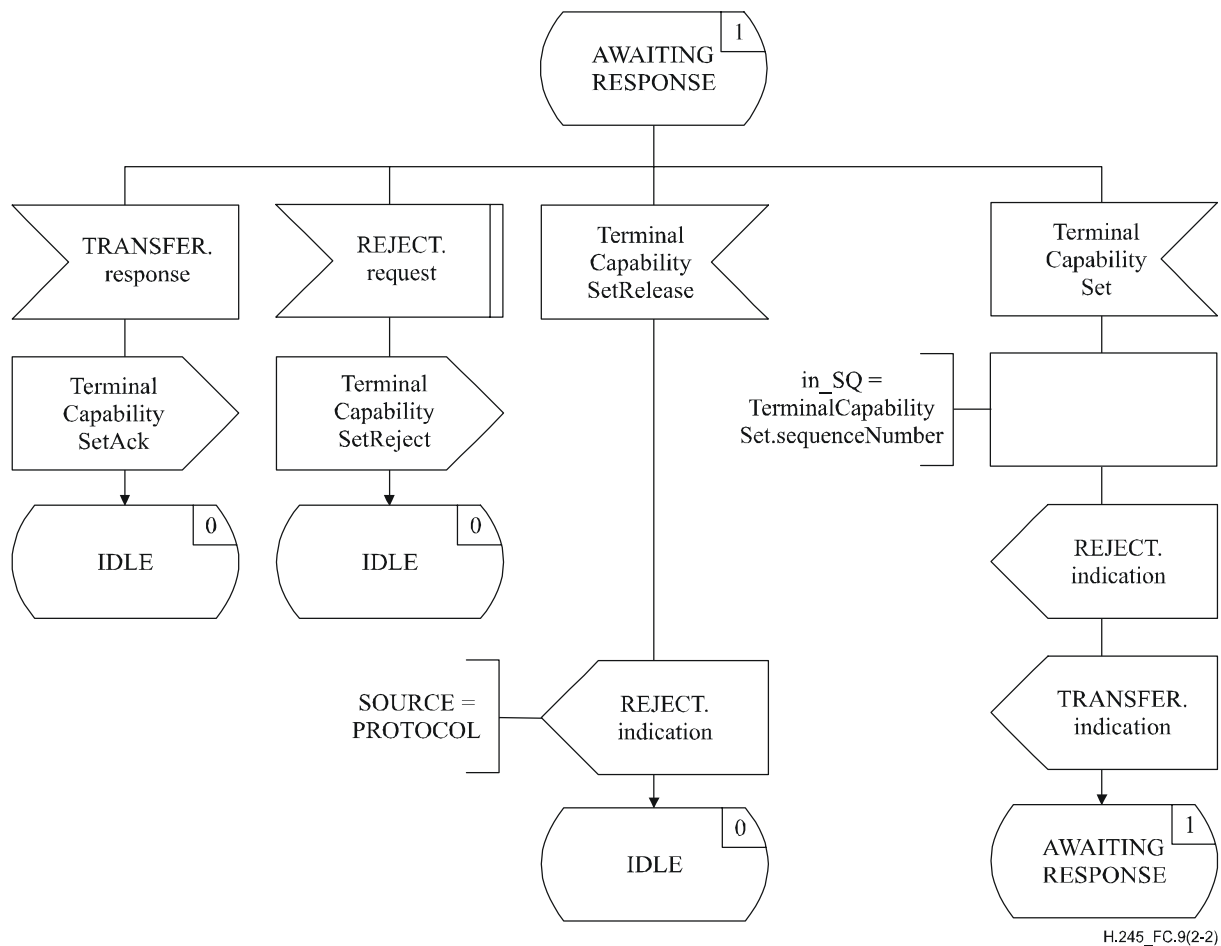


Figura C.9/H.245 – Diagrama SDL de la CESE de entrada (hoja 2 de 2)

## C.4 Procedimientos de señalización de canal lógico unidireccional

### C.4.1 Introducción

El protocolo aquí especificado proporciona la apertura y el cierre de canales lógicos unidireccionales mediante procedimientos que incluyen acuse de recibo.

El protocolo aquí especificado se denomina la entidad de señalización de canal lógico (LCSE, *logical channel signalling entity*). Los procedimientos se especifican mediante primitivas en la interfaz entre la LCSE y el usuario de la LCSE, y estados de la LCSE. La información de protocolo se transfiere a la LCSE por medio de mensajes apropiados definidos en el anexo A.

Hay una LCSE de salida y una LCSE de entrada. Tanto en el lado de salida como en el lado de entrada hay un ejemplar de la LCSE para cada canal lógico unidireccional. Entre una LCSE de entrada y una LCSE de salida, en un lado, la única comunicación se realiza mediante primitivas hacia y desde el usuario de la LCSE. Se informan las condiciones de error de la LCSE.

Los datos sólo podrán transmitirse por un canal lógico en el estado ESTABLECIDO. Si se reciben datos por un canal lógico que no está en el estado ESTABLECIDO, se descartarán dichos datos, sin que esto signifique que se ha producido un fallo.

La conmutación de modo deberá realizarse abriendo y cerrando canales lógicos existentes o abriendo nuevos canales lógicos.

NOTA – Algunas Recomendaciones que utilizan la presente Recomendación pueden definir algunos canales lógicos supletorios. Se considerará que estos canales se encuentran en el estado ESTABLECIDO desde el principio de la comunicación y no serán abiertos utilizando estos procedimientos. Estos canales podrán, sin embargo, ser cerrados por estos procedimientos y abiertos de nuevo con el mismo fin o con fines diferentes.

Un terminal que ya no sea capaz de procesar las señales transmitidas por un canal lógico deberá ejecutar las acciones apropiadas: entre éstas debe estar la de cerrar el canal lógico y transmitir la correspondiente información de capacidad (modificada) al terminal distante.

El siguiente texto proporciona una visión general del funcionamiento del protocolo LCSE. En caso de discrepancia entre esta exposición y la especificación formal, prevalecerá la especificación formal.

#### **C.4.1.1 Visión general del protocolo**

Se inicia la apertura de un canal lógico cuando la primitiva ESTABLECIMIENTO.petición es emitida por el usuario en la LCSE de salida. Se envía un mensaje OpenLogicalChannel, que contiene los parámetros de canal lógico hacia adelante pero que no incluye los parámetros de canal lógico inverso, a la LCSE de entrada par y se arranca el temporizador T103. Si se recibe un mensaje OpenLogicalChannelAck en respuesta al mensaje OpenLogicalChannel, se detiene el temporizador T103 y se informa al usuario con la primitiva ESTABLECIMIENTO.confirmación que el canal lógico se ha abierto con éxito. El canal lógico puede ahora utilizarse para transmitir información de usuario. Sin embargo, si se recibe un mensaje OpenLogicalChannelReject en respuesta al mensaje OpenLogicalChannel, se detiene el temporizador T103 y se informa al usuario con la primitiva LIBERACIÓN.indicación de que el usuario de la LCSE par ha rechazado el establecimiento del canal lógico.

Si expira el temporizador T103 en este periodo, se informa al usuario con la primitiva LIBERACIÓN.indicación y se envía a la LCSE de entrada par un mensaje CloseLogicalChannel.

Un canal lógico que ha sido establecido con éxito puede ser cerrado cuando la primitiva LIBERACIÓN.petición es emitida por el usuario en la LCSE de salida. Se envía un mensaje CloseLogicalChannel a la LCSE de entrada par y se arranca el temporizador T103. Cuando se recibe un mensaje CloseLogicalChannelAck, se detiene el temporizador T103 y se informa al usuario de que el canal lógico ha sido cerrado con éxito con la primitiva LIBERACIÓN.confirmación.

Si expira el temporizador T103 en este periodo, se informa al usuario con la primitiva LIBERACIÓN.indicación.

Antes de que se hayan recibido los mensajes OpenLogicalChannelAck u OpenLogicalChannelReject en respuesta a un mensaje OpenLogicalChannel enviado previamente, el usuario en la LCSE puede cerrar el canal lógico utilizando la primitiva LIBERACIÓN.petición.

Antes de que se reciba el mensaje CloseLogicalChannelAck en respuesta a un mensaje CloseLogicalChannel enviado previamente, el usuario en la LCSE de salida puede establecer un nuevo canal lógico emitiendo la primitiva ESTABLECIMIENTO.petición.

#### **C.4.1.2 Visión general del protocolo – LCSE de entrada**

Cuando se recibe un mensaje OpenLogicalChannel en la LCSE de entrada, se informa al usuario sobre la petición de abrir un nuevo canal lógico con la primitiva ESTABLECIMIENTO.indicación. El usuario de la LCSE de entrada señala la aceptación de la petición de establecer el canal lógico emitiendo la primitiva ESTABLECIMIENTO.respuesta, y se envía un mensaje OpenLogicalChannelAck a la LCSE de salida par. El canal lógico se puede utilizar ahora para recibir información de usuario. El usuario de la LCSE de entrada señala el rechazo de la petición de establecer el canal lógico emitiendo la primitiva LIBERACIÓN.petición y se envía un mensaje OpenLogicalChannelReject a la LCSE de salida par.

Un canal lógico que ha sido establecido con éxito se puede cerrar cuando se recibe el mensaje CloseLogicalChannel en la LCSE de entrada. El usuario de la LCSE de entrada es informado con la primitiva LIBERACIÓN.indicación y se envía el mensaje acuse de clausura de canal lógico a la LCSE de salida par.

#### **C.4.1.3 Resolución de conflictos**

Pueden surgir conflictos cuando se inician al mismo tiempo peticiones de abrir canales lógicos. Puede ser posible determinar que hay un conflicto a partir del conocimiento de las capacidades intercambiadas.

Los terminales serán capaces de detectar cuándo ha surgido, o podría surgir, un conflicto y actuarán como sigue.

Antes de que se puedan abrir canales lógicos debe determinarse que un terminal es el principal y el otro el subordinado. El protocolo definido en C.2 proporciona un medio para tomar esta decisión. El terminal principal rechazará inmediatamente cualquier petición del subordinado que identifique como una petición conflictiva. El terminal subordinado puede identificar estos conflictos, pero responderá a la petición del terminal principal con el conocimiento de que su anterior petición será rechazada.

NOTA – Estos conflictos podrían ser causados por limitaciones de los recursos de los terminales, por ejemplo, cuando son dependientes de las capacidades de transmisión y recepción, como en el caso de un terminal que pueda soportar cierto número de algoritmos de audio, pero que sólo puede decodificar el mismo algoritmo que está codificando.

Se recomiendan los siguientes procedimientos para reducir al mínimo la posibilidad de que los puntos extremos intenten abrir canales lógicos en conflicto cuando el punto extremo subordinado tiene limitaciones de capacidades simétricas. Cuando el terminal director y el terminal subordinado han indicado la posibilidad de recibir capacidades para un determinado tipo de medios, el terminal subordinado debería tratar de abrir un canal lógico destinado a la capacidad preferencial del terminal director para la que tiene capacidad, según el orden en que el terminal director ha expresado sus capacidades; y el terminal director debería tratar de abrir un canal lógico destinado a su capacidad preferencial para la que el subordinado tiene capacidad, según el orden en que ha expresado sus capacidades.

Por ejemplo, si el terminal director ha declarado la capacidad de seguir los procedimientos G.723.1, G.729 y G.711 y el subordinado la capacidad de seguir los procedimientos G.711 y G.729, con las capacidades preferenciales enumeradas primero en ambos casos, tanto el terminal director como el subordinado deberían tratar de abrir canales lógicos para los procedimientos G.729.

Una vez que el terminal director rechaza la petición de abrir un canal lógico, con causa igual a masterSlaveConflict, el terminal subordinado es responsable de abrir un canal que no presente conflictos.

Cuando el terminal subordinado detecta un conflicto y el terminal director no rechaza un canal lógico abierto en conflicto, el subordinado debería cerrar el canal en conflicto. Si hay canales lógicos en conflicto debido a limitaciones de capacidades simétricas, el terminal subordinado debería abrir un canal lógico adecuado utilizando el procedimiento sustitutivo y, a su debido tiempo, cerrar el canal lógico en conflicto.

#### **C.4.1.4 Resolución de conflictos de canales unidireccionales y bidireccionales**

Otro tipo de conflicto que podría ocurrir es cuando ambos puntos extremos intentan abrir un canal del mismo tipo, pero uno de ellos trata de abrir el canal como canal unidireccional y el otro terminal trata de abrirlo como canal bidireccional.

En tal caso el principal rechazará el canal con causa igual a masterSlaveConflict y el subordinado deberá determinar si considera necesario abrir un canal que no presenta conflictos o no efectuar acción alguna.

Cuando el subordinado detecta un conflicto y el principal no rechaza un canal lógico abierto en conflicto, el subordinado deberá cerrar el canal en conflicto.

En el apéndice X se presentan escenarios que ayudan a clarificar cómo resolver tales conflictos.

## C.4.2 Comunicación entre la LCSE y el usuario de LCSE

### C.4.2.1 Primitivas entre la LCSE y el usuario de LCSE

La comunicación entre la LCSE y el usuario de LCSE se realiza mediante las primitivas indicadas en el cuadro C.10.

**Cuadro C.10/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
ESTABLECIMIENTO	FORWARD_PARAM	FORWARD_PARAM	– (nota 1)	–
LIBERACIÓN	CAUSE	SOURCE CAUSE	no definida (nota 2)	–
ERROR	no definida	ERRCODE	no definida	no definida
NOTA 1 – "-" Indica que no hay parámetros.				
NOTA 2 – "no definida" significa que esta primitiva no existe.				

### C.4.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- Se utilizan primitivas ESTABLECIMIENTO para establecer un canal lógico para comunicaciones audiovisuales y de datos.
- Se utilizan primitivas LIBERACIÓN para liberar un canal lógico.
- Se utiliza la primitiva ERROR para notificar errores de la LCSE a alguna entidad de gestión.

### C.4.2.3 Definición de los parámetros

La definición de los parámetros de las primitivas del cuadro C.10 son es la siguiente:

- El parámetro FORWARD\_PARAM especifica los parámetros asociados con el canal lógico. Este parámetro se pone en correspondencia con el campo forwardLogicalChannelParameters del mensaje OpenLogicalChannel y se transporta transparentemente al usuario de la LCSE par.
- El parámetro SOURCE indica al usuario de la LCSE el origen de la liberación del canal lógico. El parámetro SOURCE tiene el valor "USER" o "LCSE", indicando si se trata del usuario de la LCSE o de la LCSE. Este último puede producirse como consecuencia de un error de protocolo.
- El parámetro CAUSE indica el motivo por el cual el usuario de la LCSE par rechazó una petición de establecimiento de un canal lógico. El parámetro CAUSE no estará presente cuando el parámetro SOURCE indique "LCSE".
- El parámetro ERRCODE indica el tipo de error de la LCSE. En el cuadro C.14 se indican los valores permitidos del parámetro ERRCODE.



#### **C.4.2.4 Estados de la LCSE**

Se utilizan los estados que siguen para especificar la secuencia permitida de primitivas entre la LCSE y el usuario de la LCSE, así como para el intercambio de mensajes entre LCSE pares. Tales estados se especifican de forma separada para una LCSE de salida y una LCSE de entrada. Para las LCSE de salida los estados son:

Estado 0: LIBERADO

El canal lógico está liberado. El canal lógico no se utilizará para el envío de datos de salida.

Estado 1: ESPERA DE ESTABLECIMIENTO

La LCSE de salida está a la espera de establecer un canal lógico con una LCSE de entrada par. No se utilizará el canal lógico para enviar datos de salida.

Estado 2: ESTABLECIDO

Se ha establecido la conexión del canal lógico entre LCSE pares. Puede utilizarse el canal lógico para enviar datos de salida.

Estado 3: ESPERA DE LIBERACIÓN

La LCSE de salida está a la espera de liberar un canal lógico con la LCSE de entrada par. No se utilizará el canal lógico para enviar datos de salida.

Los estados para la LCSE de entrada son:

Estado 0: LIBERADO

El canal lógico está liberado. No se utilizará el canal lógico para recibir datos de entrada.

Estado 1: ESPERA DE ESTABLECIMIENTO

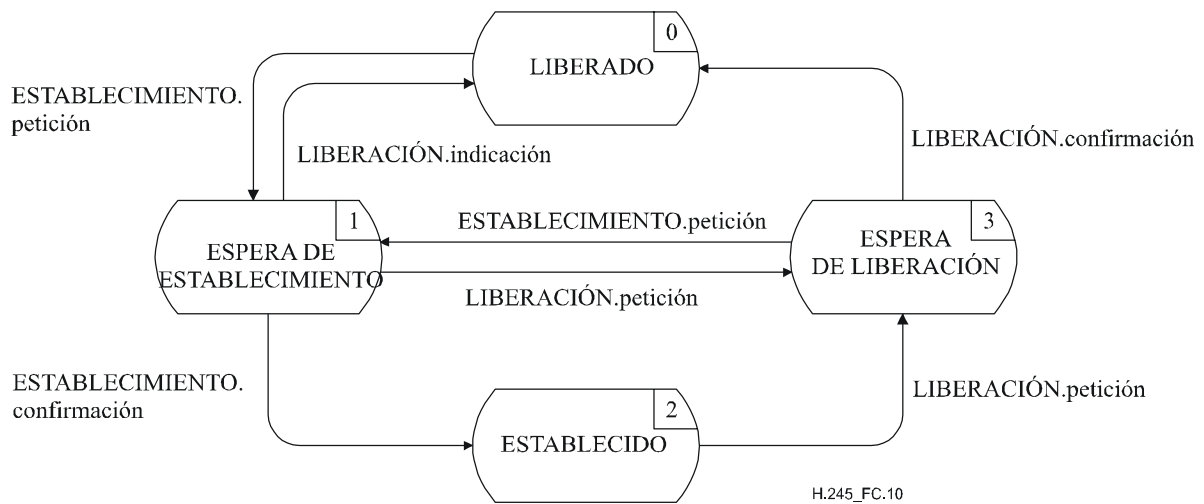
La LCSE de entrada está a la espera de establecer un canal lógico con una LCSE de salida par. No se utilizará el canal lógico para recibir datos de entrada.

Estado 2: ESTABLECIDO

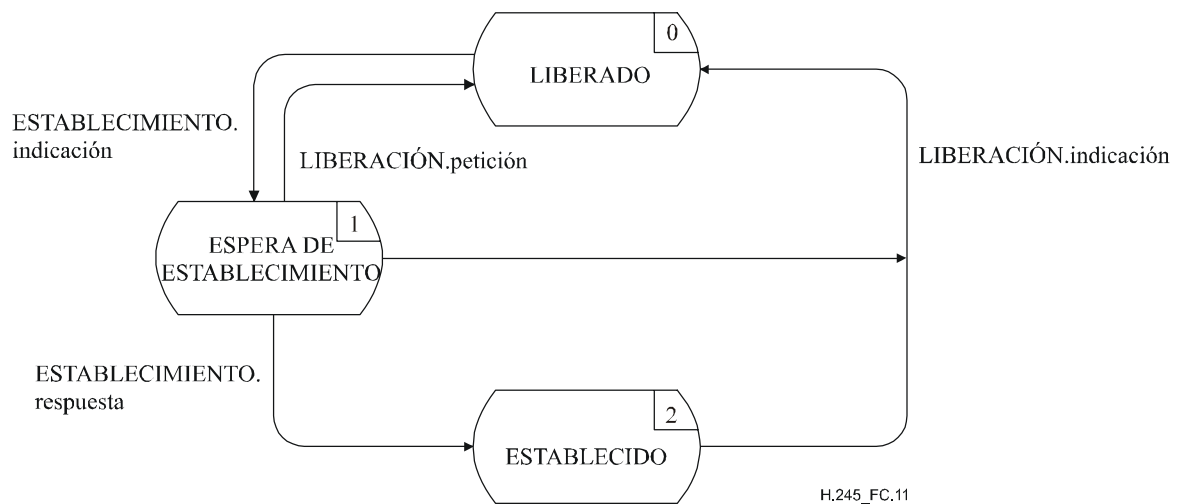
Se ha establecido una conexión de canal lógico entre las LCSE pares. Puede utilizarse el canal lógico para recibir datos de entrada.

#### **C.4.2.5 Diagrama de transición de estados**

Se define aquí la secuencia permitida de primitivas entre la LCSE y el usuario de la LCSE. Esta secuencia se relaciona con los estados de la LCSE tal y como los percibe el usuario de la LCSE. Las secuencias permitidas se especifican por separado para cada LCSE de salida y LCSE de entrada, como se representa en las figuras C.10 y C.11 respectivamente.



**Figura C.10/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una LCSE de salida**



**Figura C.11/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una LCSE de entrada**

### C.4.3 Comunicación entre LCSE pares

#### C.4.3.1 Mensajes de LCSE

El cuadro C.11 muestra los mensajes y campos de la LCSE, definidos en el anexo A, que son apropiados para el protocolo de la LCSE.

**Cuadro C.11/H.245 – Nombres de mensajes y campos de la LCSE**

<b>Función</b>	<b>Mensajes</b>	<b>Sentido</b>	<b>Campo</b>
establecimiento	OpenLogicalChannel	O → I (nota)	forwardLogicalChannelNumber forwardLogicalChannelParameters
	OpenLogicalChannelAck	O ← I	forwardLogicalChannelNumber
	OpenLogicalChannelReject	O ← I	forwardLogicalChannelNumber cause
liberación	CloseLogicalChannel	O → I	forwardLogicalChannelNumber source
	CloseLogicalChannelAck	O ← I	forwardLogicalChannelNumber
NOTA – Sentido: O – de salida; I – de entrada.			

### **C.4.3.2 Variables de estado de la LCSE**

En la LCSE de salida se define la siguiente variable de estado:

out\_LCN

Esta variable de estado permite diferenciar las LCSE de salida. Se inicializa en el proceso de inicialización de la LCSE de salida. Se utiliza el valor de out\_LCN para fijar el campo forwardLogicalChannelNumber de los mensajes LCSE enviados desde una LCSE de salida. Para mensajes LCSE recibidos en una LCSE de salida, el valor del campo forwardLogicalChannelNumber es idéntico al valor de out\_LCN.

En la LCSE de entrada se define la siguiente variable de estado:

in\_LCN

Esta variable de estado permite diferenciar las LCSE de entrada. Se inicializa en el proceso de inicialización de la LCSE de entrada. Se utiliza el valor de in\_LCN para fijar el campo forwardLogicalChannelNumber de los mensajes LCSE enviados desde una LCSE de entrada. Para mensajes LCSE recibidos en una LCSE de entrada, el valor del campo forwardLogicalChannelNumber es idéntico al valor de in\_LCN.

### **C.4.3.3 Temporizadores de la LCSE**

Para la LCSE de salida se especifica el siguiente temporizador:

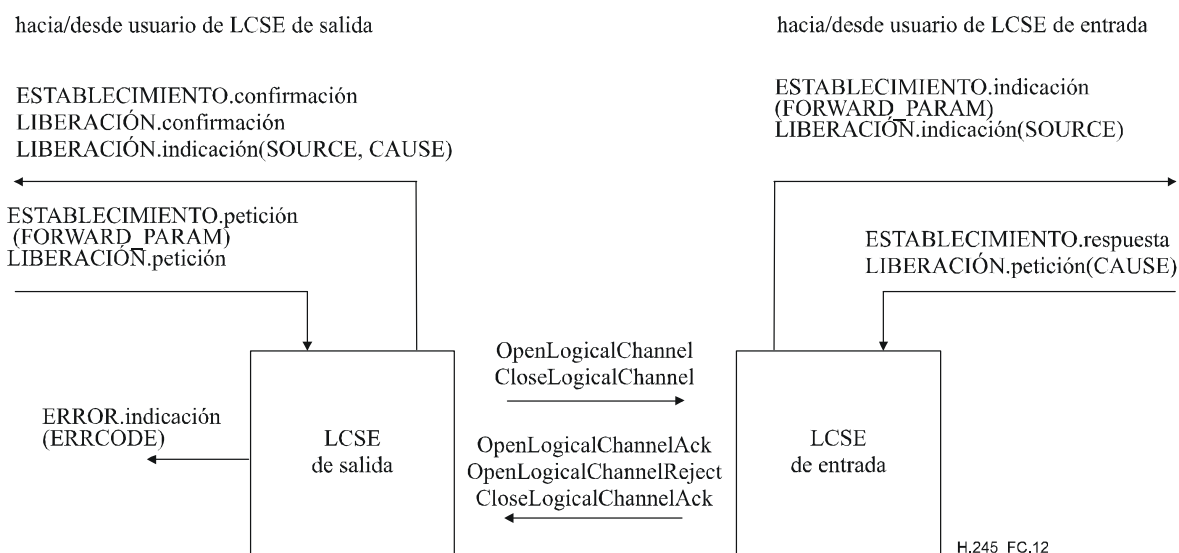
T103

Este temporizador se utiliza en los estados ESPERA DE ESTABLECIMIENTO y ESPERA DE LIBERACIÓN. Especifica el intervalo de tiempo máximo permitido durante el cual no se reciben mensajes de OpenLogicalChannelAck, OpenLogicalChannelReject ni CloseLogicalChannelAck.

## **C.4.4 Procedimientos de LCSE**

### **C.4.4.1 Introducción**

En la figura C.12 se resumen las primitivas, sus parámetros y los mensajes para cada una de las LCSE de salida y de entrada.



**Figura C.12/H.245 – Primitivas y mensajes en la entidad de señalización de canal lógico**

#### C.4.4.2 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas SDL, los valores de los parámetros de las primitivas indicación y confirmación serán los del cuadro C.12.

**Cuadro C.12/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio (nota)
ESTABLECIMIENTO.indicación	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
LIBERACIÓN.indicación	SOURCE	CloseLogicalChannel.source
	CAUSE	null
NOTA – Un parámetro de primitiva se codificará como nulo si no está presente en el mensaje un campo de mensaje indicado.		

### C.4.4.3 Valores supletorios de los campos de mensajes

Cuando no se indique de forma explícita en los diagramas SDL, los valores supletorios de los campos de mensajes serán los del cuadro C.13.

**Cuadro C.13/H.245 – Valores supletorios de los campos de mensajes**

Mensaje	Campo	Valor supletorio (nota 1)
OpenLogicalChannel (nota 2)	forwardLogicalChannelNumber	out_LCN
	forwardLogicalChannelParameters	ESTABLISH.request (FORWARD_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
OpenLogicalChannelReject	forwardLogicalChannelNumber	in_LCN
	cause	RELEASE.request (CAUSE)
CloseLogicalChannel	forwardLogicalChannelNumber	out_LCN
	source	user
CloseLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
<p>NOTA 1 – No se codificará un campo de mensaje si el parámetro de la primitiva correspondiente es nulo, es decir no está presente.</p> <p>NOTA 2 – reverseLogicalChannelParameters no se codifican en procedimientos de señalización de canal lógico unidireccional.</p>		

### C.4.4.4 Valores del parámetro ERRCODE

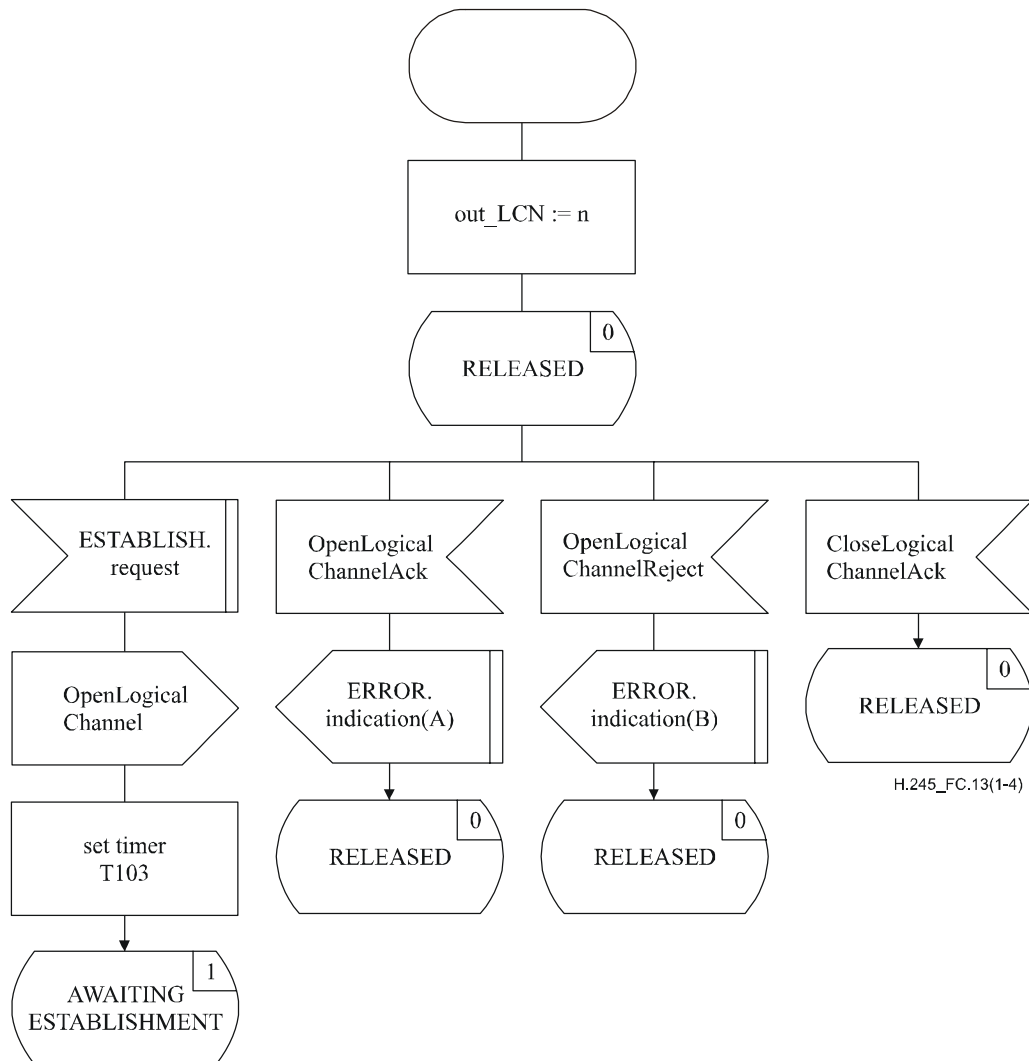
El parámetro ERRCODE de la primitiva ERROR.indicación señala una condición de error determinada. En el cuadro C.14 se muestran los valores que puede tomar el parámetro ERRCODE en la LCSE de salida. No hay ninguna primitiva ERROR.indicación asociada con la LCSE de entrada.

**Cuadro C.14/H.245 – Valores del parámetro ERRCODE en la LCSE de salida**

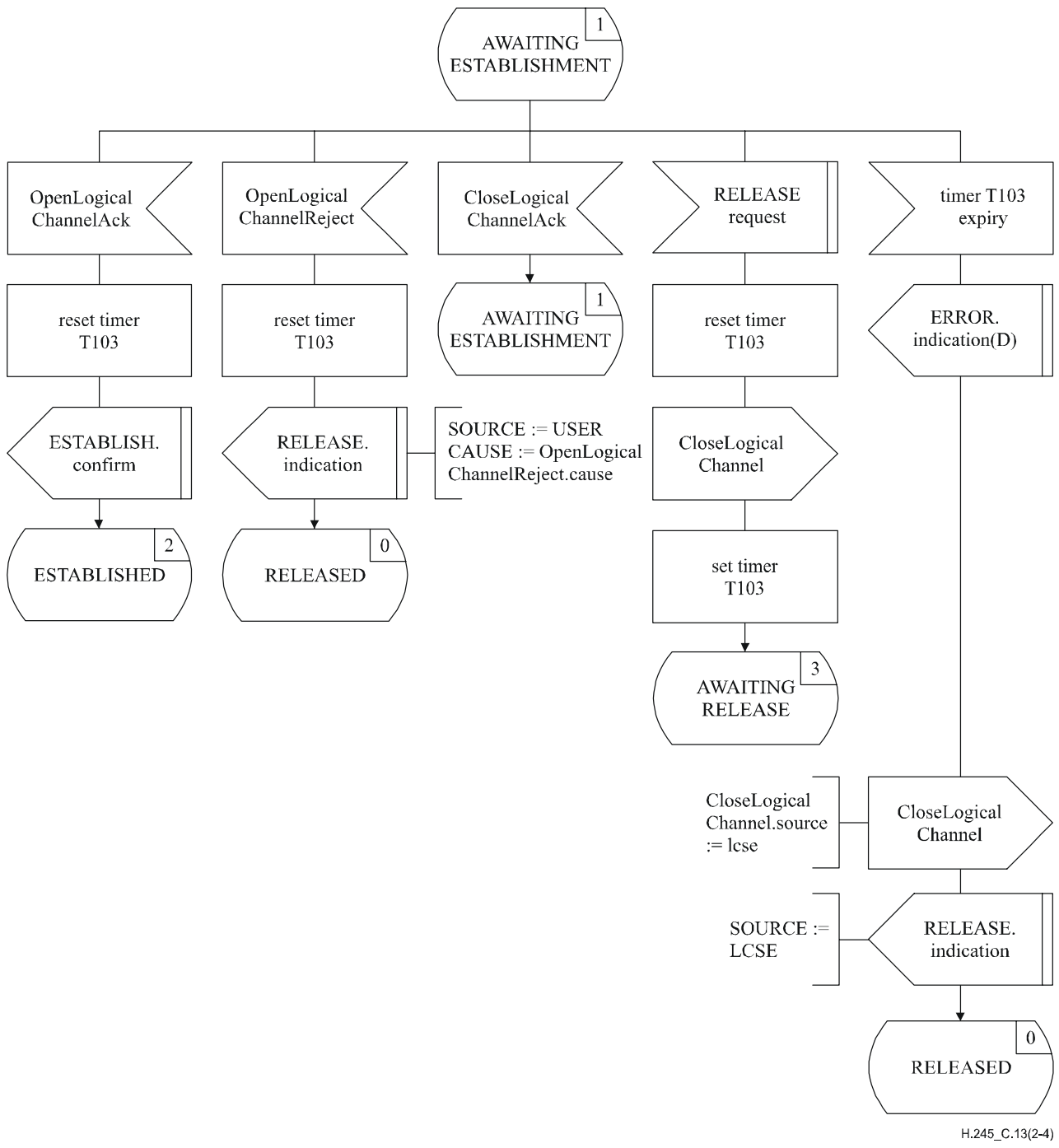
Tipo de error	Código de error	Condición de error	Estado
mensaje inapropiado	A	OpenLogicalChannelAck	LIBERADO
	B	OpenLogicalChannelReject	LIBERADO ESTABLECIDO
	C	CloseLogicalChannelAck	ESTABLECIDO
no hay respuesta de la LCSE par	D	Expiración del temporizador T103	ESPERA DE ESTABLECIMIENTO ESPERA DE LIBERACIÓN

### C.4.4.5 Diagramas de SDL

En las figuras C.13 y C.14 se representan, en SDL, los procedimientos en la LCSE de salida y en la LCSE de entrada, respectivamente.

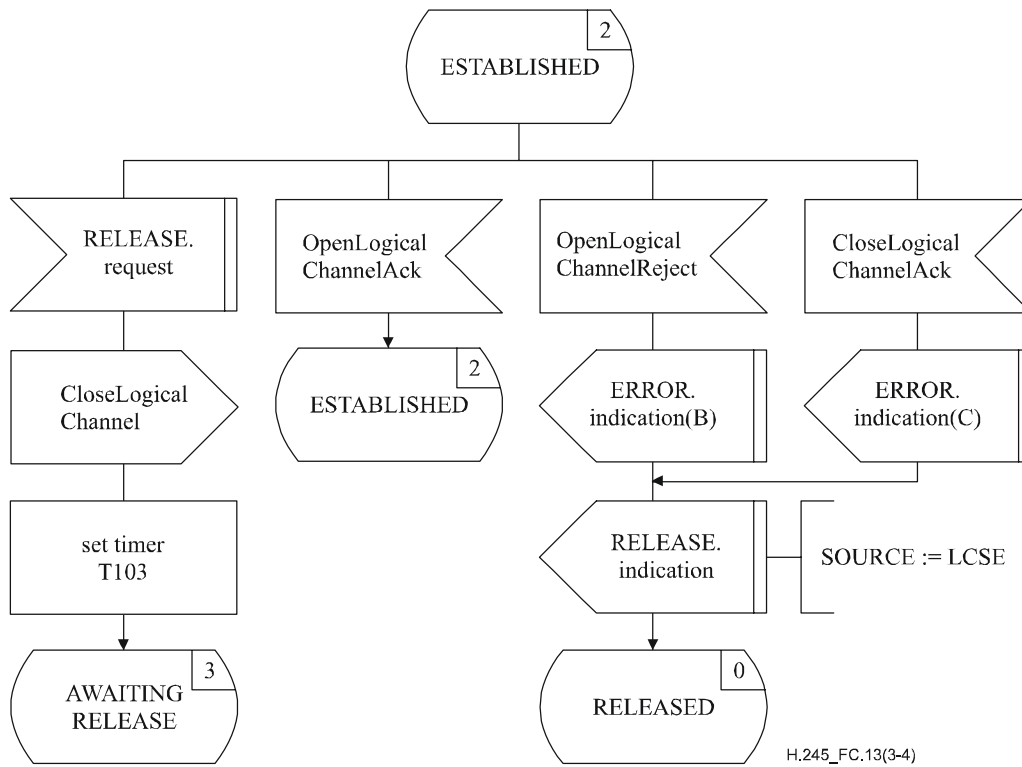


**Figura C.13/H.245 – Diagrama SDL de la LCSE de salida (hoja 1 de 4)**

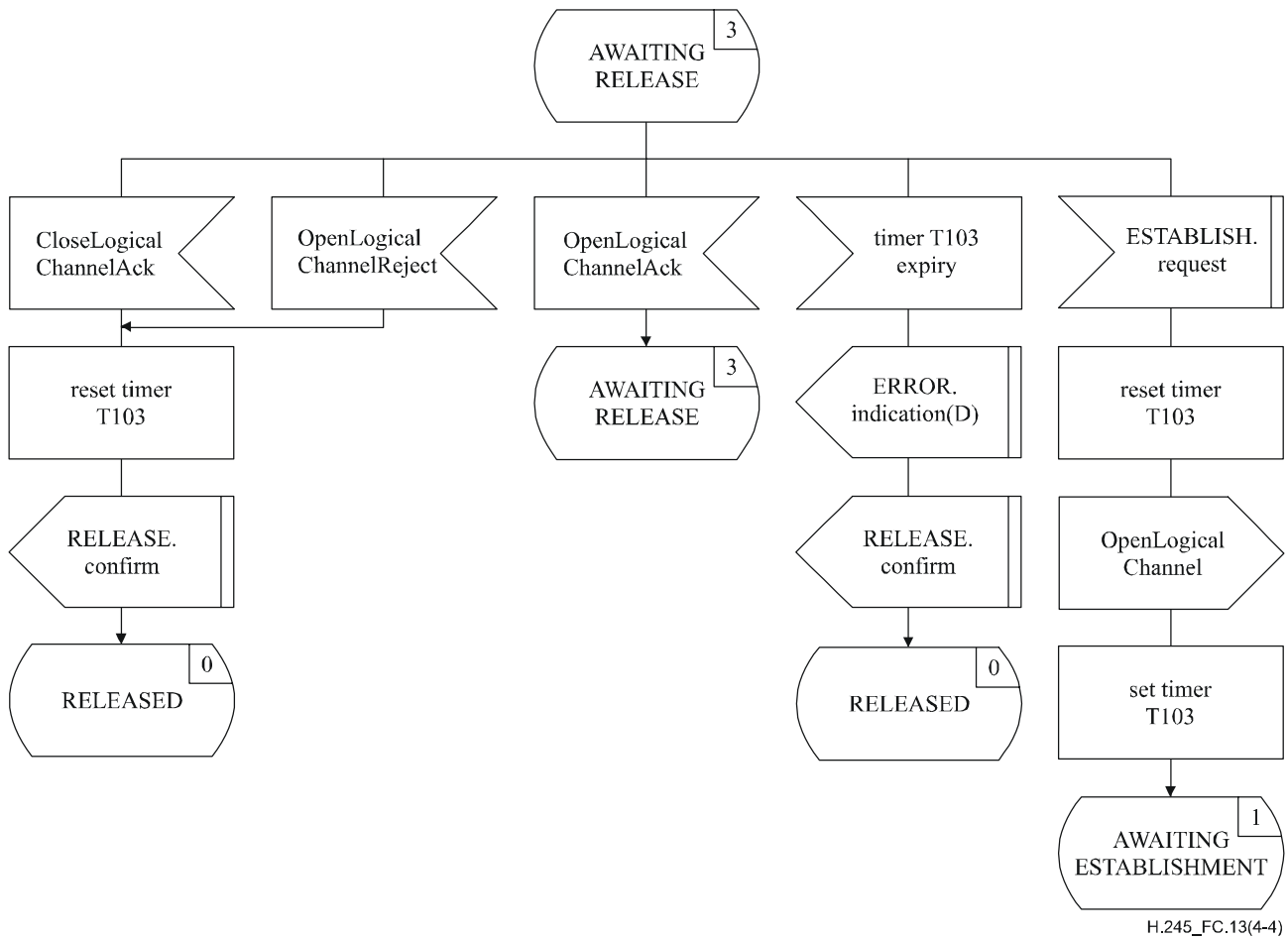


H.245\_C.13(2-4)

Figura C.13/H.245 – Diagrama SDL de la LCSE de salida (hoja 2 de 4)

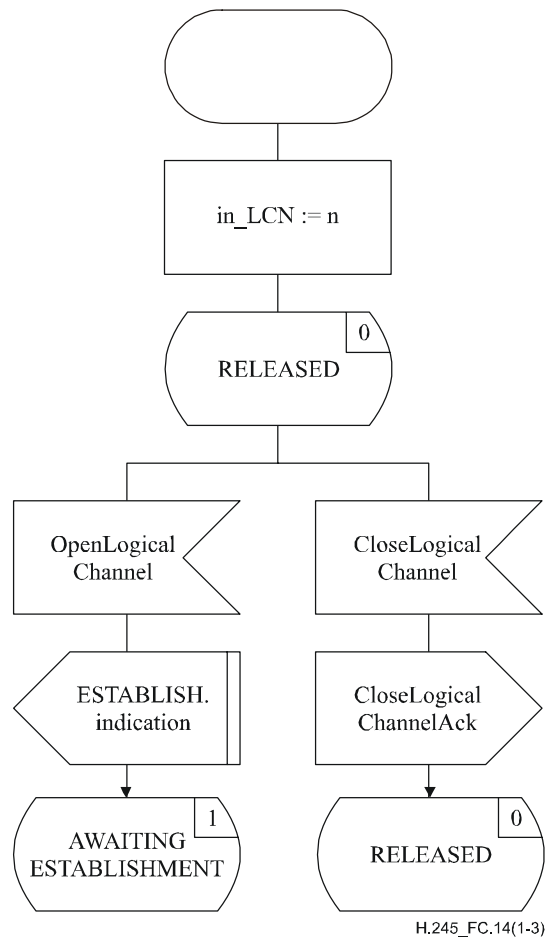


**Figura C.13/H.245 – Diagrama SDL de la LCSE de salida (hoja 3 de 4)**

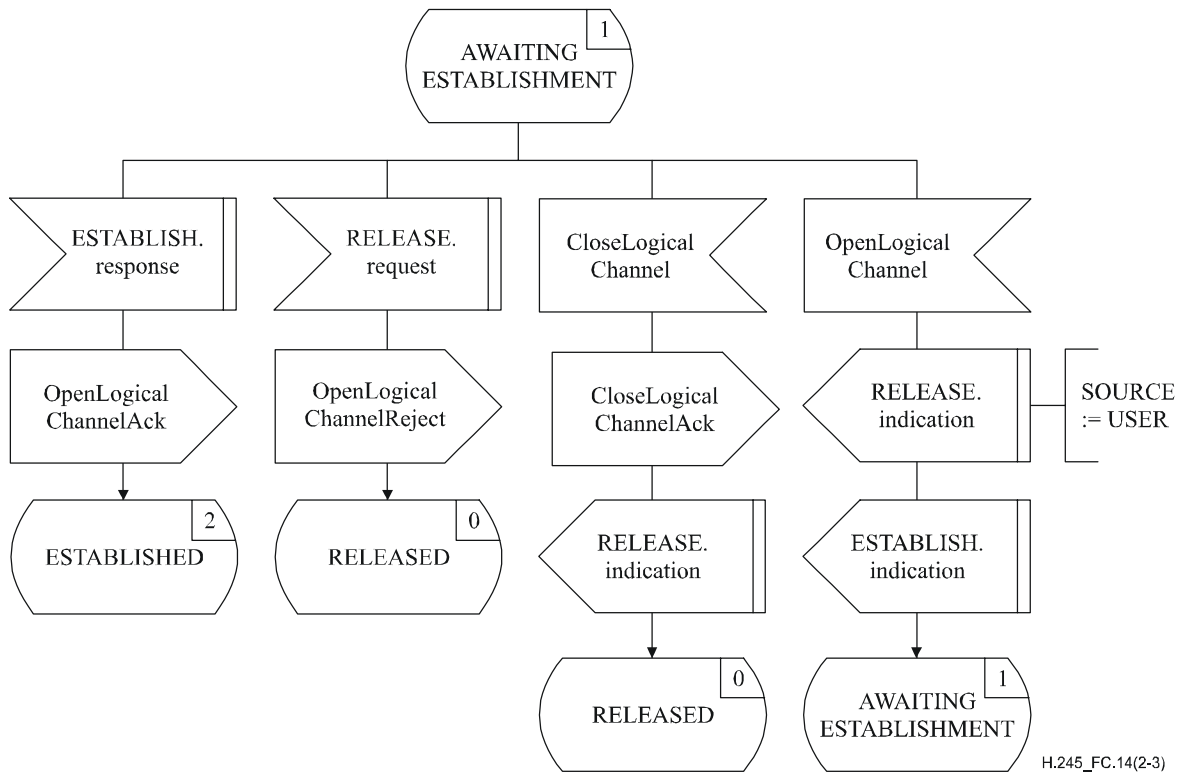


**Figura C.13/H.245 – Diagrama SDL de la LCSE de salida (hoja 4 de 4)**

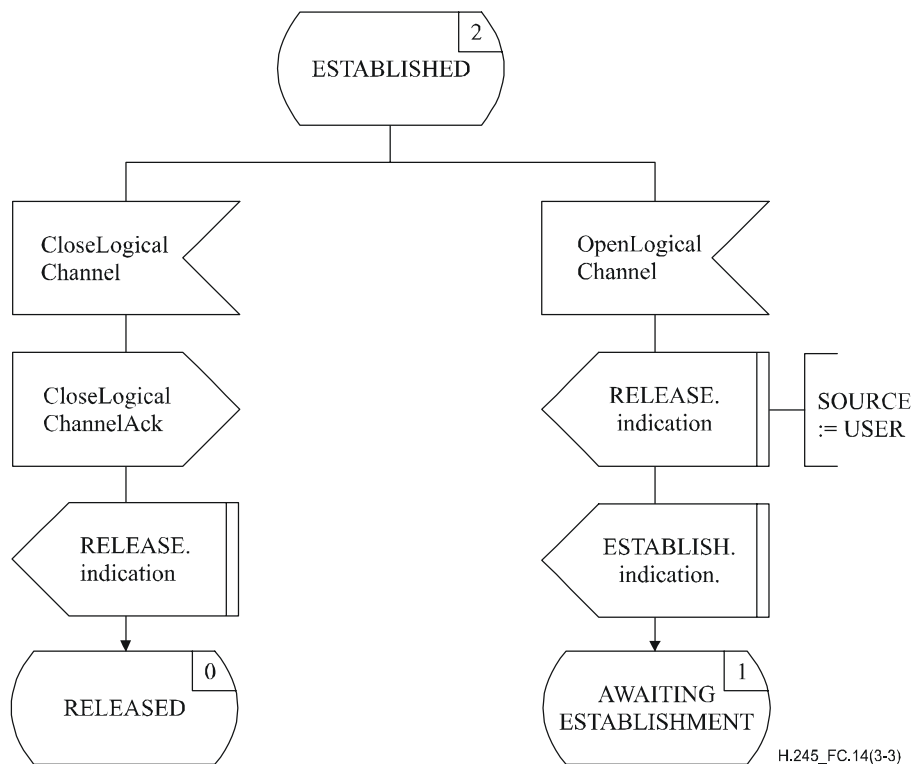




**Figura C.14/H.245 – Diagrama SDL de la LCSE de entrada (hoja 1 de 3)**



**Figura C.14/H.245 – Diagrama SDL de la LCSE de entrada (hoja 2 de 3)**



**Figura C.14/H.245 – Diagrama SDL de la LCSE de entrada (hoja 3 de 3)**

## C.5 Procedimientos de señalización de canal lógico bidireccional

### C.5.1 Introducción

El protocolo aquí especificado permite la apertura y cierre fiables de canales lógicos bidireccionales mediante procedimientos con acuse de recibo.

Estos procedimientos se designan como entidad de señalización de canal lógico bidireccional (B-LCSE, *bidirectional logical channel signalling entity*). Los procedimientos se especifican mediante primitivas en las interfaces entre la B-LCSE y el usuario de la B-LCSE, y estados de la B-LCSE. La información de protocolo se transfiere a la B-LCSE por medio de mensajes apropiados, definidos en el anexo A.

Hay una B-LCSE salida y una B-LCSE de entrada. Tanto en el lado de salida como en el de entrada hay un ejemplar de la B-LCSE para cada canal lógico bidireccional. La única conexión que existe entre una B-LCSE de entrada y una B-LCSE de salida en un lado es a través de primitivas hacia y desde el usuario de la B-LCSE. Se informan las condiciones de error de la B-LCSE.

Un canal lógico bidireccional consiste en un par de canales unidireccionales asociados. Se utiliza "de ida" (en el lado de salida) para hacer referencia al sentido de transmisión desde el terminal que hace la petición de un canal lógico bidireccional al otro terminal e "inverso" (en el lado de entrada) para hacer referencia al sentido opuesto de transmisión.

Los datos sólo se transmitirán por un canal lógico en el estado ESTABLECIDO. En cambio, se pueden recibir datos por el canal de ida cuando la B-LCSE está en el estado ESPERA DE CONFIRMACIÓN. Los datos que se reciban en otros estados que no sean los de ESTABLECIDO y ESPERA DE CONFIRMACIÓN se descartarán, sin que por ello deba considerarse que se ha producido un fallo.

Un terminal puede rechazar una petición de apertura de un canal lógico bidireccional por el solo hecho de que no puede soportar los parámetros de canal inverso solicitados. En este caso rechazará la petición con cause igual a `unsuitableReverseParameters` e iniciará inmediatamente

procedimientos para establecer un canal lógico bidireccional como lo solicita el terminal distante, en el que los parámetros del canal inverso son idénticos a los expresados para el canal de ida en la petición fracasada del terminal distante, y con parámetros de canal de ida que este terminal puede soportar y que se sabe que el terminal distante podrá soportar.

La conmutación de modo deberá efectuarse cerrando y abriendo canales lógicos existentes o abriendo nuevos canales lógicos.

NOTA – Algunas Recomendaciones que utilizan esta Recomendación pueden definir algunos canales lógicos supletorios. Estos canales se considerarán en el estado ESTABLECIDO desde el principio de la comunicación y no se abrirán mediante estos procedimientos. Sin embargo, podrán ser cerrados por estos procedimientos y abiertos de nuevo, ulteriormente, con los mismos fines o con fines diferentes.

Un canal que deje de ser capaz de procesar las señales en un canal lógico debe ejecutar acciones apropiadas: entre ellas está la de cerrar el canal lógico y transmitir la información de capacidad apropiada (modificada) al terminal distante.

El texto siguiente proporciona una visión general del funcionamiento del protocolo de la B-LCSE. En caso de discrepancia entre esta exposición y la especificación formal, prevalecerá la especificación formal.

### **C.5.1.1 Visión general del protocolo**

Se inicia la apertura de un canal lógico cuando el usuario en la B-LCSE de salida emite una primitiva ESTABLECIMIENTO.petición. Se envía a la B-LCSE de entrada par un mensaje OpenLogicalChannel que contiene los parámetros del canal de ida y del canal inverso, y se arranca el temporizador T103. Si se recibe un mensaje OpenLogicalChannelAck en respuesta al mensaje OpenLogicalChannel, se detiene el temporizador T103, se envía un mensaje OpenLogicalChannelConfirm a la B-LCSE de entrada par y se informa al usuario con la primitiva ESTABLECIMIENTO.confirmación que el canal lógico ha sido abierto correctamente. El canal lógico puede utilizarse ahora para transmitir y recibir información de usuario. Si, en cambio, se recibe un mensaje OpenLogicalChannelReject en respuesta al mensaje OpenLogicalChannel, se detiene el temporizador T103 y se informa al usuario con la primitiva LIBERACIÓN.indicación que el usuario de la B-LCSE par ha rehusado el establecimiento del canal lógico.

Si el temporizador T103 expira en este periodo, se informa de esto al usuario con la primitiva LIBERACIÓN.indicación y se envía un mensaje CloseLogicalChannel a la B-LCSE de entrada par.

Un canal lógico que ha sido correctamente establecido se puede cerrar cuando el usuario en la B-LCSE de salida emite una primitiva LIBERACIÓN.petición. Se envía un mensaje CloseLogicalChannel a la B-LCSE de entrada par y se arranca el temporizador T103. Cuando se recibe un mensaje CloseLogicalChannelAck, se detiene el temporizador T103 y se informa al usuario, mediante la primitiva LIBERACIÓN.confirmación, que el canal lógico ha sido correctamente cerrado.

Si el temporizador T103 expira en este periodo, se informa de esto al usuario con la primitiva LIBERACIÓN.indicación.

Antes de recibirse el mensaje OpenLogicalChannelAck o el mensaje OpenLogicalChannelReject en respuesta al mensaje OpenLogicalChannel anteriormente enviado, el usuario en la B-LCSE de salida puede cerrar el canal lógico emitiendo la primitiva LIBERACIÓN.petición.

Antes de recibirse el mensaje CloseLogicalChannelAck en respuesta al mensaje CloseLogicalChannel enviado anteriormente, el usuario en la B-LCSE de salida puede establecer un nuevo canal lógico emitiendo la primitiva ESTABLECIMIENTO.petición.

### **C.5.1.2 Visión general del protocolo – En la B-LCSE de entrada**

Cuando se recibe un mensaje `OpenLogicalChannel` en la B-LCSE de entrada, se informa al usuario de la petición de apertura de un nuevo canal lógico con la primitiva `ESTABLECIMIENTO.indicación`. El usuario de la B-LCSE de entrada señala la aceptación de la petición de establecimiento de canal lógico emitiendo la primitiva `ESTABLECIMIENTO.respuesta` y se envía el mensaje `OpenLogicalChannelAck` a la B-LCSE de salida par. El canal de ida del canal lógico bidireccional puede utilizarse ahora para recibir información de usuario. El usuario de la B-LCSE de llegada señala el rechazo de la petición de establecimiento de canal lógico emitiendo la primitiva `LIBERACIÓN.petición` y se envía el mensaje `OpenLogicalChannelReject` a la B-LCSE de salida par.

Cuando se recibe un mensaje `OpenLogicalChannelConfirm` en la B-LCSE de entrada, se informa al usuario que el canal lógico bidireccional está establecido mediante la primitiva `ESTABLECIMIENTO.confirmación`. El canal inverso del canal lógico bidireccional puede utilizarse ahora para transmitir información de usuario.

Un canal lógico que haya sido establecido correctamente puede cerrarse cuando se recibe el mensaje `CloseLogicalChannel` en la B-LCSE de entrada. Se informa de esto al usuario de la B-LCSE de entrada mediante la primitiva `LIBERACIÓN.indicación` y se envía el mensaje `CloseLogicalChannelAck` a la B-LCSE de salida par.

### **C.5.1.3 Resolución de conflictos**

Pueden presentarse conflictos cuando se envían al mismo tiempo peticiones de apertura de canal lógico. Es posible determinar, a partir del conocimiento de las capacidades intercambiadas, que se ha presentado un conflicto. En otras ocasiones, ambos terminales pueden iniciar la apertura de un canal lógico bidireccional con la misma finalidad, aunque los parámetros solicitados sean diferentes, y ambos terminales pueden tener capacidad suficiente para atender ambas peticiones. Los terminales deberán poder detectar cuándo se han presentado estas dos situaciones, en cuyo caso cualquiera de los dos actuará como sigue.

Antes de que se puedan abrir canales lógicos, es preciso determinar que uno de los dos terminales es el terminal principal y el otro el subordinado. El protocolo definido en C.2 proporciona un medio para decidir esta situación. El terminal principal rechazará inmediatamente toda petición del terminal subordinado que identifique como conflictiva. El terminal subordinado podrá identificar esos conflictos, pero responderá a la petición del terminal principal, aunque sepa que su anterior petición será rechazada.

En el segundo tipo de conflicto antes mencionado es imposible distinguir el caso en que realmente se desean dos canales bidireccionales del caso en que se desea uno solo. Los terminales deberán responder suponiendo que se desea un solo canal, pero un terminal puede ulteriormente repetir su petición si tal suposición era incorrecta.

Se recomiendan los siguientes procedimientos para reducir al mínimo la posibilidad de que los puntos extremos intenten abrir canales lógicos en conflicto cuando el punto extremo subordinado tiene limitaciones de capacidades simétricas. Cuando el terminal director y el terminal subordinado han indicado la posibilidad de recibir capacidades para un determinado tipo de medios, el terminal subordinado debería tratar de abrir un canal lógico destinado a la capacidad preferencial del terminal director para la que tiene capacidad, según el orden en que el terminal director ha expresado sus capacidades; y el terminal director debería tratar de abrir un canal lógico destinado a su capacidad preferencial para la que el subordinado tiene capacidad, según el orden en que ha expresado sus capacidades.

Por ejemplo, si el terminal director ha declarado la capacidad de seguir los procedimientos G.723.1, G.729 y G.711 y el subordinado la capacidad de seguir los procedimientos G.711 y G.729, con las

capacidades más preferenciales enumeradas primero en ambos casos, tanto el terminal director como el subordinado deberían tratar de abrir canales lógicos para los procedimientos G.729.

Una vez que el terminal director rechaza la petición de abrir un canal lógico, con causa igual a `masterSlaveConflict`, el terminal subordinado es responsable de abrir un canal que no presente conflictos.

Cuando el terminal subordinado detecta un conflicto y el terminal director no rechaza un canal lógico abierto en conflicto, el subordinado debería cerrar el canal en conflicto. Si hay canales lógicos en conflicto debido a limitaciones de capacidades simétricas, el terminal subordinado debería abrir un canal lógico adecuado utilizando el procedimiento sustitutivo y, a su debido tiempo, cerrar el canal lógico en conflicto.

## C.5.2 Comunicación entre la B-LCSE y el usuario de B-LCSE

### C.5.2.1 Primitivas entre la B-LCSE y el usuario de la B-LCSE

La comunicación entre la B-LCSE y el usuario de la B-LCSE se realiza utilizando las primitivas indicadas en el cuadro C.15.

**Cuadro C.15/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
ESTABLECIMIENTO	FORWARD_PARAM REVERSE_PARAM	FORWARD_PARAM REVERSE_PARAM	REVERSE_DATA	REVERSE_DATA
LIBERACIÓN	CAUSE	SOURCE CAUSE	no definida (nota 2)	– (nota 1)
ERROR	no definida	ERRCODE	no definida	no definida
NOTA 1 – "-" indica que no hay parámetros.				
NOTA 2 – "no definida" significa que esta primitiva no existe.				

### C.5.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- Se utilizan las primitivas ESTABLECIMIENTO para establecer un canal lógico para comunicaciones audiovisuales y de datos.
- Se utilizan las primitivas LIBERACIÓN para liberar un canal lógico.
- Se utilizan las primitivas ERROR para informar errores B-LCSE a una entidad de gestión.

### C.5.2.3 Definición de los parámetros

La definición de los parámetros de las primitivas del cuadro C.15 es la siguiente:

- El parámetro FORWARD\_PARAM especifica los parámetros asociados con el canal de ida, es decir, del terminal que contiene la B-LCSE de salida al terminal que contiene la B-LCSE de entrada. Este parámetro se hace corresponder con el campo `forwardLogicalChannelParameters` del mensaje `OpenLogicalChannel` y se transporta transparentemente al usuario de la LCSE par.
- El parámetro REVERSE\_PARAM especifica los parámetros asociados con el canal inverso, es decir, del terminal que contiene la B-LCSE de entrada al terminal que contiene la B-LCSE de salida. Este parámetro se hace corresponder con el campo `reverseLogicalChannelParameters` del mensaje `OpenLogicalChannel` y se transporta transparentemente al usuario de la LCSE par.

- c) El parámetro REVERSE\_DATA especifica los datos asociados con el canal inverso, es decir, del terminal que contiene la B-LCSE de entrada al terminal que contiene la B-LCSE de salida. Este parámetro se hace corresponder con el campo reverseLogicalChannelParameters del mensaje OpenLogicalChannelAck y se transporta transparentemente al usuario de la B-LCSE par.
- d) El parámetro SOURCE indica al usuario de la B-LCSE la fuente de la liberación del canal lógico. Este parámetro toma el valor "USER" o "B-LCSE", indicando el usuario de la B-LCSE o la B-LCSE. Este último puede producirse como consecuencia de un error de protocolo.
- e) El parámetro CAUSE indica el motivo por el cual el usuario de la B-LCSE rechazó una petición de establecimiento de un canal lógico. El parámetro CAUSE únicamente está presente cuando el parámetro SOURCE indica "B-LCSE".
- f) El parámetro ERRCODE indica el tipo de error B-LCSE. En el cuadro C.19 se indican los valores del parámetro ERRCODE.

#### **C.5.2.4 Estados de la B-LCSE**

Los siguientes estados se utilizan para especificar la secuencia permitida de primitivas entre la B-LCSE y el usuario de la B-LCSE, y el intercambio de mensajes entre las B-LCSE. Los estados se especifican separadamente para la B-LCSE de salida y la B-LCSE de entrada. Los estados para una B-LCSE de salida son:

Estado 0: LIBERADO

El canal lógico está liberado. No se utilizará para enviar ni recibir datos.

Estado 1: ESPERA DE ESTABLECIMIENTO

La B-LCSE de salida está esperando el establecimiento de un canal lógico con una B-LCSE de entrada par. El canal lógico no se utilizará para enviar ni recibir datos.

Estado 2: ESTABLECIDO

Se ha establecido la conexión de canal lógico entre las B-LCSE pares. El canal lógico puede utilizarse para enviar y recibir datos.

Estado 3: ESPERA DE LIBERACIÓN

La B-LCSE de salida está esperando la liberación de un canal lógico con la B-LCSE de entrada par. El canal lógico no se utilizará para enviar datos, pero puede continuar utilizándose para recibir datos.

Los estados para una B-LCSE de entrada son:

Estado 0: LIBERADO

El canal lógico está liberado. No se utilizará para enviar ni recibir datos.

Estado 1: ESPERA DE ESTABLECIMIENTO

La B-LCSE de entrada está esperando el establecimiento de un canal lógico con una B-LCSE de salida par. El canal lógico no se utilizará para enviar ni recibir datos.

Estado 2: ESPERA DE CONFIRMACIÓN

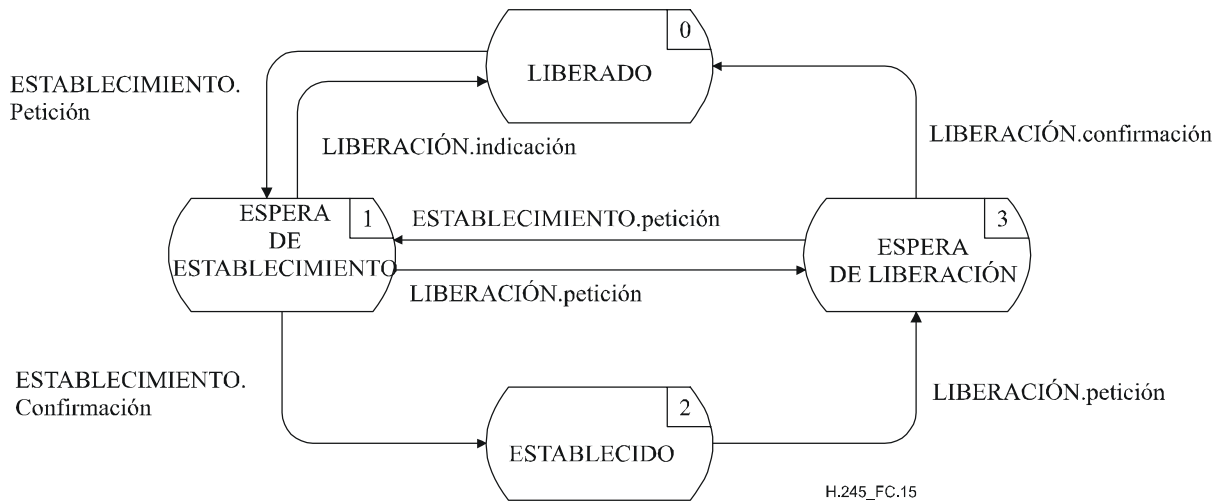
La B-LCSE de entrada está esperando la confirmación de que se ha establecido el canal lógico con la B-LCSE de salida par. El canal lógico no se utilizará para enviar datos, pero puede utilizarse para recibir datos.

Estado 3: ESTABLECIDO

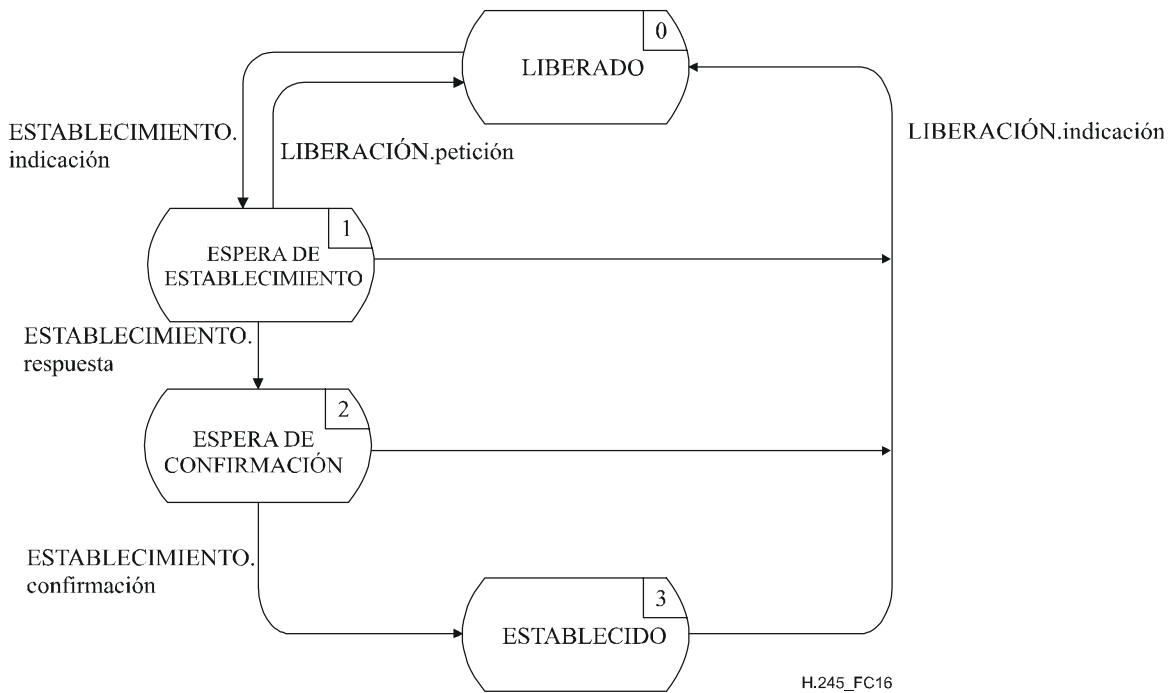
Se ha establecido la conexión de canal lógico entre las B-LCSE pares. El canal lógico puede utilizarse para recibir y enviar datos.

**C.5.2.5 Diagramas de transición de estados**

Se define aquí la secuencia permitida de primitivas entre la B-LCSE y el usuario de la B-LCSE. La secuencia permitida de primitivas se refiere a los estados de la B-LCSE tal como son percibidos por el usuario de la B-LCSE. Las secuencias permitidas se especifican separadamente para la B-LCSE de salida y la B-LCSE de entrada, como se muestra en las figuras C.15 y C.16.



**Figura C.15/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una B-LCSE de salida**



**Figura C.16/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una B-LCSE de entrada**

### C.5.3 Comunicaciones entre B-LCSE pares

#### C.5.3.1 Mensajes de la B-LCSE

El cuadro C.16 muestra los mensajes y campos de la B-LCSE, definidos en el anexo A, que son apropiados para el protocolo de la B-LCSE.

**Cuadro C.16/H.245 – Nombres de mensajes y campos de la B-LCSE**

Función	Mensaje	Sentido	Campo
establecimiento	OpenLogicalChannel	O → I (nota)	forwardLogicalChannelNumber forwardLogicalChannelParameters reverseLogicalChannelParameters
	OpenLogicalChannelAck	O ← I	forwardLogicalChannelNumber reverseLogicalChannelParameters
	OpenLogicalChannelReject	O ← I	forwardLogicalChannelNumber cause
	OpenLogicalChannelConfirm	O → I	forwardLogicalChannelNumber
liberación	CloseLogicalChannel	O → I	forwardLogicalChannelNumber source
	CloseLogicalChannelAck	O ← I	forwardLogicalChannelNumber

NOTA – Sentido: O – salida; I – entrada.

#### C.5.3.2 Variables de estado de la B-LCSE

Se define la siguiente variable de estado en la B-LCSE de salida:

out\_LCN

Esta variable de estado distingue entre las B-LCSE de salida. Se inicializa en el momento de la inicialización de la B-LCSE de salida. El valor de out\_LCN se utiliza para fijar el campo forwardLogicalChannelNumber de los mensajes B-LCSE enviados desde una B-LCSE de salida. En el caso de mensajes B-LCSE recibidos en una B-LCSE de salida, el valor del campo forwardLogicalChannelNumber es idéntico al valor de out\_LCN.

Se define la siguiente variable de estado en la B-LCSE de entrada:

in\_LCN

Esta variable de estado distingue entre las B-LCSE de entrada. Se inicializa en el momento de la inicialización de la B-LCSE de entrada. El valor de in\_LCN se utiliza para fijar el campo forwardLogicalChannelNumber de los mensajes B-LCSE enviados desde una B-LCSE de entrada. En el caso de mensajes B-LCSE recibidos en una B-LCSE de entrada, el valor del campo forwardLogicalChannelNumber es idéntico al valor de in\_LCN.

#### C.5.3.3 Temporizadores de la B-LCSE

Se especifica el siguiente temporizador para la B-LCSE de salida y la de entrada:

T103

En la B-LCSE de salida se utiliza este temporizador durante los estados ESPERA DE ESTABLECIMIENTO y ESPERA DE LIBERACIÓN. Especifica el máximo periodo de tiempo durante el cual no se pueden recibir los mensajes OpenLogicalChannelAck, OpenLogicalChannelReject o CloseLogicalChannelAck.

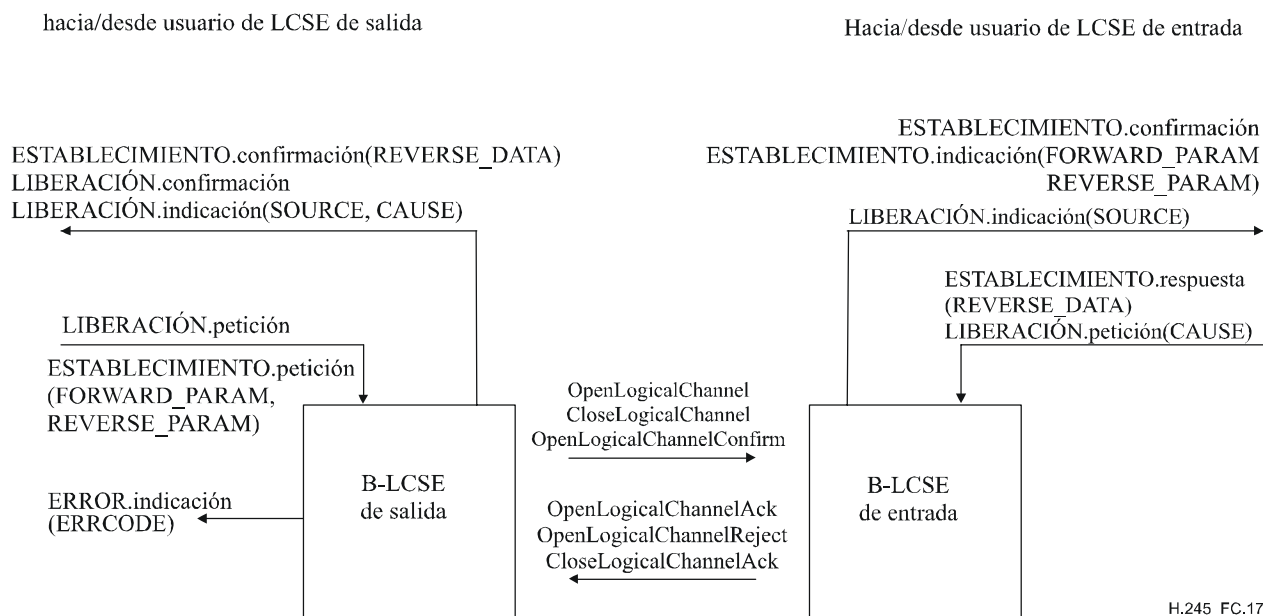


En la B-LCSE de entrada se utiliza este temporizador durante el estado CONFIRMACIÓN DE ESTABLECIMIENTO. Especifica el máximo periodo de tiempo durante el cual no se puede recibir el mensaje OpenLogicalChannelConfirm.

#### C.5.4 Procedimientos de B-LCSE

##### C.5.4.1 Introducción

La figura C.17 resume las primitivas de la B-LCSE y sus parámetros, así como los mensajes, para la B-LCSE de salida y la de entrada.



H.245\_FC.17

**Figura C.17/H.245 – Primitivas y mensajes en la entidad de señalización de canal lógico bidireccional**

##### C.5.4.2 Valores supletorios de los parámetros de las primitivas

Donde no se indiquen explícitamente en los diagramas SDL, los parámetros de las primitivas indicación y confirmación tomarán los valores del cuadro C.17.

**Cuadro C.17/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio (nota)
ESTABLECIMIENTO.indicación	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
	REVERSE_PARAM	OpenLogicalChannel.reverseLogicalChannelParameters
ESTABLECIMIENTO.confiración	REVERSE_DATA	OpenLogicalChannelAck.reverseLogicalChannelParameters
LIBERACIÓN.indicación	SOURCE	CloseLogicalChannel.source
	CAUSE	null

NOTA – Un parámetro de primitiva se codificará como nulo si un determinado campo de mensaje no está presente en el mensaje.

### C.5.4.3 Valores supletorios de los campos de los mensajes

Donde no se indiquen explícitamente en los diagramas SDL, los campos de los mensajes tomarán los valores del cuadro C.18.

**Cuadro C.18/H.245 – Valores supletorios de los campos de los mensajes**

Mensaje	Campo	Valor supletorio (nota)
OpenLogicalChannel	forwardLogicalChannelNumber	out_LCN
	forwardLogicalChannelParameters	ESTABLECIMIENTO.petición (FORWARD_PARAM)
	reverseLogicalChannelParameters	ESTABLECIMIENTO.petición (REVERSE_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
	reverseLogicalChannelParameters	ESTABLECIMIENTO.respuesta (REVERSE_DATA)
OpenLogicalChannelReject	forwardLogicalChannelNumber cause	in_LCN LIBERACIÓN.petición(CAUSE)
OpenLogicalChannelConfirm	forwardLogicalChannelNumber	out_LCN
CloseLogicalChannel	forwardLogicalChannelNumber	out_LCN
	source	user
CloseLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
NOTA – Un campo de mensaje no se codificará si el correspondiente parámetro de primitiva es nulo, es decir, no está presente.		

### C.5.4.4 Valores del parámetro ERRCODE

El parámetro ERRCODE de la primitiva ERROR.indicación indica una determinada condición de error. El cuadro C.19 muestra los valores que puede tomar el parámetro ERRCODE en la B-LCSE de salida y el cuadro C.20 muestra los valores que puede tomar el parámetro ERRCODE en la B-LCSE de entrada.

**Cuadro C.19/H.245 – Valores del parámetro ERRCODE en la B-LCSE de salida**

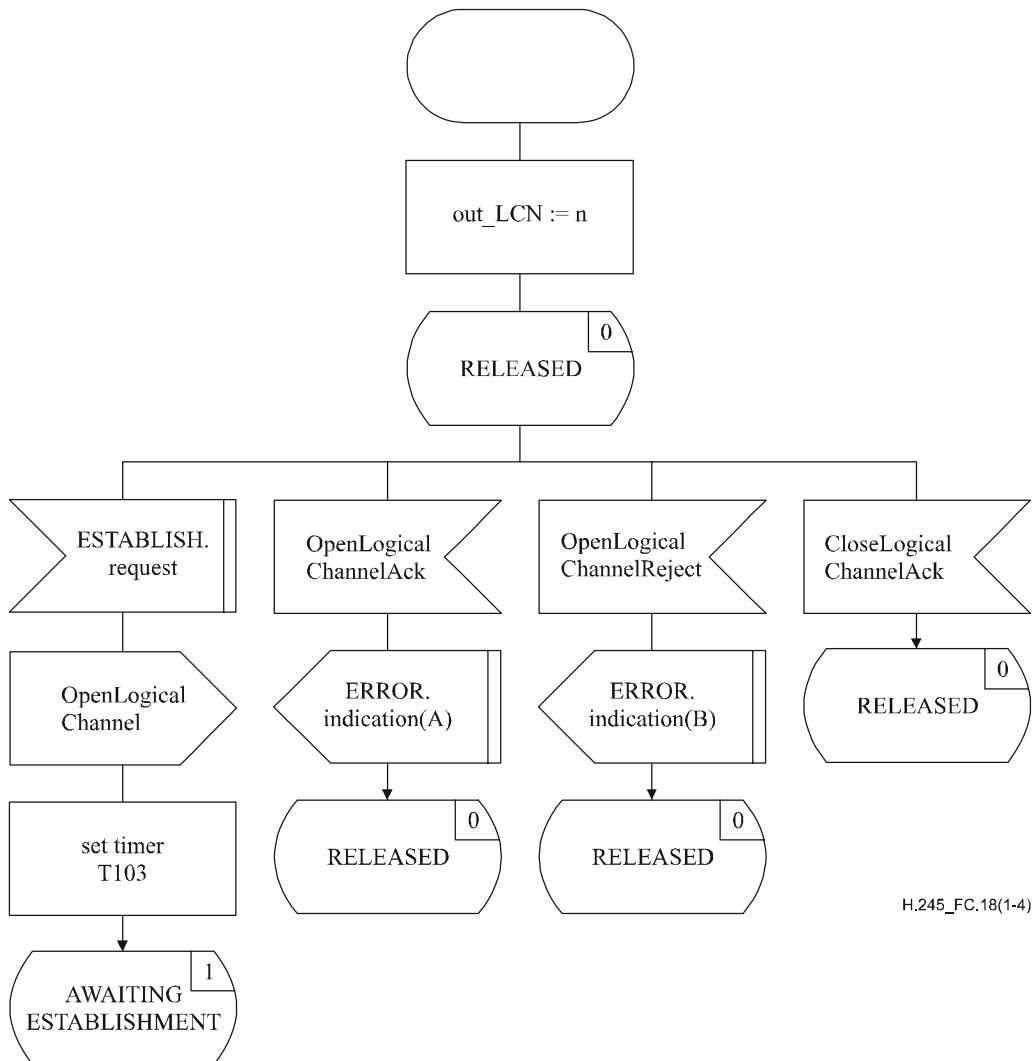
Tipo de error	Código de error	Condición de error	Estado
mensaje no apropiado	A	OpenLogicalChannelAck	LIBERADO
	B	OpenLogicalChannelReject	LIBERADO ESTABLECIDO
	C	CloseLogicalChannelAck	ESTABLECIDO
no hay respuesta de la B-LCSE par	D	expiración del temporizador T103	ESPERA DE ESTABLECIMIENTO ESPERA DE LIBERACIÓN

**Cuadro C.20/H.245 – Valores del parámetro ERRCODE en la B-LCSE de entrada**

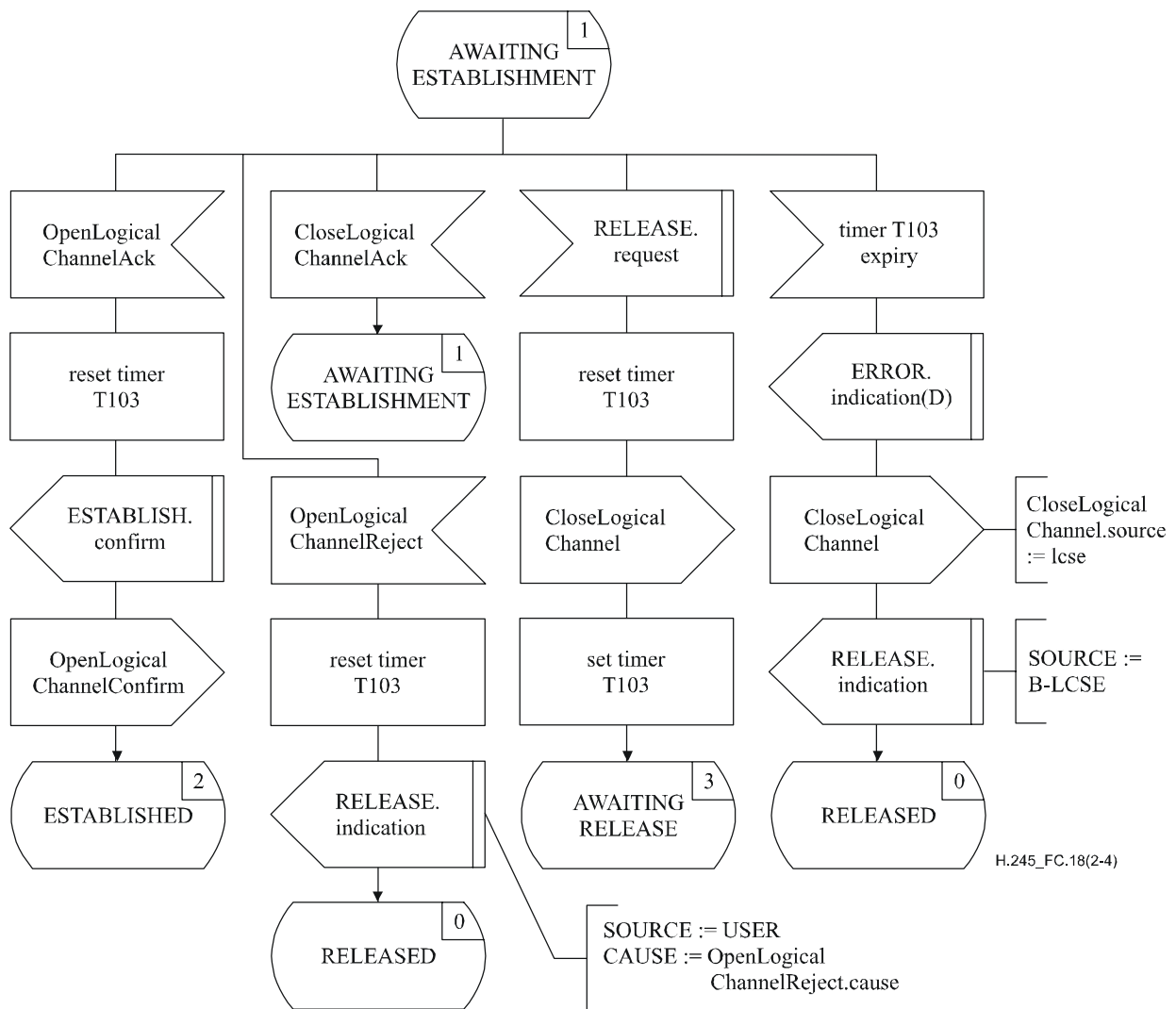
Tipo de error	Código de error	Condición de error	Estado
mensaje no apropiado	E	OpenLogicalChannelConfirm	ESPERA DE ESTABLECIMIENTO
no hay respuesta de la B-LCSE par	F	expiración del temporizador T103	ESPERA DE CONFIRMACIÓN

### C.5.4.5 Diagramas SDL

Los procedimientos de la B-LCSE de salida y de la B-LCSE de entrada se expresan en forma de diagramas SDL en las figuras C.18 y C.19, respectivamente.



**Figura C.18/H.245 – Diagrama SDL de la B-LCSE de salida (hoja 1 de 4)**



**Figura C.18/H.245 – Diagrama SDL de la B-LCSE de salida (hoja 2 de 4)**

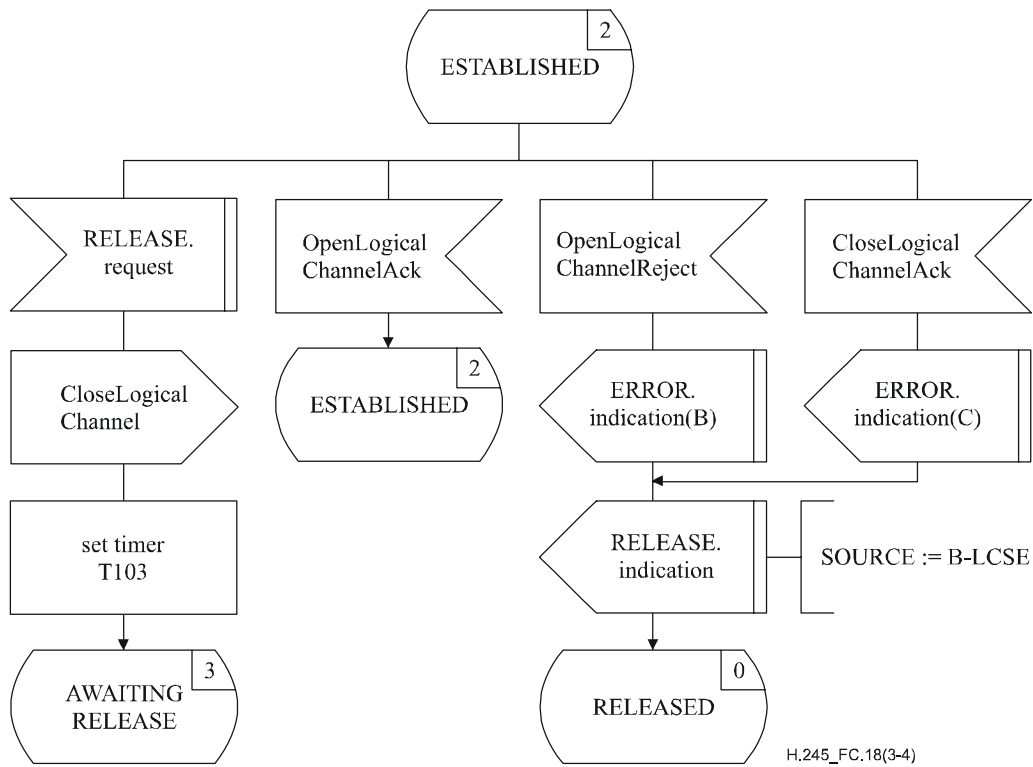


Figura C.18/H.245 – Diagrama SDL de la B-LCSE de salida (hoja 3 de 4)

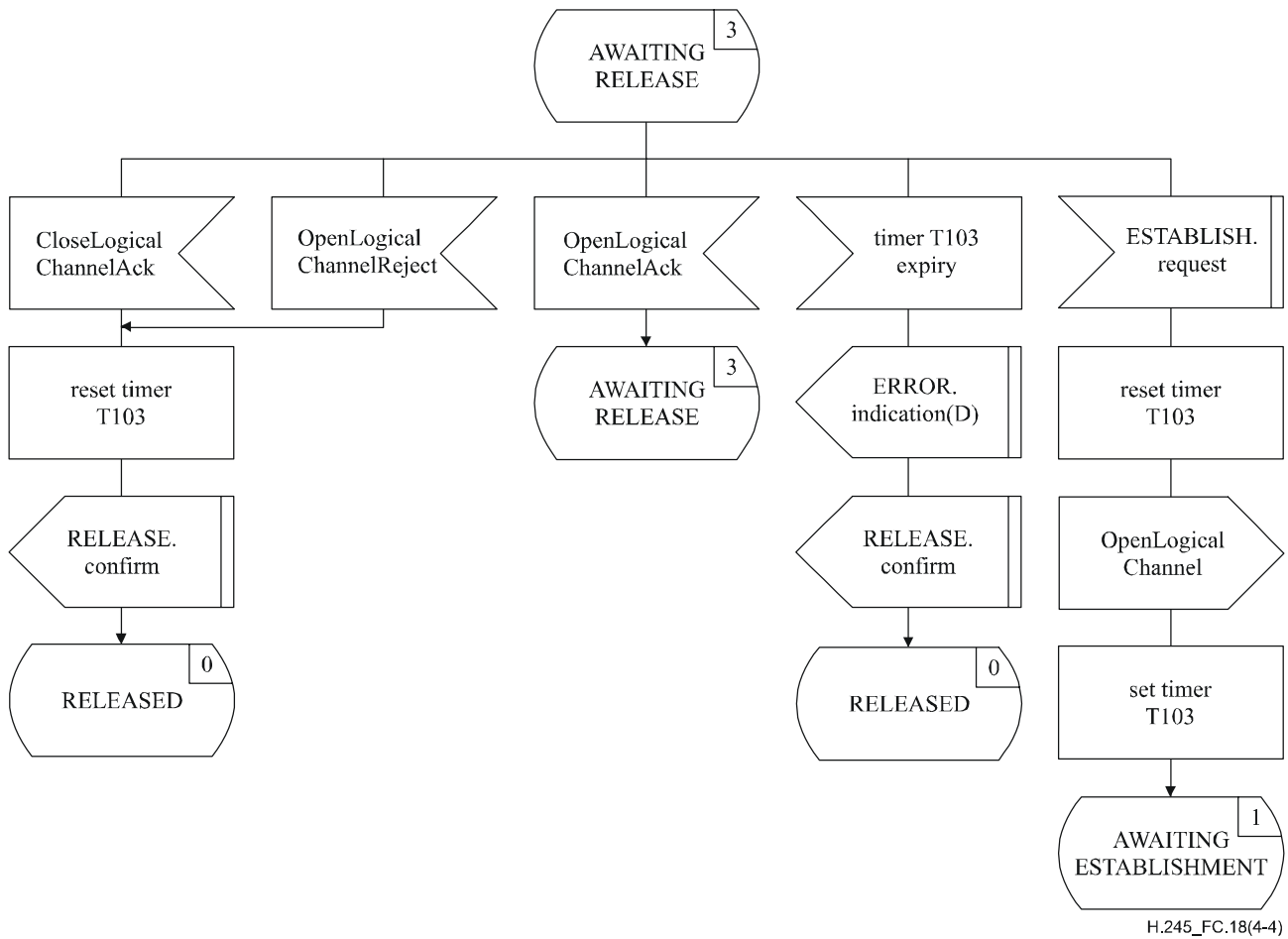
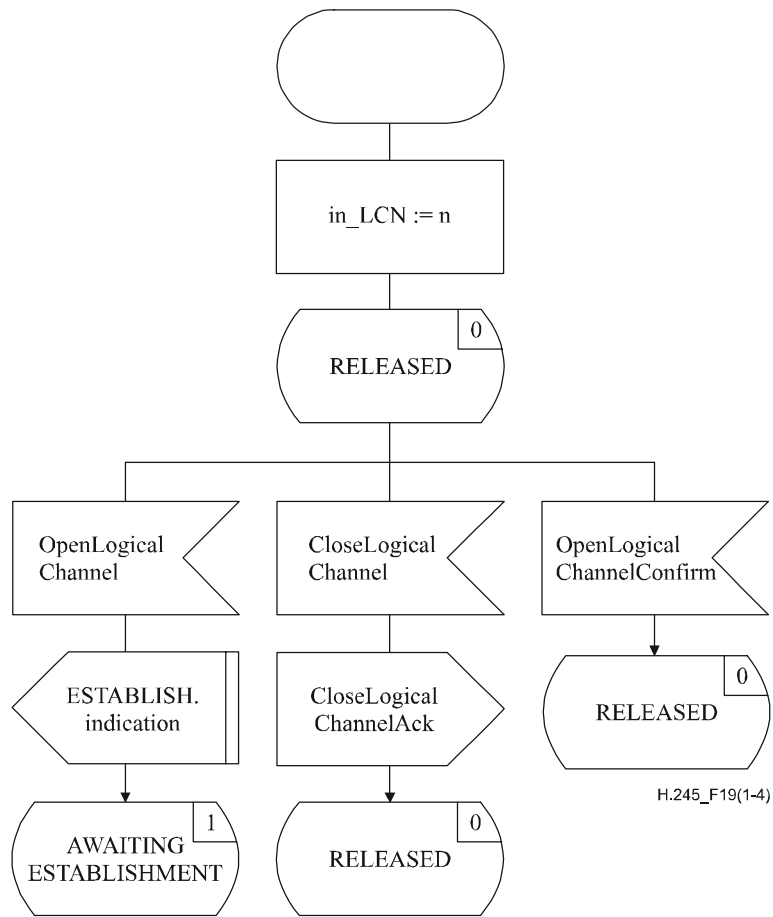
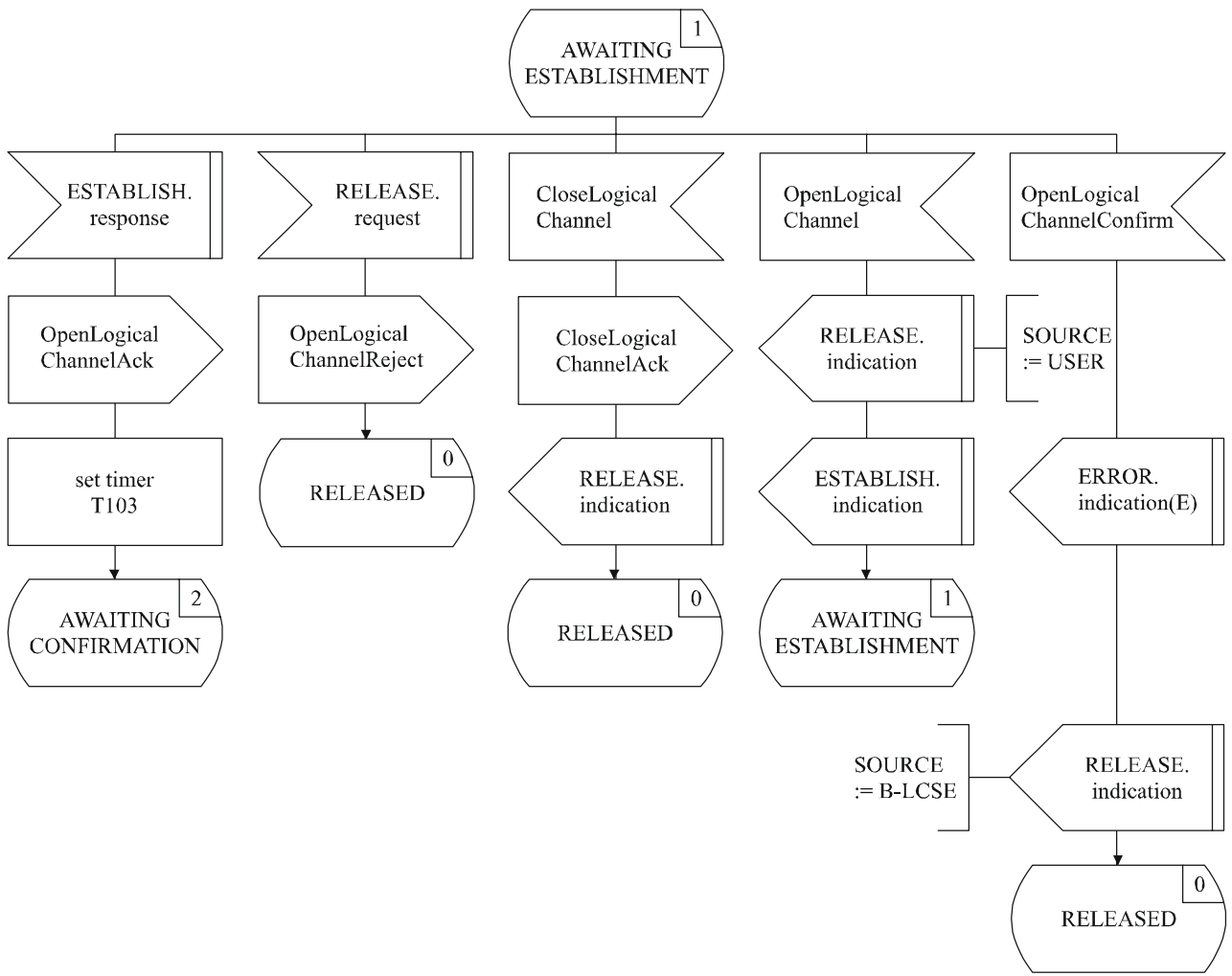


Figura C.18/H.245 – Diagrama SDL de la B-LCSE de salida (hoja 4 de 4)

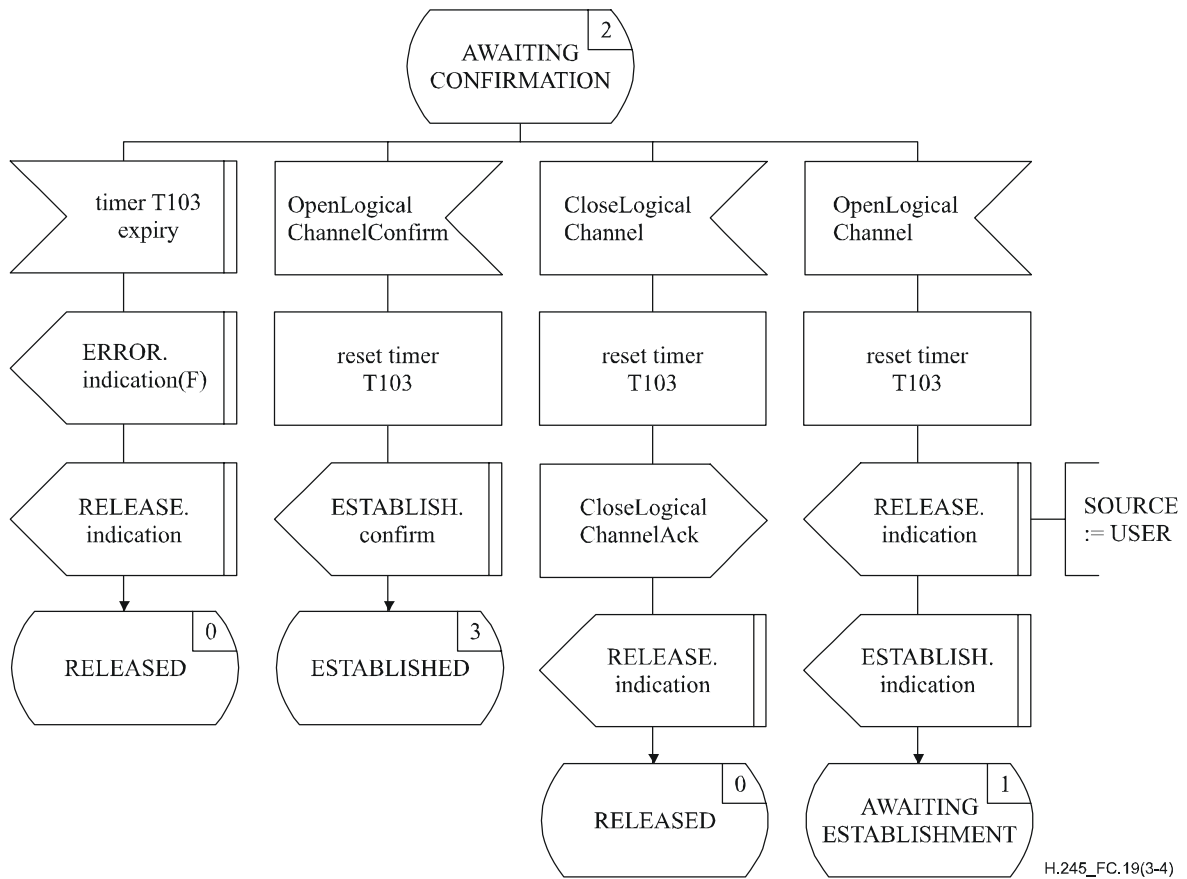


**Figura C.19/H.245 – Diagrama SDL de la B-LCSE de entrada (hoja 1 de 4)**

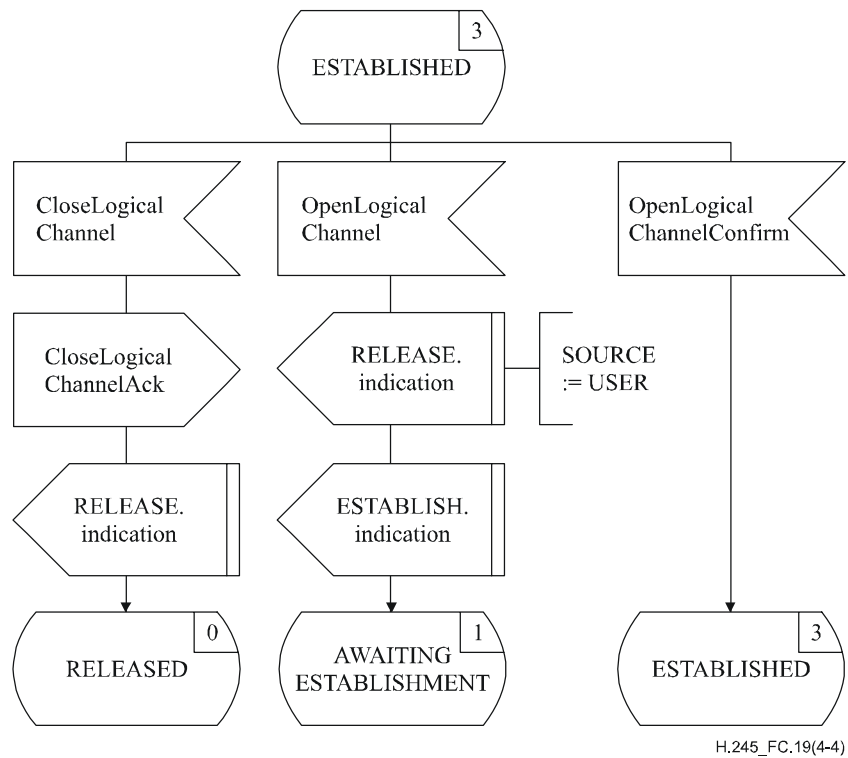


H.245\_FC.19(2-4)

**Figura C.19/H.245 – Diagrama SDL de la B-LCSE de entrada (hoja 2 de 4)**



**Figura C.19/H.245 – Diagrama SDL de la B-LCSE de entrada (hoja 3 de 4)**



**Figura C.19/H.245 – Diagrama SDL de la B-LCSE de entrada (hoja 4 de 4)**



## **C.6 Procedimientos de cierre de canal lógico**

### **C.6.1 Introducción**

Estos procedimientos los utiliza un terminal para pedir al terminal distante que cierre un canal lógico. Obsérvese que estos procedimientos son únicamente de petición de cierre; el cierre efectivo del canal lógico se obtiene aplicando los procedimientos de la LCSE y de la B-LCSE. Estos procedimientos se denominan aquí entidad de señalización de cierre de canal lógico (CLCSE, *close logical channel signalling entity*). Los procedimientos se especifican mediante primitivas y estados en las interfaces entre la CLCSE y el usuario de la CLCSE. La información de protocolo se transfiere a la CLCSE por medio de mensajes apropiados definidos en el anexo A. Hay una CLCSE de salida y una CLCSE de entrada. En cada uno de los extremos de salida y de entrada hay un ejemplar de CLCSE para cada canal lógico.

Si un terminal es incapaz de procesar las señales de entrada, puede utilizar estos procedimientos para pedir el cierre de los canales lógicos pertinentes.

Un terminal que contesta a esta respuesta positivamente, es decir, emitiendo la primitiva CIERRE.respuesta, iniciará lo antes posible el cierre del canal lógico enviando la primitiva LIBERACIÓN.petición a la LCSE o B-LCSE apropiadas.

El siguiente texto proporciona una visión general del funcionamiento del protocolo. En caso de discrepancia entre esta exposición y la especificación formal del protocolo que sigue, prevalecerá la especificación formal.

#### **C.6.1.1 Visión general del protocolo – CLCSE de salida**

Se inicia un procedimiento de petición de cierre de canal lógico cuando la primitiva CIERRE.petición es emitida por el usuario en la CLCSE de salida. Se envía un mensaje RequestChannelClose a la CLCSE de entrada por y se arranca el temporizador T108. Si se recibe un mensaje RequestChannelCloseAck en respuesta al mensaje RequestChannelClose, se detiene el temporizador T108 y el usuario es informado con una primitiva CIERRE.confirmación de que el procedimiento de petición de cierre de canal lógico tuvo éxito. Sin embargo, si se recibe un mensaje RequestChannelCloseReject en respuesta al mensaje RequestChannelClose, se detiene el temporizador T108 y se informa al usuario con la primitiva RECHAZO.indicación de que el usuario de la CLCSE par ha rehusado cerrar el canal lógico.

Si expira el temporizador T108, el usuario CLCSE de salida es informado con la primitiva RECHAZO.indicación y se envía un mensaje RequestChannelCloseRelease.

#### **C.6.1.2 Visión general del protocolo – CLCSE de entrada**

Cuando se recibe un mensaje RequestChannelClose en la CLCSE de entrada, el usuario es informado de la petición de cierre de canal lógico con la primitiva CIERRE.indicación. El usuario de la CLCSE de entrada señala la aceptación de la petición de cierre de canal lógico emitiendo la primitiva CIERRE.respuesta, y se envía un mensaje RequestChannelCloseAck a la CLCSE de salida par. El usuario de la CLCSE de entrada señala el rechazo de la petición de cierre de canal lógico emitiendo la primitiva RECHAZO.petición y se envía un mensaje RequestChannelCloseReject a la CLCSE de salida par.

### **C.6.2 Comunicación entre la CLCSE y el usuario de CLCSE**

#### **C.6.2.1 Primitivas entre la CLCSE y el usuario de CLCSE**

La comunicación entre la CLCSE y el usuario de la CLCSE se realiza mediante las primitivas del cuadro C.21.

**Cuadro C.21/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
CIERRE	– (nota 1)	–	–	–
RECHAZO	CAUSE	SOURCE CAUSE	no definida (nota 2)	no definida
NOTA 1 – "–" significa que no hay parámetros.				
NOTA 2 – "no definida" significa que no se ha definido esta primitiva.				

**C.6.2.2 Definición de las primitivas**

La definición de estas primitivas es la siguiente:

- a) Se utilizan las primitivas CIERRE para la petición de cierre de un canal lógico.
- b) Se utilizan las primitivas RECHAZO para el rechazo del cierre de un canal lógico.

**C.6.2.3 Definición de los parámetros**

La definición de los parámetros de las primitivas del cuadro C.21 es la siguiente:

- a) El parámetro SOURCE indica el origen de la primitiva RECHAZO.indicación. El parámetro SOURCE toma los valores "USER" o "PROTOCOL". Este último puede producirse como consecuencia de la expiración de un temporizador.
- b) El parámetro CAUSE indica el motivo del rechazo del cierre de un canal lógico. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "PROTOCOL".

**C.6.2.4 Estados de la CLCSE**

Para especificar la secuencia permitida de primitivas entre la CLCSE y el usuario de la CLCSE se emplean los siguientes estados:

Los estados para una CLCSE de salida son:

Estado 0: REPOSO

La CLCSE está en reposo.

Estado 1: ESPERA DE RESPUESTA

La CLCSE se encuentra a la espera de una respuesta de la CLCSE distante.

Los estados para una CLCSE de entrada son:

Estado 0: REPOSO

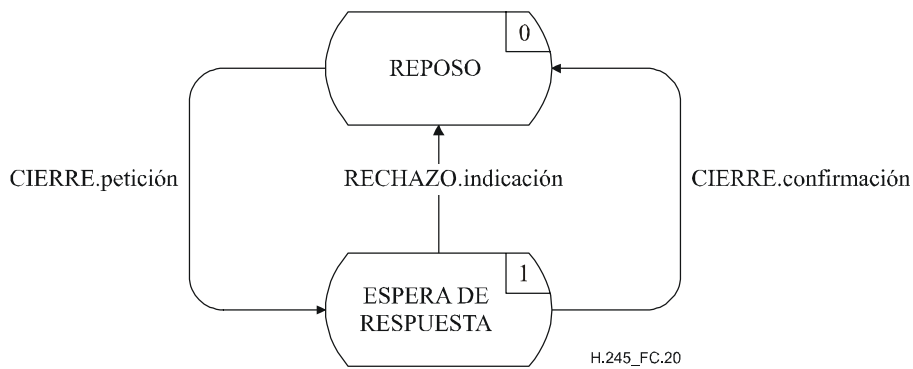
La CLCSE está en reposo.

Estado 1: ESPERA DE RESPUESTA

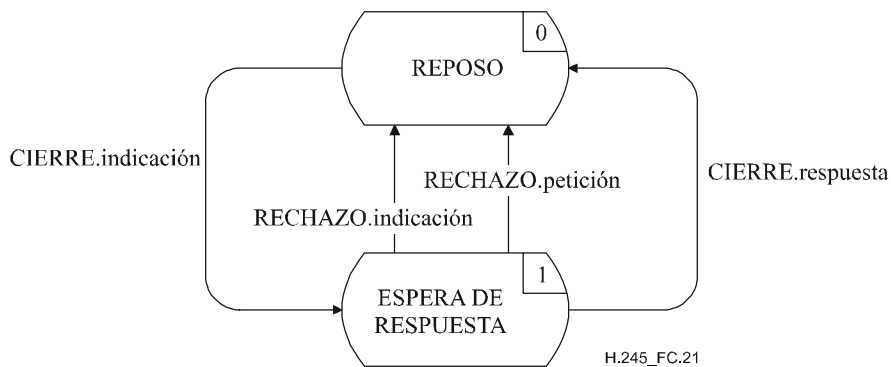
La CLCSE se encuentra a la espera de una respuesta del usuario de la CLCSE.

**C.6.2.5 Diagramas de transición de estados**

Se define aquí la secuencia permitida de primitivas entre la CLCSE y el usuario de la CLCSE. Se especifican por separado las secuencias permitidas para una CLCSE de salida y una CLCSE de entrada, como se muestra en las figuras C.20 y C.21, respectivamente.



**Figura C.20/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CLCSE de salida**



**Figura C.21/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CLCSE de entrada**

### C.6.3 Comunicación entre CLCSE pares

#### C.6.3.1 Mensajes

El cuadro C.22 muestra los mensajes y campos de la CLCSE, definidos en el anexo A, que son apropiados para el protocolo de la CLCSE.

**Cuadro C.22/H.245 – Nombres de mensajes y campos de la CLCSE**

Función	Mensaje	Sentido	Campo
transferencia	RequestChannelClose	O → I (nota)	forwardLogicalChannelNumber
	RequestChannelCloseAck	O ← I	forwardLogicalChannelNumber
	RequestChannelCloseReject	O ← I	forwardLogicalChannelNumber
reiniciación	RequestChannelCloseRelease	O → I	forwardLogicalChannelNumber

NOTA – Sentido: O – de salida; I – de entrada.

#### C.6.3.2 Variables de estado de la CLCSE

Se define la siguiente variable de estado en la CLCSE de salida:

out\_LCN

Esta variable de estado distingue entre las CLCSE de salida. Se inicializa en el momento de la inicialización de la CLCSE de salida. El valor de out\_LCN se utiliza para fijar el campo forwardLogicalChannelNumber de los mensajes CLCSE enviados desde una CLCSE de salida. En

el caso de mensajes CLCSE recibidos en una CLCSE de salida, el valor del campo forwardLogicalChannelNumber del mensaje es idéntico al valor de out\_LCN.

Se define la siguiente variable de estado en la CLCSE de entrada:

in\_LCN

Esta variable de estado distingue entre las CLCSE de entrada. Se inicializa en el momento de la inicialización de la CLCSE de entrada. El valor de in\_LCN se utiliza para fijar el campo forwardLogicalChannelNumber de los mensajes CLCSE enviados desde una CLCSE de entrada. En el caso de mensajes CLCSE recibidos en una CLCSE de entrada, el valor del campo forwardLogicalChannelNumber del mensaje es idéntico al valor de in\_LCN.

### C.6.3.3 Temporizadores de la CLCSE

Para la CLCSE de salida se especifica el siguiente temporizador:

T108

Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el intervalo de tiempo que puede transcurrir sin que se reciba un mensaje RequestChannelCloseAck o RequestChannelCloseReject.

### C.6.4 Procedimientos de la CLCSE

En la figura C.22 se resumen las primitivas de la CLCSE, sus parámetros y mensajes para cada CLCSE de salida y de entrada.

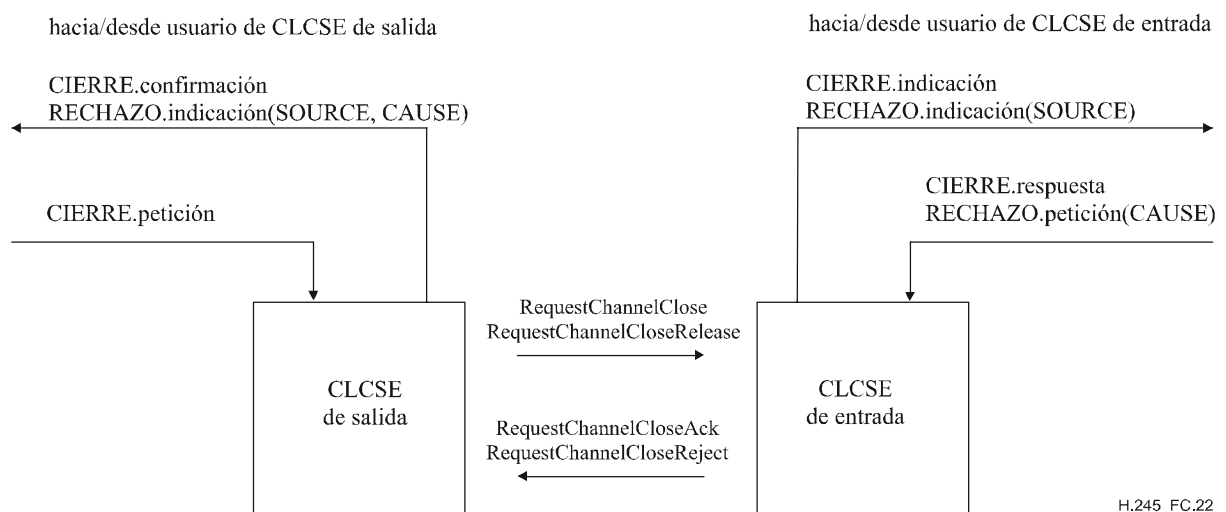


Figura C.22/H.245 – Primitivas y mensajes en la entidad señalización cierre de canal lógico

#### C.6.4.1 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL, los valores de los parámetros de las primitivas de indicación y confirmación serán los del cuadro C.23.

Cuadro C.23/H.245 – Valores supletorios de los parámetros de las primitivas

Primitiva	Parámetro	Valor supletorio
RECHAZO.indicación	SOURCE CAUSE	USER nulo

### C.6.4.2 Valores supletorios de los campos de mensajes

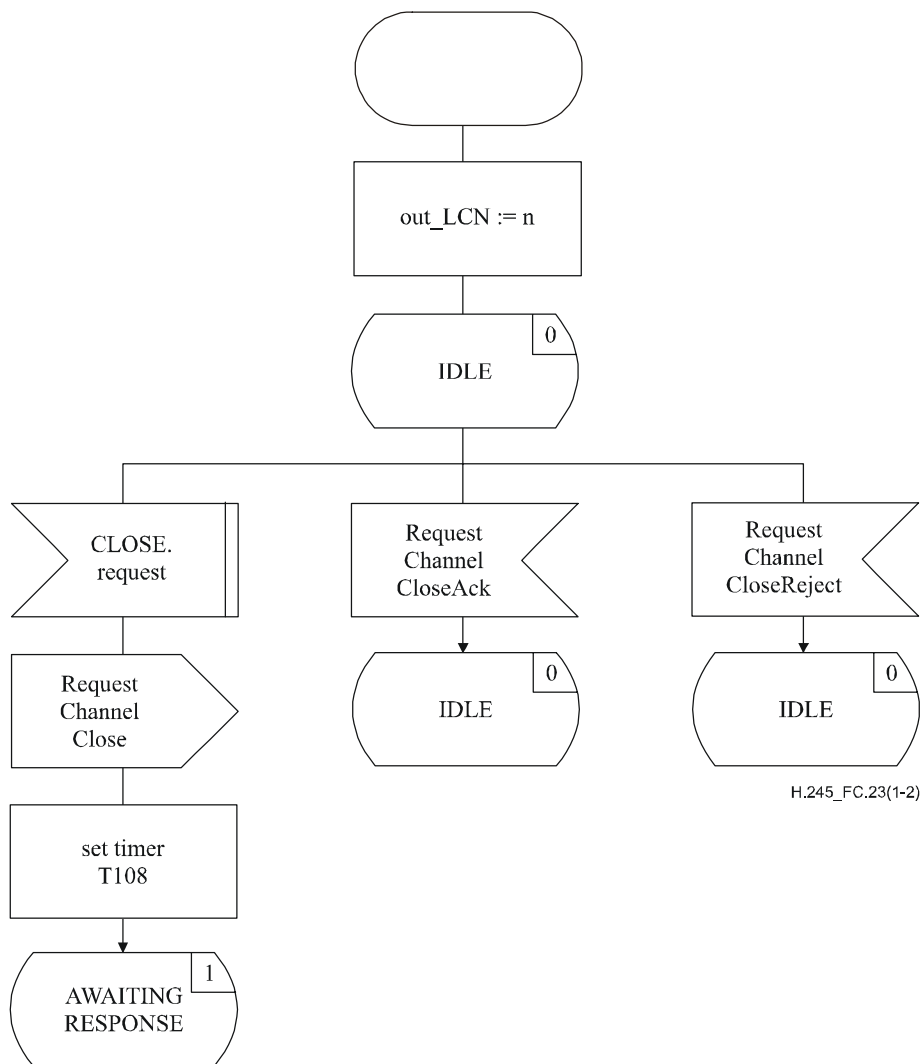
Cuando no se indiquen explícitamente en los diagramas de SDL, los valores supletorios de los campos de mensajes serán los del cuadro C.24.

**Cuadro C.24/H.245 – Valores supletorios de los campos de mensajes**

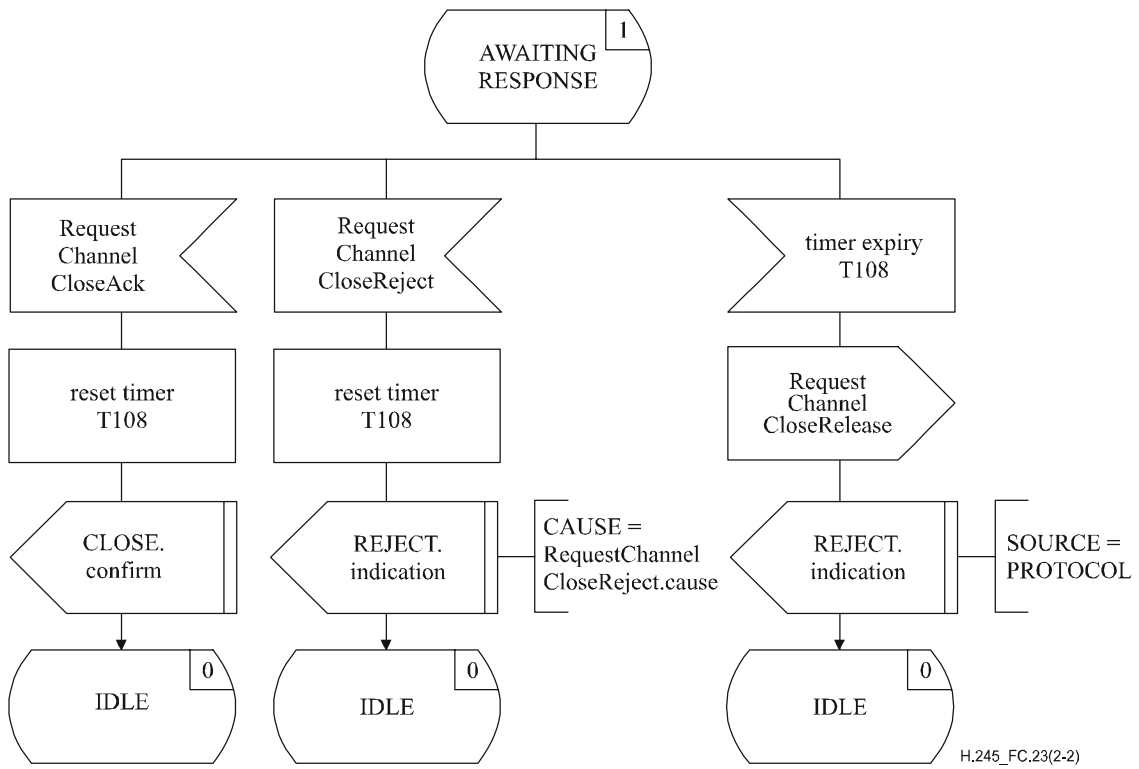
Mensaje	Campo	Valor supletorio
RequestChannelClose	forwardLogicalChannelNumber	out_LCN
RequestChannelCloseAck	forwardLogicalChannelNumber	in_LCN
RequestChannelCloseReject	forwardLogicalChannelNumber cause	in_LCN RECHAZO.petición(CAUSE)
RequestChannelCloseRelease	forwardLogicalChannelNumber	out_LCN

### C.6.4.3 Diagramas de SDL

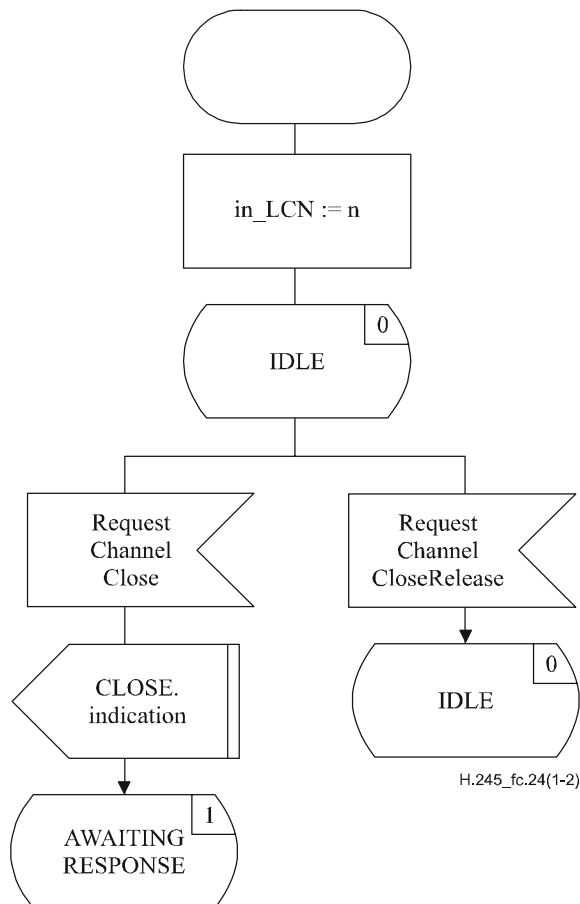
Los procedimientos en la CLCSE de salida y en la CLCSE de entrada, expresados en SDL, se indican en las figuras C.23 y C.24, respectivamente.



**Figura C.23/H.245 – Diagrama SDL de la CLCSE de salida (hoja 1 de 2)**



**Figura C.23/H.245 – Diagrama SDL de la CLCSE de salida (hoja 2 de 2)**



**Figura C.24/H.245 – Diagrama SDL de la CLCSE de entrada (hoja 1 de 2)**

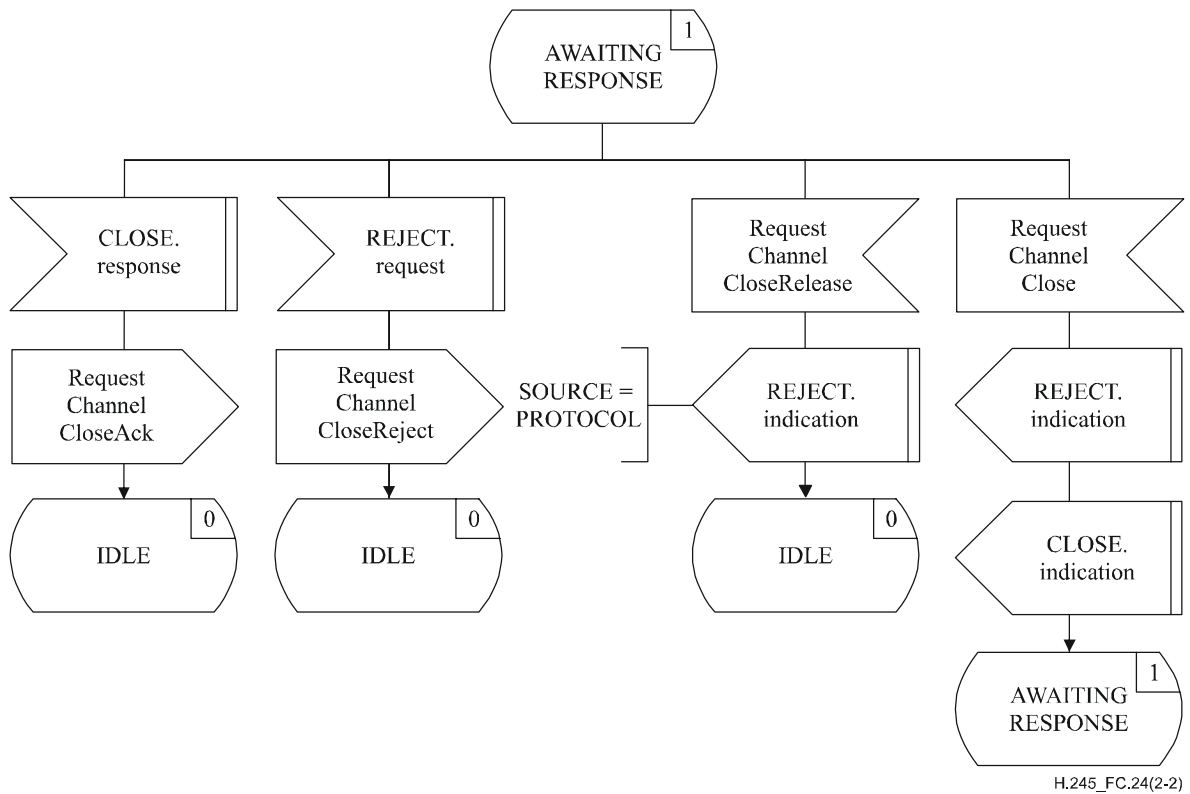


Figura C.24/H.245 – Diagrama SDL de la CLCSE de entrada (hoja 2 de 2)

## C.7 Procedimientos de la tabla múltiplex H.223

### C.7.1 Introducción

La tabla múltiplex se emplea para asociar cada octeto dentro de una MUX-PDU H.223 [10] con un determinado número de canal lógico. La tabla múltiplex H.223 puede tener hasta 16 entradas, numeradas de 0 a 15. Las entradas de tabla del 1 al 15 se enviarán de transmisores a receptores como se especifica en los siguientes procedimientos.

A los procedimientos aquí descritos se hace referencia como la entidad de señalización de tabla múltiplex (MTSE, *multiplex table signalling entity*). Los procedimientos se especifican mediante primitivas y estados en la interfaz entre la MTSE y el usuario de la MTSE. La información de protocolo se transfiere a la MTSE por medio de mensajes apropiados descritos en el anexo A.

Hay una MTSE de salida y una MTSE de entrada. Para cada entrada de tabla múltiplex hay un ejemplar de la MTSE.

Un terminal transmisor utiliza este protocolo para señalar a un terminal distante una o más entradas nuevas de la tabla múltiplex. El terminal distante puede aceptar o rechazar las nuevas entradas de tabla múltiplex. Si el terminal distante acepta una entrada de tabla múltiplex, la anterior entrada en el número de entrada dado se sustituye por la nueva entrada.

El transmisor puede desactivar una entrada de tabla múltiplex enviando un MultiplexEntryDescriptor sin una elementList. El transmisor no utilizará en ningún momento una tabla múltiplex que esté desactivada. Antes de transmitir un MultiplexEntrySend, el transmisor dejará de utilizar las entradas descritas por este mensaje. No rearrancará utilizando esas entradas hasta que haya recibido un MultiplexEntrySendAck. Se utiliza este procedimiento porque si no se detiene la utilización de estas entradas de tabla múltiplex antes de enviar el MultiplexEntrySend, los errores pueden producir ambigüedad en el receptor.

El transmisor dejará de utilizar las entradas desactivadas antes de enviar el MultiplexEntrySend indicando que han sido desactivadas. Las entradas desactivadas pueden utilizarse de nuevo en cualquier momento transmitiendo un mensaje MultiplexEntrySend para activar esas entradas. La desactivación de las entradas que ya no necesite el transmisor puede aumentar la probabilidad de detectar errores en el campo de código múltiplex H.223.

NOTA – Mientras se actualizan algunas entradas de tabla múltiplex, se puede continuar utilizando otras entradas (activas). Asimismo, una entrada de tabla múltiplex puede suprimirse en el mismo MultiplexEntrySend que se utiliza para modificar otras entradas de tabla múltiplex.

Al comienzo de la comunicación, a menos que se especifique otra cosa en una Recomendación apropiada, sólo la entrada 0 de la tabla está disponible para transmisión y las entradas 1 a 15 de la tabla están desactivadas.

Un procedimiento de petición de entrada de tabla múltiplex puede utilizarse en cualquier momento para provocar la retransmisión de determinadas entradas de tabla múltiplex desde el terminal distante, por ejemplo tras una interrupción o cualquier otro evento que cause incertidumbre.

En el texto siguiente se proporciona una visión general del funcionamiento del protocolo. En caso de discrepancia con la descripción formal del protocolo que sigue, prevalecerá la especificación formal.

#### **C.7.1.1 Visión general del protocolo – MTSE de salida**

Se inicia un procedimiento de petición de envío de entrada de tabla múltiplex cuando la primitiva TRANSFERENCIA.petición es emitida por el usuario en la MTSE de salida. Se envía un mensaje MultiplexEntrySend a la MTSE de entrada par y se arranca al temporizador T104. Si se recibe un mensaje MultiplexEntrySendAck en respuesta al mensaje MultiplexEntrySend, se detiene el temporizador T104 y se informa al usuario con la primitiva TRANSFERENCIA.confirmación de que la petición de envío de entrada de tabla múltiplex tuvo éxito. Sin embargo, si se recibe un mensaje MultiplexEntrySendReject en respuesta al mensaje MultiplexEntrySend, se detiene el temporizador T104 y se informa al usuario con la primitiva RECHAZO.indicación de que el usuario de la MTSE par ha rehusado aceptar la entrada de tabla múltiplex.

Si expira el temporizador T104, se informa al usuario MTSE de salida con la primitiva RECHAZO.indicación y se envía un mensaje MultiplexEntrySendRelease.

Sólo se aceptan los mensajes MultiplexEntrySendAck y MultiplexEntrySendReject que son respuesta al mensaje MultiplexEntrySend más reciente. Se ignoran los mensajes MultiplexEntrySend anteriores.

Un nuevo procedimiento de petición de envío de entrada de tabla múltiplex puede ser iniciado con la primitiva TRANSFERENCIA.petición por el usuario en el MTSE de salida antes de que se haya recibido un mensaje MultiplexEntrySendAck o un mensaje MultiplexEntrySendReject.

#### **C.7.1.2 Visión general del protocolo – MTSE de entrada**

Cuando se recibe un mensaje MultiplexEntrySend en la MTSE de entrada, se informa al usuario de la petición de envío de entrada de tabla múltiplex con la primitiva TRANSFERENCIA.indicación. El usuario MTSE de entrada señala la aceptación de la entrada de tabla múltiplex emitiendo la primitiva TRANSFERENCIA.respuesta y se envía un mensaje MultiplexEntrySendAck a la MTSE de salida par. El usuario MTSE de entrada señala el rechazo de la entrada de tabla múltiplex emitiendo la primitiva RECHAZO.petición y se envía un mensaje MultiplexEntrySendReject a la MTSE de salida par.

Un nuevo mensaje envío de entrada múltiplex puede ser recibido antes de que el usuario MTSE de entrada haya respondido a un mensaje MultiplexEntrySend anterior. El usuario de la MTSE de entrada es informado con la primitiva RECHAZO.indicación, seguida de la primitiva



TRANSFERENCIA.indicación, y el usuario de la MTSE de entrada responde a la nueva entrada de tabla múltiplex.

Si se recibe un mensaje MultiplexEntrySendRelease antes de que el usuario MTSE de entrada haya respondido a un mensaje MultiplexEntrySend anterior, se informa al usuario MTSE de entrada con la primitiva RECHAZO.indicación y se descarta la anterior entrada de tabla múltiplex anterior.

## C.7.2 Comunicación entre la MTSE y el usuario de MTSE

### C.7.2.1 Primitivas entre la MTSE y el usuario de MTSE

La comunicación entre la MTSE y el usuario de la MTSE se realiza mediante las primitivas indicadas en el cuadro C.25.

**Cuadro C.25/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
TRANSFERENCIA	MUX-DESCRIPTOR	MUX-DESCRIPTOR	– (nota 1)	–
RECHAZO	CAUSE	SOURCE CAUSE	no definida (nota 2)	no definida
NOTA 1 – "-" indica que no hay parámetros.				
NOTA 2 – "no definida" indica que esta primitiva no está definida.				

### C.7.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- Las primitivas TRANSFERENCIA se utilizan para transferir entradas de tabla múltiplex.
- Las primitivas RECHAZO se utilizan para rechazar una entrada de tabla múltiplex y para terminar una transferencia de entrada de tabla múltiplex.

### C.7.2.3 Definición de los parámetros

La definición de los parámetros de las primitivas del cuadro C.25 es la siguiente:

- El parámetro MUX-DESCRIPTOR es una entrada de tabla múltiplex. Este parámetro se hace corresponder con el campo MultiplexEntryDescriptor del mensaje MultiplexEntrySend y se transporta transparentemente desde el usuario MTSE en la MTSE de salida al usuario MTSE en la MTSE de entrada. Puede haber varios MUX-DESCRIPTOR asociados con la primitiva TRANSFERENCIA.
- El parámetro SOURCE indica la fuente de la primitiva RECHAZO.indicación. El parámetro SOURCE toma los valores "USER" o "PROTOCOL". Este último puede ser consecuencia de la expiración de un temporizador.
- El parámetro CAUSE indica el motivo del rechazo de una entrada de tabla múltiplex. El parámetro CAUSE está ausente cuando el parámetro SOURCE indica "PROTOCOL".

### C.7.2.4 Estados de la MTSE

Los siguientes estados se utilizan para especificar la secuencia permitida de primitivas entre la MTSE y el usuario de la MTSE. Los estados se especifican separadamente para la MTSE de salida y la MTSE de entrada. Los estados para una MTSE de salida son:

Estado 0: REPOSO

No hay transferencia de MTSE en curso. La entrada de tabla múltiplex puede ser utilizada por el transmisor.

### Estado 1: ESPERA DE RESPUESTA

El usuario de la MTSE ha solicitado la transferencia de una entrada de tabla múltiplex y se espera una respuesta de la MTSE par. La entrada de tabla múltiplex no será utilizada por el transmisor.

Los estados para una MTSE de entrada son:

### Estado 0: REPOSO

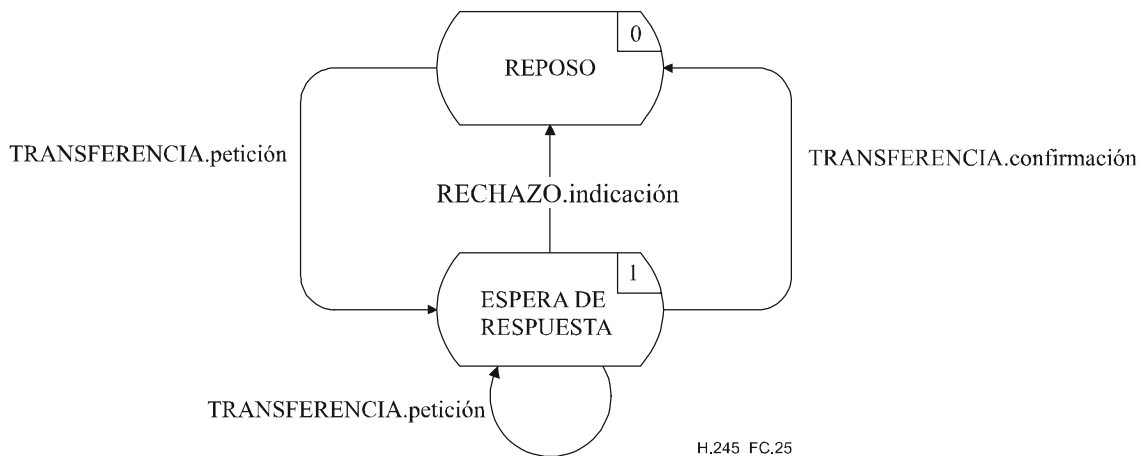
No hay transferencia de MTSE en curso. La entrada de tabla múltiplex puede ser utilizada por el transmisor.

### Estado 1: ESPERA DE RESPUESTA

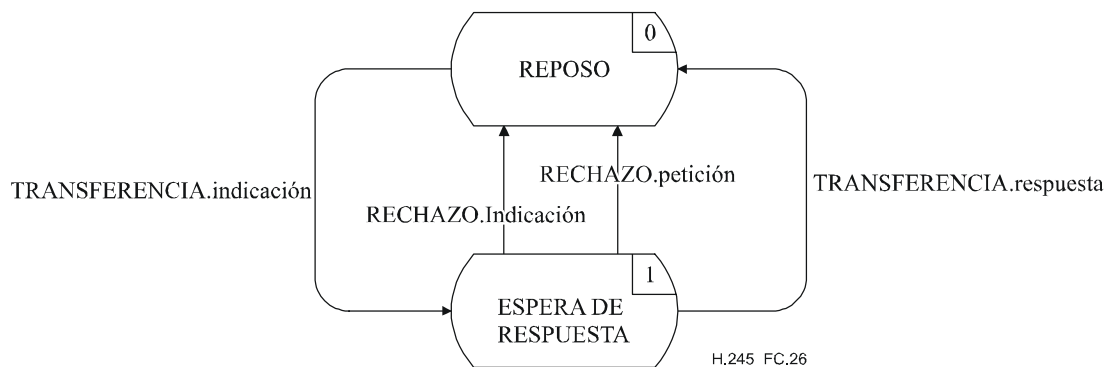
La MTSE par ha transferido una entrada de tabla múltiplex y se espera una respuesta del usuario de la MTSE. La entrada de tabla múltiplex no puede ser utilizada por el transmisor.

### C.7.2.5 Diagramas de transición de estados

Se define aquí la secuencia permitida de primitivas entre la MTSE y el usuario de la MTSE. Las secuencias permitidas se especifican separadamente para la MTSE de salida y la MTSE de entrada, como se indica en las figuras C.25 y C.26, respectivamente.



**Figura C.25/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MTSE de salida**



**Figura C.26/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MTSE de entrada**

### C.7.3 Comunicación entre MTSE pares

#### C.7.3.1 Mensajes

El cuadro C.26 muestra los mensajes y campos de la MTSE, definidos en el anexo A, apropiados para el protocolo de la MTSE.

**Cuadro C.26/H.245 – Nombres de mensajes y campos de la MTSE**

Función	Mensaje	Sentido	Campo
transferencia	MultiplexEntrySend	O → I (nota)	sequenceNumber multiplexEntryDescriptors.multiplexTableEntryNumber multiplexEntryDescriptors.elementList
	MultiplexEntrySendAck	O ← I	sequenceNumber multiplexTableEntryNumber
rechazo	MultiplexEntrySendReject	O ← I	sequenceNumber multiplexTableEntryNumber rejectionDescriptions.cause
reiniciación	MultiplexEntrySendRelease	O → I	multiplexTableEntryNumber

NOTA – Sentido: O – de salida; I – de entrada.

#### C.7.3.2 Variables de estado de la MTSE

Se definen las siguientes variables de estado en la MTSE de salida:

out\_ENUM

Esta variable de estado distingue entre las MTSE de salida. Se inicializa en el momento de la inicialización de la MTSE de salida. El valor de out\_ENUM se utiliza para fijar el campo multiplexTableEntryNumber de los mensajes MTSE enviados desde una MTSE de salida. En el caso de mensajes MTSE recibidos en una MTSE de salida, el valor del campo multiplexTableEntryNumber es idéntico al valor de out\_ENUM.

out\_SQ

Esta variable de estado se utiliza para indicar el mensaje MultiplexEntrySend enviado más recientemente. Esta variable se incrementa en una unidad y se hace corresponder con el campo sequenceNumber del mensaje MultiplexEntrySend antes de la transmisión de un mensaje MultiplexEntrySend. Se aplica aritmética módulo 256 a out\_SQ.

Se definen las siguientes variables de estado en la MTSE de entrada:

in\_ENUM

Esta variable de estado distingue entre las MTSE de entrada. Se inicializa en el momento de la inicialización de la MTSE de entrada. El valor de in\_ENUM se utiliza para fijar el campo multiplexTableEntryNumber de los mensajes MTSE enviados desde una MTSE de entrada. En el caso de mensajes MTSE recibidos en una MTSE de entrada, el valor del campo multiplexTableEntryNumber es idéntico al valor de in\_ENUM.

in\_SQ

Esta variable de estado se utiliza para contener el valor del campo sequenceNumber del mensaje MultiplexEntrySend recibido más recientemente. El campo sequenceNumber de los mensajes MultiplexEntrySendAck y MultiplexEntrySendReject se fija al valor de in\_SQ antes de enviarlos a la MTSE par.

### C.7.3.3 Temporizadores de la MTSE

Para la MTSE de salida se especifica el siguiente temporizador:

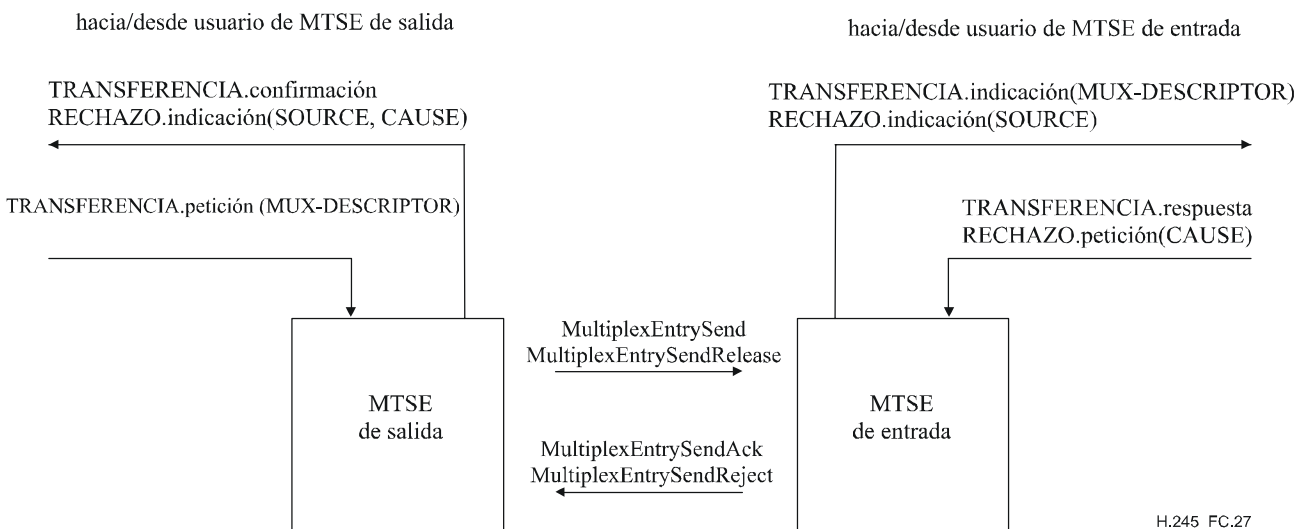
T104

Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el intervalo de tiempo que puede transcurrir sin que se reciba el mensaje MultiplexEntrySendAck o MultiplexEntrySendReject.

### C.7.4 Procedimientos de MTSE

#### C.7.4.1 Introducción

En la figura C.27 se resumen las primitivas de la MTSE, sus parámetros y mensajes para las MTSE de salida y de entrada.



**Figura C.27/H.245 – Primitivas y mensajes en la entidad de señalización de tabla múltiplex**

#### C.7.4.2 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL, los valores de los parámetros de las primitivas indicación y confirmación serán los del cuadro C.27.

**Cuadro C.27/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio
TRANSFERENCIA.indicación	MUX-DESCRIPTOR	MultiplexEntrySend.multiplexEntryDescriptors.elementList
RECHAZO.indicación	SOURCE CAUSE	USER null

### C.7.4.3 Valores supletorios de los campos de mensajes

Cuando no se indiquen explícitamente en los diagramas de SDL, los valores de los campos de mensajes serán los del cuadro C.28.

**Cuadro C.28/H.245 – Valores supletorios de los campos de mensajes**

<b>Mensaje</b>	<b>Campo</b>	<b>Valor supletorio (Nota)</b>
MultiplexEntrySend	sequenceNumber multiplexEntryDescriptors.multiplexTableEntryNumber multiplexEntryDescriptors.elementList	out_SQ out_ENUM TRANSFERENCIA.petición (MUX-DESCRIPTOR)
MultiplexEntrySendAck	sequenceNumber multiplexTableEntryNumber	in_SQ in_ENUM
MultiplexEntrySendReject	sequenceNumber rejectionDescriptions.multiplexTableEntryNumber rejectionDescriptions.cause	in_SQ in_ENUM RECHAZO.petición(CAUSE)
MultiplexEntrySendRelease	multiplexTableEntryNumber	out_ENUM
NOTA – El campo de mensajes no se codificará si el parámetro de la primitiva correspondiente es nulo, esto es, no está presente.		

### C.7.4.4 Diagrama SDL

Los procedimientos de la MTSE de salida y de la MTSE de entrada se expresan en forma SDL en las figuras C.28 y C.29, respectivamente.

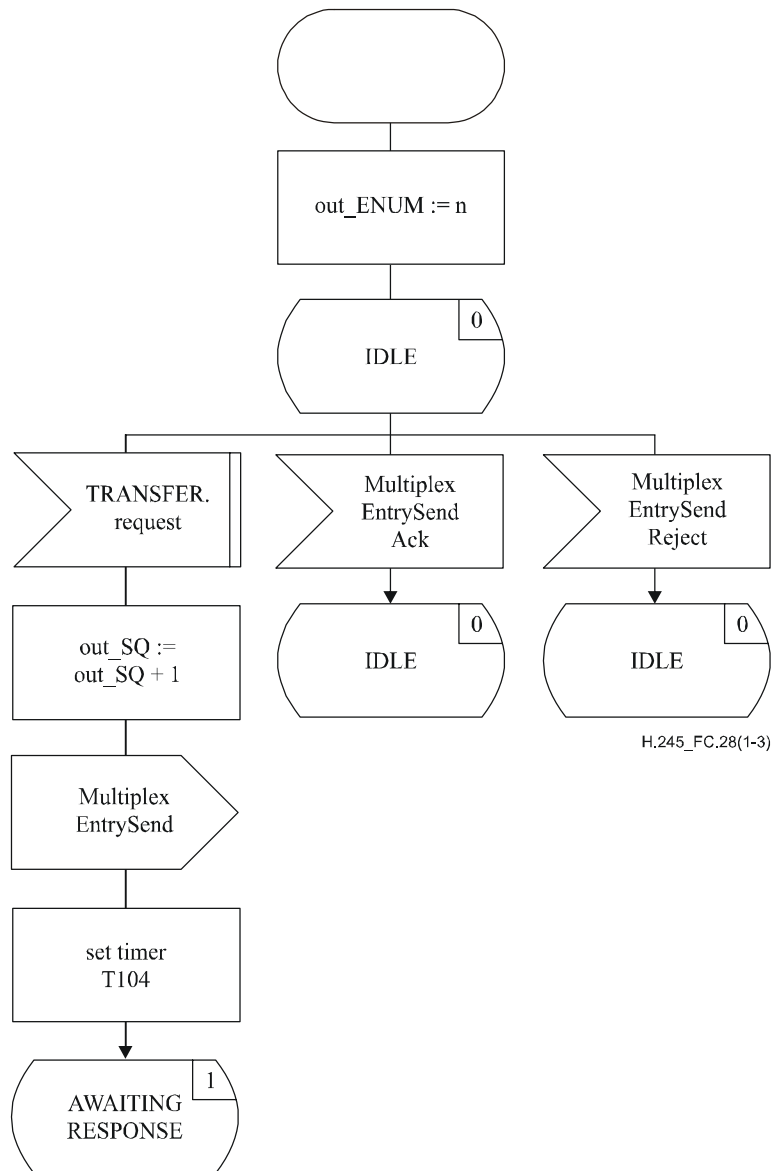
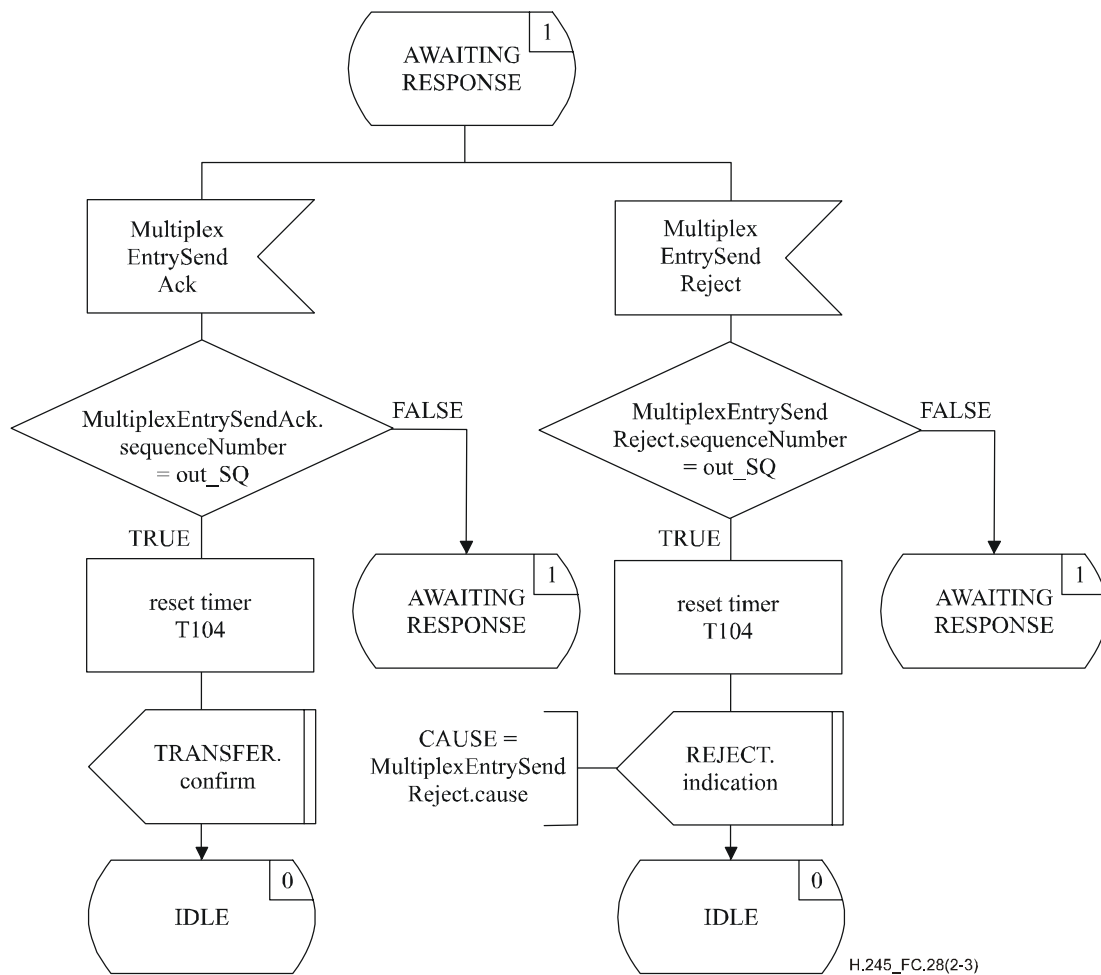
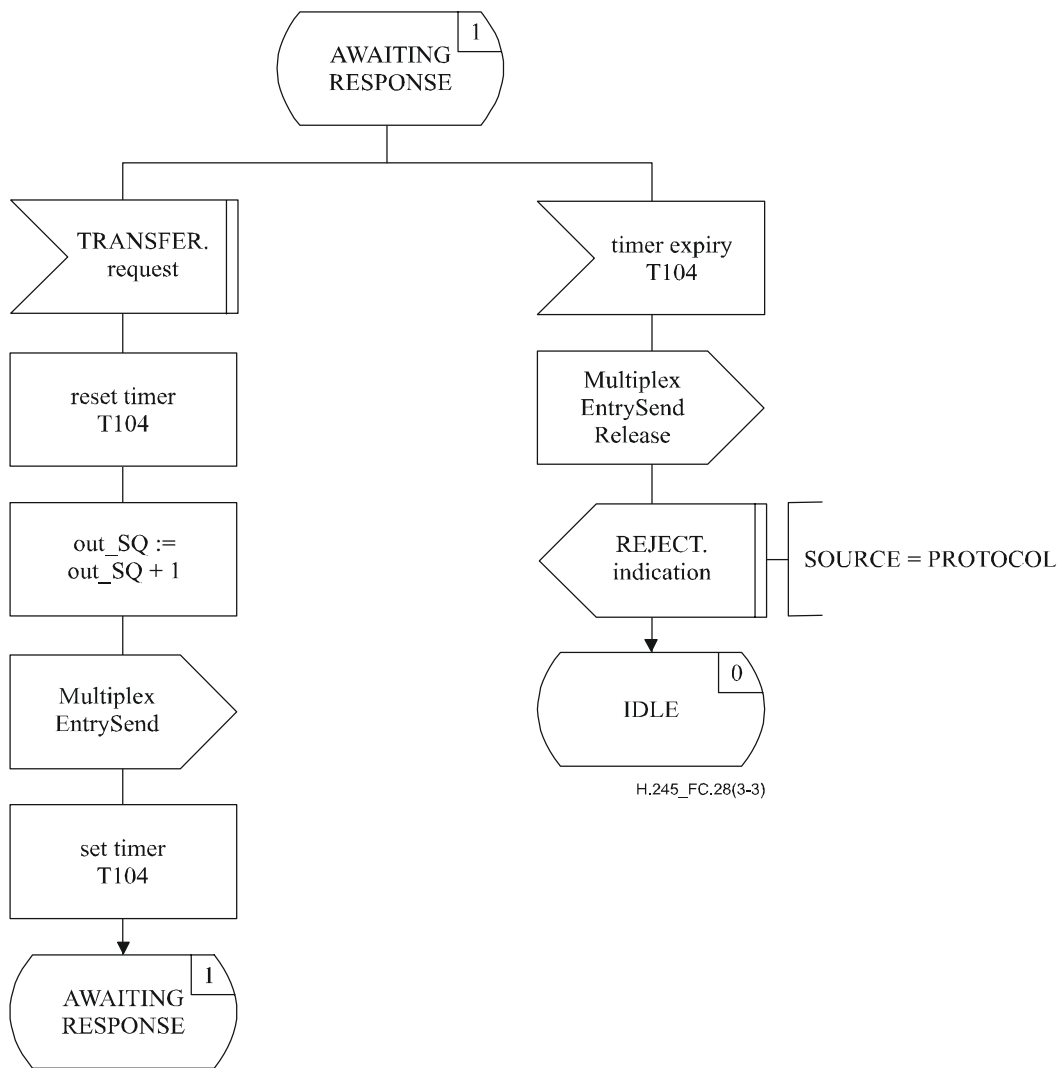


Figura C.28/H.245 – Diagrama SDL de la MTSE de salida (hoja 1 de 3)

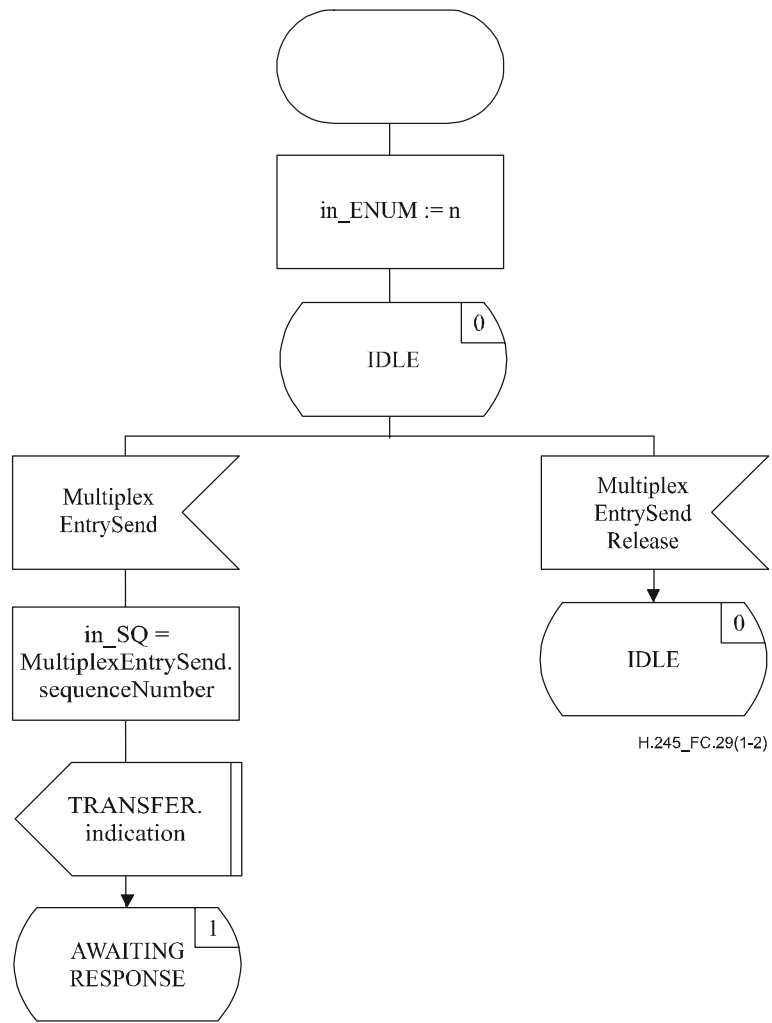


**Figura C.28/H.245 – Diagrama SDL de la MTSE de salida (hoja 2 de 3)**

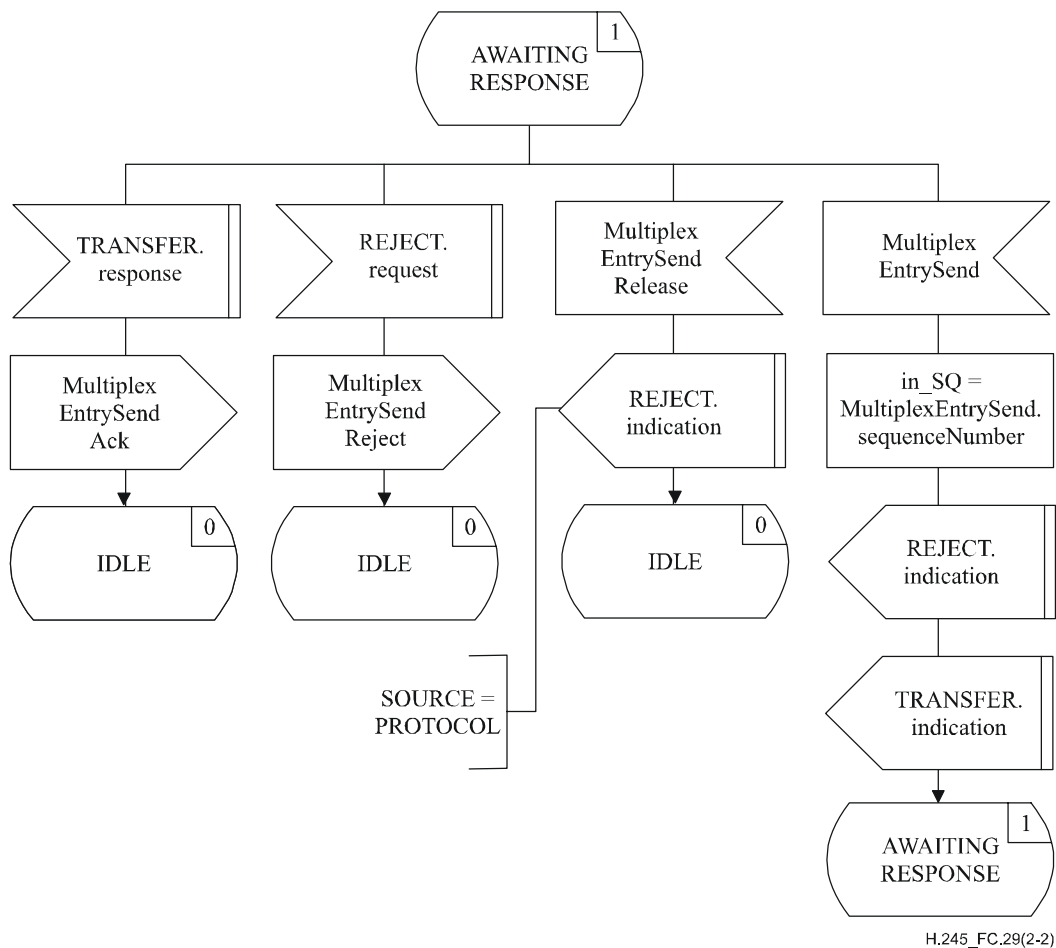


**Figura C.28/H.245 – Diagrama SDL de la MTSE de salida (hoja 3 de 3)**





**Figura C.29/H.245 – Diagrama SDL de la MTSE de entrada (hoja 1 de 2)**



**Figura C.29/H.245 – Diagrama SDL de la MTSE de entrada (hoja 2 de 2)**

## C.8 Procedimientos de petición de entrada múltiplex

### C.8.1 Introducción

Estos procedimientos los utiliza un terminal para solicitar la retransmisión de uno o más MultiplexEntryDescriptor. Estos procedimientos están designados aquí como la entidad de señalización de petición de entrada múltiplex (RMESE, *request multiplex entry signalling entity*). Los procedimientos se especifican mediante primitivas y estados en la interfaz entre la RMESE y el usuario de la RMESE. La información de protocolo se transfiere a la RMESE por medio de mensajes apropiados descritos en el anexo A. Hay una RMESE de salida y una RMESE de entrada. Para cada entrada de tabla múltiplex hay una RMESE.

Un terminal que contesta a tal respuesta positivamente, es decir, emitiendo una primitiva ENVIO.respuesta, iniciará los procedimientos de tabla múltiplex para enviar lo antes posible la entrada de tabla múltiplex.

El texto siguiente proporciona una visión general del funcionamiento del protocolo. En caso de discrepancia con la especificación formal del protocolo que sigue, prevalecerá la especificación formal.

NOTA – Este protocolo se ha definido de modo que haya una RMESE independiente para cada entrada de tabla múltiplex y la sintaxis se ha definido de modo que permita a un solo mensaje transportar información relativa a una o más entradas de tabla múltiplex. La manera de construir los mensajes es una cuestión a decidir en la implementación: por ejemplo, un terminal puede responder a un mensaje RequestMultiplexEntry pidiendo que se envíen tres entradas con uno, dos o tres mensajes de respuesta.

### C.8.1.1 Visión general del protocolo – RMESE de salida

Se inicia un procedimiento de petición de entrada múltiplex cuando un usuario en la RMESE de salida emite una primitiva ENVIO.petición. Envía un mensaje RequestMultiplexEntry a la RMESE de entrada par y se arranca el temporizador T107. Si se recibe un mensaje RequestMultiplexEntryAck en respuesta al mensaje RequestMultiplexEntry, se detiene el temporizador T107 y se informa al usuario con la primitiva ENVIO.confirmación que el procedimiento de petición de entrada múltiplex tuvo éxito. Si, en cambio, se recibe el mensaje RequestMultiplexEntryReject en respuesta al mensaje RequestMultiplexEntry, se detiene el temporizador T107 y se informa al usuario con la primitiva RECHAZO.indicación que el usuario de la RMESE par ha rehusado enviar la entrada múltiplex.

Si expira el temporizador T107 se informa al usuario de la RMESE de salida con la primitiva RECHAZO.indicación y se envía un mensaje RequestMultiplexEntryRelease.

### C.8.1.2 Visión general del protocolo – RMESE de entrada

Cuando se recibe un mensaje RequestMultiplexEntry en la RMESE de entrada, se informa al usuario sobre la petición de entrada múltiplex mediante la primitiva ENVIO.indicación. El usuario de la RMESE de entrada señala la aceptación de la petición de entrada múltiplex emitiendo la primitiva ENVIO.respuesta y se envía un mensaje RequestMultiplexEntryAck a la RMESE de salida par. El usuario de la RMESE de entrada señala el rechazo de la petición de entrada múltiplex emitiendo la primitiva RECHAZO.petición y se envía un mensaje RequestMultiplexEntryReject a la RMESE de salida par.

## C.8.2 Comunicación entre la RMESE y el usuario de RMESE

### C.8.2.1 Primitivas entre la RMESE y el usuario de RMESE

La comunicación entre la RMESE y el usuario de la RMESE se realiza mediante las primitivas del cuadro C.29.

**Cuadro C.29/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
ENVÍO	– (nota 1)	–	–	–
RECHAZO	CAUSE	SOURCE CAUSE	no definida (nota 2)	no definida
NOTA 1 – "–" indica que no hay parámetros.				
NOTA 2 – "no definida" indica que esta primitiva no está definida.				

### C.8.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- Se utilizan las primitivas ENVÍO para pedir la transmisión de una entrada de tabla múltiplex.
- Se utilizan las primitivas RECHAZO para rechazar la petición de transmisión de una entrada de tabla múltiplex.

### C.8.2.3 Definición de los parámetros

La definición de los parámetros de las primitivas del cuadro C.29 es la siguiente:

- a) El parámetro SOURCE indica la fuente de la primitiva RECHAZO.indicación. El parámetro SOURCE toma los valores "USER" o "PROTOCOL". Este último caso puede ser consecuencia de la expiración de un temporizador.
- b) El parámetro CAUSE indica el motivo por el cual se rehúsa a enviar una entrada de tabla múltiplex. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "PROTOCOL".

### C.8.2.4 Estados de la RMESE

Se utilizan los estados que siguen para especificar la secuencia permitida de primitivas entre la RMESE y el usuario de la RMESE.

Para la RMESE de salida, los estados son:

Estado 0: REPOSO

La RMESE está en reposo.

Estado 1: ESPERA DE RESPUESTA

La RMESE está en espera de una respuesta de la RMESE distante.

Para la RMESE de entrada, los estados son:

Estado 0: REPOSO

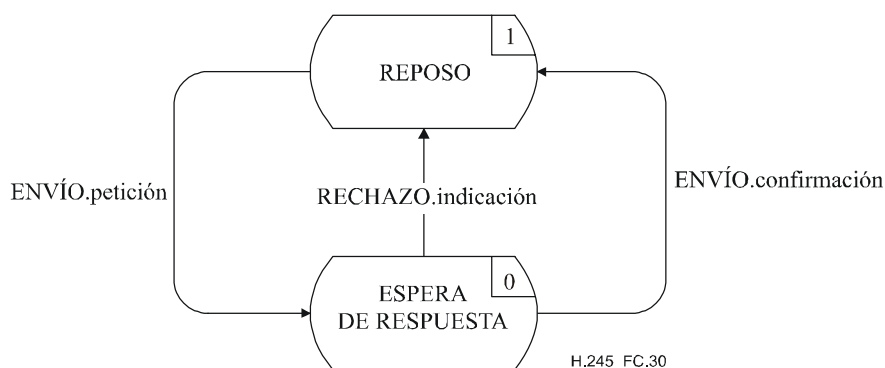
La RMESE está en reposo.

Estado 1: ESPERA DE RESPUESTA

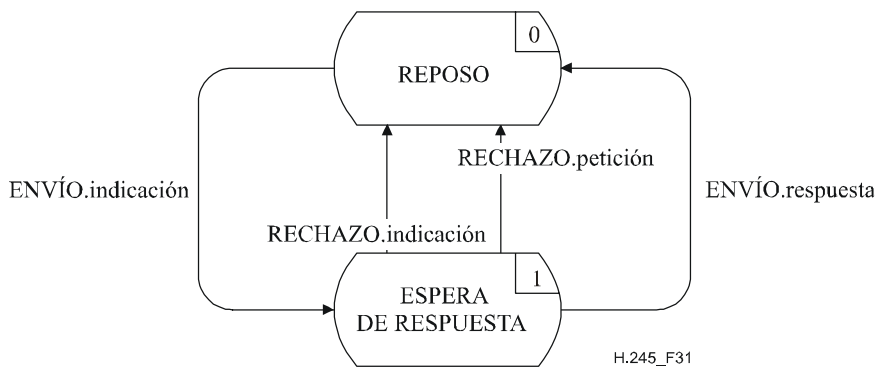
La RMESE está en espera de una respuesta del usuario de la RMESE.

### C.8.2.5 Diagramas de transición de estados

Se define aquí la secuencia permitida de primitivas entre la RMESE y el usuario de la RMESE. Las secuencias permitidas se especifican separadamente para la RMESE de salida y la RMESE de entrada como se indica en las figuras C.30 y C.31, respectivamente.



**Figura C.30/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una RMESE de salida**



**Figura C.31/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una RMESE de entrada**

### C.8.3 Comunicación entre RMESE pares

#### C.8.3.1 Mensajes

El cuadro C.30 muestra los mensajes y campos de la RMESE, definidos en el anexo A que son apropiados para el protocolo de la RMESE.

**Cuadro C.30/H.245 – Nombres de mensajes y campos de la RMESE**

<b>Función</b>	<b>Mensaje</b>	<b>Sentido</b>	<b>Campo</b>
transferencia	RequestMultiplexEntry	O → I (nota)	multiplexTableEntryNumber
	RequestMultiplexEntryAck	O ← I	multiplexTableEntryNumber
	RequestMultiplexEntryReject	O ← I	multiplexTableEntryNumber rejectionDescriptions.cause
reiniciación	RequestMultiplexEntryRelease	O → I	
NOTA – Sentido: O – de salida; I – de entrada.			

#### C.8.3.2 Variables de estado de la RMESE

Para la RMESE de salida se definen las siguientes variables de estado:

out\_ENUM

Esta variable de estado permite efectuar la distinción entre las RMESE de salida. Se inicializa en una inicialización de la RMESE de salida. Se utiliza el valor de out\_ENUM para establecer el campo multiplexTableEntryNumber de los mensajes de la RMESE enviados desde una RMESE de salida. Para los mensajes de RMESE recibidos en una RMESE de salida, el valor del campo multiplexTableEntryNumber del mensaje es idéntico al valor de out\_ENUM.

En la RMESE de entrada se definen las siguientes variables de estado:

in\_ENUM

Esta variable de estado permite efectuar la distinción entre las RMESE de entrada. Se inicializa en una inicialización de la RMESE de entrada. Se utiliza el valor de in\_ENUM para establecer el campo multiplexTableEntryNumber de los mensajes de la RMESE enviados desde una RMESE de entrada. Para los mensajes de RMESE recibidos en una RMESE de entrada, el valor del campo multiplexTableEntryNumber del mensaje es idéntico al valor de in\_ENUM.

### C.8.3.3 Temporizadores de la RMESE

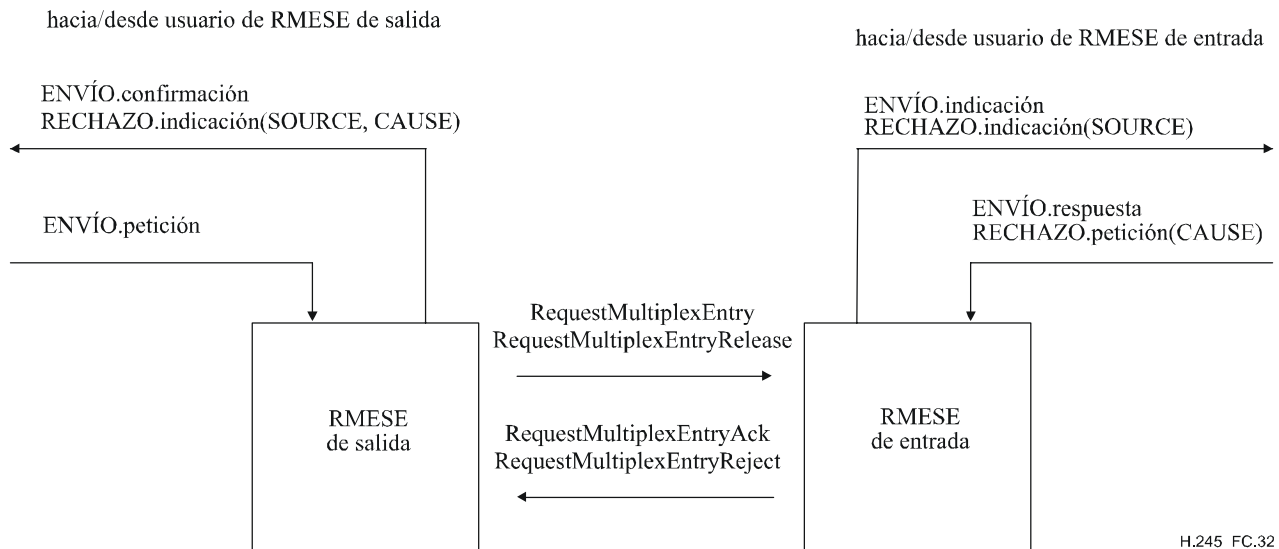
Para la RMESE de salida, se especifica el siguiente temporizador:

T107

Se utiliza este temporizador en el estado ESPERA DE RESPUESTA. Especifica el máximo intervalo de tiempo que puede transcurrir antes de que se reciba un mensaje RequestMultiplexEntryAck o RequestMultiplexEntryReject.

### C.8.4 Procedimientos de RMESE

La figura C.32 resume las primitivas de la RMESE y sus parámetros, así como los mensajes, para la RMESE de salida y la de entrada.



**Figura C.32/H.245 – Primitivas y mensajes en entidades de señalización de petición de entrada múltiple**

#### C.8.4.1 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL, los valores de los parámetros de las primitivas indicación y confirmación serán los del cuadro C.31.

**Cuadro C.31/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio
RECHAZO.indicación	SOURCE	USER
	CAUSE	null

#### C.8.4.2 Valores supletorios de los campos de mensajes

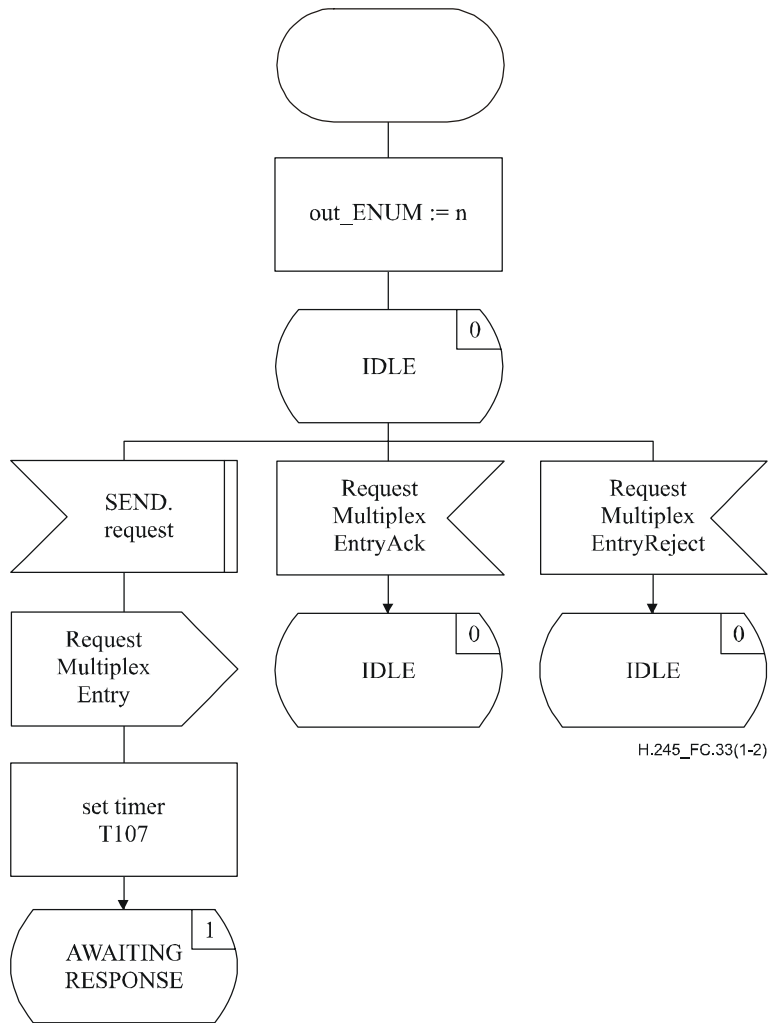
Cuando no se indiquen explícitamente en los diagramas SDL, los valores de los campos de mensajes serán los del cuadro C.32.

**Cuadro C.32/H.245 – Valores supletorios de los campos de mensajes**

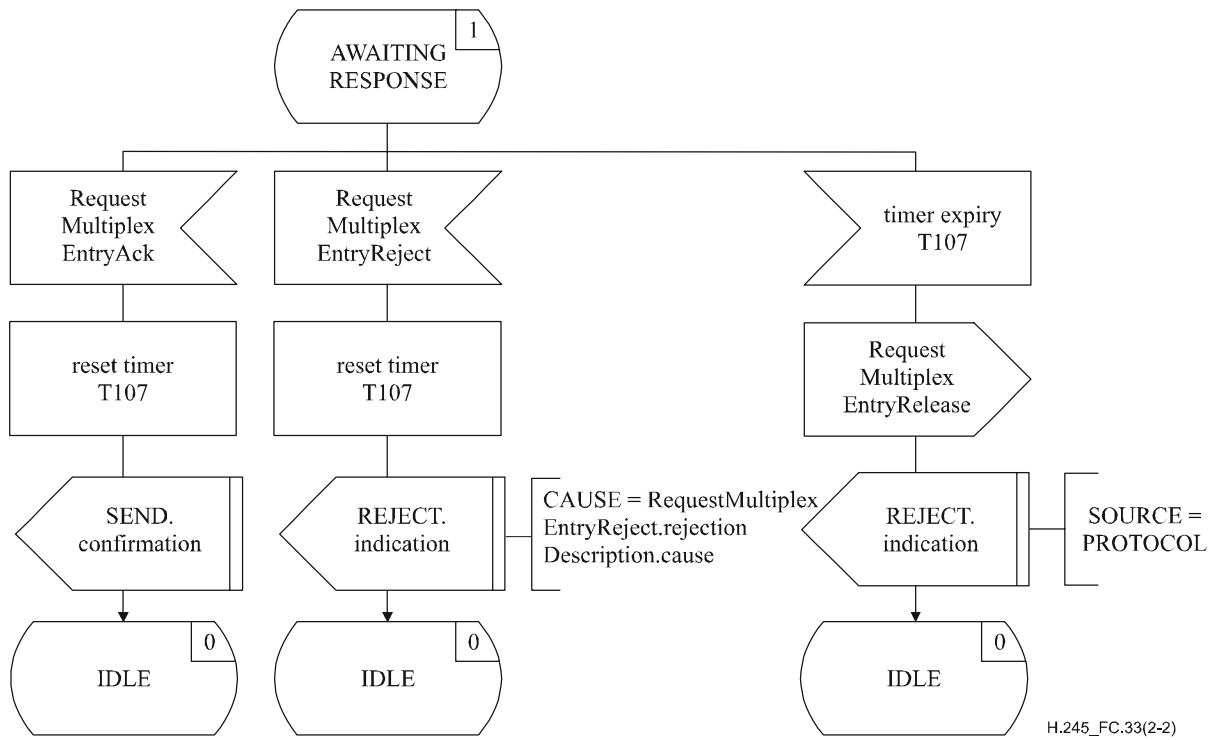
Mensajes	Campo	Valor supletorio
RequestMultiplexEntry	multiplexTableEntryNumber	out_ENUM
RequestMultiplexEntryAck	multiplexTableEntryNumber	in_ENUM
RequestMultiplexEntryReject	multiplexTableEntryNumber cause	in_ENUM RECHAZO.petición(CAUSE)
RequestMultiplexEntryRelease	multiplexTableEntryNumber	out_ENUM

**C.8.4.3 Diagramas SDL**

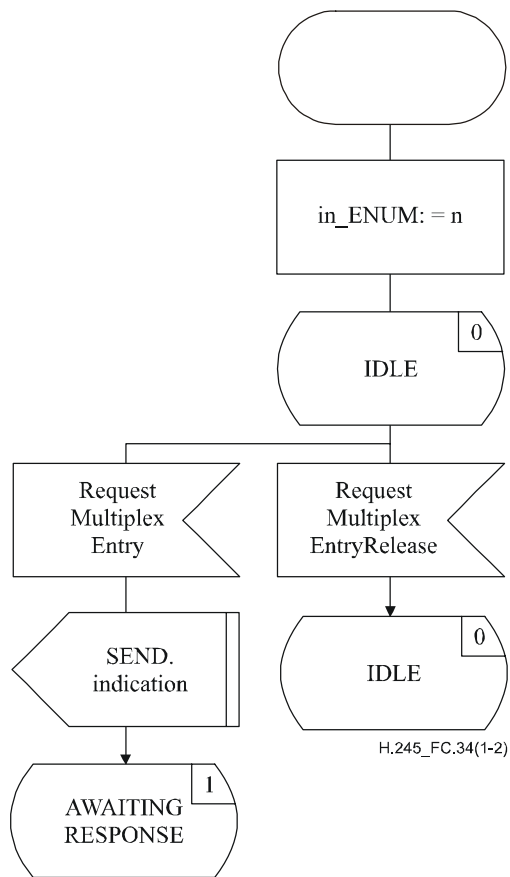
Los procedimientos de la RMESE de salida y la RMESE de entrada se expresan en forma de diagramas SDL en las figuras C.33 y C.34, respectivamente.



**Figura C.33/H.245 – Diagrama SDL de la RMESE de salida (hoja 1 de 2)**

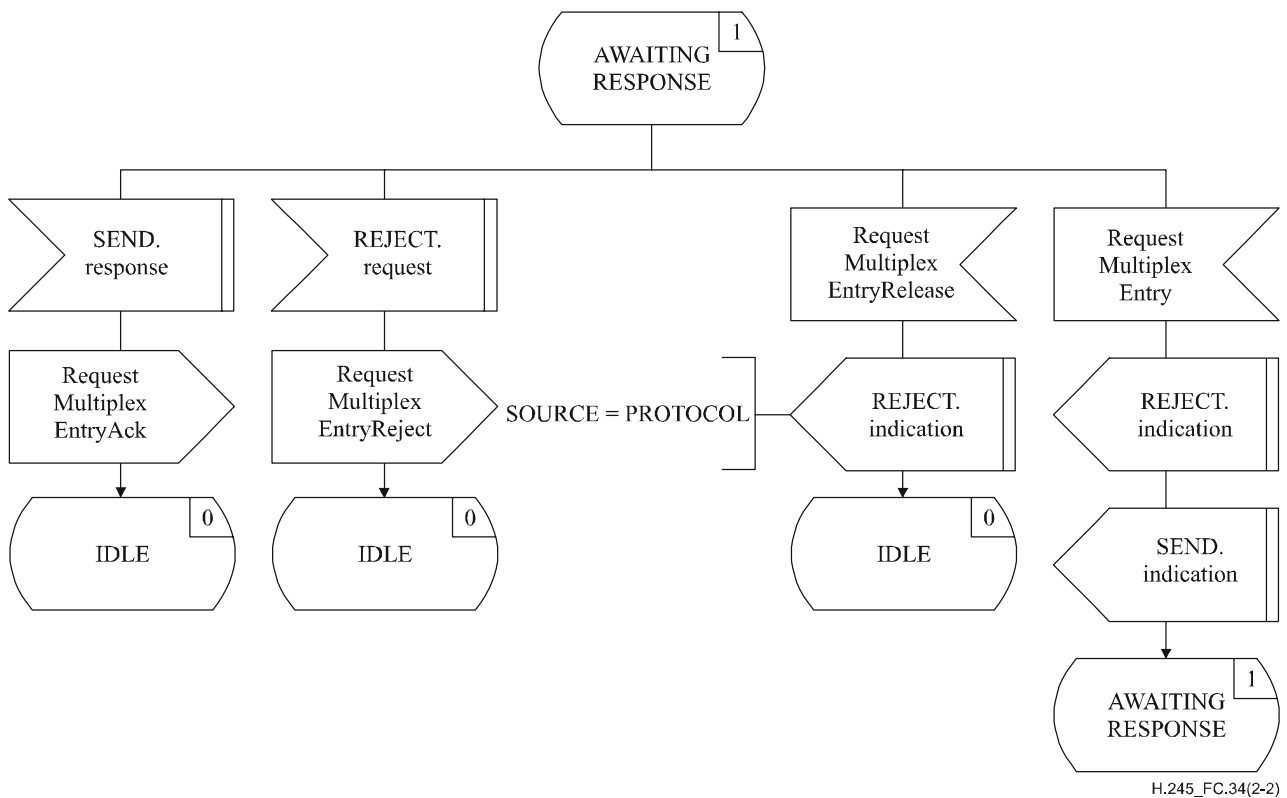


**Figura C.33/H.245 – Diagrama SDL de la RMESE de salida (hoja 2 de 2)**



**Figura C.34/H.245 – Diagrama SDL de la RMESE de entrada (hoja 1 de 2)**





**Figura C.34/H.245 – Diagrama SDL de la RMESE de entrada (hoja 2 de 2)**

## C.9 Procedimientos de petición de modo

### C.9.1 Introducción

Los procedimientos aquí descritos, y denominados entidad de señalización de petición de modo (MRSE, *mode request signalling entity*), permiten que un terminal solicite a un terminal distante el empleo de un modo determinado de operación en su sentido de transmisión. Se especifican los procedimientos mediante primitivas y estados en la interfaz entre la MRSE y el usuario de la MRSE. La información de protocolo se transfiere a la MRSE par por medio de mensajes apropiados descritos en el anexo A. Hay una MRSE de salida y una MRSE de entrada. En el extremo de salida y en el extremo de entrada hay un ejemplar de la MRSE por cada llamada.

Un terminal que responde a tal respuesta positivamente, es decir, emitiendo la primitiva TRANSFERENCIA.respuesta, iniciará los procedimientos de señalización de canal lógico para establecer el modo de transmisión apropiado lo antes posible.

Si las capacidades actualmente válidas recibidas desde el terminal distante contienen una o más capacidades de transmisión, el terminal puede seleccionar el modo en el que prefiera transmitir mediante los procedimientos de petición de modo. Todo terminal cuyas capacidades actualmente válidas contengan una o más capacidades de transmisión y que reciba tal petición, deberá cumplimentarla.

No se enviará una petición de modo a un terminal cuyas capacidades actualmente válidas no contengan capacidades de transmisión, es decir el terminal no desea ser controlado a distancia y no deberá serlo. Sin embargo, si tal terminal recibe una petición de modo, deberá cumplimentarla.

Un terminal que reciba `multipointModeCommand` cumplirá todas las peticiones de modo recibidas, hasta que se cancele la instrucción por recibo de `cancelMultipointModeCommand`. Una petición de modo puede ser enviada a un terminal cuyas capacidades válidas en ese momento no contengan capacidades de transmisión cuando se ha enviado previamente `multipointModeCommand`.

El modo solicitado puede incluir canales ya abiertos. Por ejemplo, si actualmente está abierto un canal para G.723.1 y un terminal desea recibir un canal G.728 adicional, deberá enviar una petición de modo que contenga los dos canales: G.723.1 y G.728. De estar ausente la petición del canal G.723.1 ello indicaría que ya no se desea utilizar este canal.

Cuando el parámetro `logicalChannelNumber` está presente, la petición se refiere sólo al canal lógico indicado, que estará en el estado abierto, y solicita que el modo del canal lógico indicado se cambie al modo especificado.

NOTA – La petición de descripción de modo especifica un modo completo, salvo que el parámetro `logicalChannelNumber` esté presente. Si, por ejemplo, en un momento dado se está transmitiendo vídeo y se recibe una petición de modo que no incluye ninguna especificación para vídeo, con esto se pide que se detenga la transmisión de vídeo.

Cuando una fuente está conectada con varios receptores puede que no sea capaz de responder a cualesquiera señales recibidas tales como peticiones para transmitir en un modo determinado.

El texto que sigue proporciona una visión general del funcionamiento de protocolo de MRSE. En caso de cualquier discrepancia entre ésta y la especificación formal, prevalecerá la especificación formal.

#### **C.9.1.1 Visión general del protocolo – MRSE de salida**

Se inicia un procedimiento de petición de modo cuando la primitiva `TRANSFERENCIA.petición` es emitida por el usuario en la MRSE de salida. Se envía un mensaje `RequestMode` a la MRSE de entrada par y se arranca el temporizador T109. Si se recibe un mensaje `RequestModeAck` en respuesta al mensaje `RequestMode`, se detiene entonces el temporizador T109 y se informa al usuario con la primitiva `TRANSFERENCIA.confirmación` de que la petición de modo tuvo éxito. Sin embargo, si se recibe un mensaje `RequestModeReject` en respuesta al mensaje `RequestMode`, se detiene entonces el temporizador T109 y se informa al usuario con la primitiva `RECHAZO.indicación` de que el usuario MRSE par ha rehusado aceptar la petición de modo.

Si expira el temporizador T109, se informa al usuario MRSE de salida con la primitiva `RECHAZO.indicación` y se envía un mensaje `RequestModeRelease`.

Sólo se aceptan los mensajes `RequestModeAck` y `RequestModeReject` que se han transmitido en respuesta al mensaje `RequestMode` más reciente. Los mensajes en respuesta a mensajes `RequestMode` anteriores son ignorados.

Un nuevo procedimiento de petición de modo puede ser iniciado con la primitiva `TRANSFERENCIA.petición` por el usuario en la MRSE de salida antes de que se haya recibido un mensaje `RequestModeAck` o `RequestModeReject`.

#### **C.9.1.2 Visión general del protocolo – MRSE de entrada**

Cuando se recibe un mensaje `RequestMode` en la MRSE de entrada, se informa al usuario de la petición de modo con la primitiva `TRANSFERENCIA.indicación`. El usuario de la MRSE de entrada señala la aceptación de la petición de modo emitiendo la primitiva `TRANSFERENCIA.respuesta` y se envía un mensaje `RequestModeAck` a la MRSE de salida par. El usuario de la MRSE de entrada señala el rechazo de la petición de modo emitiendo la primitiva `RECHAZO.petición` y se envía un mensaje `RequestModeReject` a la MRSE de salida par.

Puede recibirse un nuevo mensaje `RequestMode` antes de que el usuario de la MRSE de salida haya respondido a un mensaje `RequestMode` anterior. Se informa al usuario de la MRSE de entrada con

la primitiva RECHAZO.indicación, seguida de la primitiva TRANSFERENCIA.indicación, y el usuario de la MRSE de entrada responde a la nueva entrada de tabla múltiplex.

Si se recibe un mensaje RequestModeRelease antes de que el usuario de la MRSE de entrada haya respondido a un mensaje RequestMode anterior, se informa entonces al usuario de la MRSE de entrada con la primitiva RECHAZO.indicación y se descarta la anterior petición de modo.

## C.9.2 Comunicación entre la MRSE y el usuario de MRSE

### C.9.2.1 Primitivas entre la MRSE y el usuario de MRSE

La comunicación entre la MRSE y el usuario de la MRSE se realiza mediante las primitivas indicadas en el cuadro C.33.

**Cuadro C.33/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
TRANSFERENCIA	MODE-ELEMENT	MODE-ELEMENT	MODE-PREF	MODE-PREF
RECHAZO	CAUSE	SOURCE CAUSE	no definida (nota)	no definida
NOTA – "no definida" indica que esta primitiva no está definida.				

### C.9.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- Se utilizan las primitivas TRANSFERENCIA para la transferencia de la petición de modo.
- Se utilizan las primitivas RECHAZO para el rechazo de una petición de modo.

### C.9.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro C.33 son:

- El parámetro MODE-ELEMENT especifica un elemento de modo. Este parámetro se pone en correspondencia con el campo requestedModes del mensaje RequestMode y se transporta transparentemente del usuario de la MRSE de salida al usuario de la MRSE de entrada. Este parámetro es obligatorio. Puede haber múltiples MODE-ELEMENT asociados con las primitivas TRANSFERENCIA.
- El parámetro MODE-PREF informa al usuario sobre si se utilizará o no el modo preferido solicitado. Este parámetro se pone en correspondencia con el campo respuesta del mensaje RequestModeAck y se transporta transparentemente del usuario de la RMSE de entrada al usuario de la RMSE de salida. Sus dos valores son "MOST-PREFERRED" y "LESS-PREFERRED".
- El parámetro SOURCE indica la fuente de la primitiva RECHAZO.indicación. El parámetro SOURCE toma los valores "USER" o "PROTOCOL". Este último puede producirse como consecuencia de la expiración de un temporizador.
- El parámetro CAUSE indica el motivo del rechazo del cierre de un canal lógico. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "PROTOCOL".

### C.9.2.4 Estados de la MRSE

Se utilizan los siguientes estados para especificar la secuencia permitida de primitivas entre una MRSE y el usuario de una MRSE. Para la MRSE de salida los estados son:

Estado 0: REPOSO

La MRSE está en reposo.

Estado 1: ESPERA DE RESPUESTA

La MRSE espera una respuesta de la MRSE distante.

Para la MRSE de entrada los estados son:

Estado 0: REPOSO

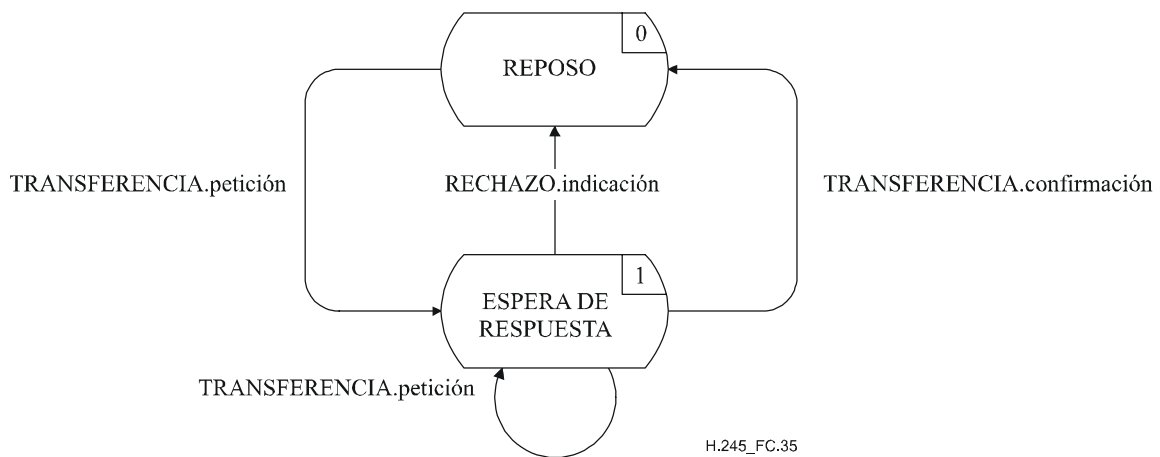
La MRSE está en reposo.

Estado 1: ESPERA DE RESPUESTA

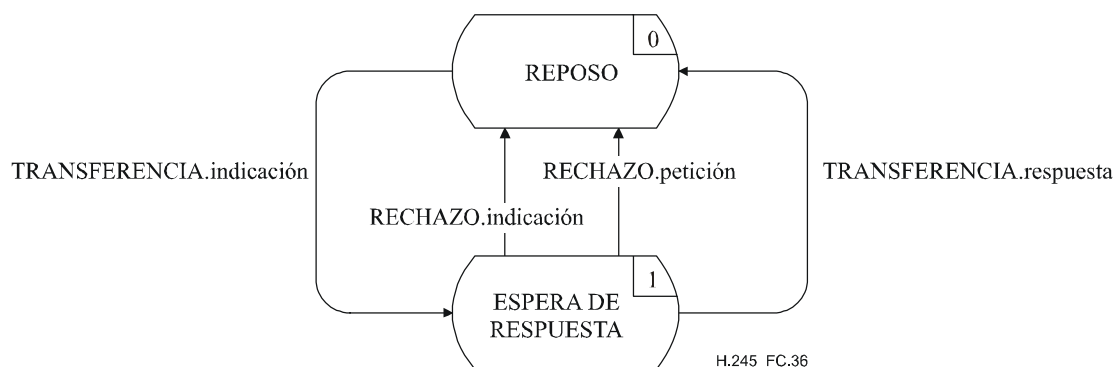
La MRSE espera una respuesta del usuario de la MRSE.

### C.9.2.5 Diagramas de transición de estados

Se define aquí la secuencia permitida de primitivas entre la MRSE y el usuario de la MRSE. Las secuencias permitidas se especifican separadamente para la MRSE de salida y la MRSE de entrada como se indica en las figuras C.35 y C.36, respectivamente.



**Figura C.35/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MRSE de salida**



**Figura C.36/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MRSE de entrada**

### C.9.3 Comunicación entre MRSE pares

#### C.9.3.1 Mensajes

El cuadro C.34 muestra los mensajes y campos de la MRSE, definidos en el anexo A, que son apropiados para el protocolo de la MRSE.

**Cuadro C.34/H.245 – Nombres de mensajes y campos de la MRSE**

Función	Mensaje	Sentido	Campo
petición de modo	RequestMode	O → I (nota)	sequenceNumber requestedModes
	RequestModeAck	O ← I	sequenceNumber response
	RequestModeReject	O ← I	sequenceNumber cause
reiniciación	RequestModeRelease	O → I	–

NOTA – Sentido: O – de salida; I – de entrada.

#### C.9.3.2 Variables de estado de la MRSE

En la MRSE de salida se define la siguiente variable de estado:

out\_SQ

Se emplea esta variable de estado para indicar el mensaje RequestMode más reciente. Antes de la transmisión del mensaje RequestMode, esta variable se incrementa en uno y se pone en correspondencia con el campo sequenceNumber del mensaje RequestMode. Para out\_SQ se utiliza aritmética módulo 256.

En la MRSE de entrada se define la siguiente variable de estado:

in\_SQ

Se utiliza esta variable de estado para almacenar el valor del campo sequenceNumber del mensaje RequestMode recibido más recientemente. Los mensajes RequestModeAck y RequestModeReject tienen sus campos sequenceNumber puestos al valor in\_SQ antes de su envío a la MRSE par.

#### C.9.3.3 Temporizadores de la MRSE

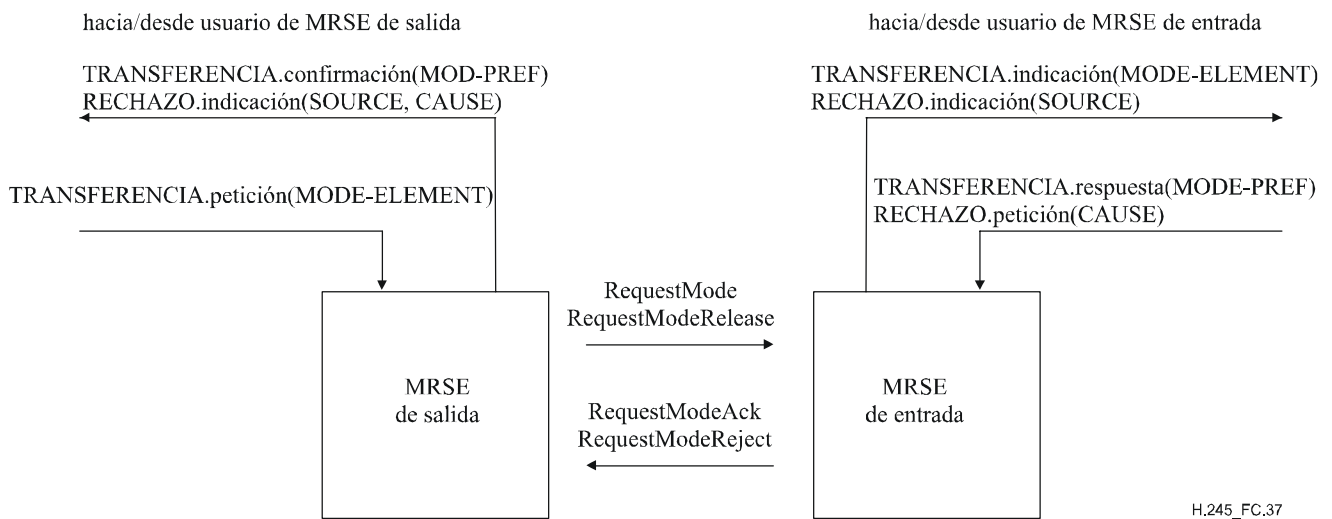
Para la MRSE de salida se define el siguiente temporizador:

T109

Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo intervalo de tiempo durante el cual no pueden recibirse mensajes RequestModeAck o RequestModeReject.

#### C.9.4 Procedimientos de MRSE

En la figura C.37 se resumen las primitivas de la MRSE, sus parámetros y los mensajes para las MRSE de salida y de entrada.



**Figura C.37/H.245 – Primitivas y mensajes en la entidad de señalización petición de modo**

#### C.9.4.1 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas SDL, los valores de los parámetros de las primitivas indicación y confirmación serán los del cuadro C.35.

**Cuadro C.35/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio
TRANSFERENCIA.indicación	MODE-ELEMENT	RequestMode.requestedModes
TRANSFERENCIA.confirmación	MODE-PREF	RequestModeAck.response
RECHAZO.indicación	SOURCE CAUSE	USER nulo

#### C.9.4.2 Valores supletorios de los campos de mensajes

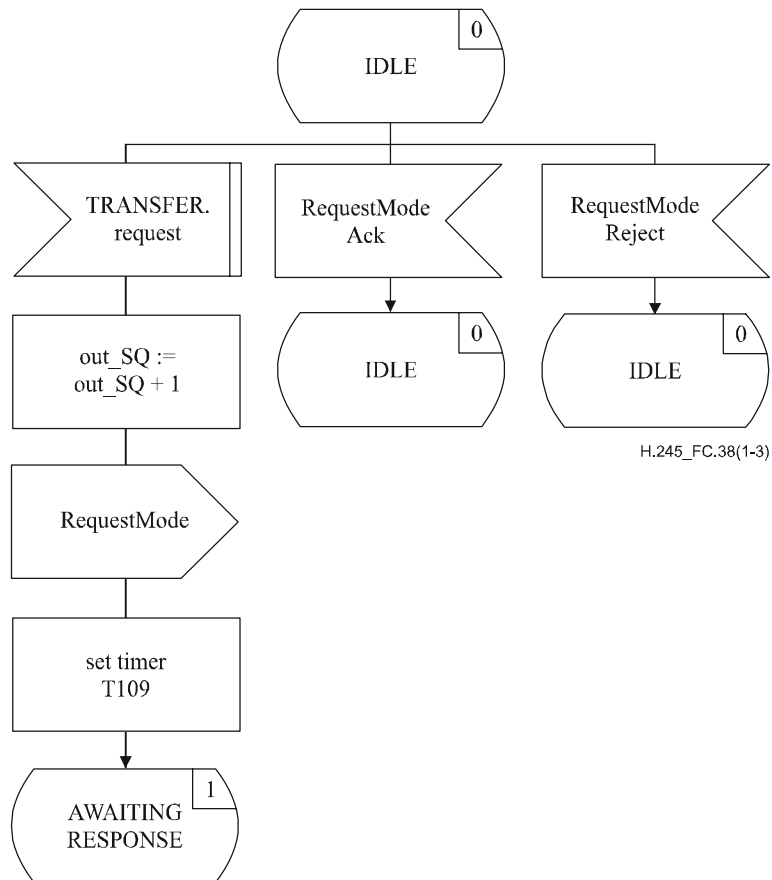
Cuando no se indiquen explícitamente en los diagramas de SDL, los valores supletorios de los campos de mensajes serán los del cuadro C.36.

**Cuadro C.36/H.245 – Valores supletorios de los campos de mensajes**

Mensaje	Campo	Valor supletorio
RequestMode	sequenceNumber	out_SQ
	requestedModes	TRANSFERENCIA.petición (MODE-ELEMENT)
RequestModeAck	sequenceNumber	in_SQ
	response	TRANSFERENCIA.respuesta(MODE-PREF)
RequestModeReject	sequenceNumber	in_SQ
	cause	RECHAZO.petición(CAUSE)
RequestModeRelease	–	–

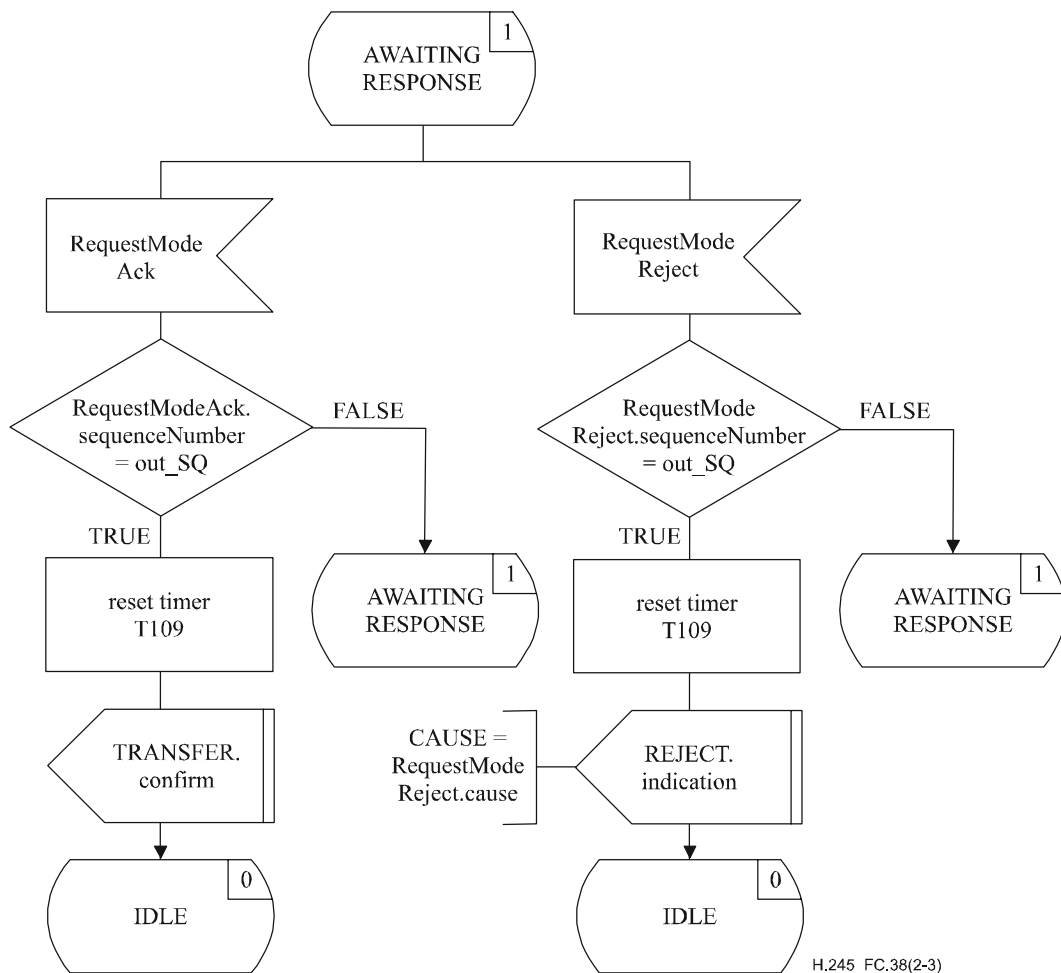
### C.9.4.3 Diagramas SDL

Los procedimientos de la MRSE de salida y de la MRSE de entrada se expresan en forma de diagrama SDL en las figuras C.38 y C.39 respectivamente.



H.245\_FC.38(1-3)

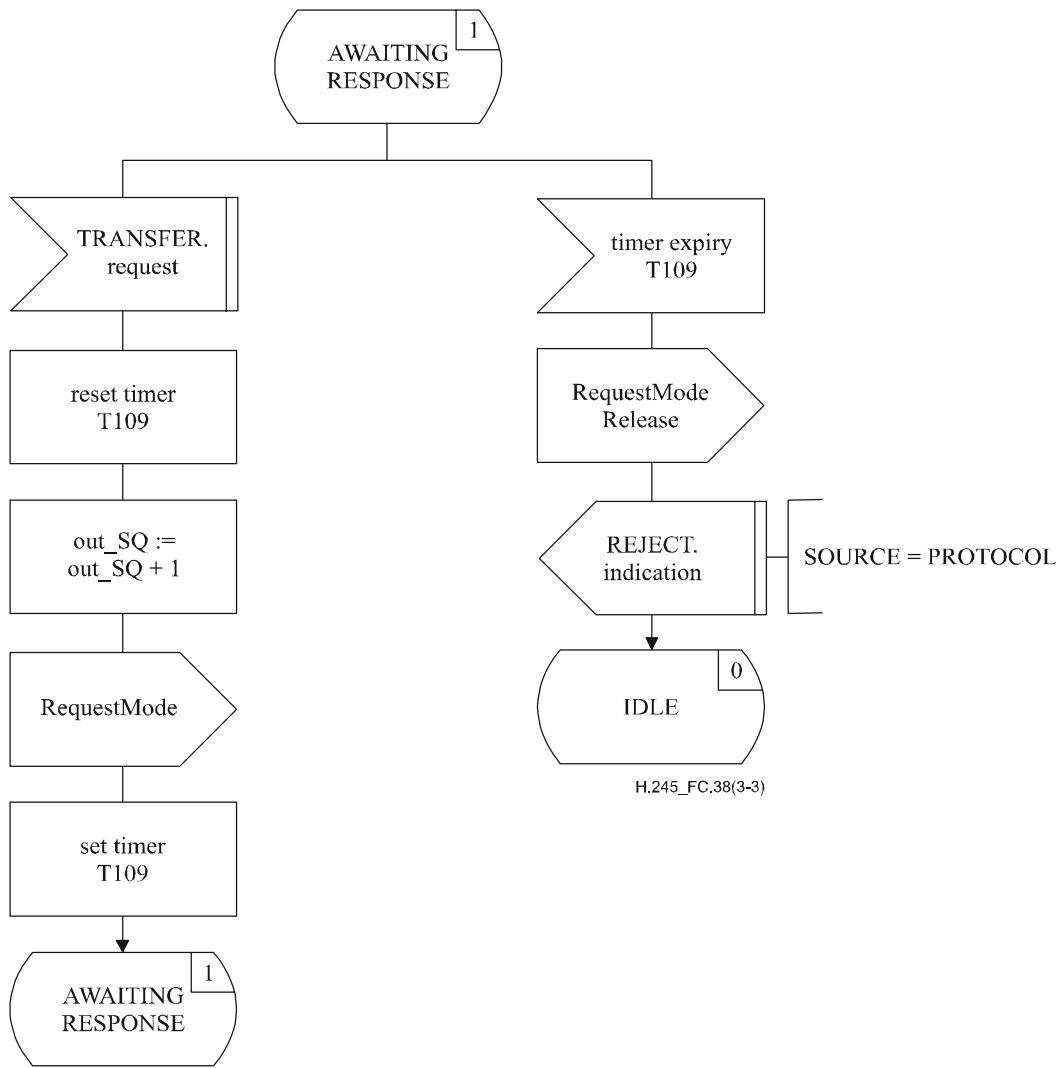
Figura C.38/H.245 – Diagrama SDL de la MRSE de salida (hoja 1 de 3)



H.245\_FC.38(2-3)

**Figura C.38/H.245 – Diagrama SDL de la MRSE de salida (hoja 2 de 3)**





**Figura C.38/H.245 – Diagrama SDL de la MRSE de salida (hoja 3 de 3)**

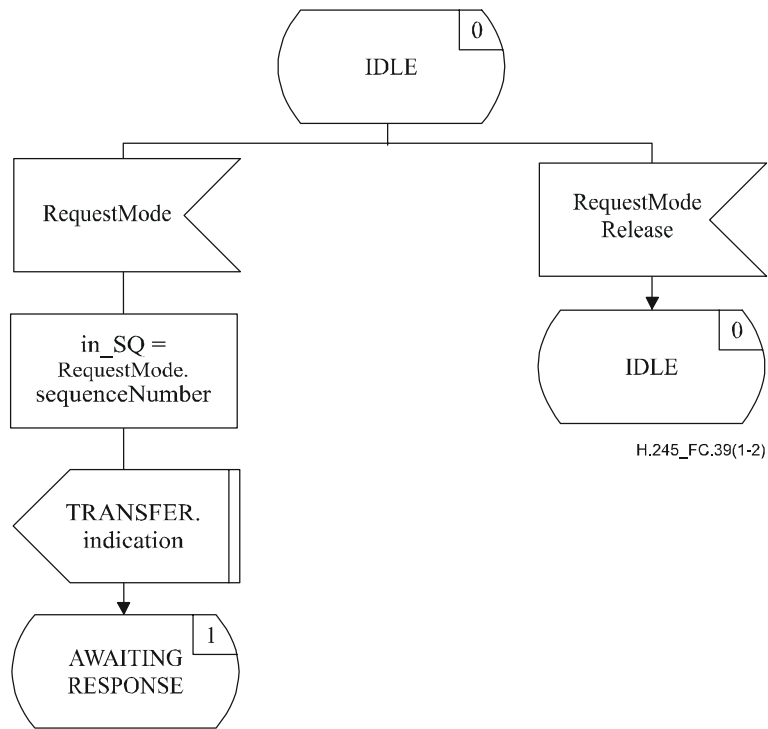


Figura C.39/H.245 – Diagrama SDL de la MRSE de entrada (hoja 1 de 2)

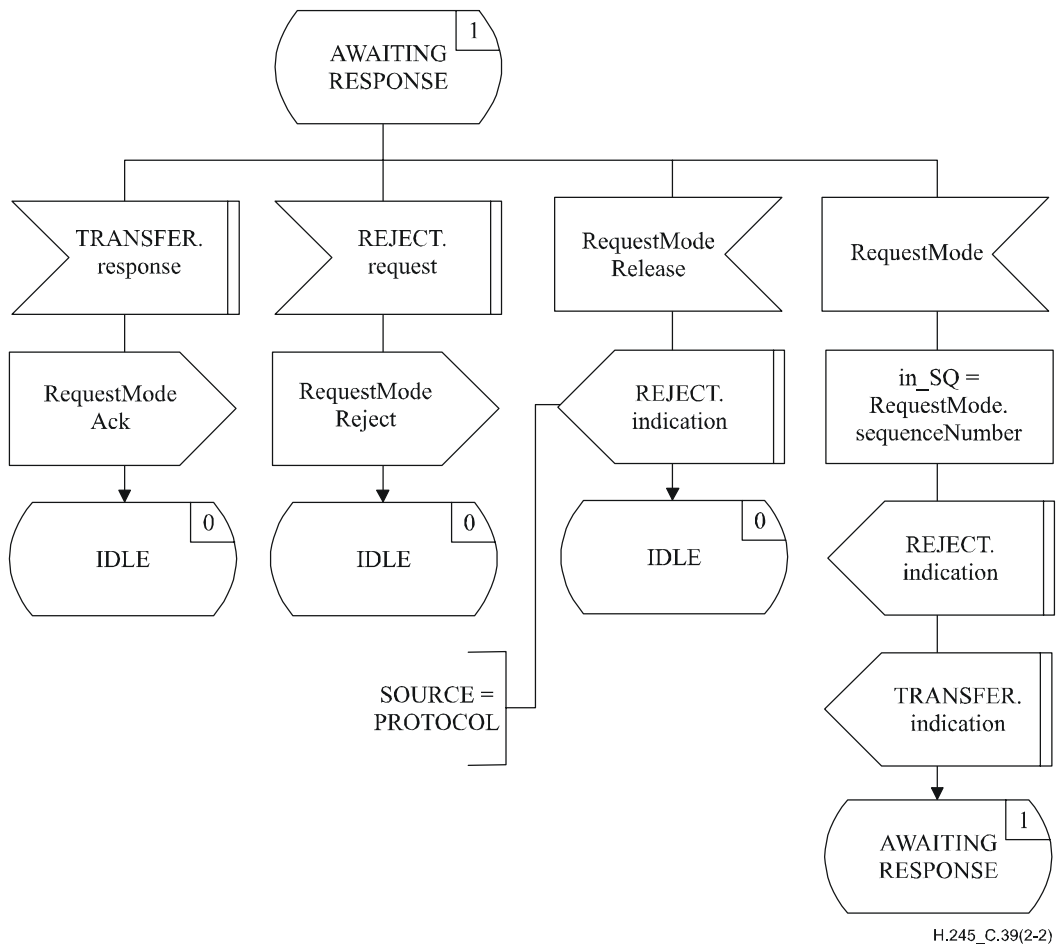


Figura C.39/H.245 – Diagrama SDL de la MRSE de entrada (hoja 2 de 2)

## C.10 Procedimientos de retardo de ida y vuelta

### C.10.1 Introducción

Se describen aquí procedimientos para la determinación del retardo de ida y vuelta entre dos terminales en comunicación. Esta función permite asimismo a un usuario H.245 determinar si continúa activa la entidad de protocolo H.245 par.

La función aquí descrita se conoce como entidad de señalización del retardo de ida y vuelta (RTDSE, *round-trip delay signalling entity*). Se especifican los procedimientos mediante primitivas y estados en la interfaz entre la RTDSE y el usuario de la RTDSE. En cada terminal hay un ejemplar de la RTDSE. Cualquier terminal puede realizar la determinación del retardo de ida y vuelta.

El siguiente texto presenta una visión general del funcionamiento de protocolo de RTDSE. En caso de cualquier discrepancia entre éste y la especificación formal, prevalecerá la especificación formal.

#### C.10.1.1 Visión general del protocolo – RTDSE

Se inicia un procedimiento de determinación del retardo de ida y vuelta cuando la primitiva TRANSFERENCIA.petición es emitida por el usuario de la RTDSE. Se envía un mensaje RoundTripDelayRequest a la RTDSE par y se arranca el temporizador T105. Si se recibe un mensaje RoundTripDelayResponse en respuesta al mensaje RoundTripDelayRequest, se detiene el temporizador y se informa al usuario con la primitiva TRANSFERENCIA.confirmación del retardo de ida y vuelta, que es el valor del temporizador T105.

Si se recibe en cualquier momento un mensaje RoundTripDelayRequest procedente de la RTDSE par, se envía inmediatamente a la RTDSE par un mensaje RoundTripDelayResponse.

Si expira el temporizador T105, se informa entonces al usuario RTDSE con la primitiva EXPIRACIÓN.indicación.

Sólo se acepta el mensaje RoundTripDelayResponse que es en respuesta al mensaje RoundTripDelayRequest más reciente. Los mensajes en respuesta a mensajes RoundTripDelayRequest anteriores son ignorados.

Un nuevo procedimiento de determinación del retardo de ida y vuelta puede ser iniciado con la primitiva TRANSFERENCIA.petición por el usuario de la RTDSE antes de que se haya recibido un mensaje RoundTripDelayResponse.

### C.10.2 Comunicación entre la RTDSE y el usuario de RTDSE

#### C.10.2.1 Primitivas entre la RTDSE y el usuario de RTDSE

Se realiza la comunicación entre la RTDSE y el usuario de la RTDSE mediante las primitivas del cuadro C.37. Estas primitivas tienen como finalidad la definición de procedimientos en la RTDSE y no se pretende que especifiquen o limiten la implementación.

**Cuadro C.37/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
TRANSFERENCIA	– (nota 1)	no definida (nota 2)	no definida	DELAY
EXPIRACIÓN	no definida	–	no definida	no definida

NOTA 1 – "–" indica que no hay parámetros.  
NOTA 2 – "no definida" indica que esta primitiva no está definida.

### C.10.2.2 Definición de las primitivas

La definición de estas primitivas es la siguiente:

- a) Se utiliza la primitiva TRANSFERENCIA para solicitar la determinación del retardo de ida y vuelta y notificar este retardo.
- b) La primitiva EXPIRACIÓN indica que no se ha recibido respuesta desde el terminal par.

### C.10.2.3 Definición de los parámetros

La definición de los parámetros de las primitivas del cuadro C.37 es la siguiente:

- a) El parámetro DELAY proporciona el valor medido del retardo de ida y vuelta.

### C.10.2.4 Estados de la RTDSE

Para especificar la secuencia permitida de primitivas entre la RTDSE y el usuario de la RTDSE se utilizan los siguientes estados:

Estado 0: REPOSO

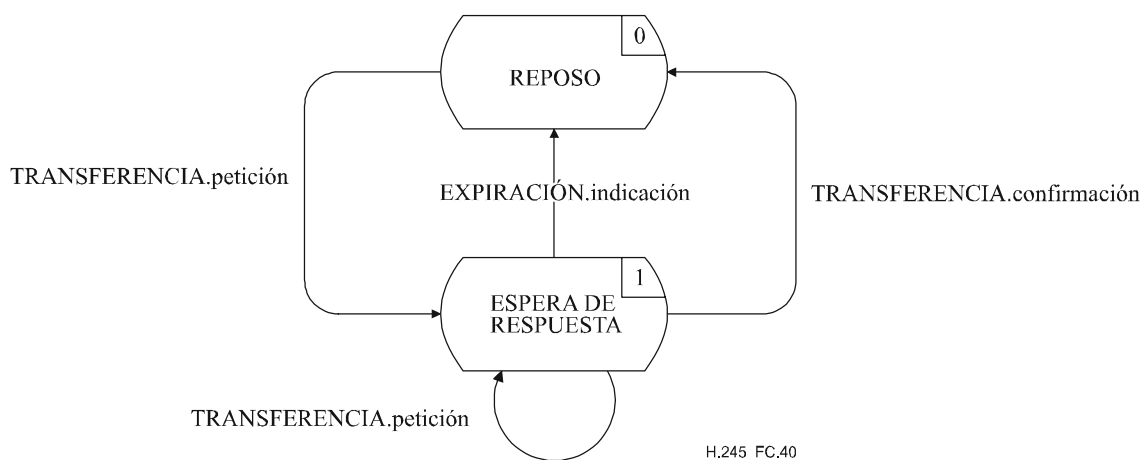
No hay ninguna transferencia de RTDSE en curso.

Estado 1: ESPERA DE RESPUESTA

El usuario de la RTDSE ha solicitado la medición del retardo de ida y vuelta. Se espera una respuesta de la RTDSE par.

### C.10.2.5 Diagramas de transición de estados

Se define aquí la secuencia permitida de primitivas entre la RTDSE y el usuario de la RTDSE. En la figura C.40 se muestra la secuencia permitida.



**Figura C.40/H.245 – Diagrama de transición de estados para la secuencia de primitivas en la RTDSE**

## C.10.3 Comunicación entre RTDSE pares

### C.10.3.1 Mensajes

El cuadro C.38 muestra los mensajes y campos de la RTDSE, definidos en el anexo A, que son apropiados para el protocolo de la RTDSE.

**Cuadro C.38/H.245 – Nombres de mensajes y campos de la RTDSE**

Función	Mensaje	Campo
transferencia	RoundTripDelayRequest	sequenceNumber
	RoundTripDelayResponse	sequenceNumber

### C.10.3.2 Variables de estado de la RTDSE

Se define la siguiente variable de estado de la RTDSE:

out\_SQ

Se utiliza esta variable de estado para indicar el mensaje RoundTripDelayRequest más reciente. Antes de la transmisión del mensaje RoundTripDelayRequest se incrementa su valor en uno y se pone en correspondencia con el campo sequenceNumber del mensaje RoundTripDelayRequest. Para out\_SQ se utiliza aritmética módulo 256.

### C.10.3.3 Temporizadores de la RTDSE

Para la RTDSE de salida se especifica el siguiente temporizador:

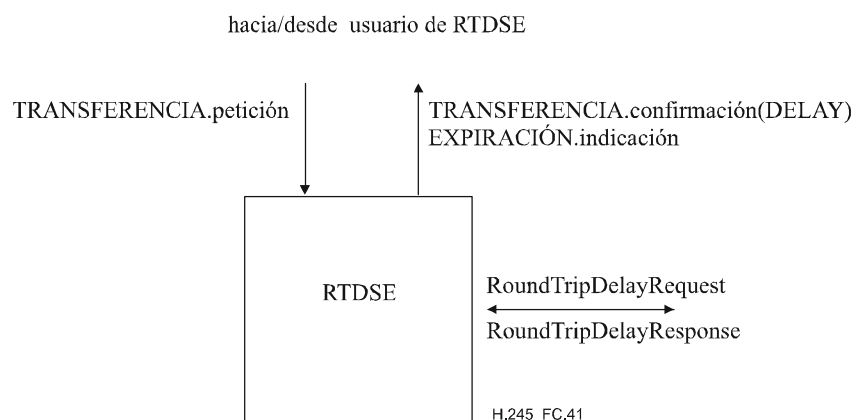
T105

Se utiliza este temporizador en el estado ESPERA DE RESPUESTA. Especifica el máximo periodo de tiempo durante el que no se pueden recibir mensajes RoundTripDelayResponse.

## C.10.4 Procedimientos de RTDSE

### C.10.4.1 Introducción

En la figura C.41 se resumen las primitivas de la RTDSE y sus parámetros, así como los mensajes.



**Figura C.41/H.245 – Primitivas y mensajes en la RTDSE**

### C.10.4.2 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas de SDL los valores de los parámetros de las primitivas indicación y confirmación, serán los indicados en el cuadro C.39.

**Cuadro C.39/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio
TRANSFERENCIA.confirmación	DELAY	valor inicial del temporizador T105 menos el valor del temporizador T105
EXPIRACIÓN.indicación	–	–

NOTA – Los contadores están diseñados para que cuenten en sentido decreciente hasta cero. El parámetro DELAY indica el tiempo durante el cual el temporizador ha estado en marcha, por lo que su valor es la diferencia entre el valor al que fue fijado inicialmente y el valor retenido del temporizador.

**C.10.4.3 Valores supletorios del campo de los mensajes**

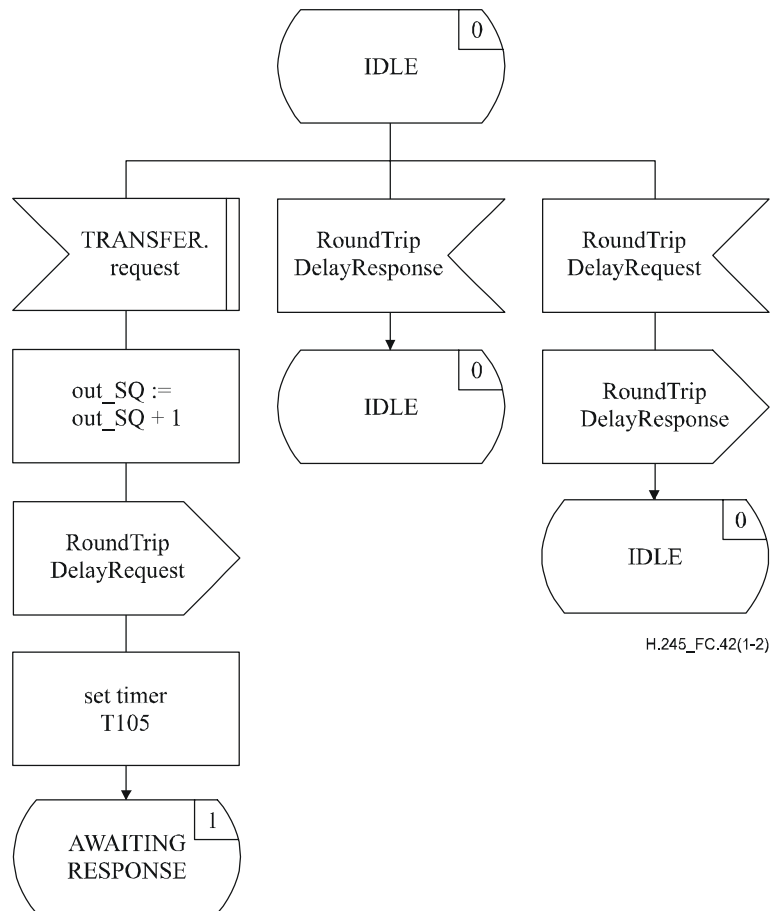
Cuando no se indiquen explícitamente en los diagramas de SDL, los valores de los campos de los mensajes serán los indicados en el cuadro C.40.

**Cuadro C.40/H.245 – Valores supletorios de los campos de los mensajes**

Mensaje	Campo	Valor supletorio
RoundTripDelayRequest	sequenceNumber	out_SQ
RoundTripDelayResponse	sequenceNumber	RoundTripDelayRequest.sequenceNumber

**C.10.4.4 Diagramas de SDL**

En la figura C.42 se representan, mediante diagramas de SDL, los procedimientos en la RTDSE.



**Figura C.42/H.245 – Diagrama SDL de la RTDSE (hoja 1 de 2)**

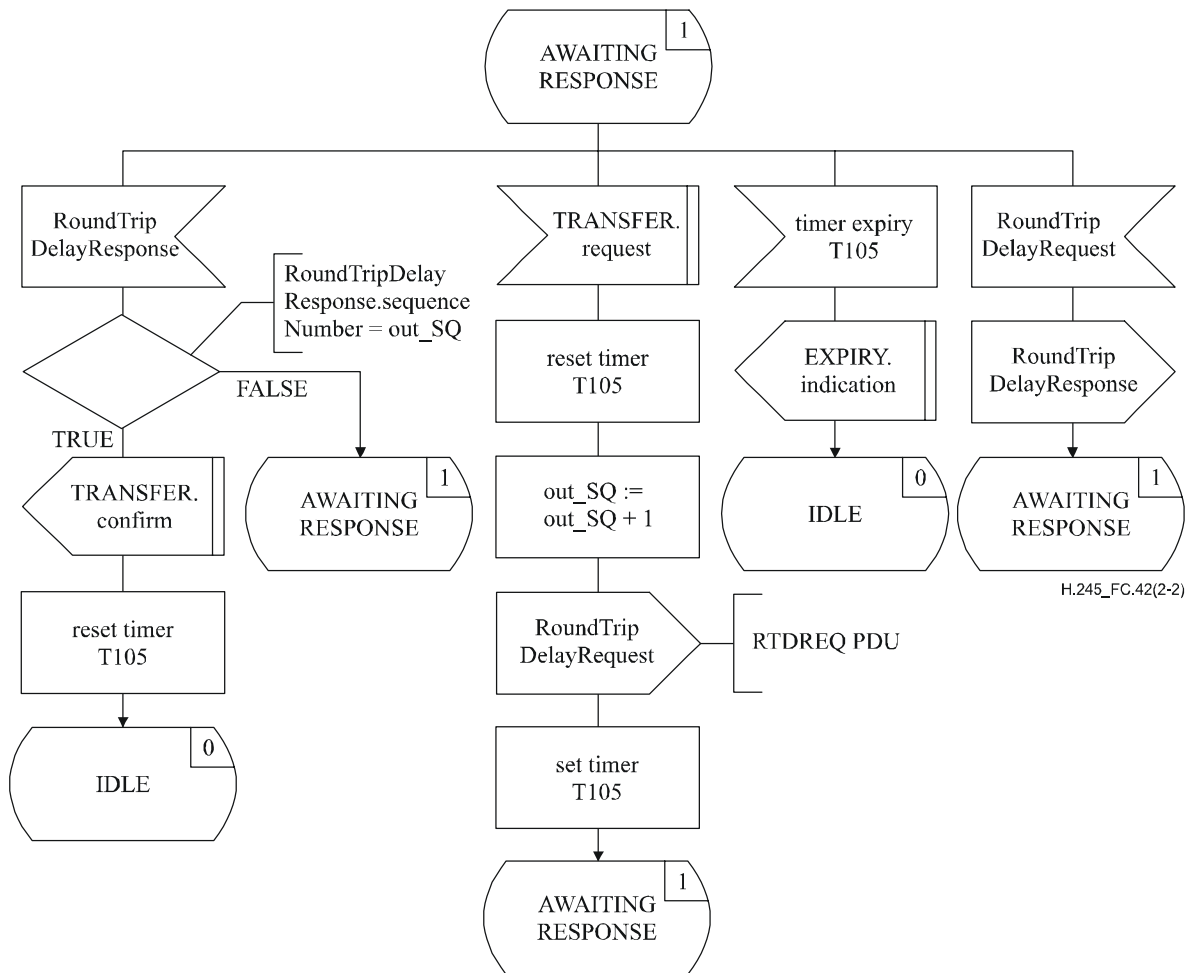


Figura C.42/H.245 – Diagrama SDL de la RTDSE (hoja 2 de 2)

## C.11 Procedimientos de bucle de mantenimiento

### C.11.1 Introducción

El protocolo aquí especificado proporciona un funcionamiento fiable de los bucles de mantenimiento mediante el empleo de procedimientos con acuse de recibo.

Este protocolo se designa aquí como la entidad de señalización de bucle de mantenimiento (MLSE, *maintenance loop signalling entity*). Los procedimientos se especifican mediante primitivas en las interfaces entre la MLSE y el usuario de la MLSE, y estados de la MLSE. La información de protocolo se transfiere a la MLSE par mediante mensajes apropiados definidos en el anexo A.

Hay una MLSE de salida y una MLSE de entrada. En el lado de salida y en el lado de entrada hay un ejemplar de la MLSE para cada canal lógico bidireccional y una para el bucle de sistema. La única conexión entre una MLSE de entrada y una MLSE de salida en un lado es mediante primitivas hacia y desde el usuario de la MLSE. Se informa de las condiciones de error.

El terminal que contiene la MLSE de entrada devolverá en bucle los datos apropiados mientras está en el estado BUCLE CONECTADO y no en cualquier otro momento. El terminal que contiene la MLSE de salida podrá recibir datos devueltos en bucle cualquiera que sea el estado en que se encuentre, pero, si se encuentra en el estado BUCLE CONECTADO, deberá recibir solamente datos devueltos en bucle.

NOTA – El mensaje MaintenanceLoopOffCommand se aplica a todas las MLSE. Se utiliza siempre para desconectar todos los bucles de mantenimiento.

El siguiente texto proporciona una visión general del funcionamiento del protocolo MLSE. En caso de discrepancia entre esta exposición y la especificación formal, prevalecerá la especificación formal.

#### **C.11.1.1 Visión general del protocolo – Salida**

Se inicia el establecimiento de un bucle de mantenimiento cuando el usuario emite una primitiva BUCLE.petición en la MLSE de salida. Se envía un mensaje MaintenanceLoopRequest a la MLSE de entrada par, y se arranca el temporizador T102. Si en respuesta al mensaje MaintenanceLoopRequest se recibe un mensaje MaintenanceLoopAck, se detiene el temporizador T102 y se informa al usuario, mediante la primitiva BUCLE.confirmación que el bucle de mantenimiento se ha establecido correctamente. Por el contrario, si en respuesta al mensaje MaintenanceLoopRequest se recibe un mensaje MaintenanceLoopReject, se detiene el temporizador T102 y se informa al usuario, mediante la primitiva LIBERACIÓN.indicación, que la MLSE par ha rehusado el establecimiento del bucle de mantenimiento.

Si el temporizador T102 expira dentro de este periodo, se informará al usuario mediante la primitiva LIBERACIÓN.indicación y se enviará un mensaje MaintenanceLoopOffCommand a la MLSE de entrada par. Este mensaje anulará todos los bucles de mantenimiento y no solamente el referente a la MLSE en cuestión.

Un bucle de mantenimiento que ha sido establecido correctamente puede anularse cuando el usuario en la MLSE de salida emite la primitiva LIBERACIÓN.petición. En este caso se envía un mensaje MaintenanceLoopOffCommand a la MLSE de entrada par.

Antes de que se haya recibido el mensaje MaintenanceLoopAck o MaintenanceLoopReject en respuesta al anterior mensaje MaintenanceLoopRequest, el usuario en la MLSE de salida puede anular el bucle de mantenimiento mediante la primitiva LIBERACIÓN.petición.

#### **C.11.1.2 Visión general del protocolo – Entrada**

Cuando se recibe un mensaje MaintenanceLoopRequest en la MLSE de entrada, se informa al usuario de la petición de establecimiento de un bucle de mantenimiento mediante la primitiva BUCLE.indicación. El usuario de la MLSE de entrada señala la aceptación de la petición de establecimiento del bucle de mantenimiento emitiendo una primitiva BUCLE.respuesta, en cuyo caso se envía un mensaje MaintenanceLoopAck a la MLSE de salida par. Con esto queda establecido el bucle de mantenimiento. El usuario de la MLSE de entrada señala el rechazo de la petición de establecimiento de bucle de mantenimiento emitiendo la primitiva LIBERACIÓN.petición, en cuyo caso se envía un mensaje MaintenanceLoopReject a la MLSE de salida par.

Un bucle de mantenimiento que ha sido establecido correctamente puede anularse cuando el mensaje MaintenanceLoopOffCommand se recibe en la MLSE de entrada. En este caso se informa al usuario de la MLSE de entrada mediante la primitiva LIBERACIÓN.indicación.

### **C.11.2 Comunicación entre la MLSE y el usuario de MLSE**

#### **C.11.2.1 Primitivas entre la MLSE y el usuario de MLSE**

La comunicación entre la MLSE y el usuario de la MLSE se efectúa mediante las primitivas indicadas en el cuadro C.41.



**Cuadro C.41/H.245 – Primitivas y parámetros**

Nombre genérico	Tipo			
	petición	indicación	respuesta	confirmación
BUCLE	LOOP_TYPE	LOOP_TYPE	– (nota 1)	–
LIBERACIÓN	CAUSE	SOURCE CAUSE	no definida (nota 2)	no definida
ERROR	no definida	ERRCODE	no definida	no definida
NOTA 1 – "-" significa que no hay parámetros. NOTA 2 – "no definida" significa que esta primitiva no existe.				

### C.11.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- a) Las primitivas BUCLE se utilizan para establecer un bucle de mantenimiento.
- b) Las primitivas LIBERACIÓN se utilizan para anular un bucle de mantenimiento.
- c) La primitiva ERROR informa de errores de la MLSE a una entidad de gestión.

### C.11.2.3 Definiciones de los parámetros

Las definiciones de los parámetros indicados en el cuadro C.41 son:

- a) El parámetro LOOP\_TYPE especifica los parámetros asociados con el bucle de mantenimiento. Puede tomar los valores de "SYSTEM", "MEDIA" y "LOGICAL\_CHANNEL". Este parámetro y el número de canal lógico determinan el valor del campo tipo del mensaje MaintenanceLoopRequest, que se transporta entonces transparentemente al usuario de la MLSE par.
- b) El parámetro SOURCE indica al usuario MLSE la fuente de la liberación del bucle de mantenimiento. El parámetro SOURCE tiene el valor "USER" o "MLSE", que indican el usuario de la MLSE o la MLSE. Este último puede producirse en el caso de un error de protocolo.
- c) El parámetro CAUSE indica el motivo por el cual el usuario de la MLSE par rechazó una petición de establecimiento de un bucle de mantenimiento. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "MLSE".
- d) El parámetro ERRCODE indica el tipo de error de la MLSE. En el cuadro C.45 se indican los valores permitidos del parámetro ERRCODE.

### C.11.2.4 Estados de la MLSE

Los siguientes estados se utilizan para especificar la secuencia permitida de primitivas entre la MLSE y el usuario de la MLSE, y el intercambio de mensajes entre MLSE pares. Los estados se especifican separadamente para la MLSE de salida y la MLSE de entrada. Los estados para la MLSE de salida son:

Estado 0: BUCLE NO CONECTADO

No hay bucle de mantenimiento.

Estado 1: ESPERA DE RESPUESTA

La MLSE de salida está en espera del establecimiento de un bucle de mantenimiento con una MLSE de entrada par.

## Estado 2: BUCLE CONECTADO

El bucle de mantenimiento entre las entidades MLSE pares ha sido establecido. Todos los datos recibidos por el canal apropiado deben ser devueltos en bucle.

Los estados para una MLSE de entrada son:

## Estado 0: BUCLE NO CONECTADO

No hay bucle de mantenimiento.

## Estado 1: ESPERA DE RESPUESTA

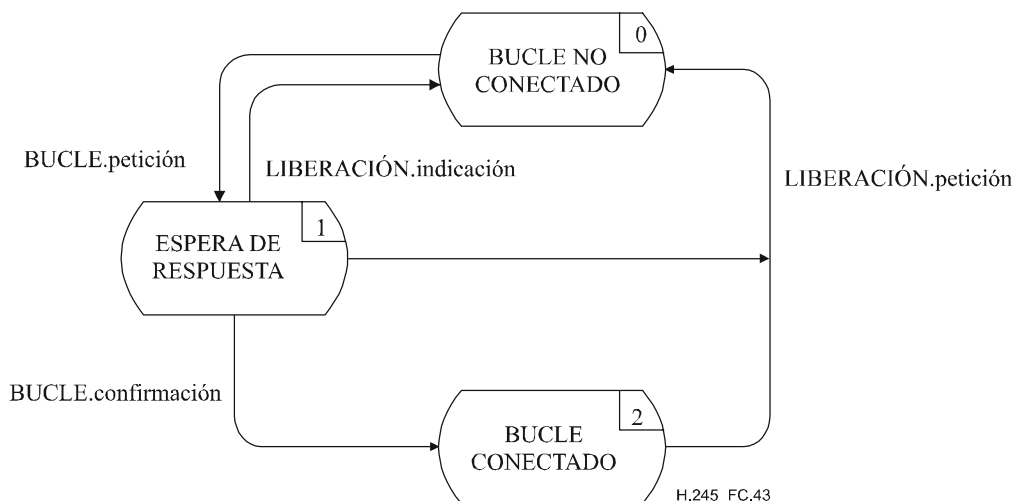
La MLSE de entrada está en espera del establecimiento de un bucle de mantenimiento con una MLSE de salida par. Los datos apropiados no deben ser devueltos en bucle.

## Estado 2: BUCLE CONECTADO

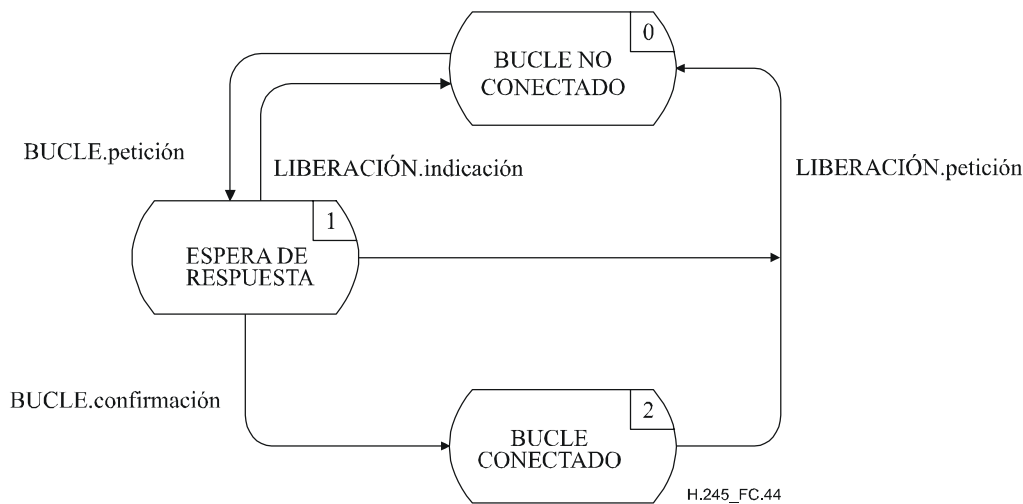
El bucle de mantenimiento entre las entidades MLSE pares ha sido establecido. Todos los datos recibidos por el canal apropiado deben ser devueltos en bucle.

### C.11.2.5 Diagramas de transición de estados

Se define aquí la secuencia permitida de primitivas entre la MLSE y el usuario de la MLSE. La secuencia permitida de primitivas se refiere a los estados de la MLSE tal como son percibidos por el usuario de la MLSE. Los estados se especifican separadamente para la MLSE de salida y la MLSE de entrada, como se indica en las figuras C.43 y C.44, respectivamente.



**Figura C.43/H.245 – Diagrama de transición de estados para la secuencia de primitivas en la MLSE de salida**



**Figura C.44/H.245 – Diagrama de transición de estados para la secuencia de primitivas en la MLSE de entrada**

### C.11.3 Comunicación entre MLSE pares

#### C.11.3.1 Mensajes de las MLSE

El cuadro C.42 muestra los mensajes y campos, definidos en el anexo A, que son apropiados para el protocolo MLSE.

**Cuadro C.42/H.245 – Nombres y campos de los mensajes de la MLSE**

Función	Mensaje	Sentido	Campo
establecimiento	MaintenanceLoopRequest	O → I (nota)	type
	MaintenanceLoopAck	O ← I	type
	MaintenanceLoopReject	O ← I	type cause
liberación	MaintenanceLoopOffCommand	O → I	–

NOTA – Sentido: O – saliente; I – entrante.

#### C.11.3.2 Variables de estado de la MLSE

Se define la siguiente variable de estado en la MLSE de salida:

out\_MLN

Esta variable de estado distingue entre las MLSE de salida. Se inicializa en el momento de la inicialización de la MLSE de salida. El valor de out\_MLN se utiliza para fijar el campo type de los mensajes MaintenanceLoopRequest enviados desde una MLSE de salida.

Se define la siguiente variable de estado en la MLSE de entrada:

in\_MLN

Esta variable de estado distingue entre las MLSE de entrada. Se inicializa en el momento de la inicialización de la MLSE de entrada. En el caso de mensajes MaintenanceLoopRequest recibidos en una MLSE de entrada, el valor del campo type del mensaje concuerda con el valor de in\_MLN.

in\_TYPE

Esta variable de estado contiene el valor de LOOP\_TYPE cuando se recibe MaintenanceLoopRequest. Esta variable de estado ayuda a fijar el valor del campo type del mensaje MaintenanceLoopAck.

### C.11.3.3 Temporizadores de la MLSE

Para la MLSE de salida se especifica el siguiente temporizador:

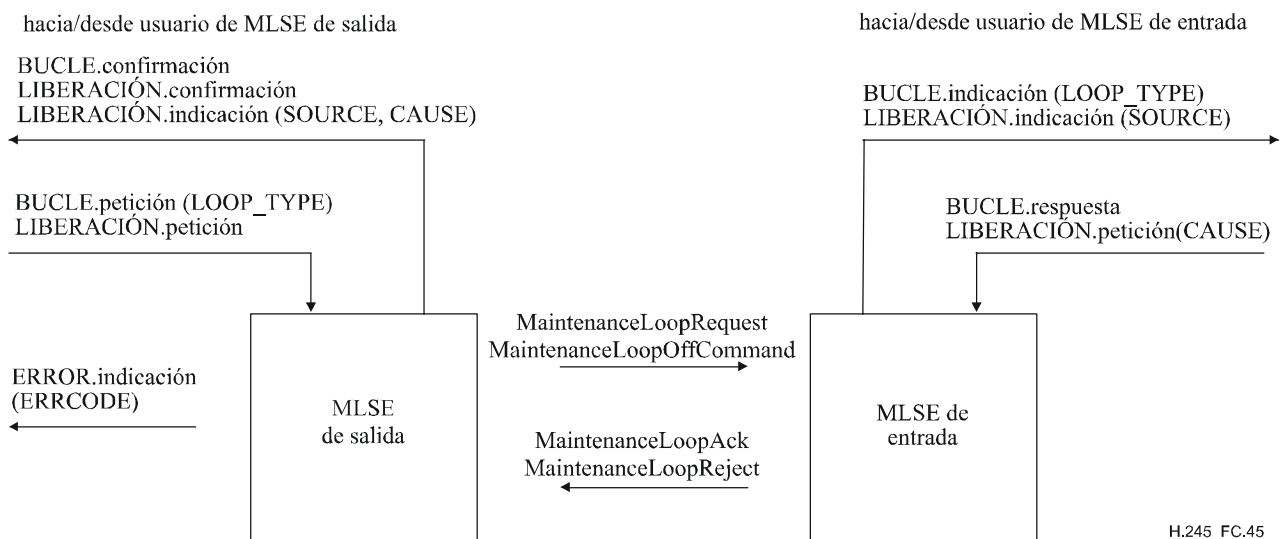
T102

Este temporizador se utiliza durante el estado ESPERA DE RESPUESTA. Determina el máximo periodo de tiempo durante el cual no puede recibirse el mensaje MaintenanceLoopAck o MaintenanceLoopReject.

### C.11.4 Procedimientos de la MLSE

#### C.11.4.1 Introducción

En la figura C.45 se resumen las primitivas y sus parámetros, así como los mensajes, para la MLSE de salida y la MLSE de entrada.



H.245\_FC.45

**Figura C.45/H.245 – Primitivas y mensajes en la entidad de señalización de bucle de mantenimiento**

#### C.11.4.2 Valores supletorios de los parámetros de las primitivas

Cuando no se indiquen explícitamente en los diagramas SDL, los parámetros de las primitivas indicación y confirmación tomarán los valores del cuadro C.43.

**Cuadro C.43/H.245 – Valores supletorios de los parámetros de las primitivas**

Primitiva	Parámetro	Valor supletorio (Nota)
BUCLE.indicación	LOOP_TYPE	MaintenanceLoopRequest.type
LIBERACIÓN.indicación	SOURCE CAUSE	USER MaintenanceLoopReject.cause
NOTA – Un parámetro de una primitiva se codificará como nulo si un campo de mensaje indicado no está presente en el mensaje.		

**C.11.4.3 Valores supletorios de los campos de los mensajes**

Cuando no se indiquen explícitamente en los diagramas SDL, los campos de los mensajes tomarán los valores del cuadro C.44.

**Cuadro C.44/H.245 – Valores supletorios de los parámetros de las primitivas**

Mensaje	Campo	Valor supletorio (Nota 1)
MaintenanceLoopRequest	type	BUCLE.petición(LOOP_TYPE) y out_MLN (nota 2)
MaintenanceLoopAck	type	in_LOOP e in_MLN (nota 3)
MaintenanceLoopReject	type cause	in_LOOP e in_MLN (nota 3) LIBERACIÓN.petición(CAUSE)
MaintenanceLoopOffCommand	–	–
NOTA 1 – Un campo de mensaje no se codificará si el correspondiente parámetro de la primitiva es nulo o no está presente.		
NOTA 2 – El valor del campo type se deriva del parámetro LOOP_TYPE y del número de canal lógico.		
NOTA 3 – El valor del campo type se deriva de las variables de estado in_LOOP e in_MLN.		

**C.11.4.4 Valores del parámetro ERRCODE**

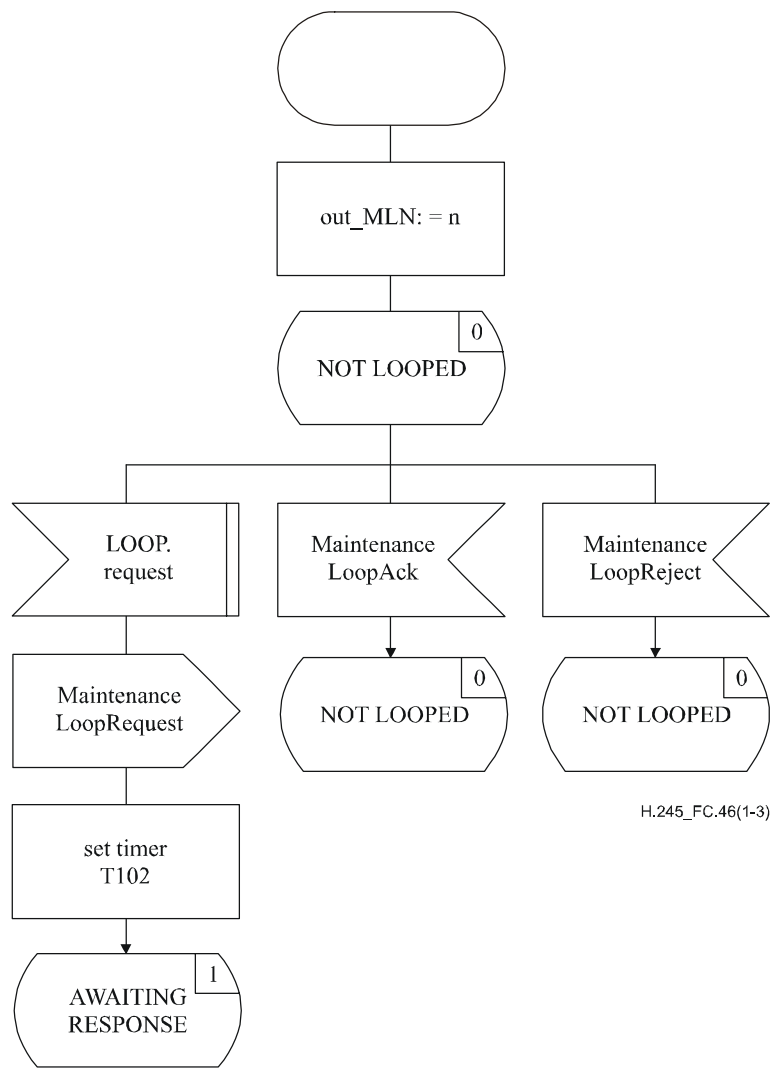
El parámetro ERRCODE de la primitiva ERROR.indicación indica una condición de error dada. En el cuadro C.45 se muestran los valores que el parámetro ERRCODE puede tomar en la MLSE de salida. No hay primitiva ERROR.indicación asociada con la MLSE de llegada.

**Cuadro C.45/H.245 – Valores del parámetro ERRCODE en la MLSE de salida**

Tipo de error	Código de error	Condición de error	Estado
mensaje no apropiado	A	MaintenanceLoopAck	BUCLE CONECTADO
no hay respuesta de la MLSE par	B	expiración del temporizador T102	ESPERA DE RESPUESTA

**C.11.4.5 Diagramas SDL**

Los procedimientos de la MLSE de salida y de la MLSE de entrada se expresan en forma de diagramas SDL en las figuras C.46 y C.47, respectivamente.



**Figura C.46/H.245 – Diagrama SDL de la MLSE de salida (hoja 1 de 3)**

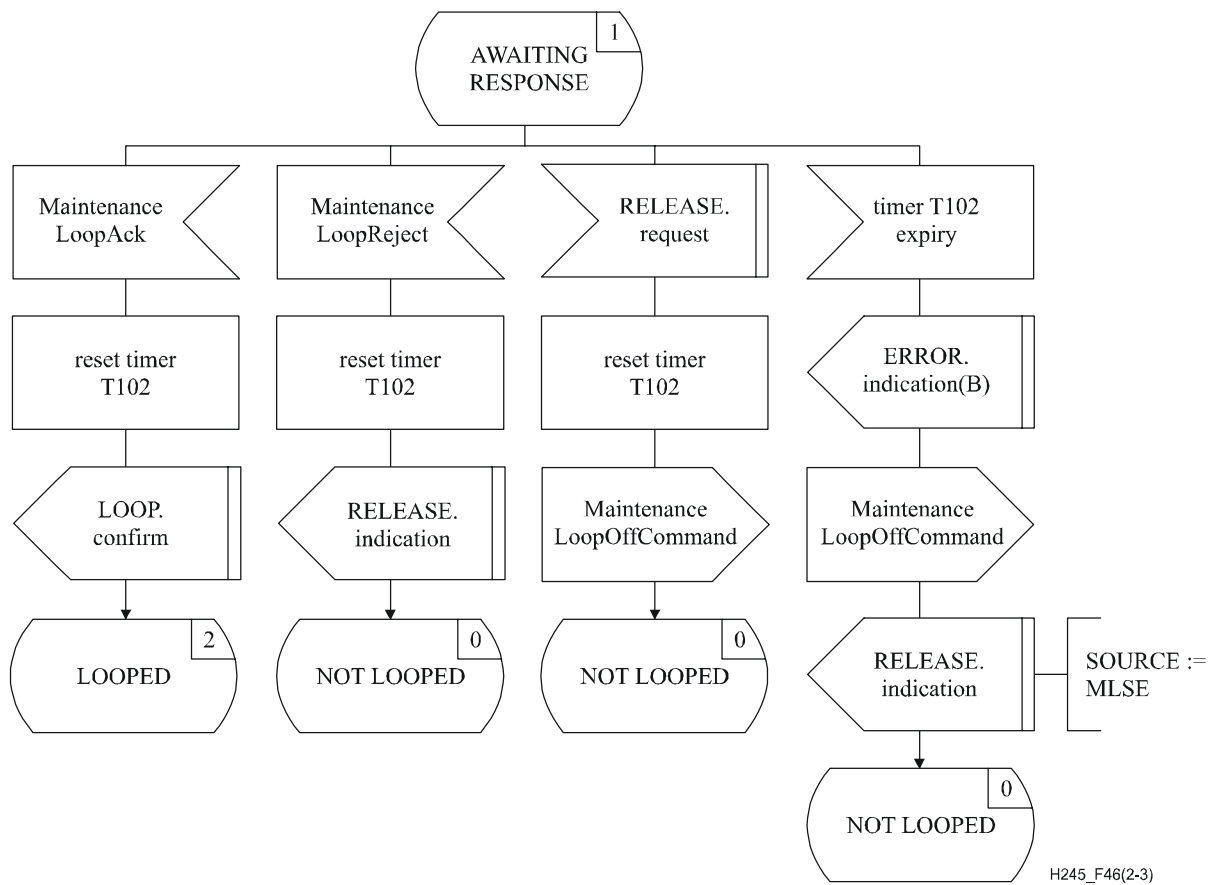


Figura C.46/H.245 – Diagrama SDL de la MLSE de salida (hoja 2 de 3)

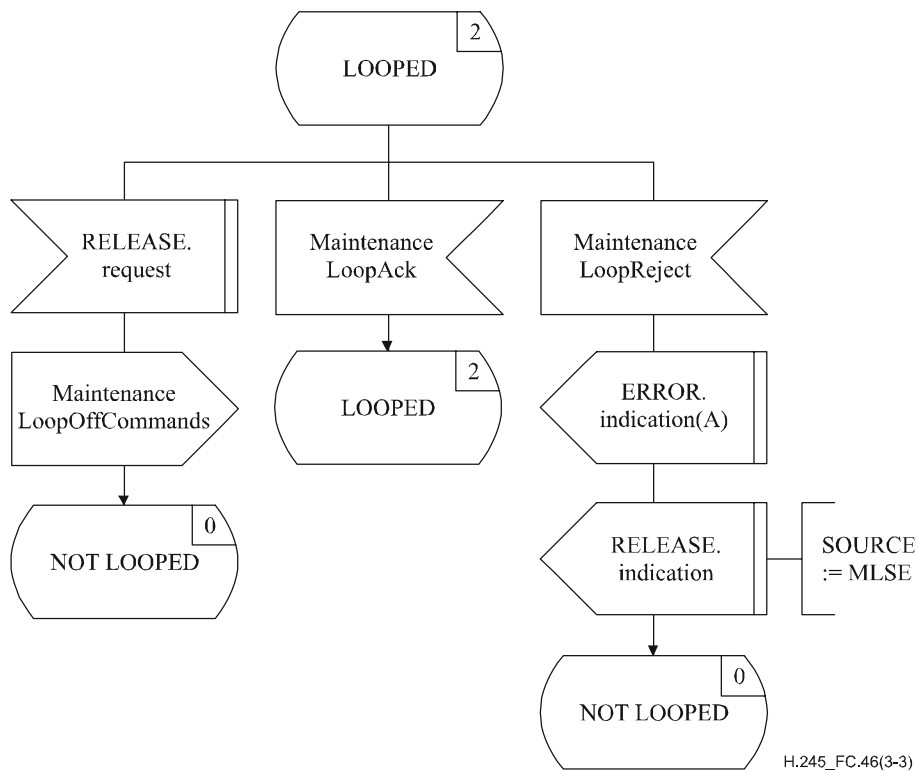
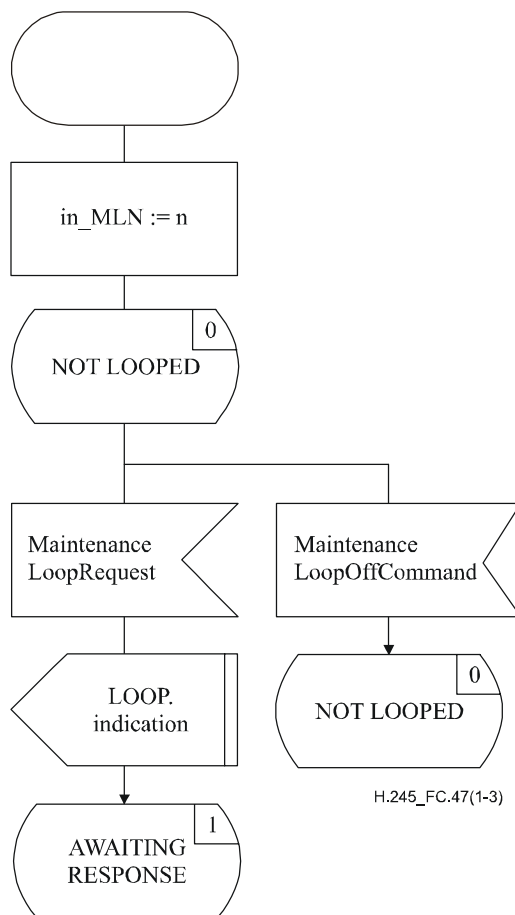
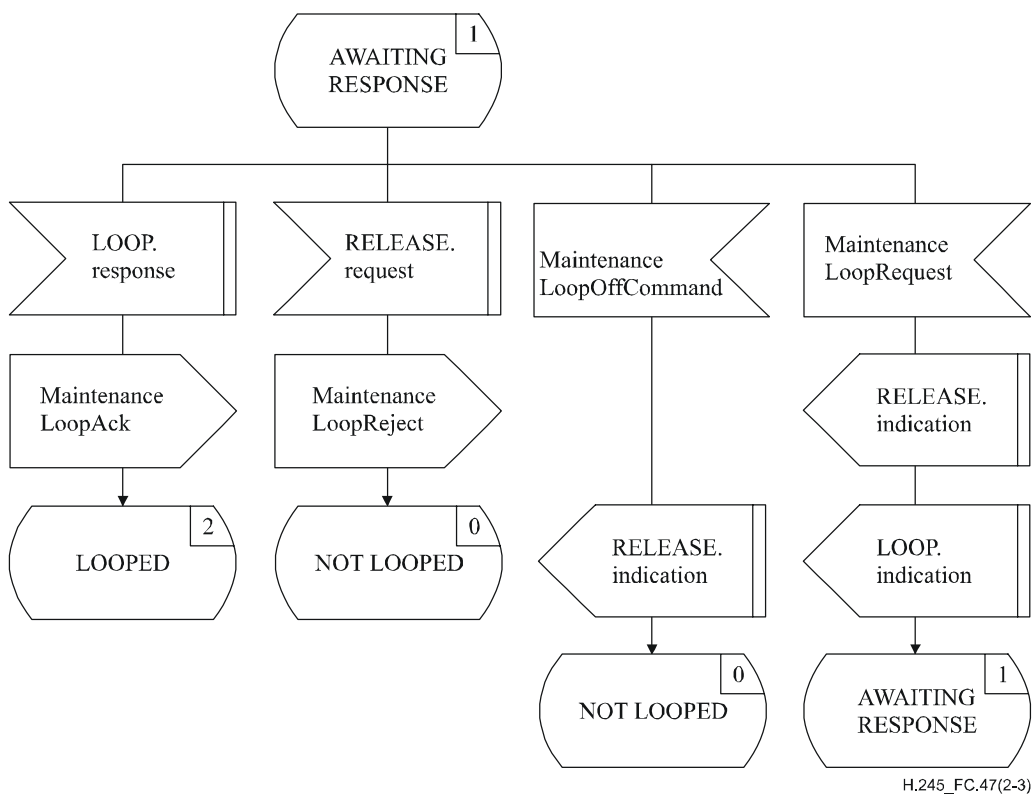


Figura C.46/H.245 – Diagrama SDL de la MLSE de salida (hoja 3 de 3)

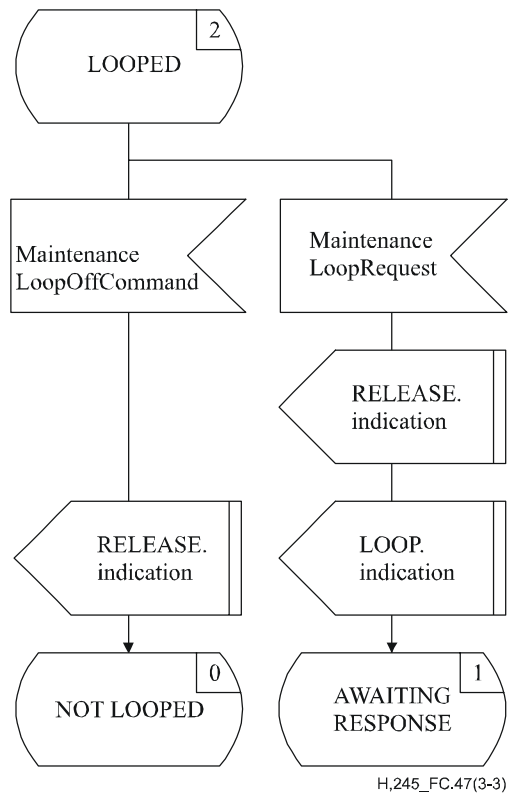


**Figura C.47/H.245 – Diagrama SDL de la MLSE de entrada (hoja 1 de 3)**



**Figura C.47/H.245 – Diagrama SDL de la MLSE de entrada (hoja 2 de 3)**





**Figura C.47/H.245 – Diagrama SDL de la MLSE de entrada (hoja 3 de 3)**

## Anexo D

### Asignaciones de identificadores de objetos

El cuadro D.1 indica los identificadores de objetos definidos para su uso en esta Recomendación.

**Cuadro D.1/H.245**

<b>Valor de identificador de objeto</b>	<b>Descripción</b>
{itu-t (0) recommendation (0) h (8) 245 version (0) 1}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. Este valor indica la primera versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 2}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la segunda versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 3}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la tercera versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 4}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la cuarta versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 5}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la quinta versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 6}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la sexta versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 7}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la séptima versión de esta Recomendación.

**Cuadro D.1/H.245**

<b>Valor de identificador de objeto</b>	<b>Descripción</b>
{itu-t (0) recommendation (0) h (8) 245 version (0) 8}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la octava versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 9}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la novena versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 10}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la décima versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 version (0) 11}	Este identificador de objeto se utiliza para indicar la versión de esta Recomendación que se está utilizando como protocolo de control de sistema multimedia. En la actualidad hay once versiones normalizadas definidas. Este valor indica la undécima versión de esta Recomendación.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) video (0) ISO/IEC 14496-2 (0)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para ISO/CEI 14496-2. Esta capacidad se define en el anexo E.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 14496-3 (0)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para ISO/CEI 14496-3. Esta capacidad se define en el anexo H.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) amr (1)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para el códec vocal a multivelocidad adaptable GSM. Esta capacidad se define en el anexo I.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) acelp (2)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para el códec vocal TIA/EIA/ANSI IS-136 ACELP. Esta capacidad se define en el anexo J.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) us1 (3)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para el códec vocal TIA/EIA/ANSI IS-136 US1. Esta capacidad se define en el anexo K.

**Cuadro D.1/H.245**

<b>Valor de identificador de objeto</b>	<b>Descripción</b>
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) is127evrc (4)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para el códec de velocidad variable mejorado TIA/EIA IS-127.  Esta capacidad se define en el anexo L.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para ISO/CEI 13818-7  Esta capacidad se define en el anexo M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) rfc3389 (6)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para la señalización del ruido de confort como se especifica en RFC 3389.  Esta capacidad se define en el anexo N.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) L-16 (7)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para codificación de audio L-16, códec lineal de velocidad variable de 16 bits/muestra como se define en IETF RFC 1890.  Esta capacidad se define en el anexo O.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) bounded-audio-stream (8)}	Este identificador de objeto se utiliza para indicar la capacidad limitada del tren de audio como capacidad genérica.  Esta capacidad se define en el anexo P.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) ISO/IEC 14496-1 (0)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para ISO/CEI 14496-1.  Esta capacidad se define en el anexo G.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) control (3) logical-channel-bit-rate-management (0)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para la gestión de velocidad binaria de canal lógico.  Esta capacidad se define en el anexo F.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para ISO/CEI 13818-7.  Esta capacidad se define en el anexo M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) itu-r bs.1196 (6)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para la Rec. UIT-R BS.1196.  Esta capacidad se define en el anexo M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) Nx64 (1)}	Este identificador de objeto se utiliza para indicar la capacidad genérica para transmisión de datos de canal libre Nx64, como se documenta en el anexo Q.

## Anexo E

### Definiciones de capacidades visuales ISO/CEI 14496-2

El cuadro E.1 define el identificador de capacidad para las capacidades ISO/CEI 14496-2 [49]. Los parámetros sólo deberán incluirse como **genericVideoCapability** dentro de la estructura **VideoCapability** y como **genericVideoMode** dentro de la estructura **VideoMode**. Los cuadros E.2 a E.6 definen los parámetros asociados a las capacidades.

Una sola ejemplificación de **profileAndLevel** de ISO/CEI 14496-2 puede soportar múltiples objetos visuales. Cada uno de esos objetos visuales se lleva como un tren elemental por su propio canal lógico separado. Puesto que es posible que múltiples entornos visuales ISO/CEI 14496-2 sean transmitidos al mismo tiempo de manera activa, y puesto que cada uno de ellos puede estar construido a partir de múltiples trenes de objetos, es necesario tener un mecanismo con el que indicar qué trenes de objetos están asociados conjuntamente en un único entorno visual ISO/CEI 14496-2. La asociación se llevará a cabo utilizando el mecanismo **forwardLogicalChannelDependency** en **OpenLogicalChannel** cuando quiera que se estén utilizando múltiples objetos visuales para el mismo entorno visual ISO/CEI 14496-2. Todos los objetos visuales asociados a un entorno visual ISO/CEI 14496-2 deberán tener el mismo perfil y nivel indicando el mismo valor de **profileAndLevel** cuando se abran canales lógicos. Si un canal lógico se abrió con una indicación de dependencia con respecto a otro canal lógico de esta manera, y el canal lógico respecto al cual se indicó su dependencia está cerrado, los restantes canales lógicos abiertos, agrupados previamente por alguna cadena de enlaces **forwardLogicalChannelDependency** deberán permanecer agrupados lógicamente como un único entorno visual ISO/CEI 14496-2.

**Cuadro E.1/H.245 – Identificador de capacidad para capacidad visual ISO/CEI 14496-2**

Nombre de la capacidad	ISO/CEI 14496-2
Clase de capacidad	Códec vídeo
Tipo de identificador de la capacidad	Standard (normalizado)
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) video (0) ISO/CEI 14496-2 (0)}
maxBitRate	El campo maxBitRate deberá estar siempre incluido.
nonCollapsingRaw	Este campo no deberá estar incluido.
transporte	Este campo no deberá estar incluido.

**Cuadro E.2/H.245 – Perfil y nivel para capacidad visual ISO/CEI 14496-2**

Nombre del parámetro	profileAndLevel
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. profileAndLevel indica la capacidad de procesar los perfiles particulares en combinación con el nivel indicado en el cuadro G-1 'FLC table for profile_and_level_indication' de ISO/CEI 14496-2.
Valor del identificador del parámetro	0
Estado del parámetro	Obligatorio
Tipo de parámetro	unsignedMax. Deberá estar comprendido entre 0..255.
Sustituye a	–

**Cuadro E.3/H.245 – Parámetro de objeto para capacidad visual ISO/CEI 14496-2**

Nombre del parámetro	objeto
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. object indica el conjunto de herramientas que han de ser utilizadas por el decodificador del tren de bits contenido en el canal lógico indicado en el cuadro 6-10 'FLC table for video_object_type indication' de ISO/CEI 14496-2.
Valor del identificador del parámetro	1
Estado del parámetro	Opcional. No deberá estar presente en el intercambio de capacidades. Deberá estar presente en la señalización de canal lógico. Puede estar presente en la petición de modo.
Tipo de parámetro	unsignedMax. Deberá estar comprendido entre 0..255.
Sustituye a	–

**Cuadro E.4/H.245 – Información de configuración de decodificador para capacidad visual ISO/CEI 14496-2**

Nombre del parámetro	decoderConfigurationInformation
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. decoderConfigurationInformation indica cómo se ha de configurar el decodificador para un determinado objeto (tren de bits) (véase la subcláusula 6.2.1 'Start Code' y las subcláusulas K.3.1 'VideoObject' a K.3.4 'FaceObject' de ISO/CEI 14496-2).
Valor del identificador del parámetro	2
Estado del parámetro	Opcional. No deberá estar presente en el intercambio de capacidades ni en la petición de modo. Puede estar presente en la señalización de canal lógico.
Tipo de parámetro	octetString
Sustituye a	–

**Cuadro E.5/H.245 – Orden de representación para capacidad visual ISO/CEI 14496-2**

Nombre del parámetro	drawingOrder
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. drawingOrder indica el orden de representación de un objeto visual dentro de una composición de objetos visuales (posiblemente superpuestos). El objeto visual que tenga el drawingOrder más bajo será el que se represente primero. Si los objetos visuales tienen el mismo drawingOrder, el objeto correspondiente al canal lógico con el número de canal lógico más bajo será el que se represente primero. Si drawingOrder no está presente durante la señalización de canal lógico, se supone que tiene el valor de 32768.
Valor del identificador del parámetro	3
Estado del parámetro	Opcional. No deberá estar presente en el intercambio de capacidades ni en la petición de modo. Puede estar presente en la señalización de canal lógico.
Tipo de parámetro	unsignedMax. Deberá estar comprendido entre 0..65535.
Sustituye a	–

**Cuadro E.6/H.245 – Tratamiento del canal de retorno visual para capacidad ISO/CEI 14496-2**

Nombre del parámetro	visualBackChannelHandle
Descripción del parámetro	Éste es un parámetro genérico Collapsing. La presencia de este parámetro indica que el transmisor recibe mensajes del canal de retorno o que el receptor envía mensajes por el canal de retorno conforme a ISO/CEI 14496-2.
Valor del identificador del parámetro	4
Estado del parámetro	Puede estar presente en el intercambio de capacidades, la señalización de canal lógico y la petición de modo.
Tipo de parámetro	Lógico
Sustituye a	–

## Anexo F

### Definiciones de capacidades de gestión de velocidad binaria de canal lógico

El cuadro F.1 define el identificador de capacidad para la gestión de la velocidad binaria. Los parámetros, que proporcionan información sobre los mensajes de gestión de la velocidad binaria que soporta el terminal, sólo deberán incluirse como **genericControlCapability** dentro de la estructura **Capability**. Los cuadros F.2 a F.4 definen los parámetros asociados a las capacidades.

**Cuadro F.1/H.245 – Identificador de capacidad para gestión de velocidad binaria de canal lógico**

Nombre de la capacidad	Gestión de velocidad binaria de canal lógico H.245
Clase de capacidad	Control
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) control (3) logical-channel-bitrate-management (0)}
maxBitRate	El campo maxBitRate no deberá estar incluido.
nonCollapsingRate	Este campo no deberá estar incluido.
transporte	Este campo no deberá estar incluido.

**Cuadro F.2/H.245 – Parámetro de la capacidad de control de flujo para gestión de velocidad binaria**

Nombre del parámetro	Capacidad de control de flujo
Descripción del parámetro	Éste es un parámetro genérico Collapsing. La presencia de este parámetro indica la capacidad de soportar el mensaje FlowControlIndication.
Valor del identificador del parámetro	0
Estado del parámetro	Opcional
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro F.3/H.245 – Parámetro de la capacidad de cambio de velocidad binaria de canal lógico para gestión de velocidad binaria**

Nombre del parámetro	Capacidad de cambio de velocidad binaria de canal lógico
Descripción del parámetro	Éste es un parámetro genérico Collapsing. La presencia de este parámetro indica la capacidad de soportar el procedimiento de cambio de velocidad de canal lógico, que utiliza los mensajes LogicalChannelRateRequest, LogicalChannelRateAcknowledge, LogicalChannelRateReject y LogicalChannelRateRelease.
Valor del identificador del parámetro	1
Estado del parámetro	Opcional
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro F.4/H.245 – Parámetro de frecuencia RTCP para gestión de velocidad binaria**

Nombre del parámetro	Capacidad de frecuencia RTCP
Descripción del parámetro	Éste es un parámetro genérico Collapsing. Indica la frecuencia a la que el terminal puede enviar informes RTCP.
Valor del identificador del parámetro	2
Estado del parámetro	Opcional
Tipo de parámetro	unsigned32Min
Sustituye a	–



## Anexo G

### Definiciones de capacidades ISO/CEI 14496-1

El cuadro G.1 define el identificador de capacidad para las capacidades ISO/CEI 14496-1 [48]. Los cuadros G.2 a G.6 definen los parámetros asociados a las capacidades para ISO/CEI 14496-1. Estos parámetros sólo deberán incluirse como **genericDataCapability** dentro de la estructura **DataCapability** y como **genericDataMode** dentro de la estructura **DataMode**. Para el intercambio de capacidades, deberán especificarse **streamType** y **profileAndLevel**, y podrá especificarse **objectType**. Cuando se abre un canal lógico (de ida o de retorno) deberán especificarse **ES\_ID** u **objectDescriptor**.

En el anexo F/H.324 se incluye más información acerca de la utilización de la capacidad genérica ISO/CEI 14496-1.

#### G.1 Identificador de capacidades

**Cuadro G.1/H.245 – Identificador de capacidad para ISO/CEI 14496-1**

Nombre de la capacidad	ISO/CEI 14496-1
Clase de la capacidad	Aplicación de datos
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) ISO/IEC 14496-1 (0)}
maxBitRate	Este campo se incluirá para indicar la velocidad binaria máxima de un tren de datos ISO/CEI 14496.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo se incluirá para indicar el protocolo de protección contra errores para un tren de datos ISO/CEI 14496 específico transportado por un canal lógico.

#### G.2 Parámetros de capacidad utilizados para negociaciones de capacidades y señalización de canal lógico

**Cuadro G.2/H.245 – Parámetro de capacidad streamType**

Nombre del parámetro	streamType
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. <b>StreamType</b> indica el tipo de tren ISO/CEI 14496 a que hace referencia un caso específico de capacidad genérica ISO/CEI 14496-1 indicado en el cuadro 9 ("streamType Values") de ISO/CEI 14496-1.
Valor del identificador del parámetro	0
Estado del parámetro	Facultativo. Estará presente para el intercambio de capacidades. No estará presente para la señalización de canal lógico ni para la petición de modo.
Tipo de parámetro	unsignedMax. Estará en la gama 0..255.
Sustituye a	–

**Cuadro G.3/H.245 – Parámetro de capacidad profileAndLevel**

Nombre del parámetro	profileAndLevel
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>profileAndLevel indica la capacidad de procesar los perfiles particulares en combinación con el nivel según se indica en:</p> <ul style="list-style-type: none"> <li>• Cuadro 3 de ISO/CEI 14496-1 ("ODProfileLevelIndication Values") para streamType = 0x01</li> <li>• Cuadro 4 de ISO/CEI 14496-1 ("sceneProfileLevelIndication Values") para streamType = 0x03</li> <li>• Cuadro 5 de ISO/CEI 14496-1 ("audioProfileLevelIndication Values") para streamType = 0x05</li> <li>• Cuadro 6 de ISO/CEI 14496-1 ("visualProfileLevelIndication Values") para streamType = 0x04</li> </ul>
Valor del identificador del parámetro	1
Estado del parámetro	Facultativo. Estará presente para el intercambio de capacidades. No estará presente para la señalización de canal lógico ni para la petición de modo.
Tipo de parámetro	unsignedMax. Estará en la gama 0..255.
Sustituye a	–

**Cuadro G.4/H.245 – Parámetro de capacidad objectType**

Nombre del parámetro	objectType
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>objectType indica el conjunto de herramientas que debe utilizar el decodificador del tren de bits contenido en un canal lógico proporcionado en:</p> <ul style="list-style-type: none"> <li>• Cuadro 8 de ISO/CEI 14496-1 ("objectTypeIndication Values ") para streamType = 0x04 ó 0x05</li> <li>• Cuadro 7 de ISO/CEI 14496-1 ("graphicsProfileLevelIndication Values") para streamType = 0x03</li> </ul> <p>Para todos los demás valores de streamType, objectType no se define y, por consiguiente, no deberá ser utilizado.</p>
Valor del identificador del parámetro	2
Estado del parámetro	<p>Opcional.</p> <p>Para streamType = 0x04 ó 0x05, no deberá estar presente en el intercambio de capacidades, deberá estar presente en la señalización de canal lógico. Puede estar presente en la petición de modo.</p> <p>Para streamType = 0x03, deberá estar presente en el intercambio de capacidades y en la señalización de canal lógico. Puede estar presente en la petición de modo.</p> <p>Para los demás valores streamType, no deberá estar presente.</p>
Tipo de parámetro	unsignedMax. Deberá estar comprendido en la gama 0..255.
Sustituye a	–

### G.3 Parámetros de capacidad utilizados para la señalización de canal lógico

**Cuadro G.5/H.245 – Parámetro de capacidad objectDescriptor**

Nombre del parámetro	objectDescriptor
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. <b>objectDescriptor</b> contiene una cadena de octetos que proporciona toda la información necesaria para configurar el decodificador para un tren de bits determinado en un canal lógico (véase ISO/CEI 14496-1). Sólo contendrá información para un tren elemental.
Valor del identificador del parámetro	3
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni la petición de modo. Puede estar presente para la señalización de canal lógico.
Tipo de parámetro	octetString
Sustituye a	–

**Cuadro G.6/H.245 – Parámetro de capacidad ES\_ID**

Nombre del parámetro	ES_ID
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. <b>ES_ID</b> indica el ID del tren elemental que está contenido en un canal lógico específico y por el cual puede ser referido por otros trenes de datos ISO/CEI 14496. Para el InitialObjectDescriptor, ES_ID se pondrá a '0' (cero).
Valor del identificador del parámetro	4
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades. Puede estar presente para la señalización de canal lógico. Estará presente para la petición de modo.
Tipo de parámetro	unsignedMax. Estará en la gama 0..65535.
Sustituye a	–

## Anexo H

### Definiciones de capacidades ISO/CEI 14496-3

El cuadro H.1 define el identificador para las capacidades ISO/CEI 14496-3 [50] e ISO/CEI 14496-3/Amd.1 [51]. Los cuadros H.2 a H.11 definen los parámetros de capacidad asociados para ISO/CEI 14496-3. Estos parámetros sólo se incluirán como `genericAudioCapability` dentro de la estructura `AudioCapability` y como `genericAudioMode` dentro de la estructura `AudioMode`. Para el intercambio de capacidades, los parámetros `profileAndLevel`, `formatType` y `maxA1-sduAudioFrames` estarán presentes, `audioObjectType` y `maxAudioObjects` pueden estar presentes y todos los demás estarán ausentes. Si `formatType` indica formato de tren de transporte ISO/CEI 14496-3, `maxAudioObjects` estará presente para el intercambio de capacidades. Al abrirse un canal lógico (directo o inverso), `profileAndLevel`, `formatType` y `audioObjectType` estarán presentes y todos los demás parámetros pueden ser especificados. Para la petición de modo, `profileAndLevel` y `formatType` estarán presentes y `audioObjectType` puede ser especificado.

`profileAndLevel` de ISO/CEI 14496-3 e ISO/CEI 14496-3/Amd.1 pueden soportar varios tipos de objetos audio. El objeto audio será transportado como uno de dos formatos de trenes de bits que son el formato de datos brutos y el formato de tren de transporte ISO/CEI 14496-3. `formatType` indica la elección de un tipo de formato de tren de bits. En aplicaciones que utilizan transmisión a multivelocidad o escalonada, es útil permitir cambios de estructura de los objetos audio en un canal lógico. Esto se puede realizar con el formato MPEG-4/Audio que permite cambiar la configuración del tren trama por trama. Para la transmisión a baja velocidad binaria, se puede utilizar el formato de datos brutos para disminuir la redundancia de transmitir la configuración del tren en cada trama.

**Cuadro H.1/H.245 – Identificador para la capacidad ISO/CEI 14496-3**

Nombre de la capacidad	ISO/CEI 14496-3
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 14496-3 (0)}
<code>maxBitRate</code>	Este campo se incluirá siempre.
<code>nonCollapsingRaw</code>	Este campo no se incluirá.
<code>transporte</code>	Este campo no se incluirá.

**Cuadro H.2/H.245 – Perfil y nivel para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	<code>profileAndLevel</code>
Descripción del parámetro	Éste es un parámetro genérico <code>nonCollapsing</code> . <code>profileAndLevel</code> indica la capacidad de procesar perfiles particulares en combinación con el nivel, según se indica en ISO/CEI 14496-1 e ISO/CEI 14496-1/Amd.1.
Valor del identificador del parámetro	0
Estado del parámetro	Obligatorio
Tipo de parámetro	<code>unsignedMax</code> . Estará en la gama 0..255.
Sustituye a	–

**Cuadro H.3/H.245 – formatType para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	formatType
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>formatType indica la elección del tipo de formato del tren de bits de un objeto audio entre el formato de datos brutos y el formato audio como sigue:</p> <ul style="list-style-type: none"> <li>• 0: el formato de datos brutos (ISO/CEI 14496-3 e ISO/CEI 14496-3/Amd.1).</li> <li>• 1: el formato definido como multiplex de transporte audio MPEG-4 de baja tara (LATM) en ISO/CEI 14496-3/Amd.1.</li> </ul>
Valor del identificador del parámetro	1
Estado del parámetro	Obligatorio
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro H.4/H.245 – maxA1-sduAudioFrames para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	maxA1-sduAudioFrames
Descripción del parámetro	Éste es un parámetro genérico Collapsing. Indica el número máximo de tramas audio por paquete AL-SDU.
Valor del identificador del parámetro	2
Estado del parámetro	Estará presente para el intercambio de capacidades y la señalización de canal lógico. No estará presente para la petición de modo.
Tipo de parámetro	unsignedMin. Estará en la gama 1..256.
Sustituye a	–

**Cuadro H.5/H.245 – audioObjectType para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	audioObjectType
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>audioObjectType indica el conjunto de herramientas que ha de utilizar el decodificador del tren de bits contenido en el canal lógico según se indica en ISO/CEI 14496-3/Amd.1. Puede utilizarse para limitar la capacidad dentro del profileAndLevel especificado en el intercambio de capacidades.</p>
Valor del identificador del parámetro	3
Estado del parámetro	Facultativo. Puede estar presente para el intercambio de capacidades. Estará presente para la señalización de canal lógico. Puede estar presente para la petición de modo.
Tipo de parámetro	unsignedMax. Estará en la gama 0..31.
Sustituye a	–

**Cuadro H.6/H.245 – audioSpecificConfig para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	audioSpecificConfig
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. audioSpecificConfig indica cómo configurar el decodificador de un objeto particular (véase ISO/CEI 14496-3/Amd.1).
Valor del identificador del parámetro	4
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni para la petición de modo. Estará presente para la señalización de canal lógico si formatType es igual a 0 (formato de datos brutos). En caso contrario, no estará presente para la señalización de canal lógico.
Tipo de parámetro	octetString
Sustituye a	–

**Cuadro H.7/H.245 – maxAudioObjects para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	maxAudioObjects
Descripción del parámetro	Éste es un parámetro genérico Collapsing. Especifica el número máximo de objetos audio multiplexados en la cabida útil audio.
Valor del identificador del parámetro	5
Estado del parámetro	Facultativo. Si formatType es igual a 0 (formato de datos brutos), no estará presente para el intercambio de capacidades y la señalización de canal lógico. En caso contrario, estará presente para el intercambio de capacidades y la señalización de canal lógico. No estará presente para la petición de modo.
Tipo de parámetro	unsignedMin. Estará en la gama 1..16.
Sustituye a	–

**Cuadro H.8/H.245 – muxConfigPresent para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	muxConfigPresent
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. muxConfigPresent indica si la configuración de cabida útil audio está multiplexada en la propia cabida útil audio como se da en ISO/CEI 14496-3/Amd.1: <ul style="list-style-type: none"> <li>• 0: configuración de cabida útil audio (streamMuxConfig) no está multiplexada en la cabida útil audio.</li> <li>• 1: streamMuxConfig está multiplexada en la cabida útil audio.</li> </ul>
Valor del identificador del parámetro	6
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni para la petición de modo. Estará presente para la señalización de canal lógico si formatType es igual a 1 (formato LATM). En caso contrario, no estará presente para la señalización de canal lógico.
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro H.9/H.245 – EP\_DataPresent para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	EP_DataPresent
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>EP_DataPresent indica si la cabida útil audio tiene elasticidad frente a los errores de bit (ninguna pérdida de paquetes) tal como se da en ISO/CEI 14496-3/Amd.1:</p> <ul style="list-style-type: none"> <li>• 0: la cabida útil audio no tiene elasticidad frente a los errores.</li> <li>• 1: la cabida útil audio tiene elasticidad frente a los errores. La configuración para la herramienta EP ISO/CEI 14496-3/Amd.1 (errorProtection_SpecificConfig) puede estar presente para la señalización de canal lógico.</li> </ul>
Valor del identificador del parámetro	7
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni para la petición de modo. Estará presente para la señalización de canal lógico si formatType es igual a 1 (formato LATM). En caso contrario, no estará presente para la señalización de canal lógico.
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro H.10/H.245 – streamMuxConfig para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	streamMuxConfig
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>streamMuxConfig indica la configuración de cabida útil audio dada en ISO/CEI 14496-3/Amd.1.</p>
Valor del identificador del parámetro	8
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni para la petición de modo. Estará presente para la señalización de canal lógico si formatType es igual a 1 (formato LATM). En caso contrario, no estará presente para la señalización de canal lógico.
Tipo de parámetro	octetString
Sustituye a	–

**Cuadro H.11/H.245 – ErrorProtection\_SpecificConfig para la capacidad ISO/CEI 14496-3**

Nombre del parámetro	errorProtection_SpecificConfig
Descripción del parámetro	<p>Éste es un parámetro genérico nonCollapsing.</p> <p>errorProtection_SpecificConfig indica el modo de configurar la herramienta EP ISO/CEI 14496-3/Amd.1 dado en la descripción LATM EP_MuxElement() de ISO/CEI 14496-3/Amd.1.</p>
Valor del identificador del parámetro	9
Estado del parámetro	Facultativo. No estará presente para el intercambio de capacidades ni para la petición de modo. Estará presente para la señalización de canal lógico si formatType es igual a 1 (formato LATM). En caso contrario, no estará presente para la señalización de canal lógico.
Tipo de parámetro	octetString
Sustituye a	–

## Anexo I

### Definiciones de la capacidad de multivelocidad adaptable GSM

El cuadro I.1 define el identificador para las capacidades multivelocidad adaptable (AMR, *adaptive multi-rate*) GSM. Los cuadros I.2 a I.7 definen los parámetros de capacidades asociados. Las especificaciones pertinentes son [58], [69], [70], [71], [72], [73], [74] y [75].

La cláusula I.1 define el modo de señalización y el empaquetado de tramas vocales en la estructura de octetos.

**Cuadro I.1/H.245 – Identificador de capacidad AMR GSM**

Nombre de la capacidad	AMR
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) amr (1)}
maxBitRate	Será 122
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

**Cuadro I.2/H.245 – Parámetro de capacidad AMR GSM – maxAl-sduAudioFrames**

Nombre del parámetro	maxAl-sduAudioFrames
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica el número máximo de tramas audio por AL-SDU.
Valor del identificador del parámetro	0
Estado del parámetro	Deberá estar presente para el intercambio de capacidades y la señalización de canal lógico. No deberá estar presente para la petición de modo.
Tipo de parámetro	unsignedMin
Sustituye a	–

**Cuadro I.3/H.245 – Parámetro de capacidad AMR GSM – bitRate**

Nombre del parámetro	bitRate
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. Especifica la velocidad binaria AMR. Este parámetro se utilizará sólo en peticiones de modo. 0 = 4,75; 1 = 5,15; 2 = 5,90; 3 = 6,70; 4 = 7,40; 5 = 7,95; 6 = 10,2; 7 = 12,2.
Valor del identificador del parámetro	1
Estado del parámetro	Facultativo
Tipo de parámetro	unsignedMin
Sustituye a	–



**Cuadro I.4/H.245 – Parámetro de capacidad AMR GSM – Ruido de confort GSM AMR**

Nombre del parámetro	gsmAmrComfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que se ha de utilizar ruido de confort AMR GSM en la petición de modo. Este parámetro se utilizará solamente en peticiones de modo, pero no en capacidades porque esta capacidad es obligatoria.
Valor del identificador del parámetro	2
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro I.5/H.245 – Parámetro de capacidad AMR GSM – Ruido de confort GSM EFR**

Nombre del parámetro	gsmEfrComfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. En una capacidad, el parámetro especifica si hay o no una capacidad ruido de confort GSM EFR. En un modo petición, el parámetro especifica que se solicita el ruido de confort GSM EFR.
Valor del identificador del parámetro	3
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro I.6/H.245 – Parámetro de capacidad AMR GSM – Ruido de confort IS-641**

Nombre del parámetro	is-641ComfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. En una capacidad, el parámetro especifica si hay o no una capacidad de ruido de confort IS-641. En un modo petición, el parámetro especifica que se solicita el ruido de confort IS-641.
Valor del identificador del parámetro	4
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro I.7/H.245 – Parámetro de capacidad AMR GSM – Ruido de confort PDC EFR**

Nombre del parámetro	pdcEFRComfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. En una capacidad, el parámetro especifica si hay o no una capacidad ruido de confort PDC EFR. PDC EFR es un codificador ACELP a 6,7 kbit/s especificado en la sección 5.4 de [74]. En un modo petición, el parámetro especifica que se solicita el ruido de confort PD EFR.
Valor del identificador del parámetro	5
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

## I.1 Definición de señalización de modo y relleno de bits para lograr la alineación de octetos

El modo de señalización AMR en los sistemas móviles se basa parcialmente en la señalización externa no definida dentro de la especificación del códec vocal AMR. Para la compatibilidad con los sistemas móviles, en esta cláusula se define la señalización de modo necesaria para utilizar AMR en las Recomendaciones de la serie H del UIT-T. El tamaño de las tramas vocales del códec AMR en los diferentes modos no es un múltiplo de ocho. Por ese motivo, se necesita el relleno de bits para lograr la estructura de octetos.

Nótese que en el futuro, el contenido de este anexo puede ser modificado para hacer referencia a la documentación de ETSI o a otra documentación de normalización apropiada.

El cuadro I.10 muestra todos los modos AMR en índices de modo específicos  $m_i(k)$ . Los índices de modo están reservados también para tramas de supresión de periodos de silencio que se utilizan en diferentes sistemas. Los cuadros I.11 a I.14 especifican los formatos de éstas. El cuadro I.15 especifica una trama sin transmisión.

Los bits entregados por el codificador vocal AMR  $\{s(1),s(2),\dots,s(K_s)\}$ , serán reordenados de acuerdo con la importancia subjetiva antes de ser alineados en octetos. Los cuadros I.16 a I.23 definen la reordenación correcta de los modos de códec vocales 12,2 kbit/s, 10,2 kbit/s, 7,95 kbit/s, 7,40 kbit/s, 6,70 kbit/s, 5,90 kbit/s, 5,15 kbit/s y 4,75 kbit/s, respectivamente. En los cuadros, los parámetros de códec vocales están numerados en el orden en que son entregados por el correspondiente codificador vocal de acuerdo con GSM 06.90 y los bits reordenados son etiquetados  $\{d(0),d(1),\dots,d(K_d - 1)\}$ , definidos en orden de importancia decreciente. El índice  $K_d$  hace referencia al número de bits entregados por el codificador vocal, véase el cuadro I.8.

**Cuadro I.8/H.245 – Número de bits de conversación en diferentes modos AMR**

Modo del códec	Número de bits de conversación entregados por bloque ( $K_d$ )
AMR12.2	244
AMR10.2	204
AMR7.95	159
AMR7.4	148
AMR6.7	134
AMR5.9	118
AMR5.15	103
AMR4.75	95

El algoritmo de ordenación está en pseudocódigo como:

$$\begin{aligned} &\text{para } j = 0 \text{ a } K_d - 1 \\ &d(j) = s(\text{table}(j) + 1) \end{aligned}$$

donde  $\text{table}(j)$  se lee línea por línea de izquierda a derecha.

Por tanto, la estructura de octeto  $b_n(k)$  para cada modo de códec AMR se define como sigue:

Número de bits de relleno:  $K_s = 8 * N - K_d - K_i$ , donde  $K_i$  es el número de bits de índice de modo

Octeto[0]:  $b_0(k) = m_i(k)$ , para  $k = 0, 1, 2, 3$  (índice de modo)

$b_0(k) = d(k - 4)$ , para  $k = 4, 5, 6, 7$

Octeto[m]:  $b_m(k) = d(8 * m - 4 + k)$ , para  $k = 0, \dots, 7$  y  $0 < m < N - 1$

Octeto[N - 1]:  $b_{N-1}(k) = d(8 * (N - 1) - 4 + k)$ , para  $k = 0, \dots, 7 - K_s$   
 Si  $K_s > 0$   
 $b_{N-1}(k) = UB$ , para  $k = 8 - K_s, \dots, 7$

**Cuadro I.9/H.245 – Ejemplo de correspondencia de modo de codificación vocal AMR 6,7 kbit/s**

Octeto	MSB Estructura de octeto							LSB
$b_0$	d(3)	d(2)	d(1)	d(0)	0	0	1	1
$b_1$	d(11)	d(10)	d(9)	d(8)	d(7)	d(6)	d(5)	d(4)
$b_2$	...	...	...	...	...	...	...	d(12)
$b_{17}$	UB	UB	UB	UB	UB	UB	d(133)	d(132)

**Cuadro I.10/H.245 – Correspondencia de los modos de codificación vocal AMR definidos en GSM 06.90 con los bits de índice de modo en octetos transmitidos**

Índice de modo (4 bits)	Denominación en GSM 06.90 y GSM 06.92
0 (Amr4-75k)	Modo 4,75 kbit/s
1 (Amr5-15k)	Modo 5,15 kbit/s
2 (Amr5-90k)	Modo 5,90 kbit/s
3 (Amr6-70k)	Modo 6,70 kbit/s (PDC-EFR)
4 (Amr7-40k)	Modo 7,40 kbit/s (IS-641)
5 (Amr7-95k)	Modo 7,95 kbit/s
6 (Amr10-2k)	Modo 10,2 kbit/s
7 (Amr12-2k)	Modo 12,2 kbit/s (GSM EFR)
8	Trama de ruido de confort GSMAMR (obligatoria)
9	Trama de ruido de confort GSM-EFR (facultativa)
10	Trama de ruido de confort IS-641 (facultativa)
11	Trama de ruido de confort PDC-EFR (facultativa)
12-14	Para uso futuro
15	Ninguna transmisión

**Cuadro I.11/H.245 – Correspondencia de los bits descriptores de ruido de confort de GSM 06.92 con octetos para el índice de modo 8 (Para los bits de s1 a s35, véase GSM 06.92)**

Octetos transmitidos	MSB	Correspondencia de bits						LSB
1	Índice de 1. <sup>er</sup> subvector de LSF s4	Índice de 1. <sup>er</sup> subvector de LSF s3	s2	s1	Índice de modo mi(3)	mi(2)	mi(1)	mi(0)
2	Índice de 2. <sup>o</sup> subvector de LSF s12	Índice de 1. <sup>er</sup> subvector de LSF s11 s10 s9 s8 s7 s6 s5						
3	Índice de 2. <sup>o</sup> subvector de LSF s20 s19 s18 s17 s16 s15 s14 s13							
4	Índice de 3. <sup>er</sup> subvector de LSF s28 s27 s26 s25 s24 s23 s22 s21							
5	bit tipo SID t1	Energía de trama s35 s34 s33 s32 s31 s30					Índice de 3. <sup>er</sup> subvector de LSF s29	
6	Bits de relleno UB UB UB UB UB				Indicación de modo conversación smi(2) smi(1) smi(0)			

Para las definiciones de bits de descriptores adicionales necesarios para el descriptor de periodo de silencio, véase el cuadro I.11:

Tipo SID-(t1) es {0=SID\_FIRST, 1=SID\_UPDATE}

La indicación de modo conversación (smi(0)-smi(2)) es el modo conversación de acuerdo con las ocho primeras entradas del cuadro de índices de modo.

**Cuadro I.12/H.245 – Correspondencia de bits de descriptores de inserción de silencio de GSM 06.60 (los parámetros están descritos también en GSM 06.62) con octetos para el índice de modo 9**  
(Para los bits s1 a s91, véase GSM 06.60)

Octetos transmitidos	MSB	Correspondencia de bits						LSB
1	Índice de 1. <sup>a</sup> submatriz de LSF s4      s3      s2      s1				Índice de modo mi(3)    mi(2)    mi(1)    mi(0)			
2	Índice de 2. <sup>a</sup> submatriz de LSF s12      s11      s10      s9      s8					Índice de 1. <sup>a</sup> submatriz de LSF s7      s6      s5		
3	Índice de 3. <sup>a</sup> submatriz de LSF s20      s19      s18      s17      s16					Índice de 2. <sup>a</sup> submatriz de LSF s15      s14      s13		
4	Índice de 4. <sup>a</sup> submatriz de LSF s28      s27      s26      s25				Signo de 3. <sup>a</sup> submatriz de LSF s24	Signo de 3. <sup>a</sup> submatriz de LSF s23      s22      s21		
5	Índice de 5. <sup>a</sup> submatriz de LSF s36      s35      s34      s33				Índice de 4. <sup>a</sup> submatriz de LSF s32      s31      s30      s29			
6	Bits de relleno	Ganancia fija de tabla de códigos					Índice de 5. <sup>a</sup> submatriz de LSF	
	UB	s91	s90	s89	s88	s87	s38	s37

**Cuadro I.13/H.245 – Correspondencia de bits de descriptores de inserción de silencio de TIA IS-641-A con octetos, para el índice de modo 10**  
(Para los bits de cn0 a cn37, véase TIA IS-641-A)

Octetos transmitidos	MSB	Correspondencia de bits						LSB
1	Índice de 1. <sup>er</sup> subvector de LSF cn3      cn2      cn1      cn0				Índice de modo mi(3)    mi(2)    mi(1)    mi(0)			
2	Índice de 2. <sup>o</sup> subvector de LSF cn11      cn10      cn9      cn8				Índice de 1. <sup>er</sup> subvector de LSF cn7      cn6      cn5      cn4			
3	Índice de 3. <sup>er</sup> subvector de LSF <sub>index</sub> cn19      cn18      cn17			Índice de 2. <sup>o</sup> subvector de LSF cn16      cn15      cn14      cn13      cn12				
4	Ganancia aleatoria de excitación cn27      cn26		Índice de 3. <sup>er</sup> subvector de LSF cn25      cn24      cn23      cn22      cn21      cn20					
5	Índice de 1. <sup>er</sup> parámetro RESC cn35      cn34		Ganancia aleatoria de excitación cn33      cn32      cn31      cn30      cn29      cn28					
6	Bits de relleno						Índice de 2. <sup>o</sup> parámetro RESC	
	UB	UB	UB	UB	UB	UB	cn37	cn36

**Cuadro I.14/H.245 – Correspondencia de bits de descriptores de inserción de silencio de RCR STD-27H con octetos para el índice de modo 11**  
(Para los bits s1 a s35, véase RCR STD-27H)

Octetos transmitidos	MSB	Correspondencia de bits						LSB
1	Índice de 1. <sup>er</sup> subvector de LSF s4	s3	s2	s1	mi(3)	mi(2)	mi(1)	mi(0)
2	Índice de 2. <sup>o</sup> subvector de LSF s12	Índice de 1. <sup>er</sup> subvector de LSF						
3	Índice de 2. <sup>o</sup> subvector de LSF							
	s20	s19	s18	s17	s16	s15	s14	s13
4	Índice de 3. <sup>er</sup> subvector de LSF							
	s28	s27	s26	s25	s24	s23	s22	s21
5	Tipo SID t1	Energía de trama						Índice de 3. <sup>er</sup> subvector s29
	s35	s34	s33	s32	s31	s30		
6	Bits de relleno							Tipo SID
	UB	UB	UB	UB	UB	UB	UB	t2

Para la definición de bits de descriptores adicionales necesarios para PDC-EFR, véase el cuadro I.14:

El tipo SID es {0=POST0, 1=POST1(SID\_UPDATE), 2=PRE, 3=POST1\_BAD}, donde el LSB del tipo SID es t1 y el MSB del tipo SID es t2.

**Cuadro I.15/H.245 – Definición de trama de no transmisión para el índice de modo 15**

Octetos transmitidos	MSB	Contenido de trama						LSB
1	Bits de relleno				Índice de modo			
	UB	UB	UB	UB	mi(3)	mi(2)	mi(1)	mi(0)

**Cuadro I.16/H.245 – Importancia subjetiva de los bits codificados  
de conversación para AMR12.2**

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	23	15	16	17	18
19	20	21	22	24	25	26	27	28	38
141	39	142	40	143	41	144	42	145	43
146	44	147	45	148	46	149	47	97	150
200	48	98	151	201	49	99	152	202	86
136	189	239	87	137	190	240	88	138	191
241	91	194	92	195	93	196	94	197	95
198	29	30	31	32	33	34	35	50	100
153	203	89	139	192	242	51	101	154	204
55	105	158	208	90	140	193	243	59	109
162	212	63	113	166	216	67	117	170	220
36	37	54	53	52	58	57	56	62	61
60	66	65	64	70	69	68	104	103	102
108	107	106	112	111	110	116	115	114	120
119	118	157	156	155	161	160	159	165	164
163	169	168	167	173	172	171	207	206	205
211	210	209	215	214	213	219	218	217	223
222	221	73	72	71	76	75	74	79	78
77	82	81	80	85	84	83	123	122	121
126	125	124	129	128	127	132	131	130	135
134	133	176	175	174	179	178	177	182	181
180	185	184	183	188	187	186	226	225	224
229	228	227	232	231	230	235	234	233	238
237	236	96	199						

**Cuadro I.17/H.245 – Importancia subjetiva de los bits codificados  
de conversación para AMR10.2**

7	6	5	4	3	2	1	0	16	15
14	13	12	11	10	9	8	26	27	28
29	30	31	115	116	117	118	119	120	72
73	161	162	65	68	69	108	111	112	154
157	158	197	200	201	32	33	121	122	74
75	163	164	66	109	155	198	19	23	21
22	18	17	20	24	25	37	36	35	34
80	79	78	77	126	125	124	123	169	168
167	166	70	67	71	113	110	114	159	156
160	202	199	203	76	165	81	82	92	91
93	83	95	85	84	94	101	102	96	104
86	103	87	97	127	128	138	137	139	129
141	131	130	140	147	148	142	150	132	149
133	143	170	171	181	180	182	172	184	174
173	183	190	191	185	193	175	192	176	186
38	39	49	48	50	40	52	42	41	51
58	59	53	61	43	60	44	54	194	179
189	196	177	195	178	187	188	151	136	146
153	134	152	135	144	145	105	90	100	107
88	106	89	98	99	62	47	57	64	45
63	46	55	56						

**Cuadro I.18/H.245 – Importancia subjetiva de los bits codificados  
de conversación para AMR7.95**

8	7	6	5	4	3	2	14	16	9
10	12	13	15	11	17	20	22	24	23
19	18	21	56	88	122	154	57	89	123
155	58	90	124	156	52	84	118	150	53
85	119	151	27	93	28	94	29	95	30
96	31	97	61	127	62	128	63	129	59
91	125	157	32	98	64	130	1	0	25
26	33	99	34	100	65	131	66	132	54
86	120	152	60	92	126	158	55	87	121
153	117	116	115	46	78	112	144	43	75
109	141	40	72	106	138	36	68	102	134
114	149	148	147	146	83	82	81	80	51
50	49	48	47	45	44	42	39	35	79
77	76	74	71	67	113	111	110	108	105
101	145	143	142	140	137	133	41	73	107
139	37	69	103	135	38	70	104	136	



**Cuadro I.19/H.245 – Importancia subjetiva de los bits codificados de conversación para AMR7.4**

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	26	87	27
88	28	89	29	90	30	91	51	80	112
141	52	81	113	142	54	83	115	144	55
84	116	145	58	119	59	120	21	22	23
17	18	19	31	60	92	121	56	85	117
146	20	24	25	50	79	111	140	57	86
118	147	49	78	110	139	48	77	53	82
114	143	109	138	47	76	108	137	32	33
61	62	93	94	122	123	41	42	43	44
45	46	70	71	72	73	74	75	102	103
104	105	106	107	131	132	133	134	135	136
34	63	95	124	35	64	96	125	36	65
97	126	37	66	98	127	38	67	99	128
39	68	100	129	40	69	101	130		

**Cuadro I.20/H.245 – Importancia subjetiva de los bits codificados de conversación para AMR6.7**

0	1	4	3	5	6	13	7	2	8
9	11	15	12	14	10	28	82	29	83
27	81	26	80	30	84	16	55	109	56
110	31	85	57	111	48	73	102	127	32
86	51	76	105	130	52	77	106	131	58
112	33	87	19	23	53	78	107	132	21
22	18	17	20	24	25	50	75	104	129
47	72	101	126	54	79	108	133	46	71
100	125	128	103	74	49	45	70	99	124
42	67	96	121	39	64	93	118	38	63
92	117	35	60	89	114	34	59	88	113
44	69	98	123	43	68	97	122	41	66
95	120	40	65	94	119	37	62	91	116
36	61	90	115						

**Cuadro I.21/H.245 – Importancia subjetiva de los bits codificados de conversación para AMR5.9**

0	1	4	5	3	6	7	2	13	15
8	9	11	12	14	10	16	28	74	29
75	27	73	26	72	30	76	51	97	50
71	96	117	31	77	52	98	49	70	95
116	53	99	32	78	33	79	48	69	94
115	47	68	93	114	46	67	92	113	19
21	23	22	18	17	20	24	111	43	89
110	64	65	44	90	25	45	66	91	112
54	100	40	61	86	107	39	60	85	106
36	57	82	103	35	56	81	102	34	55
80	101	42	63	88	109	41	62	87	108
38	59	84	105	37	58	83	104		

**Cuadro I.22/H.245 – Importancia subjetiva de los bits codificados de conversación para AMR5.15**

7	6	5	4	3	2	1	0	15	14
13	12	11	10	9	8	23	24	25	26
27	46	65	84	45	44	43	64	63	62
83	82	81	102	101	100	42	61	80	99
28	47	66	85	18	41	60	79	98	29
48	67	17	20	22	40	59	78	97	21
30	49	68	86	19	16	87	39	38	58
57	77	35	54	73	92	76	96	95	36
55	74	93	32	51	33	52	70	71	89
90	31	50	69	88	37	56	75	94	34
53	72	91							

**Cuadro I.23/H.245 – Importancia subjetiva de los bits codificados de conversación para AMR4.75**

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	23	24	25	26
27	28	48	49	61	62	82	83	47	46
45	44	81	80	79	78	17	18	20	22
77	76	75	74	29	30	43	42	41	40
38	39	16	19	21	50	51	59	60	63
64	72	73	84	85	93	94	32	33	35
36	53	54	56	57	66	67	69	70	87
88	90	91	34	55	68	89	37	58	71
92	31	52	65	86					

## Anexo J

### Definiciones del códec vocal TDMA ACELP

El cuadro J.1 define el identificador para las capacidades del códec vocal TIA/EIA 136 ACELP [75]. Los cuadros J.2 a J.4 definen los parámetros de capacidad asociados. Este códec se utiliza en estaciones de base celulares TDMA y PCS y en teléfonos móviles. Las especificaciones técnicas de este códec figuran en la norma TIA/EIA 136 Parte 410. Esta norma ha sido publicada por TIA (North American) Telecommunications Industry Association y aprobada por ANSI, American National Standards Institute.

**Cuadro J.1/H.245 – Identificador de capacidad TIA/EIA 136 ACELP**

Nombre de la capacidad	TIA/EIA 136 ACELP Vocoder
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) acelp (2)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

**Cuadro J.2/H.245 – Parámetro de capacidad TIA/EIA 136 ACELP – maxAl-sduAudioFrames**

Nombre del parámetro	maxAl-sduAudioFrames
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica el número máximo de tramas de audio por AL-SDU.
Valor del identificador del parámetro	0
Estado del parámetro	Estará presente para intercambio de capacidad y señalización de canal lógico. No estará presente para petición de modo.
Tipo de parámetro	unsignedMin
Sustituye a	–

**Cuadro J.3/H.245 – Parámetro de capacidad TIA/EIA 136 ACELP – Ruido de confort**

Nombre del parámetro	comfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que se ha de utilizar el ruido de confort TIA/EIA 136 (IS-641) en la petición de modo. Este parámetro se utilizará solamente en peticiones de modo, pero no en capacidades porque esta capacidad es obligatoria.
Valor del identificador del parámetro	1
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

### Cuadro J.4/H.245 – Capacidad de parámetro TIA/EIA 136 ACELP – Aleatorizado

Nombre del parámetro	scrambled
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que se ha de utilizar la aleatorización en la petición de modo. Este parámetro se utilizará solamente en peticiones de modo, pero no en capacidades porque esta capacidad es obligatoria.
Valor del identificador del parámetro	2
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

## Anexo K

### Definiciones del códec vocal TDMA US1

El cuadro K.1 define el identificador para las capacidades del códec vocal TIA/EIA 136 US1 [76]. Los cuadros K.2 a K.4 definen los parámetros de capacidad asociados. Este códec se utiliza en estaciones de base celulares TDMA y PCS y en teléfonos móviles. Las especificaciones técnicas de este códec figuran en la norma TIA/EIA-136 Parte 430. Esta norma ha sido publicada por TIA (North American) Telecommunications Industry Association y aprobada por ANSI, American National Standards Institute.

### Cuadro K.1/H.245 – Identificador de capacidad TIA/EIA 136 US1

Nombre de la capacidad	TIA/EIA 136 US1 Vocoder
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) us1 (3)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

**Cuadro K.2/H.245 – Parámetro de capacidad TIA/EIA 136 US1 –  
maxAl-sduAudioFrames**

Nombre del parámetro	maxAl-sduAudioFrames
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica el número máximo de tramas audio por AL-SDU.
Valor del identificador del parámetro	0
Estado del parámetro	Estará presente para intercambio de capacidad y señalización de canal lógico. No estará presente para petición de modo.
Tipo de parámetro	unsignedMin
Sustituye a	–

**Cuadro K.3/H.245 – Parámetro de capacidad TIA/EIA 136 US1 – Ruido de confort**

Nombre del parámetro	comfortNoise
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que se ha de utilizar ruido de confort en la petición de modo. Este parámetro se utilizará solamente en peticiones de modo, pero no en capacidades porque esta capacidad es obligatoria.
Valor del identificador del parámetro	1
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

**Cuadro K.4/H.245 – Parámetro de capacidad TIA/EIA 136 US1 – Aleatorizado**

Nombre del parámetro	scrambled
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que se ha de utilizar aleatorización en la petición de modo. Este parámetro se utilizará solamente en peticiones de modo, pero no en capacidades porque esta capacidad es obligatoria.
Valor del identificador del parámetro	2
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

## Anexo L

### Definiciones del códec vocal EVRC CDMA

El cuadro L.1 define el identificador de capacidades del códec de velocidad variable mejorado (EVRC, *enhanced variable rate codec*) TIA/EIA IS-127 que se utiliza en los teléfonos móviles y estaciones de base PCS y celulares CDMA TIA/EIA IS-95. La descripción técnica completa y las especificaciones detalladas de este códec se dan en la norma TIA/EIA IS-127 publicada por la North American Telecommunications Industry Association (TIA). En los cuadros L.2 a L.4 se definen los parámetros de capacidad asociados.

**Cuadro L.1/H.245 – Identificador de capacidad EVRC CDMA**

Nombre de la capacidad	TIA/EIA IS-127 CDMA EVRC
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) is127evrc (4)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

**Cuadro L.2/H.245 – Parámetro de capacidad TIA/EIA IS-127 EVRC CDMA – maxAl-sduAudioFrames**

Nombre del parámetro	maxAl-sduAudioFrames
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica el número máximo de tramas audio por AL-SDU.
Valor del identificador del parámetro	0
Estado del parámetro	Estará presente para intercambio de capacidad y señalización de canal lógico. No estará presente para petición de modo.
Tipo de parámetro	unsignedMin
Sustituye a	–

**Cuadro L.3/H.245 – Parámetro de capacidad EVRC CDMA – Velocidad binaria EVRC**

Nombre del parámetro	EVRCRate
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. Especifica el modo de velocidad binaria de salida del vocodificador. Este parámetro se utilizará en las peticiones de modo: 1 = velocidad total; 2 = un medio de la velocidad; 3 = un octavo de la velocidad; 4 = modo en blanco.
Valor del identificador del parámetro	1
Estado del parámetro	Facultativo
Tipo de parámetro	unsignedMin
Sustituye a	–

### Cuadro L.4/H.245 – Parámetro de capacidad EVRC CDMA – Aleatorizado

Nombre del parámetro	scrambled
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica que la aleatorización se utilizará en la petición de modo.
Valor del identificador del parámetro	2
Estado del parámetro	Facultativo
Tipo de parámetro	Lógico
Sustituye a	–

## Anexo M

### Definiciones ISO/CEI 13818-7 e UIT-R BS.1196

El cuadro M.1 define el identificador de capacidades para ISO/CEI 13818-7. El cuadro M.2 define los parámetros de capacidad asociados.

El cuadro M.3 define el identificador de capacidades para UIT-R BS.1196. No hay parámetros de capacidad asociados.

#### Cuadro M.1/H.245 – Identificador de capacidad ISO/CEI 13818-7

Nombre de la capacidad	ISO/CEI 13818-7
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

#### Cuadro M.2/H.245 – Perfil y nivel para las capacidades ISO/CEI 13818-7

Nombre del parámetro	profileAndLevel
Descripción del parámetro	Éste es un parámetro genérico nonCollapsing. profileAndLevel indica la capacidad de procesar los perfiles considerados en combinación con el nivel indicado en ISO/CEI 13818-7.
Valor del identificador del parámetro	0
Estado del parámetro	Obligatorio
Tipo de parámetro	unsignedMax. Estará en la gama 0..255.
Sustituye a	–

**Cuadro M.3/H.245 – Identificador de capacidades UIT-R BS.1196**

Nombre de la capacidad	UIT-R BS.1196
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) itu-r-bs.1196 (6)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

## **Anexo N**

### **RFC 3389 – Cabida útil RTP para ruido de confort**

En el cuadro N.1 se define el identificador de capacidad para RFC 3389. No existen parámetros de capacidad asociados.

RFC 3389 del IETF especifica un medio de señalización del ruido de confort entremezclado con paquetes de audio regulares y se basa en el apéndice II/G.711. El propósito principal de esta RFC es el de especificar la señalización del ruido de confort para códecs que no definen su propia señalización y es generalmente aceptado como el medio de señalización de este ruido de confort en sistemas basados en el protocolo de transporte en tiempo real (RTP).

La capacidad del ruido de confort se puede señalar como cualquier otra capacidad y los medios de ese tipo se pueden abrir conjuntamente con G.711, G.726, etc., mediante la especificación de esta capacidad como parte de un canal del tren de cabida útil múltiple (MPS, *multiple payload stream*).

**Cuadro N.1/H.245 – Ruido de confort RFC 3389**

Nombre de la capacidad	Ruido de confort RFC 3389
Clase de capacidad	Códec de audio
Tipo de identificador de la capacidad	Estándar
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) rfc3389 (6)}
maxBitRate	Este campo se incluirá.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.



## Anexo O

### Identificador de capacidad L16

El cuadro O.1 define el identificador de capacidad para L16. El códec L16 es un códec de datos de audio sin comprimir que utiliza una representación de 16 bits con signo, con 65535 pasos divididos por igual entre el nivel de señal máximo y el mínimo que varían de -32768 a 32767. El valor se representa en notación de complemento de dos y orden de byte de red. Este códec se utiliza para verificación de la calidad acústica y posiblemente para aplicaciones LAN de banda ancha de bajo costo. Se define en la sección 4.4.8 de IETF RFC 1890.

El cuadro O.2 define el parámetro de capacidad asociado.

**Cuadro O.1/H.245 – Identificador de capacidad L16**

Nombre de la capacidad	Códec de audio L16
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) L-16 (7)}
maxBitRate	Este campo se incluirá. La velocidad define el número de muestras por segundo; ésta es también la velocidad de reloj de indicación de tiempo RTP. Los valores recomendados de velocidad son 8000, 11025, 16000, 22050, 24000, 32000, 44100 y 48000 muestras por segundo. Se admiten otros valores.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

**Cuadro O.2/H.245 – Parámetro de capacidad L16 – Canales**

Nombre del parámetro	channels
Descripción del parámetro	Éste es un parámetro genérico collapsing. Especifica el número de trenes de audio intercalados. El valor por defecto es 1, estéreo es 2. El intercalado se produce entre muestras de dos bytes individuales.
Valor del identificador del parámetro	0
Estado del parámetro	Estará presente para intercambio de capacidad y señalización de canal lógico. No estará presente para petición de modo.
Tipo de parámetro	unsignedMin
Sustituye a	–

## Anexo P

### Capacidad limitada del tren de audio

Esta capacidad genérica se puede utilizar como elección `genericAudioCapability` de `AudioCapability` en mensajes `TerminalCapabilitySet`, `OpenLogicalChannel` y `CommunicationModeCommand`, y como `genericAudioMode` en un mensaje `RequestMode`.

Para la capacidad limitada del tren de audio se definen dos parámetros:

- El tamaño mínimo del paquete (tramas por paquete) como parámetro `collapsing`;
- La `audioCapability` definida en H.245 para la cual se requiere un tamaño de paquete mínimo. Este valor define el códec así como el tamaño máximo del paquete para el tren de audio deseado.

**Cuadro P.1/H.245 – Identificador de capacidad para tren de audio limitado**

Nombre de la capacidad	Tren de audio limitado
Clase de capacidad	Códec audio
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) bounded-audio-stream (8)}
<code>maxBitRate</code>	El campo <code>maxBitRate</code> siempre estará incluido.
<code>nonCollapsingRaw</code>	Este campo no se incluirá.
<code>transporte</code>	Este campo no se incluirá.

**Cuadro P.2/H.245 – Parámetro de tamaño de paquete mínimo**

Nombre del parámetro	<code>minimumPacketSize</code>
Descripción del parámetro	Éste es un parámetro genérico <code>collapsing</code> . Especifica el número máximo de tramas de audio por paquete.
Valor del identificador del parámetro	0
Estado del parámetro	Obligatorio
Tipo de parámetro	<code>unsignedMax</code> . Estará en la gama 1..256.
Sustituye a	–

**Cuadro P.3/H.245 – Capacidad de audio**

Nombre del parámetro	<code>audioCapability</code>
Descripción del parámetro	Éste es un parámetro genérico <code>nonCollapsing</code> . Contiene una codificación PER <code>audioCapability</code> válida conforme al anexo B.
Valor del identificador del parámetro	1
Estado del parámetro	Obligatorio
Tipo de parámetro	<code>octetString</code>
Sustituye a	–

## Anexo Q

### Capacidad genérica para retransmisión de circuitos Nx64K por el protocolo Internet

#### Q.1 Introducción

Este anexo describe la utilización de una capacidad genérica para retransmisión de circuitos Nx64 de estructura bloqueada a través de una red IP. No se formulan premisas con relación al contenido o la estructura de la información que se ha de retransmitir. Se definen diversos parámetros para simplificar la negociación de un canal satisfactorio.

Esta capacidad tiene la intención de satisfacer los siguientes requisitos:

- 1) El transporte debe ser capaz de llevar información digital restringida o no restringida a 64 kbit/s o 56 kbit/s, como se indica en la Rec. UIT-T Q.931. Se debe señalar que aquí sólo se transporta la información de portadora, exclusiva de las señales de alineación de trama de la red o canales de control.
- 2) La transmisión debe ser transparente para los bytes (denominado "transporte enganchado a la estructura").
- 3) En la interfaz TDM/IP no se debería efectuar el procesamiento de señales de ningún tipo (relleno, compensación de eco, detección de tonos, supresión de silencio o codificación, etc.).
- 4) Se debe soportar el transporte a través del protocolo RTP (o SRTP).
- 5) El tamaño del paquete debe ser negociable.
- 6) Los mecanismos y el orden de recuperación de pérdida/corrupción de paquetes deben ser negociables.
  - a) Corrección de errores en recepción conforme a RFC 2733, en la cual cada paquete "de paridad" FEC lleva consigo  $R$  paquetes sucesivos – esto duplica la anchura de banda del canal.
  - b) Transmisión redundante conforme a RFC 2198 en la cual se transmiten copias adicionales de cada bloque de medios.
- 7) La especificación de la anchura de banda se debe expresar como producto del número de canales y capacidad de canal (Nx64 kbit/s).
- 8) El número de canales de circuito no variará durante la vigencia de una sesión de medios.

Esta capacidad utiliza un formato de paquete definido en la Rec. UIT-T Y.1413 [80]. Se efectuaron algunas consideraciones a las actividades del grupo de trabajo del PWE3 del IETF, pero parece haber mayor interés en el transporte eficiente de los circuitos de audio agrupados que en los datos digitales (véase, por ejemplo, [PWE3]).

#### Q.2 Descripción

##### Q.2.1 Terminología

**Trama** es el agregado de una muestra de 8 bits por canal en orden de canales de 1 a N. Para canales de 64 kbit/s muestreados a 8 kHz, la trama se compone de N octetos. Se consideró conveniente que el tamaño de la trama sea de una muestra por canal a fin de otorgar flexibilidad en la dimensión total del paquete para una gama de N de 1 a 31 o más, y muestras por canal de 1 a 2047.

- **M** es el número de tramas por cabida útil de paquete.
- **N** es el número de canales TDM de 64 kbit/s por trama.

## Q.2.2 Identificador de capacidad

La característica de capacidad genérica propuesta para la transmisión de datos en canal disponible Nx64K se identificará a través de **GenericCapability.capabilityIdentifier.standard** puesto a:

{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) Nx64 (10)}

Para utilización en la capacidad genérica:

**Cuadro Q.1/H.245 – Identificador de capacidad Nx64**

Nombre de la capacidad	Retransmisión de circuitos Nx64
Clase de capacidad	Códec de datos
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) Nx64 (10)}
maxBitRate	El campo maxBitRate se incluirá siempre.
nonCollapsingRaw	Este campo no se incluirá.
transporte	Este campo no se incluirá.

## Q.2.3 Parámetros para intercambio de capacidad

Los parámetros definidos para la capacidad Nx64 se describirán en los cuadros Q.2 y Q.3:

**Cuadro Q.2/H.245 – Parámetro número de canales Nx64**

Nombre del parámetro	Número de canales
Descripción del parámetro	Éste es un parámetro genérico collapsing. El número de canales especifica la cantidad de canales portadores de 64 kbit/s que se han de transportar en el tren.
Valor del identificador del parámetro	1
Estado del parámetro	Opcional, el valor por defecto es 1.
Tipo de parámetro	unsignedMax. Estará en la gama 1..255. unsignedMin (en intercambio de capacidad solamente).
Sustituye a	–

**Cuadro Q.3/H.245 – Parámetro tamaño de cabida útil Nx64**

Nombre del parámetro	Tamaño de cabida útil
Descripción del parámetro	Éste es un parámetro genérico collapsing. El tamaño de cabida útil especifica el número de tramas que se han de transportar en un paquete del tren.
Valor del identificador del parámetro	2
Estado del parámetro	unsignedMax es obligatorio. unsignedMin es opcional.
Tipo de parámetro	unsignedMax. Estará en la gama 1..2047. unsignedMin (en intercambio de capacidad solamente).
Sustituye a	–

El tamaño máximo de bloque en tramas es el único parámetro que se debe especificar y, en todos los casos, si el valor mínimo no se especifica tomará por defecto el valor máximo. Esto permite la especificación de una amplia gama de combinaciones con una capacidad simple. **GenericCapability.maxBitRate** se puede fijar al valor 640 del máximo número de canales.

## Q.2.4 Parámetros para capacidades en el establecimiento de canal

### Q.2.4.1 En H.245

Cuando se utiliza como **DataType** en un **OpenLogicalChannel** H.245 u **OpenLogicalChannelAck** o como **mode** en un **RequestMode**, la capacidad genérica Nx64 no incluirá ningún valor mínimo; es decir, todos los parámetros especificados incluirán los valores deseados para el canal de datos especificado. Estos valores deberán, por supuesto, ser compatibles con las capacidades ya intercambiadas.

### Q.2.4.2 En conexión rápida o conexión rápida ampliada

Cuando se utiliza como **DataType** de medios en una propuesta **OpenLogicalChannel** H.245 contenido en un elemento **fastStart** H.225.0 [12], la capacidad genérica Nx64 debe contener sólo los valores máximos de los parámetros a fin de indicar los valores reales solicitados. Cualquier parámetro omitido se ofrece implícitamente sólo con el valor por defecto.

## Q.2.5 Formato del paquete

El formato del paquete será RTP [83], con la cabida útil comprendida por una o más tramas en ordenación de bytes de la red, junto con los elementos de redundancia negociados, si los hubiera. Una trama se define como el agregado de una muestra de 8 bits de cada uno de los N canales TDM en alguna ordenación fija (el orden viene determinado por la red con conmutación de circuitos). Este formato de trama es idéntico al formato de encapsulación enganchada a la estructura (sin señalización) especificada en 9.2.1/Y.1413, figura Q.1 que se reproduce más abajo. Esta alineación de trama evita el conocimiento de cualquier estructura interna de los datos, sin embargo retiene la sincronización del tren de paquetes con el circuito de origen. El tamaño de la trama (M) permanecerá constante durante el tiempo de la conexión.

8	7	6	5	4	3	2	1	Trama
Bits pertenecientes al intervalo de tiempo 1								1
Bits pertenecientes al intervalo de tiempo 2								
...								
Bits pertenecientes al intervalo de tiempo N								2
Bits pertenecientes al intervalo de tiempo 1								
Bits pertenecientes al intervalo de tiempo 2								
...								...
Bits pertenecientes al intervalo de tiempo N								
...								
Bits pertenecientes al intervalo de tiempo 1								M
Bits pertenecientes al intervalo de tiempo 2								
...								
Bits pertenecientes al intervalo de tiempo N								

NOTA 1 – El bit 8 es el bit más significativo.

NOTA 2 – El paquete contiene M tramas TDM con N intervalos de tiempo por trama.

**Figura Q.1/H.245 – Formato de cabida útil para encapsulación enganchada a la estructura sin CAS (según la Rec. Y.1413)**

## **Q.2.6 Restricciones del encabezamiento RTP**

Las siguientes limitaciones son adaptadas de 8.4/Y.1413 y se han de observar en el formateo de paquetes conforme a esta capacidad.

- 1) No se utiliza relleno, ampliación de encabezamiento, fuentes de sincronización múltiples ni marcadores.
- 2) Los tipos de cabida útil se han de seleccionar de la gama dinámica.
- 3) Los números de secuencia serán consecutivos para paquetes consecutivos; conjuntamente con el tamaño de cabida útil fijo, esto permite a un receptor calcular la cantidad exacta de tramas perdidas por un paquete perdido.
- 4) La indicación de tiempo RTP conjuntamente con el tamaño y la velocidad del paquete se puede utilizar para transportar información de temporización sobre la red IP; la frecuencia de reloj utilizada para generar indicaciones de tiempo debe ser un múltiplo entero de 8 kHz. Las indicaciones para la selección adecuada de esta frecuencia de reloj figuran en el apéndice V/Y.1413.
- 5) El campo fuente de sincronización en el encabezamiento RTP se puede utilizar para la detección de conexiones erróneas.

## **Q.2.7 Formateo de redundancia**

Esta capacidad permite efectuar diversos métodos facultativos de corrección de errores a través de redundancia o corrección de errores en recepción. Como parte del intercambio de capacidad, se pueden especificar uno o varios métodos que utilizan capacidades H.245 normalizadas.

## **Q.2.8 Consideraciones sobre temporización**

El transporte de información síncrona a través de la red IP está sujeto a diversos tipos de problemas. El transporte del protocolo Internet experimenta problemas de fluctuación de fase en los tiempos de propagación de paquetes, que pueden ser compensados mediante la introducción de un retardo adicional (almacenamiento en memoria tampón en el receptor). Los mecanismos de corrección de errores pueden proporcionar redundancia.

La temporización en los TDM de origen y de destino también afecta el funcionamiento de extremo a extremo de la conexión. Si los sistemas TDM de origen y de destino funcionan con relojes diferentes se producirá una deriva relativa entre ellos que conducirá a un rebasamiento o insuficiencia de datos. La aparición de dichos eventos se puede limitar en parte con el almacenamiento en memoria tampón, pero esto producirá, tarde o temprano, la pérdida de alguna información o la inserción de datos no significativos. Se pueden considerar diversas situaciones con relojes diferentes.

## **Q.2.9 Reloj común**

Los sistemas de origen y de destino están vinculados a un reloj de red común. En estas circunstancias no se deberían producir rebasamientos o insuficiencia de datos, de modo tal que el transporte de paquetes se efectúa satisfactoriamente.

### **Q.2.9.1 Relojes independientes**

En este caso el emisor y el receptor funcionan en velocidades diferentes, pero la deriva de un reloj con respecto al otro se puede compensar durante un tiempo mediante almacenamiento en memoria tampón. La utilización de relojes calibrados precisos puede reducir la aparición de deslizamientos a un nivel satisfactorio, dependiendo de la aplicación que utiliza el transporte.

## Q.2.9.2 Capacidad portadora

Este formato se puede utilizar para transportar datos a 64 kbit/s o 56 kbit/s con o sin restricciones incorporados en un canal a 64 kbit/s, de acuerdo con la capacidad portadora Q.931. Para la codificación de la capacidad portadora en el mensaje establecimiento véase 7.2.2.1/H.225.0.

# Apéndice I

## Visión general de la sintaxis en ASN.1

### I.1 Introducción a la ASN.1

La notación de sintaxis abstracta uno (ASN.1, *abstract syntax notation one*) es un lenguaje de especificación de datos. Se normalizó inicialmente como parte de las Recomendaciones de la serie X.400 relativas al correo electrónico como Rec. UIT-T X.409. Este lenguaje evolucionó a la forma en que figura en la Rec. UIT-T X.208 y más recientemente en la Rec. UIT-T X.680. La ASN.1 permite una especificación inequívoca de estructuras de datos complejas incluidas las que incluyen campos de longitud variable, campos facultativos y recursión.

Las Recomendaciones antes mencionadas tratan solamente la sintaxis y la semántica de especificaciones ASN.1. La codificación binaria de las estructuras de datos se trata en otras Recomendaciones, sobre todo en la Rec. UIT-T X.690 (reglas de la codificación básica o BER (*basic encoding rules*)) y X.691 (reglas de codificación compactada o PER (*packed encoding rules*)). La BER permite que los datos sean descifrados por sistemas que tienen un conocimiento general de ASN.1 pero no conocen los detalles de la especificación utilizada para formar los datos. En otras palabras, los tipos de datos se codifican junto con los valores de datos. La PER es mucho más eficiente pues sólo se codifican/decodifican valores de datos y la codificación está diseñada con muy poca redundancia. Este método puede utilizarse cuando tanto el transmisor como el receptor confían en que los datos se ajustarán a alguna estructura conocida.

La presente Recomendación se implementa utilizando reglas de codificación compactada. Puesto que ambos lados de la llamada saben que los mensajes serán conformes con la especificación H.245, no es necesario codificar esa especificación para formar mensajes. Para simplificar la decodificación, se utiliza la variante alineada de la PER. En esta variante, los campos que necesitan ocho o más bits tienen que estar alineados con fronteras de octetos y ocupar un número entero de octetos. La alineación se consigue rellenando los datos con ceros antes de campos grandes.

### I.2 Tipos de datos ASN.1 básicos

El tipo de datos más simple es el BOOLEAN, que representa los valores FALSE (FALSO) y TRUE (VERDADERO) los cuales se codifican mediante un único bit como 0 y 1, respectivamente. Por ejemplo, el tipo de datos `segmentableFlag` BOOLEAN se codifica como sigue:

Valor	Codificación
FALSO	0
VERDADERO	1

El tipo de datos fundamental es INTEGER, que representa valores numéricos enteros. Los enteros pueden no estar limitados como en:

`bitRate`

`INTEGER`

o pueden estar restringidos a una gama de valores, por ejemplo:

**maximumA12SDUSize** **INTEGER (0..65535)**

Los enteros limitados se codifican de distinta forma en función del tamaño de la gama. Si N es el número de enteros de la gama, esto es, el límite superior menos el límite inferior más uno, dependiendo de los valores de N el entero limitado se codificará de alguna de las siguientes formas:

N	Codificación
1	No se necesitan bits
2-255	Campo no alineado de 1 a 8 bits
256	Campo alineado de 8 bits
257-65536	Alineado de 16 bits
mayor que 65536	Número mínimo de octetos alineados precedido de la codificación anterior del número de octetos

En todos los casos, el número utilizado realmente es el valor que debe codificarse menos el límite inferior de la gama. En estos ejemplos "relleno" representa el conjunto de cero a siete bits 0 que debe añadirse a la codificación de forma que el campo siguiente comience con un límite de 8 bits.

**firstGOB** **INTEGER (0..17)**

Valor	Codificación
0	00000
3	00011

**h233IVResponseTime** **INTEGER (0..255)**

Valor	Codificación
3	relleno 00000011
254	relleno 11111110

**skew** **INTEGER (0..4095)**

Valor	Codificación
3	relleno 00000000 00000011
4095	relleno 00001111 11111111

Los valores enteros no limitados (complemento a 2) que se pueden representar con 127 octetos o menos se codifican con el número mínimo de octetos necesario. Se codifica el número de octetos (longitud) en forma de un octeto alineado que precede al propio número. Por ejemplo:

-1	relleno 00000001 11111111
0	relleno 00000001 00000000
128	relleno 00000010 00000000 10000000
1000000	relleno 00000011 00001111 01000010 01000000

La notación ASN.1 soporta una gran variedad de tipos de datos de cadena. Se trata de listas de bits de longitud variable, octetos y otros tipos de datos cortos. Estas estructuras se codifican típicamente



en forma de una longitud seguida de los datos. La longitud puede codificarse como un entero no limitado o un entero limitado si se especifica el tamaño de la cadena. Por ejemplo,

```
data                OCTET STRING
```

Como la longitud de la cadena de octetos no está limitada, deberá codificarse como un *número entero semilimitado* (con límite inferior, pero sin límite superior). En primer lugar se rellenan los datos para que quede alineada la codificación. El resto del código será como sigue:

Longitud	Codificación
0 a 127	Longitud de 8 bits seguida de los datos
128 a 16K-1	Longitud de 16 bits con el bit más significativo (MSB) activado, seguido de los datos
16K a 32K-1	11000001, 16K octetos de datos, a continuación codificación del resto
32K a 48K-1	11000010, 32K octetos de datos, a continuación codificación del resto
48K a 64K-1	11000011, 48K octetos de datos, a continuación codificación del resto
64K o mayor	11000100, 64K octetos de datos, a continuación codificación del resto

Se denomina a este método "fragmentación". Obsérvese que si la longitud es un múltiplo de 16K, la representación terminará con un octeto de ceros indicativo de una cadena de longitud cero.

### I.3 Tipos de datos agregados

La ASN.1 comprende varios tipos de datos agregados o de contenedor que son conceptualmente similares a los tipos vector, estructuración y unión de C. Tales tipos son, respectivamente, CHOICE, SEQUENCE y SEQUENCE OF. En todos los casos se codifican con algunos bits específicos del contenedor seguidos de las codificaciones normales de los contenidos.

Se utiliza CHOICE para seleccionar exactamente un tipo de datos de un grupo de tipos. Por ejemplo:

```
VideoCapability ::= CHOICE
{
    nonStandard          NonStandardParameter,
    h261VideoCapability  H261VideoCapability,
    h262VideoCapability  H262VideoCapability,
    h263VideoCapability  H263VideoCapability,
    is11172VideoCapability IS11172VideoCapability,
    ...
}
```

A cada elección se le asigna un número índice comenzando por cero. El índice de la selección en curso se codifica en forma de entero limitado. El índice va seguido de la codificación de la selección en curso, la cual no existe si la selección es NULL. Si está presente el indicador de ampliación (como más arriba), el índice va precedido de un bit que es cero si la selección en curso procede de la lista original.

SEQUENCE es, sencillamente, una agrupación de tipos de datos diferentes. Los elementos individuales de la secuencia pueden ser OPTIONAL. La codificación es muy sencilla. Si hay un indicador de ampliación, el primer bit señala la presencia de elementos adicionales. Va seguido de una serie de bits uno para cada elemento facultativo que indica que ese dato está presente. A continuación va seguido de las codificaciones de los componentes de la secuencia. Por ejemplo:

```
H261VideoCapability ::= SEQUENCE
{
    qcifMPI    INTEGER (1..4) OPTIONAL, -- units 1/29.97 Hz
    cifMPI     INTEGER (1..4) OPTIONAL, -- units 1/29.97 Hz
}
```

```

temporalSpatialTradeOffCapability    BOOLEAN,
...
}

```

La codificación asigna un bit al indicador de ampliación, dos bits a los campos facultativos, dos bits a cada uno de los campos facultativos que esté presente, un bit para la variable booleana y seguidamente los datos de ampliación. Obsérvese que en esta secuencia no existe relleno para la alineación de octetos.

Los tipos SEQUENCE OF y SET OF describen una colección de componentes similares (un vector). El tipo SEQUENCE OF implica que el orden de los elementos es significativo en tanto que en el caso de SET OF el orden de los elementos es arbitrario. La codificación PER es la misma para ambos tipos.

Estos tipos pueden tener un número de elementos no limitado o limitado por SIZE. Si el número se conoce *a priori* y es inferior a 64K, no se codifica. En cualquier otro caso se codifica el número real de componentes con una longitud limitada o semilimitada. A continuación se codifican los datos. Si la longitud es de 16K como mínimo y está codificada, la lista de datos se subdividirá en fragmentos como en el caso de la cadena de octetos. En este caso se subdividen los fragmentos tras cierto número de campos componentes (16K, 32K, etc.) y no tras cierto número de octetos.

#### I.4 Tipo identificador de objeto

Normalmente, el tipo de un valor se da en la especificación ASN.1, por lo que la única información que es necesario codificar y transmitir es el dato propiamente dicho. Sin embargo, a veces es deseable codificar el tipo de datos así como el valor del dato. Por ejemplo, **protocolIdentifier** contiene:

```

protocolIdentifier    OBJECT IDENTIFIER,
-- shall be set to the value
-- {itu-t (0) recommendation (0) h (8) 245 version (0) 1}

```

Se codifican todos los enteros que figuran entre llaves {}, estén o no entre paréntesis (). En este ejemplo, se codifican los enteros 0, 0, 8, 245, 0, 1.

Se codifica como los datos codificados con la BER (X.690) precedidos por la longitud de esa codificación, en octetos. La longitud se codifica como un número entero semiconstreñido (véase el anterior ejemplo de OCTET STRING). A continuación se explica su codificación.

El primer octeto indica la longitud de la codificación que sigue.

Los dos primeros componentes del identificador de objeto se combinan entre sí según la expresión  $40 \times \text{primero} + \text{segundo}$ , en este caso  $40 \times 0 + 0 = 0$ . Los otros se codifican tal como son. Cada uno se codifica en una serie de octetos, el primero de los cuales indica si siguen más o no. Así:

0 → 0000 0000

8 → 0000 1000

mientras que 245, por ser mayor que 127 se codifica 1000 0001 0111 0101.

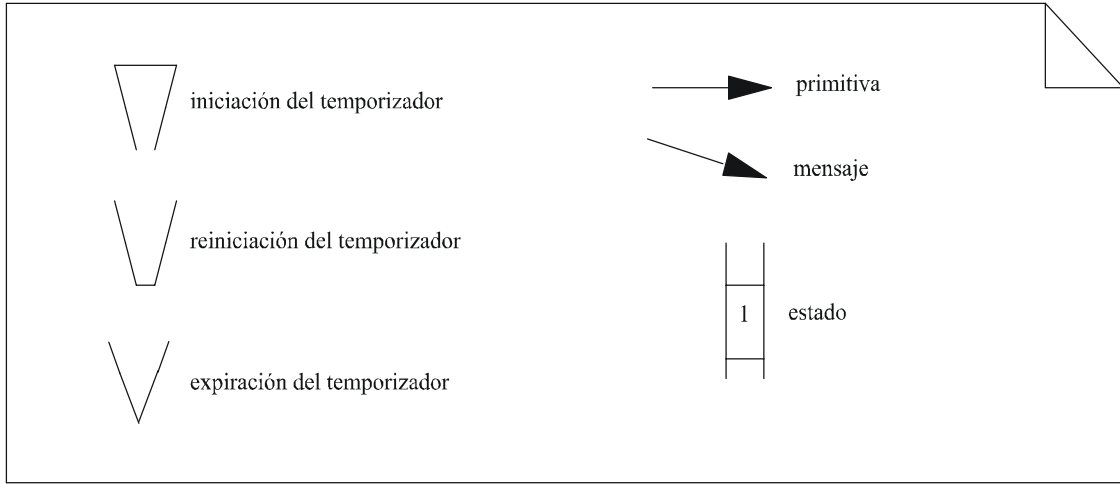
Por tanto, la codificación completa en hexadecimal consta de siete octetos 06000881 750001.

## Apéndice II

### Ejemplos de procedimientos H.245

#### II.1 Introducción

Este apéndice ilustra ejemplos de los procedimientos definidos en el anexo C. La figura II.1-1 muestra los símbolos gráficos utilizados en los diagramas de este apéndice.



H.245\_FII.1-1

**Figura II.1-1/H.245 – Clave de las figuras**

#### II.2 Entidad de señalización de determinación principal-subordinado

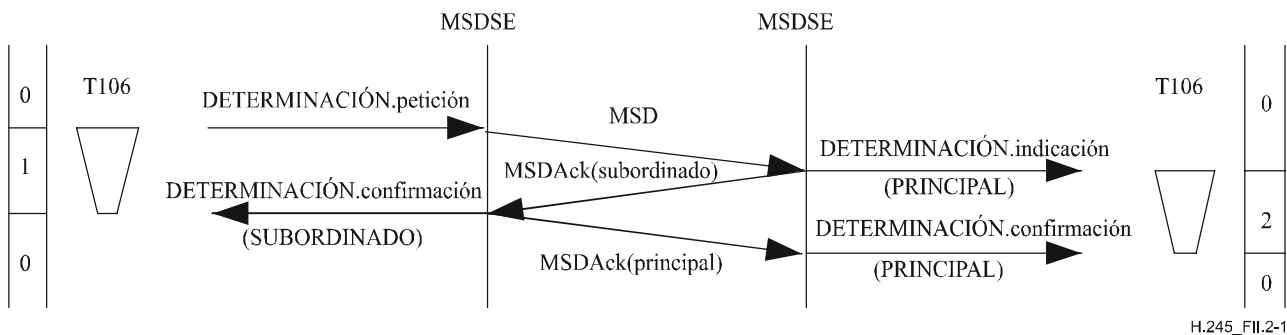
En las figuras II.2-1 a II.2-10 los mensajes se representan por los nombres abreviados indicados en el cuadro II.2-1.

**Cuadro II.2-1/H.245 – Nombres abreviados de determinación principal-subordinado**

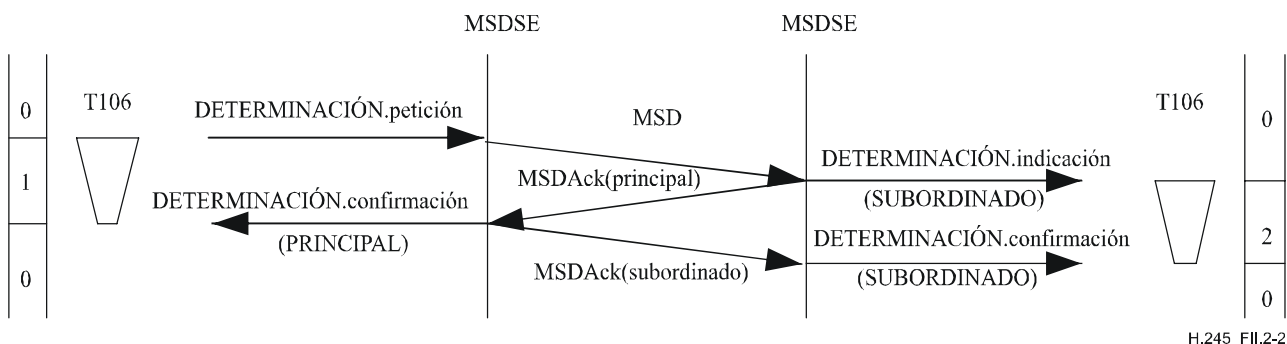
Mensaje	Nombre en los ejemplos
MasterSlaveDetermination	MSD
MasterSlaveDeterminationAck	MSDAck
MasterSlaveDeterminationReject	MSDReject
MasterSlaveDeterminationRelease	MSDRelease

En las figuras II.2-1 a II.2-10, REPOSO, ESPERA DE RESPUESTA DE SALIDA y ESPERA DE RESPUESTA DE ENTRADA se designan por "0", "1" y "2", respectivamente.

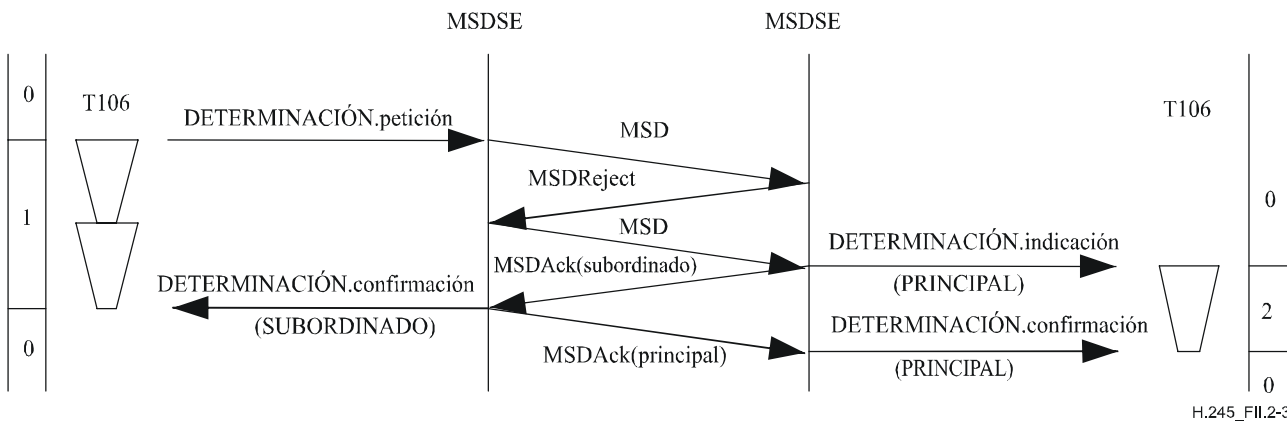
En las figuras siguientes el valor de parámetro asociado con las primitivas DETERMINACIÓN.indicación y DETERMINACIÓN.confirmación es la del parámetro TIPO. El valor de campo asociado con el mensaje MasterSlaveDeterminationAck es el del campo de decisión.



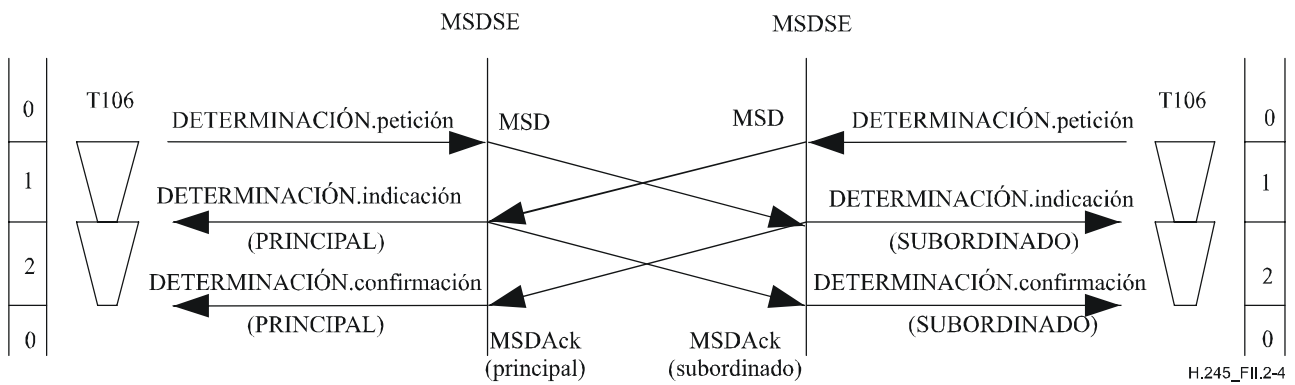
**Figura II.2-1/H.245 – Determinación principal-subordinado – Principal en la MSDSE distante**



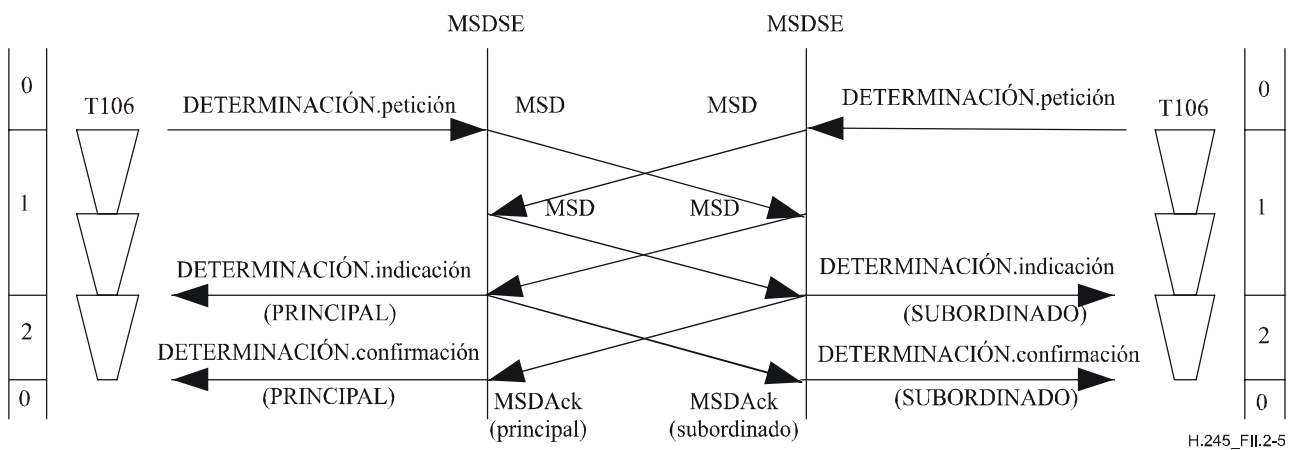
**Figura II.2-2/H.245 – Determinación principal-subordinado – Subordinado en la MSDSE distante**



**Figura II.2-3/H.245 – Determinación principal-subordinado – El primer intento produjo un resultado indeterminado. El segundo intento tuvo éxito**

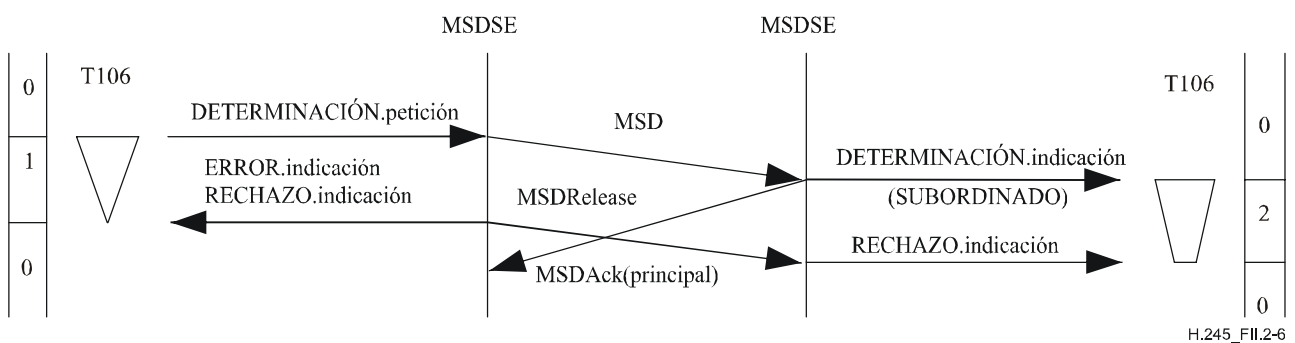


**Figura II.2-4/H.245 – Determinación principal-subordinado – Determinación simultánea**



**Figura II.2-5/H.245 – Determinación principal-subordinado – Determinación simultánea, pero el primer intento produce un resultado indeterminado**

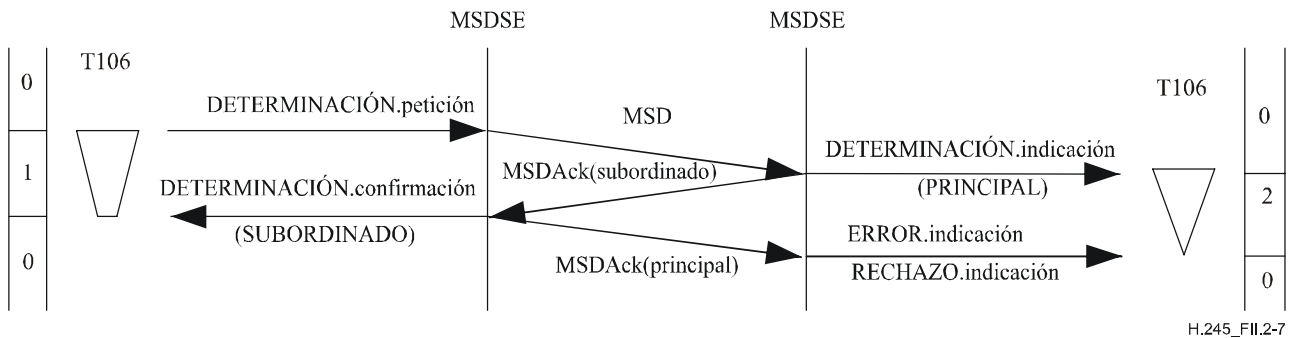
En la figura II.2-6, ha expirado el temporizador local T106. Sólo el terminal de la derecha conoce su categoría. El terminal de la derecha es capaz de recibir nuevas instrucciones pero no puede solicitar nada del otro terminal que se base en el conocimiento del resultado de la determinación de categoría. El terminal de la izquierda no puede ni aceptar ni iniciar nuevos procedimientos. Debe iniciarse un segundo procedimiento de determinación de categoría.



**Figura II.2-6/H.245 – Determinación principal-subordinado – Expiración del temporizador local T106 con subordinado en el extremo distante**

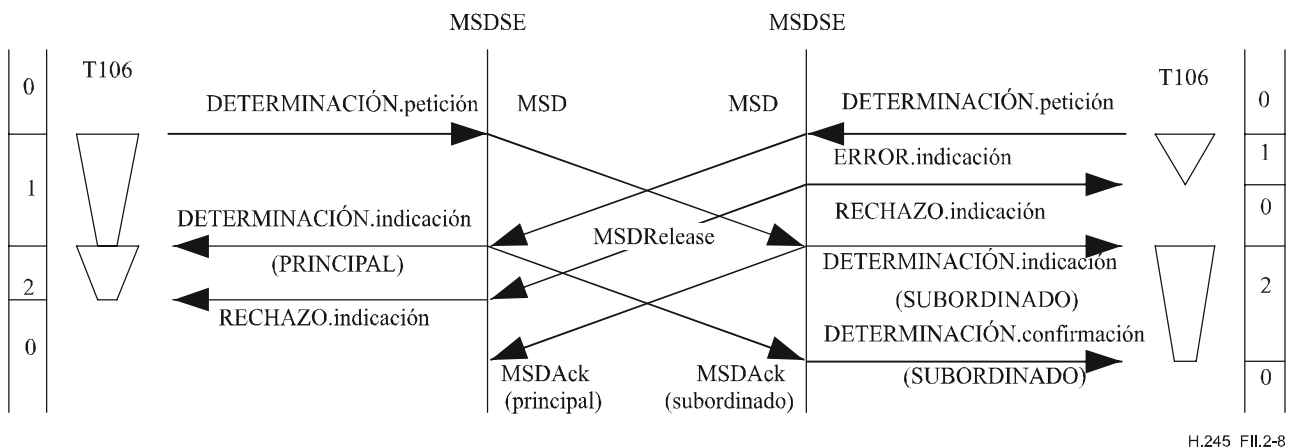
En la figura II.2-7, ha expirado el temporizador distante T106 durante el estado ESPERA DE ACUSE DE RECIBO EN ENTRADA. Ambos terminales conocen su categoría. El terminal de la

izquierda puede recibir y emitir instrucciones. Sin embargo, el terminal distante no sabe si el terminal local está listo para recibir y no puede emitir instrucciones que se basen en el conocimiento del resultado de la determinación de categoría. Debe iniciarse un segundo procedimiento de determinación de categoría.



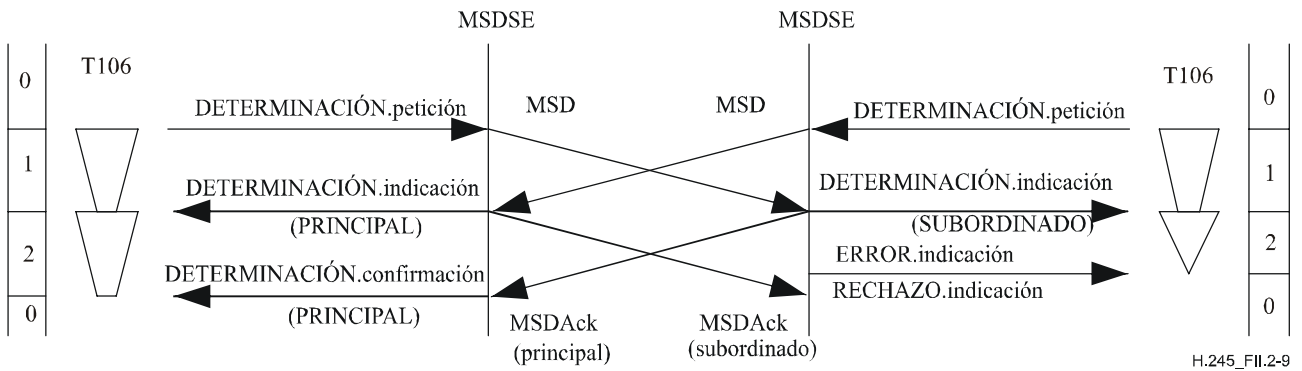
**Figura II.2-7/H.245 – Determinación principal-subordinado – Expiración del temporizador distante T106 con director en el extremo distante**

En la figura II.2-8 ha expirado el temporizador distante T106 durante el estado ESPERA DE ACUSE DE RECIBO EN SALIDA durante un procedimiento de determinación simultánea. Ambos terminales conocen su categoría. El terminal de la derecha puede recibir y emitir instrucciones. Sin embargo, el terminal de la izquierda no sabe si el otro terminal está listo para recibir y no puede emitir instrucciones que se basen en el conocimiento del resultado de la determinación de categoría. Puede recibir tales instrucciones. Debe iniciarse un segundo procedimiento de determinación de categoría.



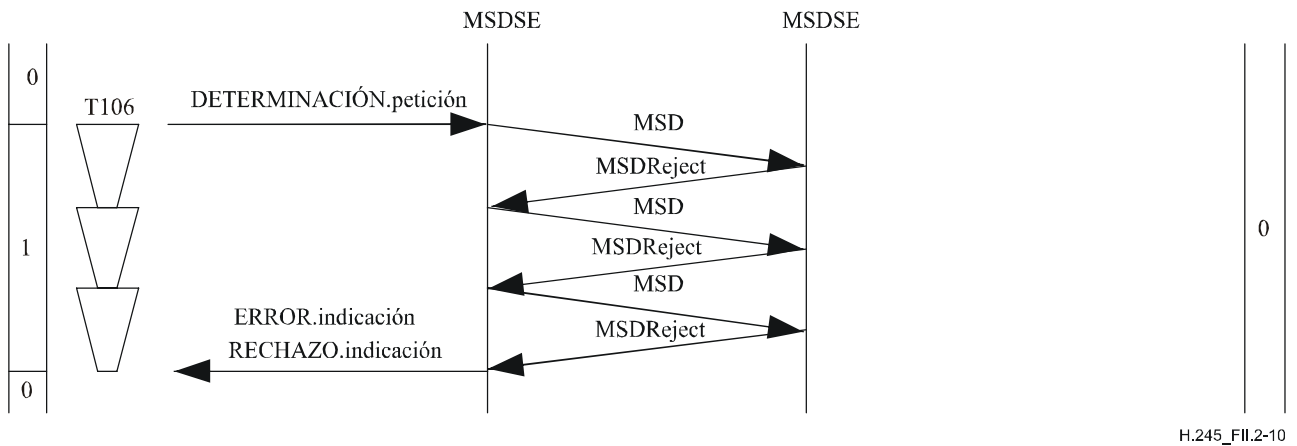
**Figura II.2-8/H.245 – Determinación principal-subordinado – Procedimientos de determinación simultánea con expiración del temporizador T106 en el subordinado**

En la figura II.2-9, ha expirado el temporizador distante T106 durante una estado ESPERA DE ACUSE DE RECIBO EN ENTRADA, durante un procedimiento de determinación simultánea. Ambos terminales conocen su categoría. El terminal de la izquierda puede recibir y emitir instrucciones. Sin embargo, el terminal de la derecha no sabe si el otro terminal está listo para recibir y no puede emitir instrucciones que se basen en el conocimiento del resultado de la determinación de categoría. Puede recibir tales instrucciones. Debe iniciarse un segundo procedimiento de determinación de categoría.



**Figura II.2-9/H.245 – Determinación principal-subordinado – Procedimientos de determinación simultánea con expiración del temporizador T106 durante el estado ESPERA DE ACUSE DE RECIBO EN ENTRADA**

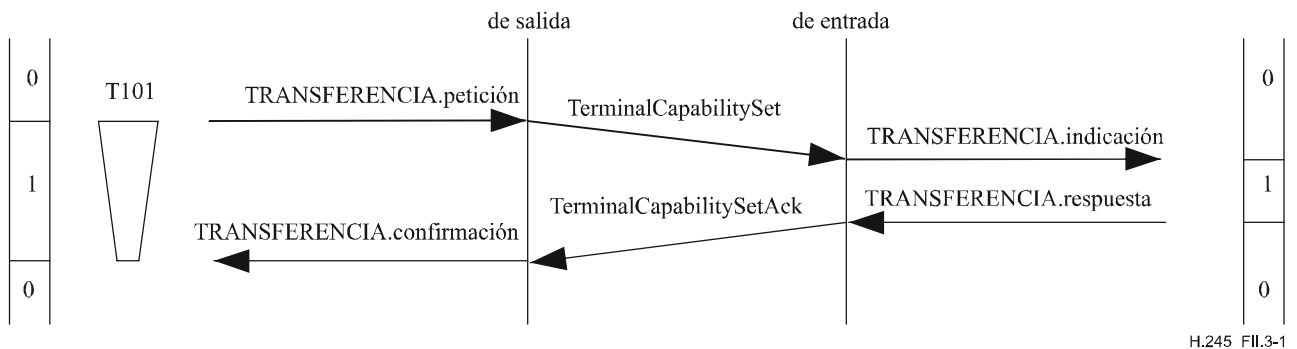
En la figura II.2-10 se obtuvo un resultado indeterminado N100 veces. En este caso,  $N100 = 3$ .



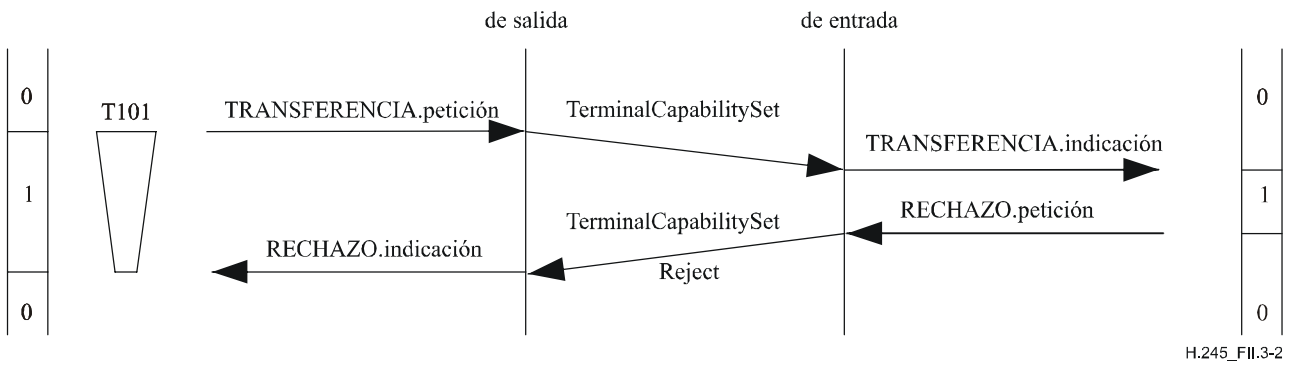
**Figura II.2-10/H.245 – Determinación principal-subordinado – Resultado indeterminado con  $N100 = 3$**

### II.3 Entidad de señalización de intercambio de capacidad

Las figuras II.3-1 a II.3-4 ilustran los procedimientos de la CESE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1", respectivamente.

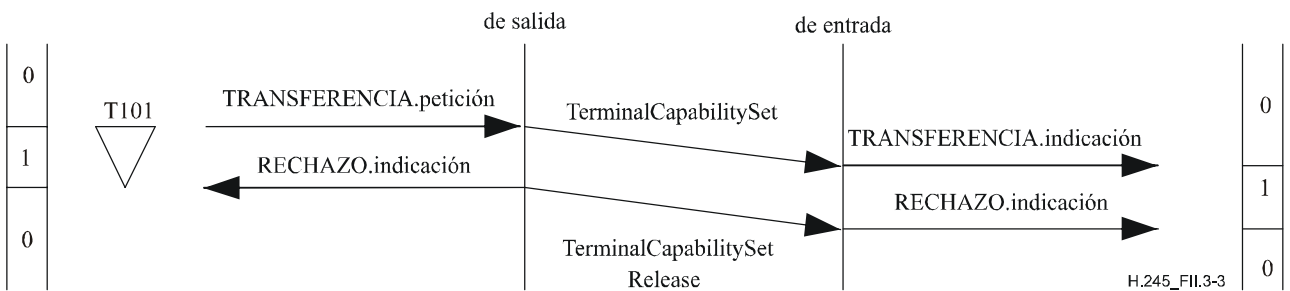


**Figura II.3-1/H.245 – Intercambio de capacidades con aceptación por el usuario de la CESE de entrada par**



H.245\_FII.3-2

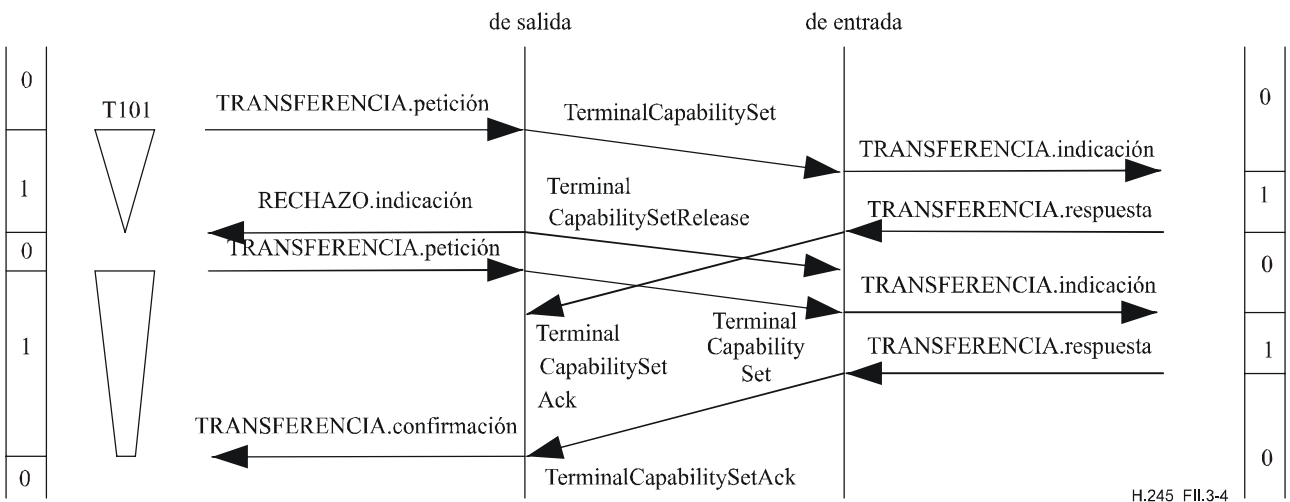
**Figura II.3-2/H.245 – Intercambio de capacidades con rechazo por el usuario de la CESE de entrada par**



H.245\_FII.3-3

NOTA – El mensaje TerminalCapabilitySetRelease llega a la CESE de entrada antes de la respuesta del usuario de la CESE de entrada.

**Figura II.3-3/H.245 – Intercambio de capacidades con expiración del temporizador T101**



H.245\_FII.3-4

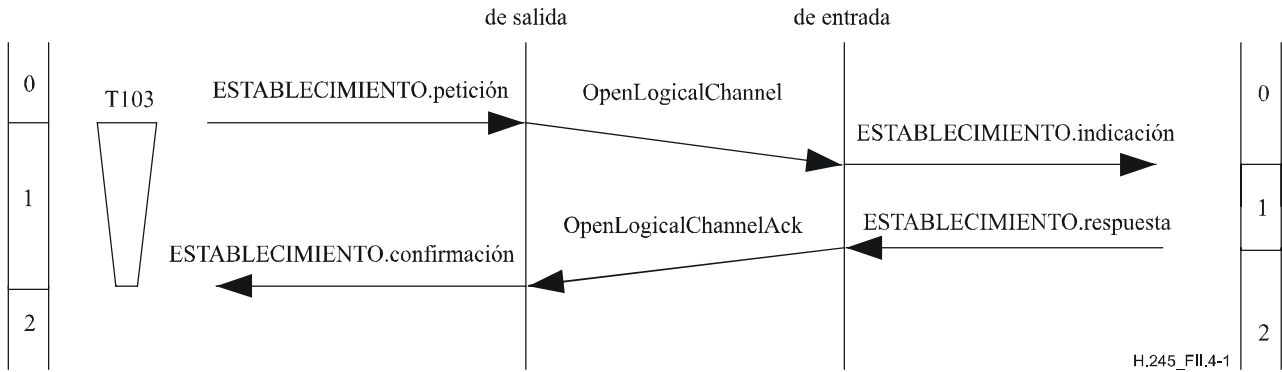
NOTA – El mensaje TerminalCapabilitySetRelease llega a la CESE de entrada después de la respuesta del usuario de la CESE de entrada. En la CESE de salida el mensaje TerminalCapabilitySetAck en respuesta al primer mensaje TerminalCapabilitySet es ignorado. Sólo tiene éxito el segundo intercambio de capacidad.

**Figura II.3-4/H.245 – Intercambio de capacidades con expiración del temporizador T101 seguido de un segundo intercambio de capacidad**

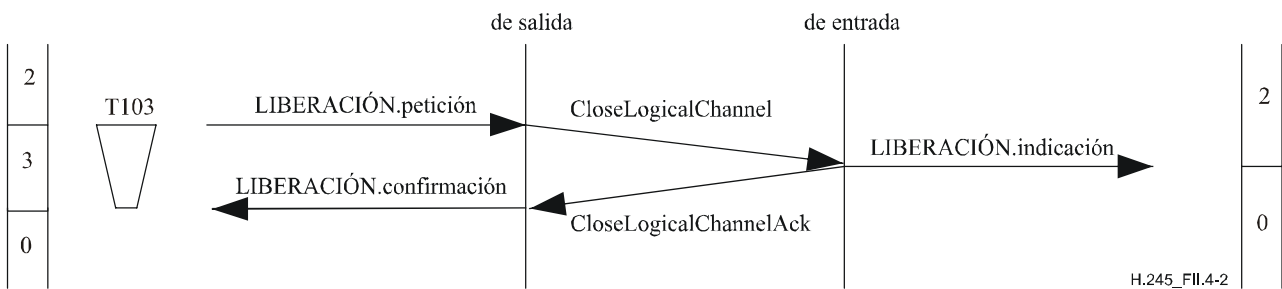


## II.4 Entidad de señalización de canal lógico

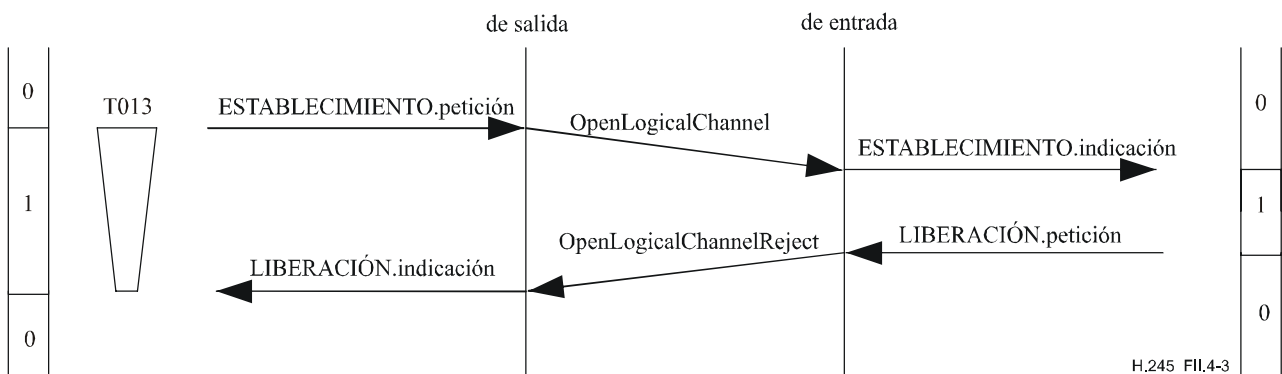
Las figuras II.4-1 a II.4-7 ilustran los procedimientos de la LCSE. Los estados de la LCSE de salida LIBERADO, ESPERA DE ESTABLECIMIENTO, ESTABLECIDO y ESPERA DE LIBERACIÓN se designan por "0", "1", "2" y "3" respectivamente. Los estados de la LCSE de entrada LIBERADO, ESPERA DE ESTABLECIMIENTO y ESTABLECIDO se designan por "0", "1" y "2", respectivamente.



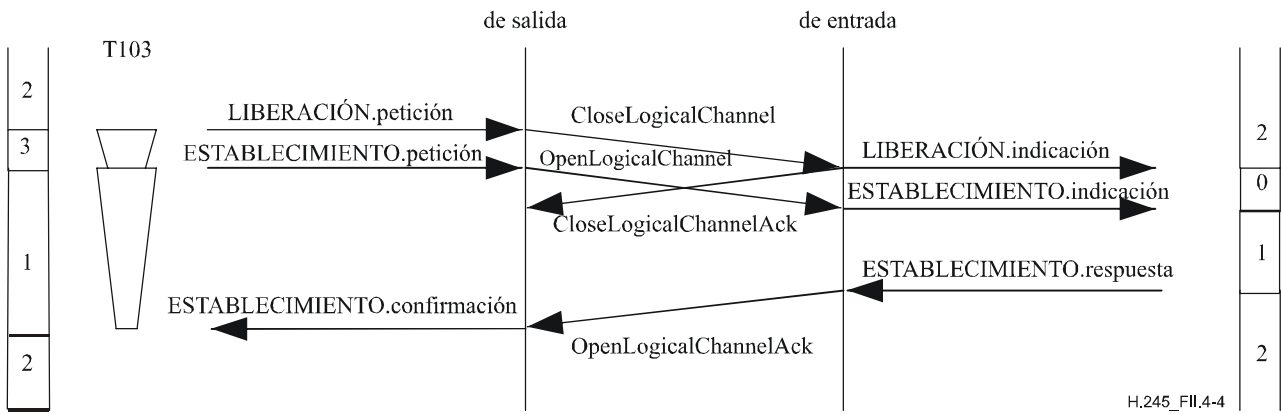
**Figura II.4-1/H.245 – Establecimiento de canal lógico**



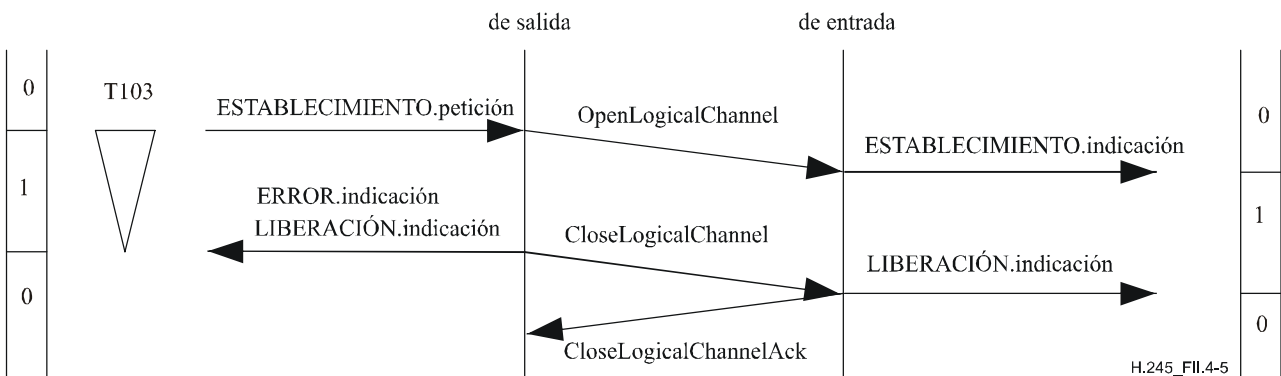
**Figura II.4-2/H.245 – Liberación de canal lógico**



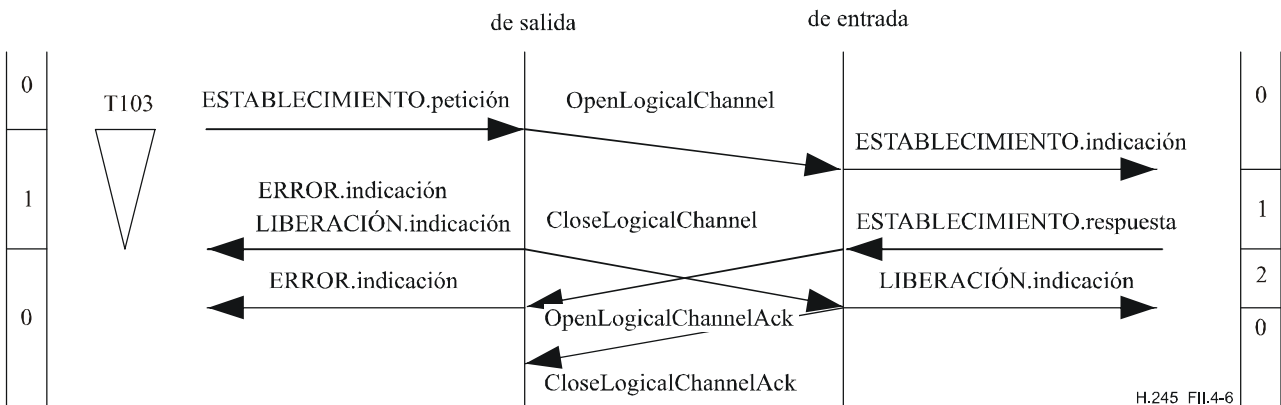
**Figura II.4-3/H.245 – Rechazo de establecimiento de canal lógico por el usuario de la LCSE par**



**Figura II.4-4/H.245 – Liberación de canal lógico seguida de nuevo establecimiento inmediato**

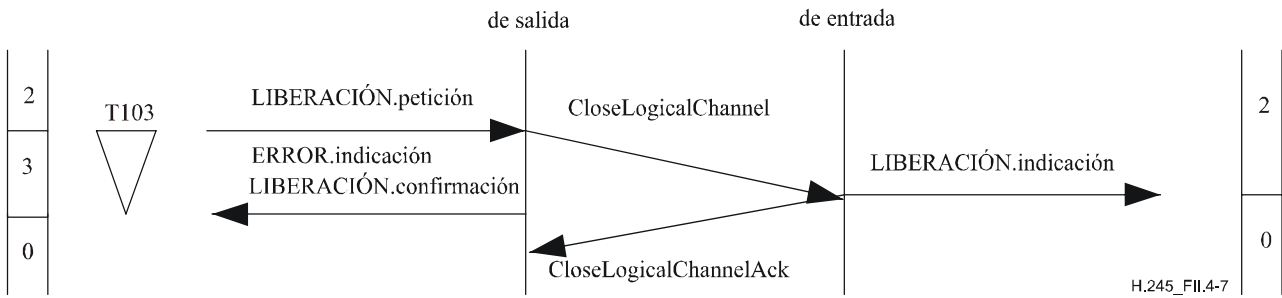


**Figura II.4-5/H.245 – Petición de establecimiento de canal lógico con expiración del temporizador T103 debido a la respuesta lenta del usuario de la LCSE de entrada par**



NOTA – El temporizador T103 ha expirado después de la transmisión del mensaje OpenLogicalChannelAck en la LCSE de entrada, pero antes de la recepción del mensaje OpenLogicalChannelAck en la LCSE de salida.

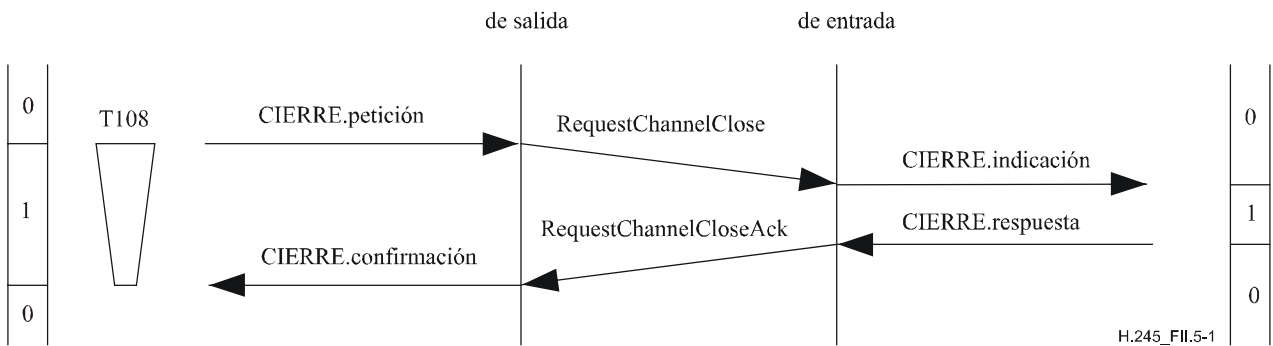
**Figura II.4-6/H.245 – Petición de establecimiento de canal lógico con expiración del temporizador T103**



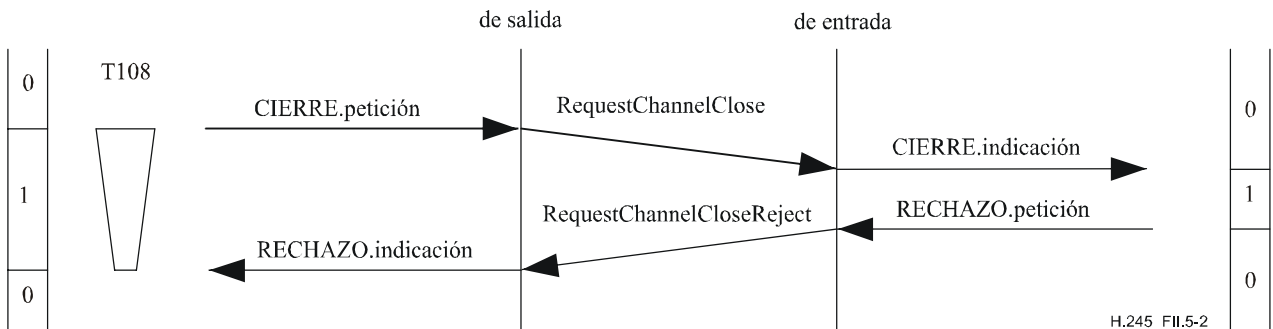
**Figura II.4-7/H.245 – Petición de liberación de canal lógico con expiración del temporizador T103**

## II.5 Entidad de señalización de cierre de canal lógico

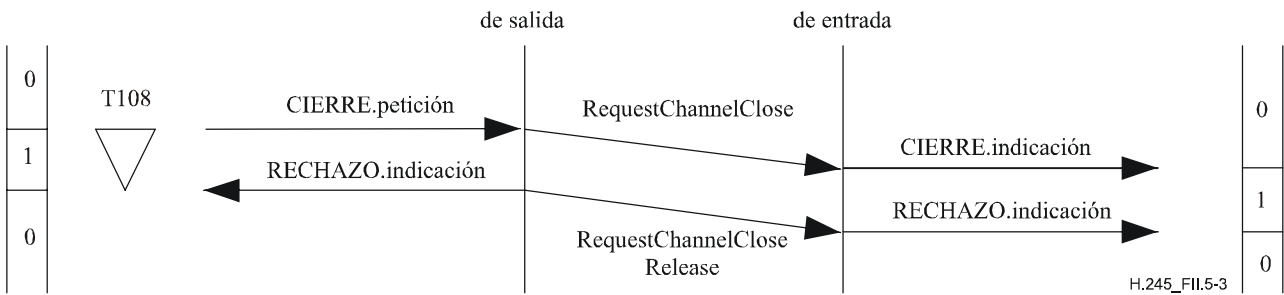
Las figuras II.5-1 a II.5-4 ilustran los procedimientos de la CLCSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1", respectivamente.



**Figura II.5-1/H.245 – Petición de cierre de canal lógico**

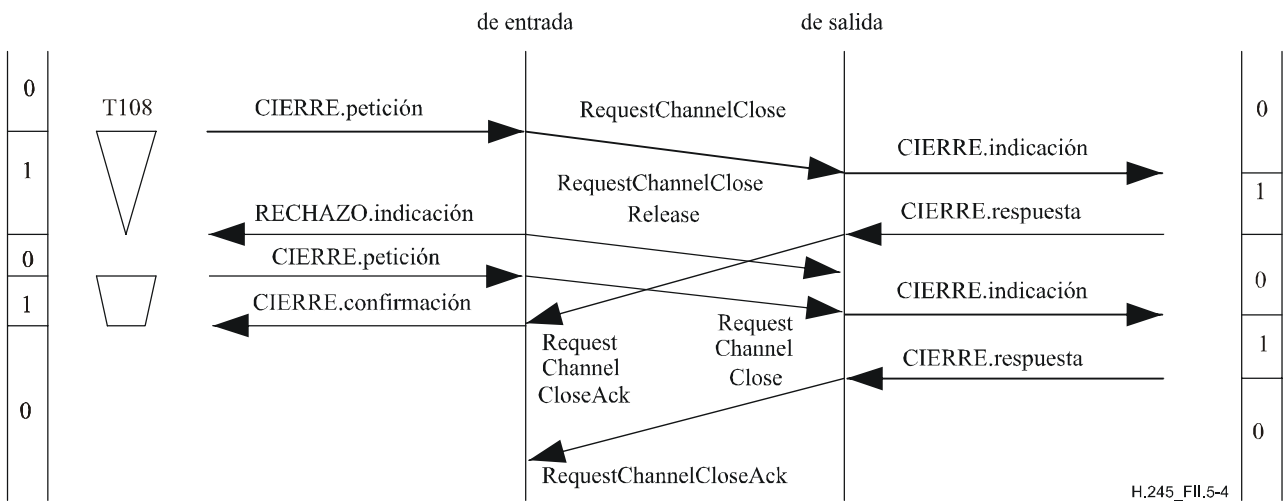


**Figura II.5-2/H.245 – Petición de cierre de canal lógico con rechazo por el usuario de la CLCSE de entrada par**



NOTA – El mensaje RequestChannelCloseRelease llega a la CLCSE de entrada antes que la respuesta del usuario de la CLCSE de entrada.

**Figura II.5-3/H.245 – Petición de cierre de canal lógico con expiración del temporizador T108**

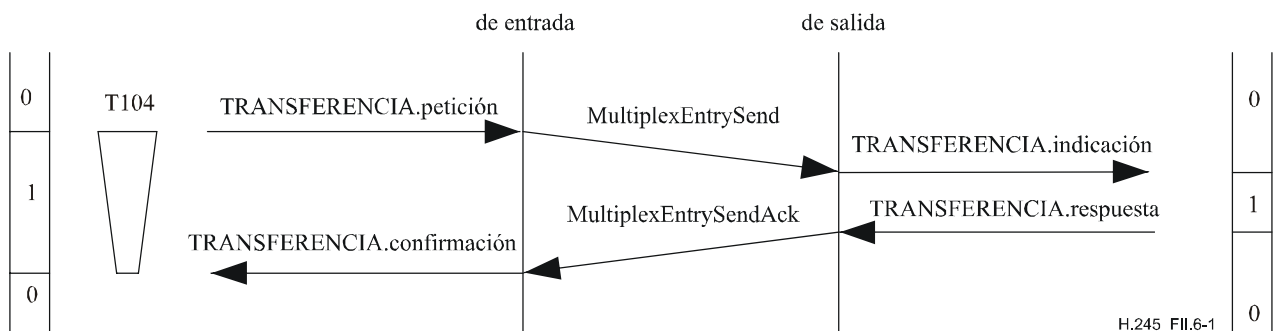


NOTA – La petición de cierre de canal es confirmada al recibo del primer mensaje RequestChannelClose.

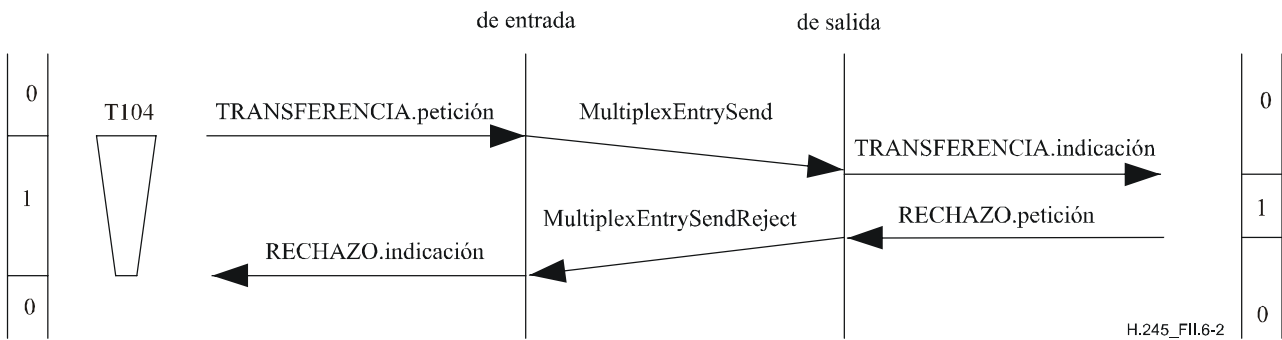
**Figura II.5-4/H.245 – Petición de cierre de canal lógico con expiración del temporizador T108 seguida de una segunda petición de cierre de canal lógico**

## II.6 Entidad de señalización de tabla múltiplex

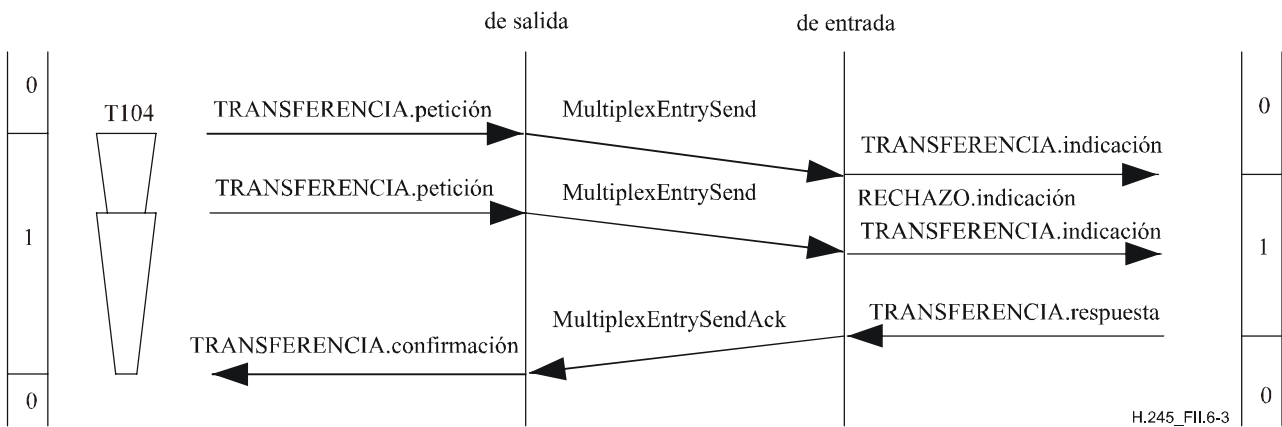
Las figuras II.6-1 a II.6-5 ilustran los procedimientos de la MTSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1", respectivamente.



**Figura II.6-1/H.245 – Petición de envío de tabla múltiplex con éxito**

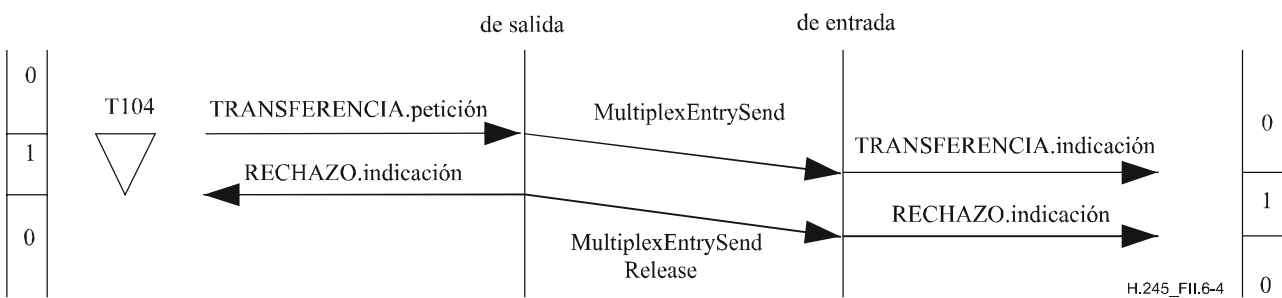


**Figura II.6-2/H.245 – Petición de envío de tabla múltiplex con rechazo por el usuario de la MTSE par**

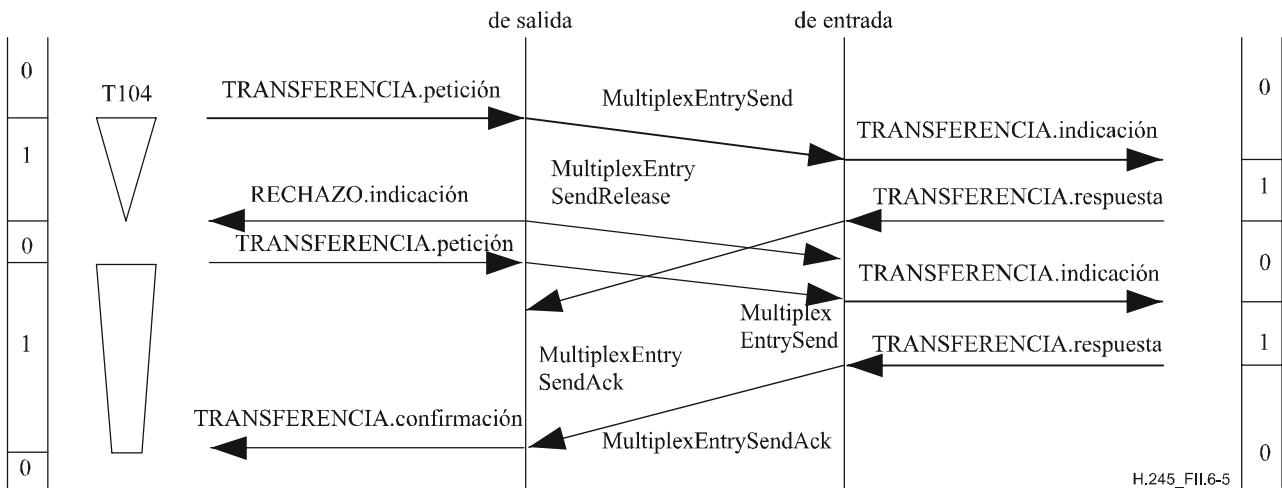


NOTA – La primera petición fracasó.

**Figura II.6-3/H.245 – Petición de envío de tabla múltiplex con una segunda petición de envío de tabla múltiplex antes del acuse de recibo de la primera petición**



**Figura II.6-4/H.245 – Petición de envío de tabla múltiplex con expiración del temporizador T104**

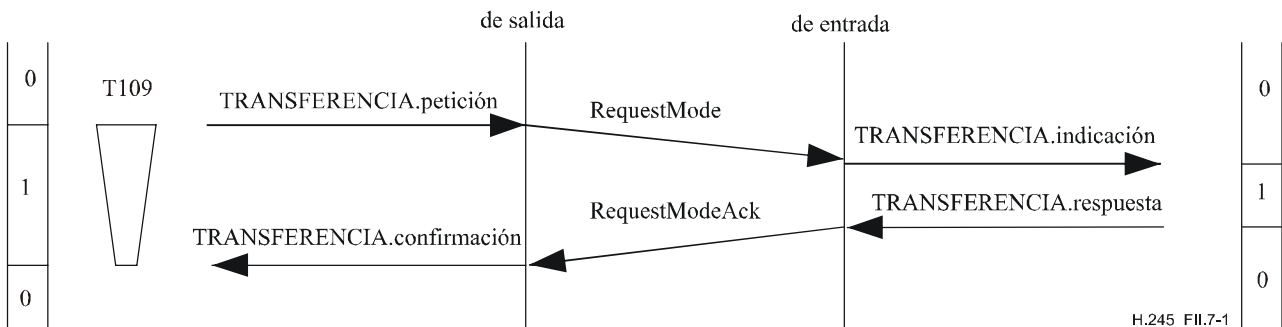


NOTA – El primer mensaje MultiplexEntrySendAck es ignorado en la MTSE de salida. Sólo tuvo éxito la segunda petición.

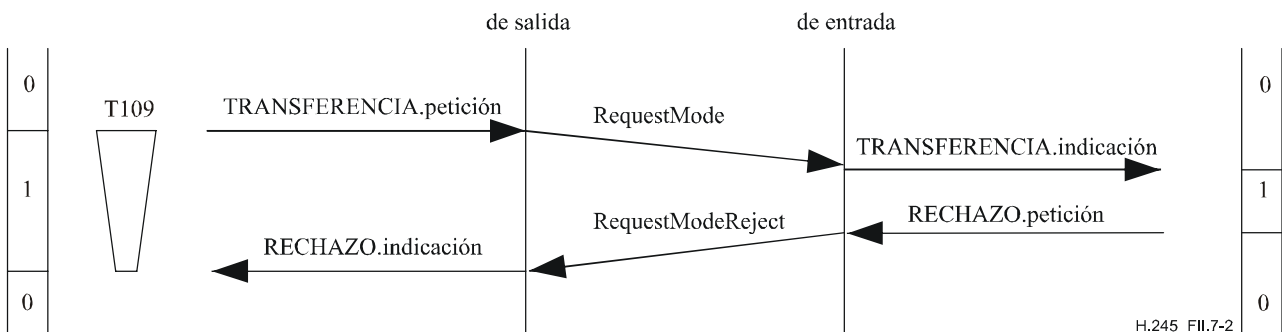
**Figura II.6-5/H.245 – Petición de envío de tabla múltiplex con expiración del temporizador T104 seguida de una segunda petición de envío de tabla múltiplex**

## II.7 Entidad de señalización de petición de modo

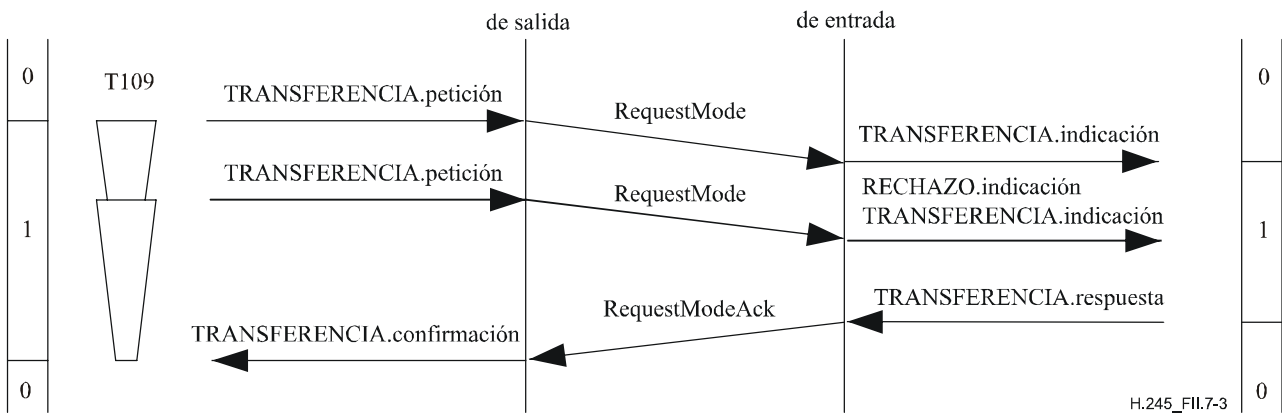
Las figuras II.7-1 a II.7-5 ilustran los intercambios de la MRSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1", respectivamente.



**Figura II.7-1/H.245 – Petición de modo con éxito**

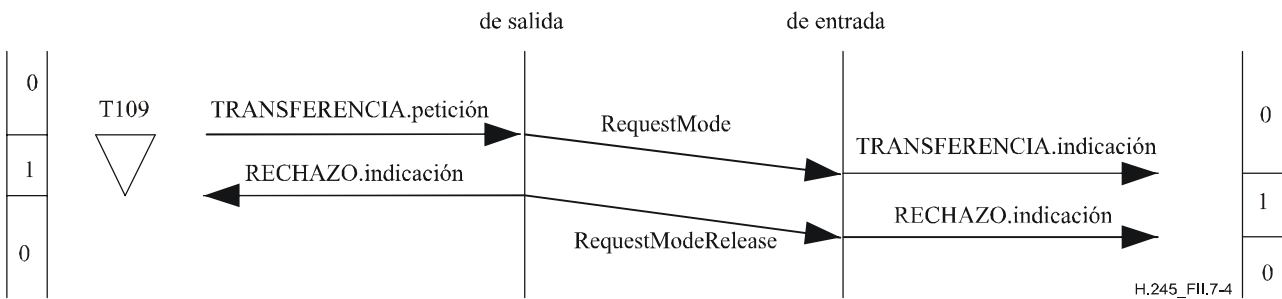


**Figura II.7-2/H.245 – Petición de modo con rechazo por el usuario de la MRSE par**



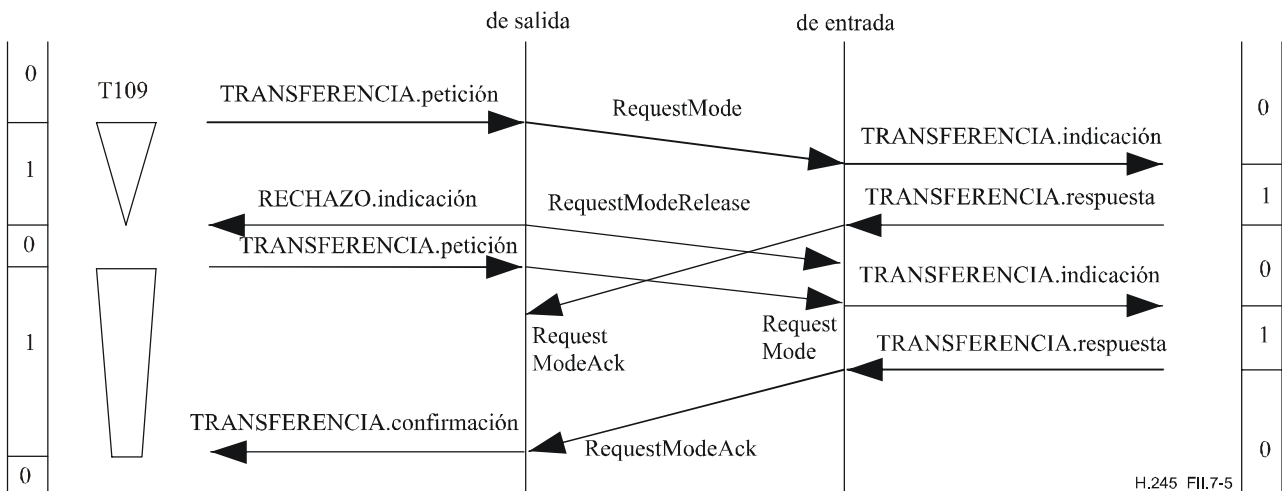
NOTA – La primera petición fracasó.

**Figura II.7-3/H.245 – Petición de modo con una segunda petición de modo antes del acuse de recibo de la primera petición**



NOTA – La petición de modo fracasó.

**Figura II.7-4/H.245 – Petición de modo con expiración del temporizador T109**

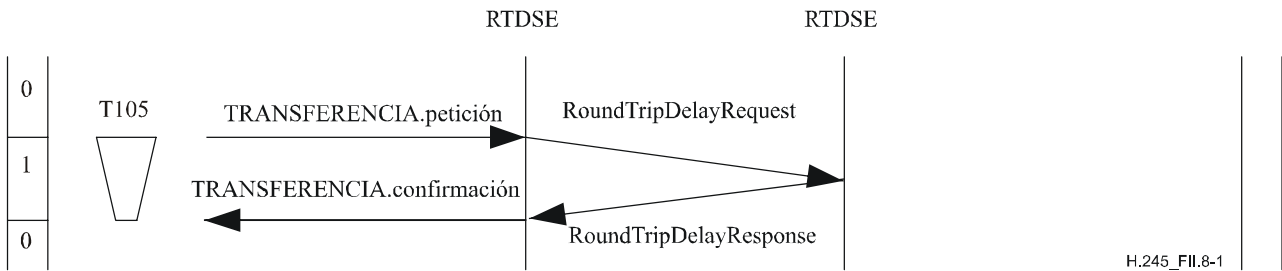


NOTA – El primer mensaje RequestModeAck es ignorado en la MRSE de salida. Sólo tuvo éxito la segunda petición.

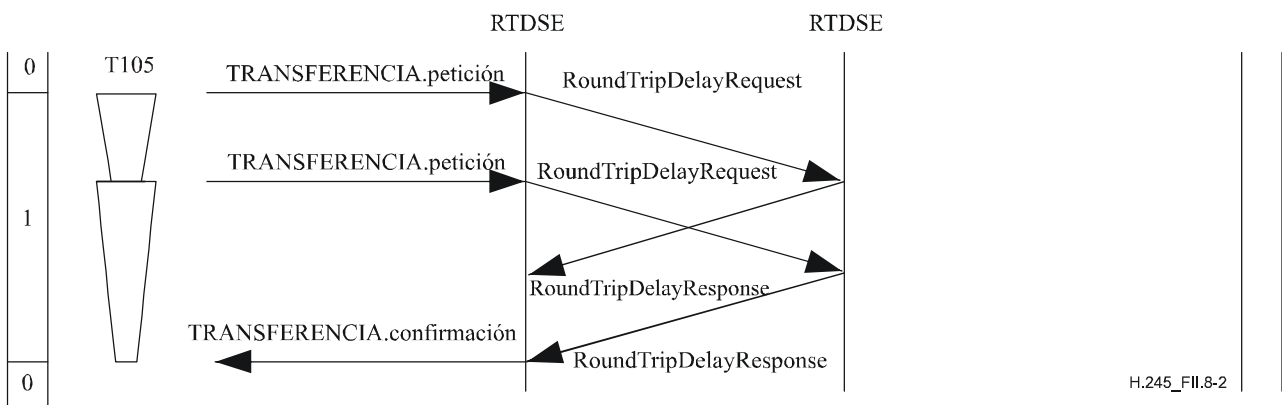
**Figura II.7-5/H.245 – Petición de modo con expiración del temporizador T109 seguida de una segunda petición de modo**

## II.8 Entidad de señalización de retardo de ida y vuelta

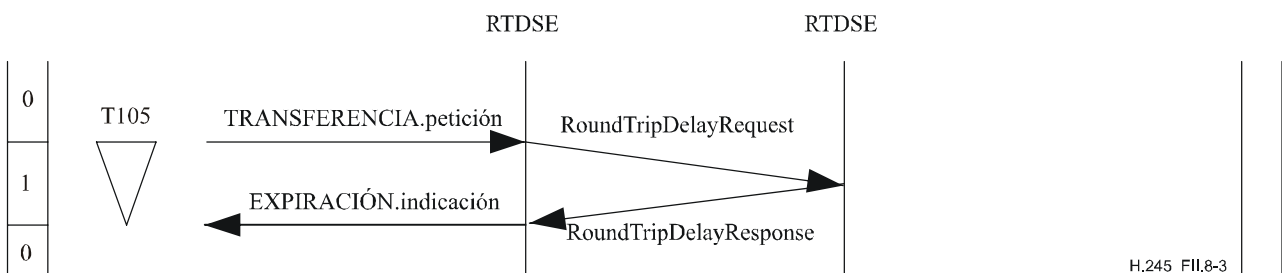
Las figuras II.8-1 a II.8-4 ilustran los procedimientos de la RTDSE. Los estados de la RTDSE REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1", respectivamente.



**Figura II.8-1/H.245 – Procedimiento de determinación de retardo de ida y vuelta**

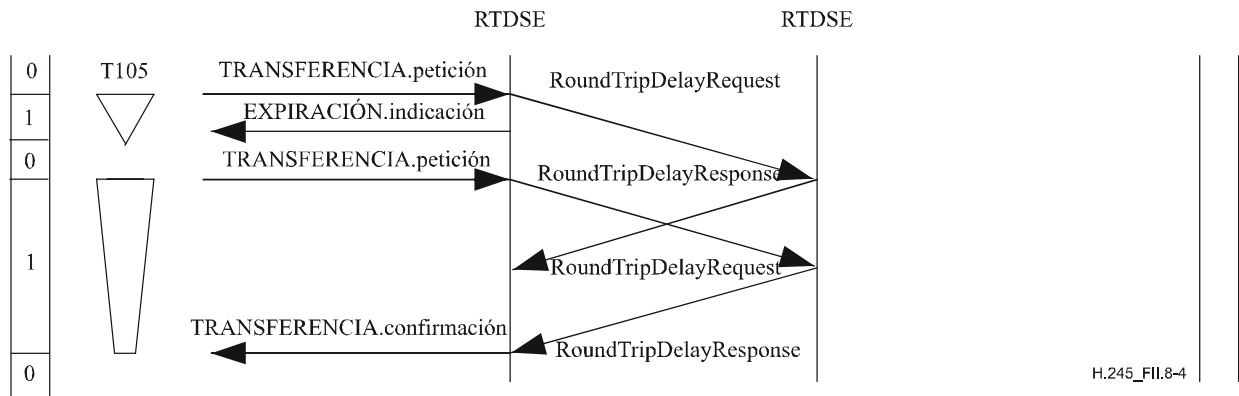


**Figura II.8-2/H.245 – Procedimiento de determinación de retardo de ida y vuelta, estando pendiente un procedimiento anterior del cual no se ha acusado recibo**



**Figura II.8-3/H.245 – Procedimiento de determinación de retardo de ida y vuelta con expiración del temporizador T105**





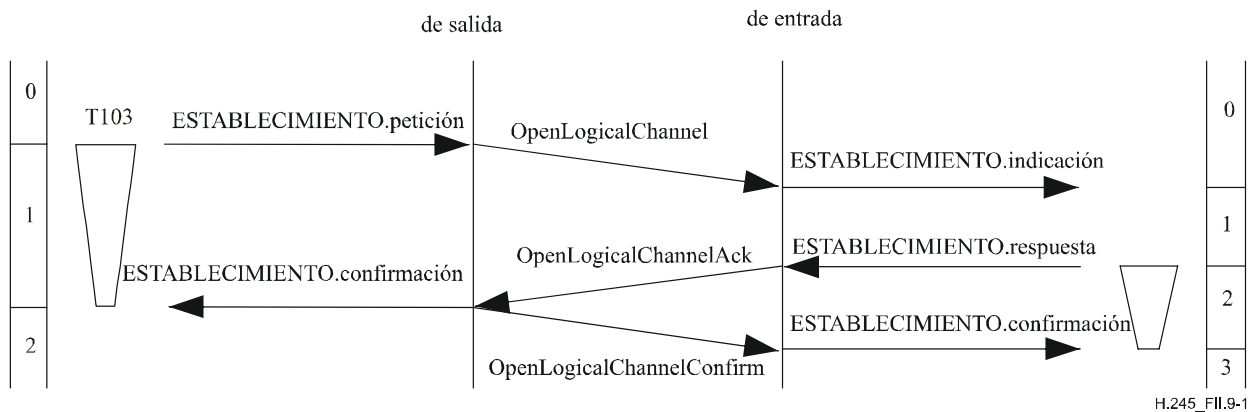
H.245\_FII.8-4

NOTA – El mensaje RoundTripDelayResponse procedente del primer procedimiento llega estando en curso el segundo procedimiento y es ignorado.

**Figura II.8-4/H.245 – Procedimiento de determinación de retardo de ida y vuelta con expiración del temporizador T105, seguido de un segundo procedimiento de determinación de retardo de ida y vuelta**

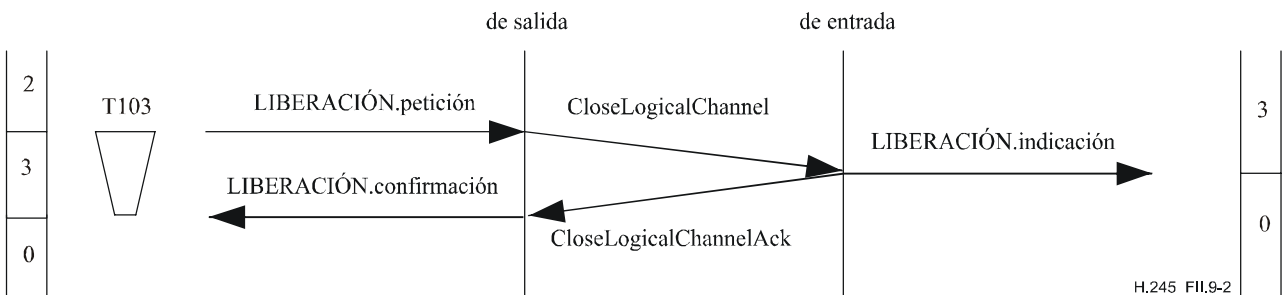
**II.9 Entidad de señalización de canal lógico bidireccional**

Las figuras II.9-1 a II.9-7 ilustran los procedimientos de la B-LCSE. Los estados LIBERADO, ESPERA DE ESTABLECIMIENTO, ESTABLECIDO y ESPERA DE LIBERACIÓN de la B-LCSE de salida se designan por "0", "1", "2" y "3", respectivamente. Los estados LIBERADO, ESPERA DE ESTABLECIMIENTO, ESPERA DE CONFIRMACIÓN y ESTABLECIDO de la B-LCSE de ENTRADA se designan por "0", "1", "2" y "3", respectivamente.



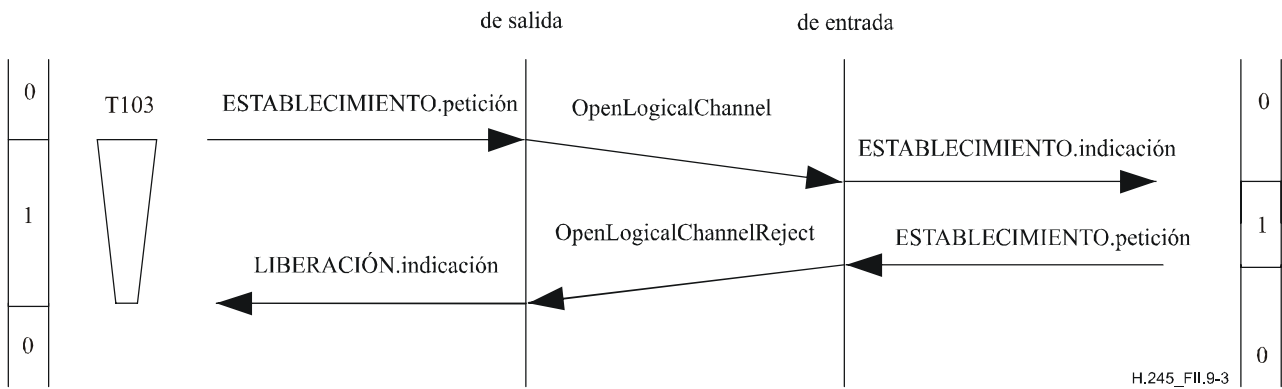
H.245\_FII.9-1

**Figura II.9-1/H.245 – Establecimiento de canal lógico bidireccional**

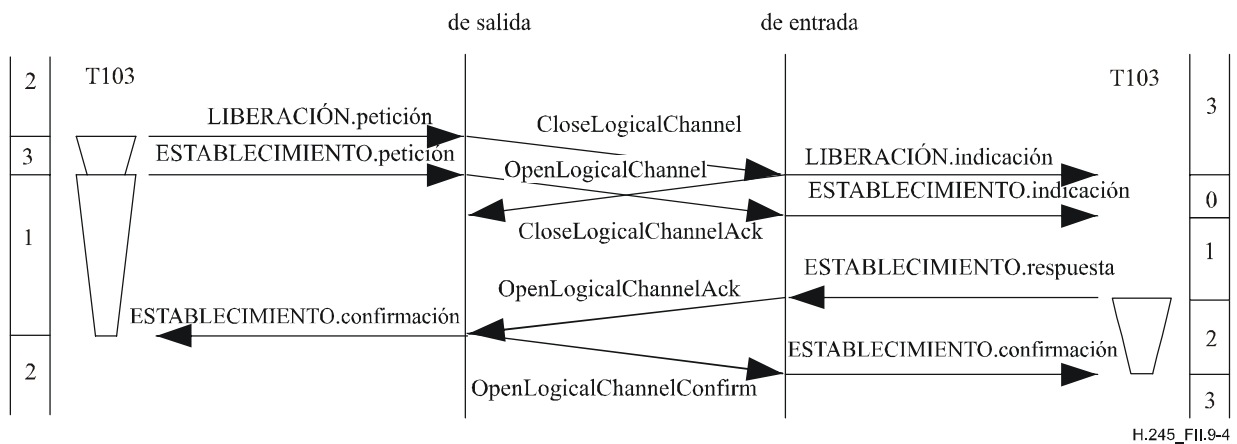


H.245\_FII.9-2

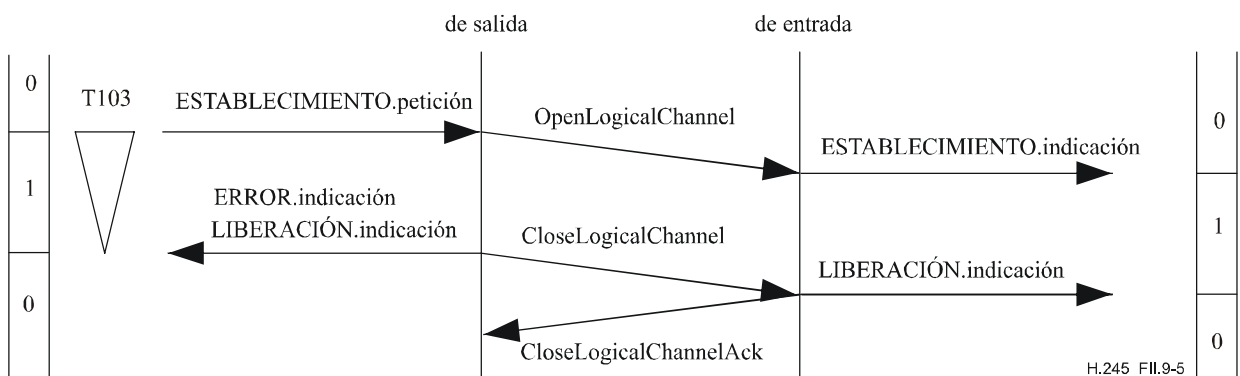
**Figura II.9-2/H.245 – Liberación de canal lógico bidireccional**



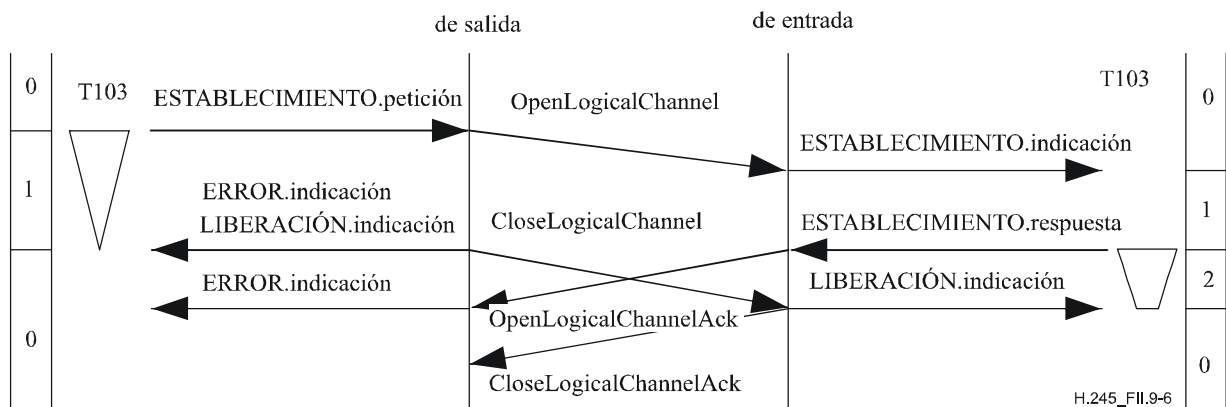
**Figura II.9-3/H.245 – Establecimiento de canal lógico bidireccional con rechazo por el usuario de la B-LCSE par**



**Figura II.9-4/H.245 – Liberación de canal lógico bidireccional seguida por un nuevo establecimiento inmediato**

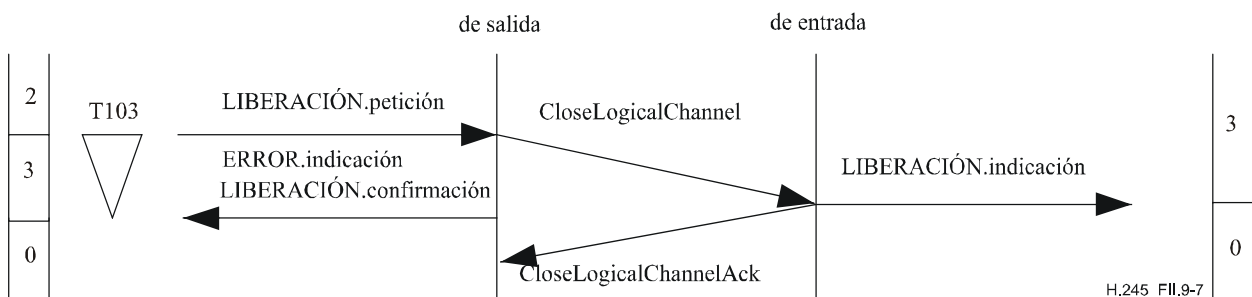


**Figura II.9-5/H.245 – Petición de establecimiento de canal lógico bidireccional con expiración del temporizador T103 en el lado de salida debido a una respuesta lenta del usuario de la B-LCSE de entrada**



NOTA – El temporizador T103 del lado salida ha expirado después de la transmisión del mensaje OpenLogicalChannelAck en la B-LCSE de entrada, pero antes de la recepción del mensaje OpenLogicalChannelAck en la B-LCSE de salida.

**Figura II.9-6/H.245 – Petición de establecimiento de canal lógico bidireccional con expiración del temporizador T103 en el lado de salida**



**Figura II.9-7/H.245 – Petición de liberación de canal lógico bidireccional con expiración del temporizador T103 en el lado de salida**

## Apéndice III

### Sumario de temporizadores y contadores de procedimiento

Este apéndice proporciona una lista de los temporizadores y contadores especificados en el anexo C.

Esta Recomendación no define los valores cargados en los temporizadores. Estos valores pueden definirse en otras Recomendaciones, como las Recs. UIT-T H.310, H.323 y H.324.

#### III.1 Temporizadores

El cuadro III.1 muestra los temporizadores especificados en esta Recomendación.

**Cuadro III.1/H.245 – Temporizadores de procedimiento**

<b>Temporizador</b>	<b>Procedimiento</b>	<b>Definición</b>
T106	Determinación principal-subordinado	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA EN SALIDA y durante el estado ESPERA DE RESPUESTA EN ENTRADA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje de acuse de recibo.
T101	Intercambio de capacidades	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje TerminalCapabilitySetAck o TerminalCapabilitySetReject.
T103	Señalización de canal lógico unidireccional y bidireccional	Este temporizador se utiliza en los estados ESPERA DE ESTABLECIMIENTO y ESPERA DE LIBERACIÓN. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje OpenLogicalChannelAck u OpenLogicalChannelReject o CloseLogicalChannelAck.
T108	Cierre canal lógico	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje RequestChannelCloseAck o RequestChannelCloseReject.
T104	Tabla múltiplex H.223	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje MultiplexEntrySendAck o MultiplexEntrySendReject.
T109	Petición de modo	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje RequestModeAck o RequestModeReject.
T105	Retardo de ida y vuelta	Este temporizador se utiliza en el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje RoundTripDelayResponse.
T107	Petición de entrada múltiplex	Este temporizador se utiliza durante el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje RequestMultiplexEntryAck o RequestMultiplexEntryReject.
T102	Bucle de mantenimiento	Este temporizador se utiliza durante el estado ESPERA DE RESPUESTA. Especifica el máximo tiempo durante el cual no puede recibirse ningún mensaje MaintenanceLoopAck o MaintenanceLoopReject.

### **III.2 Contadores**

El cuadro III.2 muestra los contadores especificados en esta Recomendación.

**Cuadro III.2/H.245 – Contadores de procedimiento**

<b>Temporizador</b>	<b>Procedimiento</b>	<b>Definición</b>
N100	Determinación principal-subordinado	Este contador especifica el máximo número de veces que se enviarán mensajes MasterSlaveDetermination durante el estado ESPERA DE RESPUESTA EN SALIDA.

## Apéndice IV

### Procedimiento de extensión de H.245

Esta Recomendación es un "documento viviente" utilizado por muchas Recomendaciones de sistemas incluidas H.310, H.323, H.324 y V.70. Se espera que será ampliado en forma retrocompatible en cada reunión de la Comisión de Estudio 16 del UIT-T. Este apéndice explica el procedimiento que debe utilizarse para añadir extensiones a esta Recomendación.

En cualquier momento dado sólo hay una sintaxis H.245 en vigor. Ninguna otra Recomendación UIT-T deberá incluir de manera normativa otras variantes de la sintaxis H.245 en sus Recomendaciones.

Las peticiones de extensiones (o ampliaciones) a esta Recomendación deberán presentarse en forma de una contribución blanca, o por un conducto oficial, a la Comisión de Estudio 16 y deberá enviarse lo antes posible una copia al Relator o editor de la H.245. En dichas peticiones se indicará:

- 1) los requisitos formales de la sintaxis que habrá de elaborar el editor o una propuesta de sintaxis basada en la versión aprobada actual de esta Recomendación; y
- 2) la semántica propuesta para el anexo B; y
- 3) los procedimientos propuestos para el anexo C si se solicitan nuevos procedimientos.

Todas las extensiones de esta Recomendación serán retrocompatibles con todas las versiones anteriores de esta Recomendación. La sintaxis, la semántica y los procedimientos preexistentes no pueden cambiarse. De manera específica, cuando se extienda una capacidad H.245, la extensión no deberá cambiar el significado de la capacidad original de manera tal que un terminal que no comprenda la extensión necesite modificar su funcionamiento para utilizar la capacidad sin la extensión. Deberán constreñirse todos los componentes ASN.1 de la extensión.

Las solicitudes deben presentarse lo antes posible para que los expertos de la Comisión de Estudio 16 responsables de la H.245 tengan tiempo para examinar las extensiones. Debe quedar entendido que la sintaxis exacta solicitada puede ser modificada por las razones siguientes:

- 1) verificación de la sintaxis ASN.1 correcta;
- 2) armonización con otras peticiones de extensiones H.245, con las que estuviera en conflicto;
- 3) retrocompatibilidad con anteriores versiones de H.245;
- 4) los expertos revisan el emplazamiento de nuevas funciones con relación a la estructura existente de H.245;
- 5) una denominación no coherente con la sintaxis preexistente;
- 6) unos componentes ASN.1 no constreñidos o ambiguos.

Deberán evitarse las abreviaturas y los acrónimos, sobre todo si una palabra o frase no está abreviada o expresada como un acrónimo en la sintaxis preexistente. Por ejemplo, la palabra "parámetros" no deberá escribirse en forma abreviada como "params". Si una palabra se ha utilizado en una sintaxis preexistente no deberá utilizarse otra palabra con el mismo significado. Por ejemplo, en el caso de componentes de la llamada de tipo agregado, "entrada" en vez de "ítem", porque entrada se ha utilizado consistentemente para describir esto. Hay que ser coherentes.

A continuación se describe cómo deben constreñirse los tipos más comunes de componentes ASN.1, aunque deben constreñirse todos ellos.

La restricción de los componentes ASN.1 SET OF y SEQUENCE OF debe hacerse proporcionando un tamaño mínimo y máximo o bien un tamaño fijo. Si no existe un tamaño máximo inherente basado en la semántica de un componente, se elige un tamaño máximo razonable, aunque sea arbitrario, de por ejemplo 256. Si un componente SET OF o SEQUENCE OF es

OPTIONAL, deberá especificarse un mínimo distinto de cero a menos que haya una diferencia semántica entre los casos, presente pero no vacío y no presente, ya que entonces deberá describirse la diferencia semántica. Si una petición de extensiones contiene el componente SET OF o el SEQUENCE OF que no están constreñidos, el editor puede utilizar SIZE (1..256) como restricción por defecto.

Deben constreñirse los componentes ASN.1 cadena de caracteres indicando un tamaño, ya sea un mínimo y máximo o bien un tamaño fijo. Si una petición de extensiones contiene componentes cadena de caracteres que no están constreñidos, el editor puede utilizar SIZE (0..255) como restricción por defecto.

Deben constreñirse los componentes INTEGER indicando una gama de valores. O si no hay una gama inherente basada en la semántica de un componente, se elige una gama razonable, aunque sea arbitraria, cuyo valor máximo sea uno de los siguientes:

255	$(2^8 - 1)$
65535	$(2^{16} - 1)$
16777215	$(2^{24} - 1)$
4294967295	$(2^{32} - 1)$

Si una petición de extensiones contiene componentes INTEGER que no están constreñidos, el editor puede utilizar INTEGER (0..4294967295) como valor por defecto.

El editor de H.245 examinará todas las peticiones de extensiones y propondrá las versiones finales de las versiones ampliadas de H.245 para su aprobación por la Comisión de Estudio 16. Tras la aprobación de cada nueva versión de esta Recomendación, el número de versión de H.245 en **protocolIdentifier** se incrementará en una unidad para identificar la nueva versión.

Obsérvese que la Comisión de Estudio 16 tiene la intención de aceptar solamente extensiones de H.245 armonizadas, procedentes del editor de H.245.

## Apéndice V

### Procedimiento replacementFor

El procedimiento **replacementFor** H.245 permite cambiar de modo de manera continua de un códec a otro, sin necesidad de dos decodificadores de medios. Este procedimiento sólo se puede utilizar si el terminal receptor ha indicado la capacidad **maxPendingReplacementFor**.

Puesto que la apertura y cierre de canales lógicos H.245 no están sincronizados con el contenido de los medios, se puede producir el abandono de medios entre el momento del cierre de un canal lógico y el momento de la apertura de su reemplazante. El parámetro **replacementFor** permite evitar ese abandono de medios.

#### Ejemplo

Supóngase que se abre el canal lógico 723, que lleva audio G.723.1, y se desea conmutar a G.711 (por el canal lógico 711), pero el receptor tiene capacidad solamente para un canal audio. El procedimiento **replacementFor** puede ser utilizado por el transmisor para efectuar un cambio de modo de manera continua, sin interrupciones, como sigue:

- 1) *Sólo para el caso de sistemas y equipos H.323 que utilizan RSVP*, puesto que el nuevo canal requerirá más anchura de banda (64 kbit/s) que el canal existente (6,4 kbit/s), el transmisor y el receptor establecerán una reserva de anchura de banda RSVP mayor.
- 2) El transmisor envía `OpenLogicalChannel` para el nuevo canal lógico 711, incluido el parámetro `replacementFor`, haciendo referencia al canal lógico existente 723.  
Esto indica al receptor que el canal lógico 711 es un *reemplazante* del canal lógico 723 y que el canal lógico 711 nunca llevará tráfico de manera simultánea con el canal lógico 723.
- 3) Mientras continúa decodificando según G.723.1 desde el canal lógico 723, el receptor se prepara para conmutar sin interrupciones a decodificación G.711.  
Esa preparación podría incluir la carga de soporte lógico de decodificador apropiado.  
Cuando el receptor ha concluido las preparaciones para aceptar el tren de audio G.711, responde con `OpenLogicalChannelAck` para el canal lógico 711. En el caso de H.323, las direcciones de los medios y de transporte del control de los medios devueltas son las mismas que las ya utilizadas para el canal lógico 723.
- 4) El transmisor detiene el envío de audio G.723 por el canal lógico 723 y empieza a enviar, de manera continua, audio G.711 por el canal lógico 711.
- 5) El transmisor envía inmediatamente `CloseLogicalChannel` para el canal lógico 723, porque este canal ya no lleva ningún tráfico y no se le necesita.
- 6) *Sólo para el caso de sistemas y equipos H.323 que utilizan RSVP*, si el nuevo canal requiere menos anchura de banda que el canal original, el transmisor y el receptor establecen una reserva de anchura de banda RSVP menor (no es aplicable en este ejemplo).

En todos los casos, las operaciones de la LCSE y la B-LCSE se atienen a los procedimientos normales. El parámetro `replacementFor` tan sólo informa al receptor del cambio de modo pendiente y de que los dos canales lógicos no se utilizarán simultáneamente y, por tanto, que el segundo canal lógico (en algunas implementaciones) puede ser aceptado en aquellos casos en los que, de otro modo, sería rechazado (por falta de la capacidad para recibir otro canal lógico independiente).

Se señala que en algunos casos el receptor puede rechazar un intento de apertura del canal lógico utilizando el mecanismo `replacementFor` (por ejemplo, si un receptor puede aceptar el mecanismo `replacementFor` para canales audio, pero no para canales vídeo). En tal caso, los transmisores deberán reintentar el cambio de modo sin `replacementFor`, por ejemplo, mediante el cierre del canal y la apertura a continuación de uno nuevo, aceptando cualquier abandono temporal de medios.

Se señala también que en los sistemas H.323, los receptores han de reutilizar las direcciones de medios y de transporte del control de medios existentes. El punto de conmutación al nuevo canal lógico se marca con el encabezamiento RTP.

## Apéndice VI

### Ejemplos de fijaciones de parámetros de la estructura de capacidad H.263

En este apéndice se dan varios ejemplos con los que se trata de aclarar la utilización de la estructura de capacidad H.263.

#### VI.1 Ejemplos de fijación de parámetros H.245 de capa de mejora

El cuadro VI.1 muestra los siguientes ejemplos de fijación de parámetros de los parámetros de la capa de mejora.

Ejemplo # 1: Se refiere a una capacidad de vídeo básica H.263 sencilla de 10 tramas/s y velocidad binaria máxima de 20 kbit/s sin ninguna opción.

Ejemplo # 2: Estas fijaciones de parámetros señalan la capacidad de un tren de canal lógico con una capa de mejora espacial con resolución de un cuarto de CIF, 10 tramas/s, velocidad binaria máxima de 5 kbit/s y ninguna otra opción fijada.

Ejemplo # 3: Estas fijaciones de parámetros señalan la capacidad de un tren de canal lógico con una capa de mejora SNR con resolución de un subcuarto de CIF, 10 tramas/s, velocidad binaria máxima de 5 kbit/s y ninguna otra opción fijada.

Ejemplo # 4: Estas fijaciones de parámetros señalan la capacidad de un tren de canal lógico con tres capas de mejora. Dos capas de mejora SNR, una con un subcuarto de CIF y la otra con un cuarto de CIF, 10 tramas/s y ninguna otra opción fijada, y una capa de mejora espacial con resolución CIF, 10 tramas/s y ninguna otra opción fijada; las tres capas combinadas a una velocidad binaria máxima de 15 kbit/s.

Ejemplo # 5: Estas fijaciones de parámetros señalan la capacidad de un tren de canal lógico con tres capas de mejora y una capa básica a una velocidad binaria máxima de 25 kbit/s. La capa básica con un cuarto de CIF y sin ninguna opción. Además, el terminal admite una capa de mejora SNR con un cuarto de CIF, 10 tramas/s y ninguna opción fijada, una capa de mejora SNR con resolución CIF, 10 tramas/s y ninguna otra opción fijada, y una capa de mejora espacial con resolución CIF, 10 tramas/s y ninguna otra opción fijada.

**Cuadro VI.1/H.245 – Ejemplos de fijación de parámetros H.245 de capa de mejora**

H263Capability parameter	Ejemplos									
	1	2	3	4			5			
sqcifMPI	3	NP	NP	NP	NP			NP		
qcifMPI	NP	NP	NP	NP	NP			3		
cifMPI	NP	NP	NP	NP	NP			NP		
cif4MPI	NP	NP	NP	NP	NP			NP		
cif16MPI	NP	NP	NP	NP	NP			NP		
maxBitRate	200	50	50	150				250		
unrestrictedVector	F	F	F	F				F		
arithmeticCoding	F	F	F	F				F		
advancedPrediction	F	F	F	F				F		
pbFrames	F	F	F	F				F		
temporalSpatialTradeOffCap	F	F	F	F				F		
hrd-B	NP	NP	NP	NP				NP		
bppMaxKb	NP	NP	NP	NP				NP		



**Cuadro VI.1/H.245 – Ejemplos de fijación de parámetros H.245 de capa de mejora**

		Ejemplos								
slowSqcifMPI		NP	NP	NP	NP			NP		
slowQcifMPI		NP	NP	NP	NP			NP		
slowCifMPI		NP	NP	NP	NP			NP		
slowCif4MPI		NP	NP	NP	NP			NP		
slowCif16MPI		NP	NP	NP	NP			NP		
errorCompensation		NP	NP	NP	NP			NP		
SET OF (EnhancementOptions <sup>a)</sup> ) =		NP	NP	1	1	2		1	2	
snrEnhancement	sqcifMPI			3	3	NP		NP	NP	
	qcifMPI			NP	NP	3		3	NP	
	cifMPI			NP	NP	NP		NP	3	
	cif4MPI			NP	NP	NP		NP	NP	
	cif16MPI			NP	NP	NP		NP	NP	
	maxbitrate			50	50	50		50	50	
SET OF (EnhancementOptions <sup>a)</sup> ) =		NP	1	NP	NP	NP	1	NP	NP	1
spatialEnhancement	sqcifMPI		NP				NP			NP
	qcifMPI		3				NP			NP
	cifMPI		NP				3			3
	cif4MPI		NP				NP			NP
	cif16MPI		NP				NP			NP
	maxbitrate		50				50			50
SET OF (EnhancementOptions <sup>a)</sup> ) =		NP	NP	NP	NP	NP	NP			
bframeEnhancement	sqcifMPI									
	qcifMPI									
	cifMPI									
	cif4MPI									
	cif16MPI									
	maxbitrate									
NP	No presente									
T	Verdadero									
F	Falso									
a)	No se muestran otras opciones por debajo de "maxbitrate" en la estructura EnhancementOptions.									

## VI.2 Ejemplos de fijación de parámetros H.245 de canal de retorno de vídeo

Esta cláusula contiene ejemplos de fijaciones de H263Capability y H263Options para el funcionamiento de canal de retorno de vídeo.

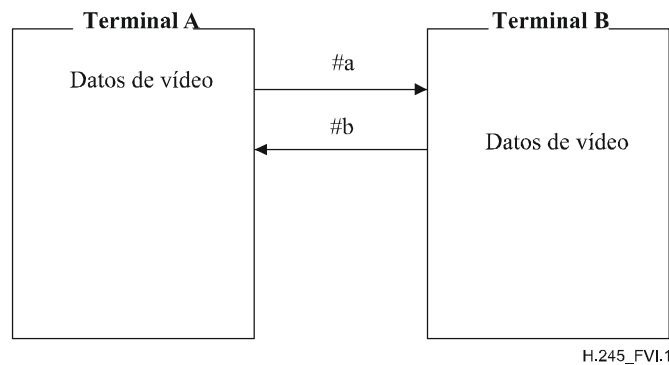
### *Ejemplo 1: Modo canal lógico separado*

En este modo se abre un canal lógico bidireccional adicional para mensajes de canal de retorno de vídeo. La dependencia entre un canal de ida de vídeo y el canal de retorno de vídeo se describe mediante **forwardLogicalChannelDependency** y **reverseLogicalChannelDependency** en el mensaje **OpenLogicalChannel**.

El canal lógico de los mensajes de canal de retorno de vídeo sólo se establecerá después del establecimiento del canal de ida de vídeo. Si se recibe un mensaje **OpenLogicalChannel** con una referencia de dependencia respecto a un canal no existente, el terminal responderá con un

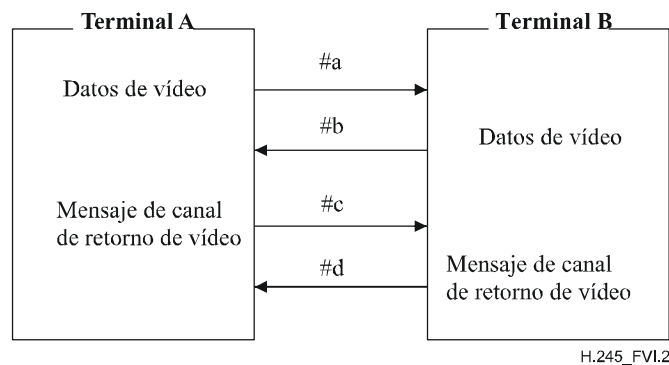
OpenLogicalChannelReject con el código de motivo invalidDependentChannel. A continuación se da un ejemplo.

- 1) Se abre un canal lógico para datos de vídeo entre el terminal A y el terminal B como se muestra en la figura VI.1. El mensaje OpenLogicalChannel para el canal lógico bidireccional incluye RefPictureSelectionCapability en H263VideoCapability.



**Figura VI.1/H.245 – Canales lógicos para datos de vídeo**

- 2) A continuación se abre un canal lógico bidireccional para mensajes de canal de retorno de vídeo, como se muestra en la figura VI.2. En este ejemplo se supone que el terminal A pide la apertura del canal lógico bidireccional. (Si el terminal B pide la apertura del canal, forwardLogicalChannelDependency es sustituido por reverseLogicalChannelDependency y viceversa.) El mensaje OpenLogicalChannel de este canal lógico incluye forwardLogicalChannelDependency en forwardLogicalChannelParameters indicando LCN de LC #a en la figura VI.2 y reverseLogicalChannelDependency en reverseLogicalChannelParameters indicando LCN de LC #b, así como separateVideoBackChannel.



**Figura VI.2/H.245 – Canales lógicos para modo canal lógico separado**

- 3) Después del establecimiento del canal lógico para mensajes de canal de retorno de vídeo, el terminal A envía los datos de vídeo a LC #a y recibe de LC #d los mensajes de canal de retorno de vídeo que corresponden a los datos de vídeo enviados a LC #a. De la misma manera, el terminal A recibe los datos de vídeo de LC #b y envía a LC #c los mensajes de canal de retorno de vídeo que corresponden a los datos de vídeo provenientes del terminal B.

En el cuadro VI.2 se muestra de forma resumida un ejemplo de fijación de parámetros de H263Capability en cada uno de los mensajes OpenLogicalChannel. Para simplificar, sólo se muestra una parte de las capacidades de H263Capability.

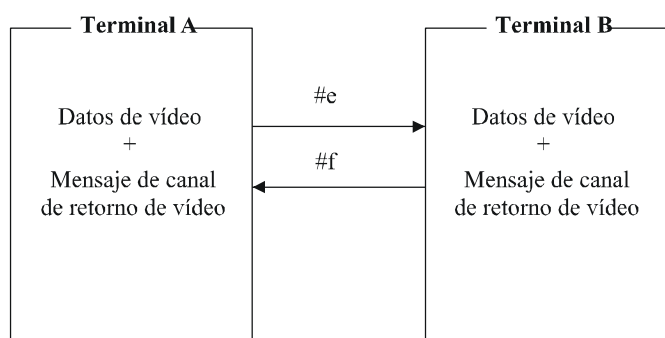
**Cuadro VI.2/H.245 – Ejemplos de fijación de parámetros H263Capability para mensajes OpenLogicalChannel**

Parámetro H263Capability	H263Capability in OpenLogicalChannel messages		
	#a, #b	#c, #d	#e, #f
sqcifMPI	NP	NP	NP
qcifMPI	3	NP	3
cifMPI	NP	NP	NP
cif4MPI	NP	NP	NP
cif16MPI	NP	NP	NP
maxBitRate	240	10	240
refPictureSelection		NP	
additionalPictureMemory	No especificado	–	No especificado
videoMuxCapability	Falso	–	(será) verdadero
videoBackChannelSendCapability	ackAndNackMessage	–	AckAndNackMessage
separateVideoBackChannel	Falso	Verdadero	Falso
NP No presente.			

*Ejemplo 2: Modo múltiplex de vídeo*

Cuando un terminal indica la videoMuxCapability en RefPictureSelectionCapability durante el intercambio de capacidades, otro terminal puede utilizar este modo para enviar mensajes de canal de retorno de vídeo. Puesto que los mensajes de canal de retorno de vídeo son multiplexados en el tren de bits de vídeo codificado, los terminales no necesitan establecer un canal lógico adicional para mensajes de canal de retorno de vídeo. A continuación se da un ejemplo.

- 1) El mensaje OpenLogicalChannel abre un canal lógico bidireccional para vídeo, incluyendo refPictureSelectionCapability con los valores verdaderos modo múltiplex de vídeo en su H263VideoCapability (véase la figura VI.3).
- 2) Después del establecimiento del canal lógico para vídeo, el terminal A envía los datos de vídeo a LC #e y recibe de LC #f los mensajes de canal de retorno de vídeo que corresponden a los datos de vídeo enviados a LC #e multiplexados en los datos de vídeo del terminal B.



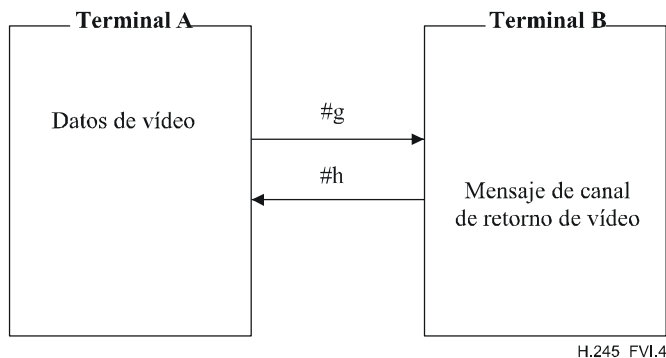
H.245\_FVI.3

**Figura VI.3/H.245 – Canales lógicos para modo múltiplex de vídeo**

En el cuadro VI.2 se muestra en forma resumida un ejemplo de fijación de parámetros de H263Capability en cada uno de los mensajes OpenLogicalChannel.

*Ejemplo 3: Modo canal lógico separado en comunicación vídeo unidireccional*

Este ejemplo muestra el caso en que sólo el terminal A envía datos de vídeo y el terminal B envía solamente mensajes de canal de retorno de vídeo (véase la figura VI.4). En el cuadro VI.3 se muestran las fijaciones de los parámetros de los canales lógicos #g y #h.

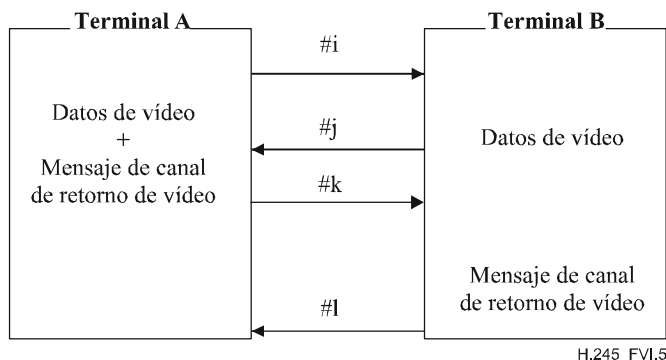


H.245\_FVI.4

**Figura VI.4/H.245 – Modo canal lógico separado en comunicación vídeo unidireccional**

*Ejemplo 4: Coexistencia de modo canal lógico separado y modo múltiplex de vídeo*

El ejemplo ilustrado en la figura VI.5 muestra el caso en que sólo el terminal A utiliza el modo canal lógico separado para recibir mensajes de canal de vídeo procedentes del terminal B vía LC #1 y el terminal B utiliza el modo múltiplex de vídeo para recibir mensajes de canal de retorno de vídeo vía LC #i. Este ejemplo puede no ser realista pero es una configuración posible. En el cuadro VI.3 se muestran las fijaciones de los parámetros de cada uno de los canales lógicos.



H.245\_FVI.5

**Figura VI.5/H.245 – Coexistencia de modo canal lógico separado con modo múltiplex de vídeo**

**Cuadro VI.3/H.245 – Ejemplos de fijaciones de parámetros de H263Capability para mensajes OpenLogicalChannel**

<b>H263Capability in OpenLogicalChannel message</b>						
<b>Parámetro H263Capability</b>	<b>#g</b>	<b>#h</b>	<b>#i</b>	<b>#j</b>	<b>#k</b>	<b>#l</b>
sqcifMPI	NP	NP	NP	NP	NP	NP
qcifMPI	3	NP	3	3	NP	NP
cifMPI	NP	NP	NP	NP	NP	NP
cif4MPI	NP	NP	NP	NP	NP	NP
cif16MPI	NP	NP	NP	NP	NP	NP
maxBitRate	240	10	240	240	10	10
refPictureSelection		NP			NP	NP
additionalPictureMemory	No especificado	–	No especificado	No especificado	–	–
videoMuxCapability	F	–	F	(será) T	–	–
videoBackChannelSendCapability	ackAndNack Message	–	AckAndNack Message	ackAndNack Message	–	–
separateVideoBackChannel	F	T	F	F	F	T
NP No presente T Verdadero F Falso						

## Apéndice VII

### Procedimiento y plantilla para definir nuevas capacidades con las capacidades genéricas H.245

Este apéndice define el procedimiento y una plantilla para definir nuevas capacidades que se expresan en la forma de las capacidades genéricas H.245. Contiene además un ejemplo de cómo podría haber sido utilizada esta plantilla para describir el códec H.261, en vez de utilizar la sintaxis ASN.1 que se ha empleado en la presente Recomendación. La aplicación prevista de este nuevo mecanismo de definición de capacidades en la presente Recomendación son todas las capacidades nuevas que se añaden a esta Recomendación; su objetivo no es redefinir capacidades existentes.

Las descripciones de capacidades relacionadas con las Recomendaciones UIT-T se definirán en los anexos de H.245 o en la Recomendación propiamente dicha (por ejemplo, la Rec. UIT-T H.283).

Otras descripciones de capacidades se pueden definir en los anexos de esta Recomendación o en otras partes.

GenericCapabilities que incluye secuencias collapsing y nonCollapsing no debe incluir estructuras GenericParameter de tipos diferentes (collapsing, nonCollapsing) que utilizan el mismo parameterIdentifier.

NOTA 1 – Esta reutilización del mismo parameterIdentifier podría causar una colisión de valores parameterIdentifier si el parámetro fuera traducido automáticamente a un sistema, por ejemplo un sistema H.320, que no tiene distinción entre parámetros collapsing y nonCollapsing.

El campo parameterIdentifier normalizado de un GenericParameter no debe tener asignado el valor 0.

NOTA 2 – La asignación al valor 0 interferiría la traducción automática a la señalización H.320, como se indica, por ejemplo, en el anexo A/H.239 y en la Rec. UIT-T H.241.

## VII.1 Procedimiento

### VII.1.1 Definición de capacidades genéricas en esta Recomendación

En el caso de que la definición se deba incluir en los anexos de esta Recomendación, se seguirá el siguiente procedimiento:

- 1) Definir un IDENTIFICADOR DE OBJETO para esta capacidad e incluirlo en la lista del anexo D.
- 2) Definir la capacidad con capacidad genérica en un nuevo anexo a la presente Recomendación.

OBJECT IDENTIFIER tiene la forma: {itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) *capability-class capability-name*}.

*capability-class* es video(0), audio(1), data(2), control(3) o multiplex(4). El valor para *capability-name* es asignado en orden numérico para cada *capability-class*.

### VII.1.2 Definición de capacidades genéricas en otras Recomendaciones UIT

En el caso de que la definición se deba incluir en otras Recomendaciones UIT, se seguirá el siguiente procedimiento:

- 1) Definir un IDENTIFICADOR DE OBJETO para esta capacidad en esa Recomendación e incluirlo en la lista del apéndice VIII.
- 2) Definir la capacidad mediante capacidad genérica en un nuevo anexo de la Recomendación apropiada.

### VII.1.3 Definición de capacidades genéricas en normas no UIT

En el caso de que la definición se deba incluir en normas no UIT, se seguirá el siguiente procedimiento:

- 1) Definir un OBJECT IDENTIFIER para esta capacidad en la norma apropiada e incluirlo en la lista del apéndice VIII.
- 2) Definir la capacidad mediante capacidad genérica en la norma apropiada.

## VII.2 Plantilla

### VII.2.1 Identificador de capacidad

Para cada descripción de GenericCapability se definirá un solo ejemplar del cuadro VII.1.

**Cuadro VII.1/H.245 – Plantilla de identificador de capacidad**

Nombre de la capacidad	El nombre del códec, por ejemplo, H.261.
Clase de capacidad	La clase de capacidad, por ejemplo, vídeo, audio, etc.
Tipo de identificador de la capacidad	El tipo del identificador que define el códec: estándar, h221NonStandard o uuid
Valor del identificador de la capacidad	El valor del rótulo del códec, por ejemplo, {itu-t (0) recommendation (0) h (8) 261 generic capabilities (1) 0}. El valor de generic-capabilities identifica un tipo o conjunto de parámetros asociados con la capacidad.  Obsérvese que el formato actual de este identificador de objeto es responsabilidad de quienes definan las capacidades, pero debe ser definido teniendo en cuenta las posibles extensiones.

**Cuadro VII.1/H.245 – Plantilla de identificador de capacidad**

maxBitRate	Si el campo maxBitRate estará incluido, si no estará incluido o si es opcional
nonCollapsingRaw	La especificación del formato de la OCTET STRING y si estará incluida o no
transporte	Si el campo transport (transporte) estará incluido, si no estará incluido o si es opcional

### VII.2.2 Parámetros de capacidad

Esta cláusula es aplicable a GenericParameters (parámetros genéricos) collapsing (colapsables) y nonCollapsing (no colapsables). Para cada GenericParameter se definirá un ejemplar del cuadro VII.2. La plantilla deberá dividirse en secciones para distinguir entre los parámetros utilizados para la negociación de capacidades y los que son específicos de la señalización de canal lógico.

**Cuadro VII.2/H.245 – Plantilla de parámetro de capacidad**

Nombre del parámetro	El nombre del parámetro, por ejemplo, cifMPI.
Descripción del parámetro	Un nombre descriptivo del parámetro, por ejemplo, "especifica el intervalo mínimo de imagen con una resolución CIF".
Valor del identificador del parámetro	Un entero que identifica a este parámetro "standard".
Estado del parámetro	Si el parámetro es obligatorio, obligatorio condicionalmente (por ejemplo dependiente de otro parámetro) u opcional.
Tipo de parámetro	El tipo del parámetro: logical (lógico), booleanArray, unsignedMin, unsignedMax, unsigned32Min, unsigned32Max, octgetString [o genericParameter].
Sustituye a	Los parámetros a los que sustituye este parámetro. Este elemento del cuadro deberá especificar cero, 1 o más parámetros a los que este parámetro sustituye. El formato deberá ser: parameter-name "( parameter-identifier-value )", por ejemplo, qcifMPI (0).
NOTA – Este cuadro no permite especificar el tipo de ParameterTag (standard, h221NonStandard o uuid) ya que sólo se utiliza para descripciones de capacidades normalizadas.	

## VII.3 Ejemplo de plantilla – H.261

### VII.3.1 Identificador de capacidad H.261

**Cuadro VII.3/H.245 – Ejemplo de identificador de capacidad H.261**

Nombre de la capacidad	Rec. UIT-T H.261
Clase de capacidad	Códec de vídeo
Tipo de identificador de la capacidad	Normalizado
Valor del identificador de la capacidad	{itu-t (0) recommendation (0) h (8) generic-capabilities (1) 0}. Éste es el primer (y único) conjunto de parámetros definidos para la Rec. UIT-T H.261.
maxBitRate	El campo maxBitRate deberá estar siempre incluido.
nonCollapsingRaw	Este campo no deberá ser incluido.
transporte	Este campo no deberá ser incluido.

### VII.3.2 Parámetros de capacidad H.261

Se señala que no hay cuadro para el campo de velocidad binaria máxima que figura en la sintaxis ASN.1 de la Rec. UIT-T H.245 para la Rec. UIT-T H.261. Esto se debe a que la velocidad binaria máxima se da al nivel máximo de la estructura GenericCapability. Se señala además que temporalSpatialTradeOffCapability y stillImageTransmission podrían combinarse en un GenericParameter del tipo booleanArray.

**Cuadro VII.4/H.245 – Ejemplo de parámetro de capacidad H.261 – qcifMPI**

Nombre del parámetro	qcifMPI
Descripción del parámetro	Si está presente, indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o la decodificación de imágenes QCIF y, si no está presente, no se indica capacidad para imágenes QCIF.
Valor del identificador del parámetro	0
Estado del parámetro	Opcional
Tipo de parámetro	unsignedMax
Sustituye a	–

**Cuadro VII.5/H.245 – Ejemplo de parámetro de capacidad H.261 – cifMPI**

Nombre del parámetro	cifMPI
Descripción del parámetro	Si está presente, indica el intervalo de imagen mínimo en unidades de 1/29,97 para la codificación y/o la decodificación de imágenes CIF y, si no está presente, no se indica capacidad para imágenes CIF.
Valor del identificador del parámetro	1
Estado del parámetro	Opcional
Tipo de parámetro	unsignedMax
Sustituye a	qcifMPI (0)

**Cuadro VII.6/H.245 – Ejemplo de parámetro de capacidad H.261 – temporalSpatialTradeOffCapability**

Nombre del parámetro	temporalSpatialTradeOffCapability
Descripción del parámetro	La presencia de este parámetro indica que el codificador puede variar su compromiso entre resolución temporal y resolución espacial según instruya el terminal distante. Carece de significado cuando es parte de una capacidad de recepción.
Valor del identificador del parámetro	2
Estado del parámetro	Opcional
Tipo de parámetro	Lógico
Sustituye a	–



**Cuadro VII.7/H.245 – Ejemplo de parámetro de capacidad H.261 – stillImageTransmission**

Nombre del parámetro	stillImageTransmission
Descripción del parámetro	La presencia de este parámetro indica capacidad para imágenes fijas como se especifica en el anexo D/H.261.
Valor del identificador del parámetro	3
Estado del parámetro	Opcional
Tipo de parámetro	Lógico
Sustituye a	–

## Apéndice VIII

### Lista de capacidades genéricas y mensajes genéricos definidos en Recomendaciones/Normas distintas de esta Recomendación

El cuadro VIII.1 presenta una lista de las capacidades genéricas definidas en Recomendaciones o Normas diferentes de esta Recomendación.

**Cuadro VIII.1/H.245 – Lista de capacidades genéricas definidas en Recomendaciones/Normas distintas de esta Recomendación**

Nombre de la capacidad	Clase de capacidad	Identificador de capacidad	Nombre de la Recomendación o Norma que define esta capacidad
H.283	Protocolo de datos	{itu-t (0) recommendation (0) h (8) 283 generic-capabilities (1) 0}	Rec. UIT-T H.283
G.722.1	Protocolo de audio	{itu-t (0) recommendation (0) g (7) 7221 generic-capabilities (1) 0}	Rec. UIT-T G.722.1
H.324	Protocolo de datos	{itu-t (0) recommendation (0) h (8) 324 generic-capabilities (1) http (0)}	Rec. UIT-T H.324
H.324 Session Reset	Protocolo de control	{itu-t (0) recommendation (0) h (8) 324 generic-capabilities (1) SessionResetCapability (1)}	Rec. UIT-T H.324

**Cuadro VIII.1/H.245 – Lista de capacidades genéricas definidas en Recomendaciones/Normas distintas de esta Recomendación**

<b>Nombre de la capacidad</b>	<b>Clase de capacidad</b>	<b>Identificador de capacidad</b>	<b>Nombre de la Recomendación o Norma que define esta capacidad</b>
H.263	Protocolo de vídeo	{itu-t (0) recommendation (0) h (8) 263 generic-capabilities (1) 0}	Rec. UIT-T H.263 NOTA – La utilización de esta capacidad para señalar los "perfiles y niveles" H.263 de conformidad con el anexo X/H.263 debería estar siempre acompañada en paralelo por la señalización de los mismos modos en H263VideoCapability, lo cual es necesario para asegurar que los sistemas que no reconocen las capacidades genéricas H.263 continúan interfaccionando con sistemas más nuevos.
H.224	Protocolo de datos	{itu-t (0) recommendation (0) h (8) 224 generic-capabilities (1) 0}	Rec. UIT-T H.224
G.722.2	Protocolo de audio	{itu-t (0) recommendation (0) g (7) 7222 generic-capabilities (1) 0}	Rec. UIT-T G.722.2
G.726	Protocolo de audio	{itu-t (0) recommendation (0) g (7) 726 generic-capabilities (1) version2003 (0)}	Rec. UIT-T G.726
H.241/H.264	Protocolo de vídeo	{itu-t (0) recommendation (0) h (8) 241 specificVideoCodecCapabilities (0) h264 (0) generic-capabilities (1)}	Rec. UIT-T H.241
h239ControlCapability	Protocolo de control	{itu-t (0) recommendation (0) h (8) 239 generic-capabilities (1) h239ControlCapability (1)}	Rec. UIT-T H.239
h239ExtendedVideoCapability	Protocolo de vídeo	{itu-t (0) recommendation (0) h (8) 239 generic-capabilities (1) h239ExtendedVideoCapability (2)}	Rec. UIT.T H.239
GenericH235securityCapability	Protocolo de seguridad	{itu-t (0) recommendation (0) h (8) 235 version (0) 372}	Rec. UIT-T H.235 anexo G – Para MIKEY-PS – (Notas 1 y 2)
GenericH235securityCapability	Protocolo de seguridad	{itu-t (0) recommendation (0) h (8) 235 version (0) 373}	Rec. UIT-T H.235 anexo G – Para MIKEYDHMAC-PS – (Notas 1 y 2)
GenericH235securityCapability	Protocolo de seguridad	{itu-t (0) recommendation (0) h (8) 235 version (0) 374}	Rec. UIT-T H.235 anexo G – Para MIKEY-PK-SIGN – (Notas 1 y 2)

**Cuadro VIII.1/H.245 – Lista de capacidades genéricas definidas en Recomendaciones/Normas distintas de esta Recomendación**

<b>Nombre de la capacidad</b>	<b>Clase de capacidad</b>	<b>Identificador de capacidad</b>	<b>Nombre de la Recomendación o Norma que define esta capacidad</b>
GenericH235security Capability	Protocolo de seguridad	{itu-t (0) recommendation (0) h (8) 235 version (0) 375}	Rec. UIT-T H.235 anexo G – Para MIKEY-DH-SIGN – (Notas 1 y 2)
NOTA 1 – Utilizado con H235SecurityCapability y en H235Mode.			
NOTA 2 – Utilizado con OLC, OLCack para los protocolos MIKEY.			

El cuadro VIII.2 presenta una lista de los mensajes genéricos definidos en Recomendaciones o Normas diferentes de esta Recomendación.

**Cuadro VIII.2/H.245 Lista de mensajes genéricos definidos en Recomendaciones/Normas distintas de esta Recomendación**

<b>Nombre del mensaje</b>	<b>Clase de mensaje</b>	<b>Identificador de mensaje</b>	<b>Nombre de la Recomendación o Norma que define este mensaje</b>
H.239	Mensaje genérico	{itu-t (0) recommendation (0) h (8) 239 generic-message (2)}	Rec. UIT-T H.239
GenericCommand, GenericIndication	Mensaje genérico	{itu-t (0) recommendation (0) h (8) 235 version (0) 372}	Rec. UIT-T H.235 anexo G (Nota)
GenericCommand, GenericIndication	Mensaje genérico	{itu-t (0) recommendation (0) h (8) 235 version (0) 373}	Rec. UIT-T H.235 anexo G (Nota)
GenericCommand, GenericIndication	Mensaje genérico	{itu-t (0) recommendation (0) h (8) 235 version (0) 374}	Rec. UIT-T H.235 anexo G (Nota)
GenericCommand, GenericIndication	Mensaje genérico	{itu-t (0) recommendation (0) h (8) 235 version (0) 375}	Rec. UIT-T H.235 anexo G (Nota)
NOTA – Utilizada para actualización/renegociación de claves TGK para los protocolos MIKEY.			

## Apéndice IX

### Utilización de la notación ASN.1 en la presente Recomendación

En este apéndice se enumeran los conceptos ASN.1 que han sido utilizados en la presente Recomendación. La Comisión de Estudio 16 tiene la intención de restringir las extensiones de esta Recomendación para utilizar sólo estos conceptos. Los conceptos ASN.1 adicionales sólo serán considerados en circunstancias excepcionales.

## IX.1 Rotulación

Todas las etiquetas de la presente Recomendación son AUTOMATIC TAGS (etiquetas automáticas).

## IX.2 Tipos

En las definiciones ASN.1 de la presente Recomendación tienen lugar los siguientes tipos.

BIT STRING	IA5String	OCTET STRING
BMPString	INTEGER	SEQUENCE
BOOLEAN	NULL	SEQUENCE OF
CHOICE	NumericString	SET
GeneralString	OBJECT IDENTIFIER	SET OF

## IX.3 Limitaciones y gamas

La presente Recomendación utiliza limitaciones de tamaño ("SIZE": strings, set-of y sequence-of), limitaciones de gamas de valores (integers) y alfabetos permitidos ("FROM").

## IX.4 Extensibilidad

La presente Recomendación utiliza el marcador de extensión (ellipsis "...").

# Apéndice X

## Resolución de escenarios de conflictos unidireccionales y bidireccionales

Este apéndice enumera los escenarios de conflictos típicos que se producen debido a la utilización de canales unidireccionales y bidireccionales, y describe el procedimiento de resolución del conflicto para cada uno de ellos.

### X.1 Ambos terminales utilizan OLC bidireccional

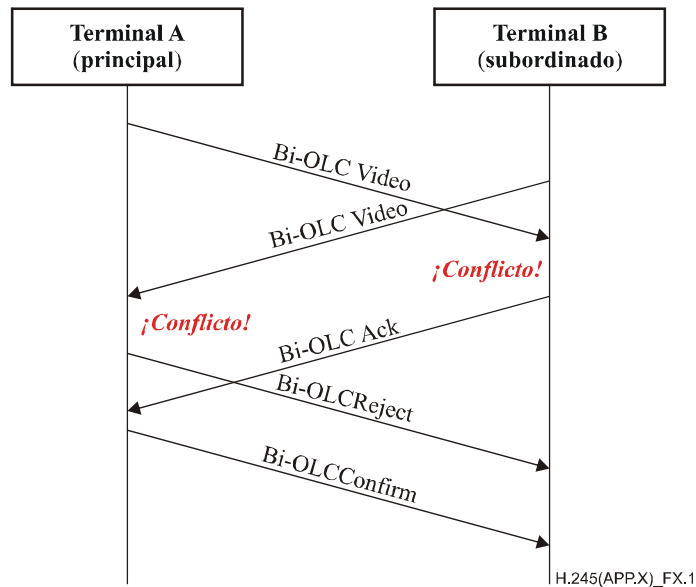
En este escenario ambos terminales proponen un canal vídeo bidireccional, como se ilustra en la figura X.1.

En razón de que ambos terminales sólo pueden procesar un tren de vídeo simple en cada sentido, esto provoca un conflicto.

En este caso, el terminal principal emitirá un mensaje de rechazo con causa igual a masterSlaveConflict en la propuesta del canal del terminal subordinado.

El terminal subordinado aceptará la propuesta del terminal principal y no tratará de abrir un canal sin conflicto.

El terminal subordinado podría también detectar parámetros inversos no apropiados en la propuesta del terminal principal, en cuyo caso rechazará la propuesta del canal principal con causa igual a unsuitableReverseParameters y volverá a emitir una propuesta con parámetros de ida y de retorno apropiados, como se describe en C.5.1.



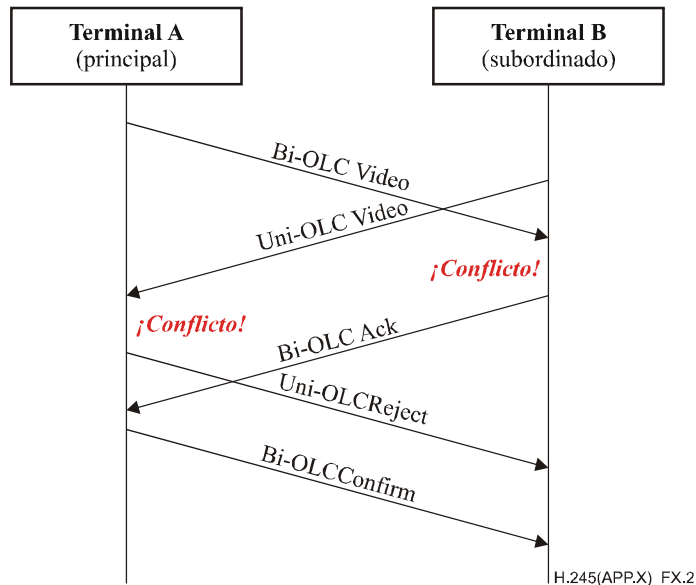
**Figura X.1/H.245 – Ambos terminales utilizan OLC bidireccional**

**X.2 El terminal principal propone OLC bidireccional y el terminal subordinado propone OLC unidireccional**

En este escenario, el terminal principal propone un canal vídeo bidireccional y el terminal subordinado propone un canal vídeo unidireccional, como se ilustra en la figura X.2.

En este caso, el terminal principal emitirá un mensaje de rechazo con causa igual a masterSlaveConflict en la propuesta de canal del terminal subordinado.

El terminal subordinado aceptará la propuesta del terminal principal y no tratará de abrir un canal sin conflicto.



**Figura X.2/H.245 – El terminal principal propone OLC bidireccional y el terminal subordinado OLC unidireccional**

### X.3 El terminal principal propone OLC unidireccional y el terminal subordinado OLC bidireccional

En este escenario, el terminal principal propone un canal vídeo unidireccional y el terminal subordinado propone un canal vídeo bidireccional, como se ilustra en la figura X.3.

El terminal principal emite un mensaje de rechazo con causa igual a masterSlaveConflict en la propuesta de canal del terminal subordinado.

El terminal subordinado aceptará la propuesta del terminal principal y abrirá entonces un canal vídeo unidireccional sin conflicto.

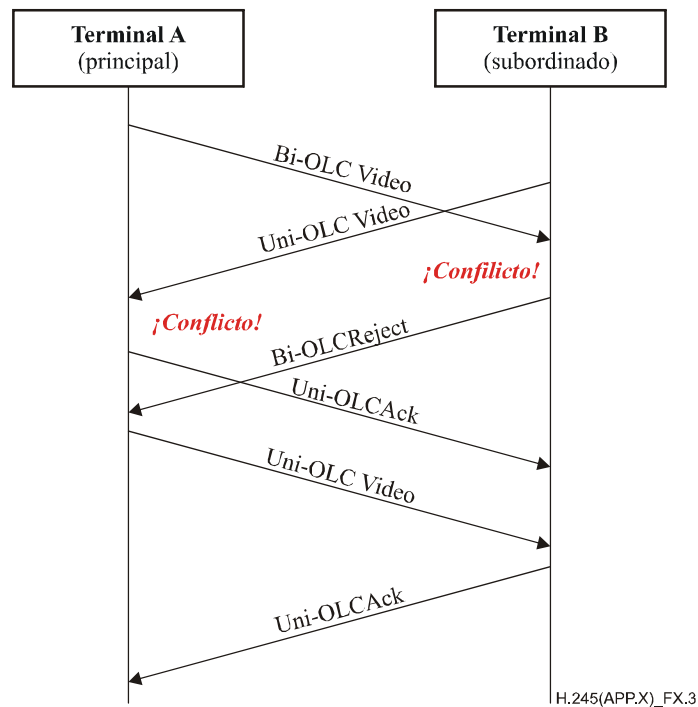
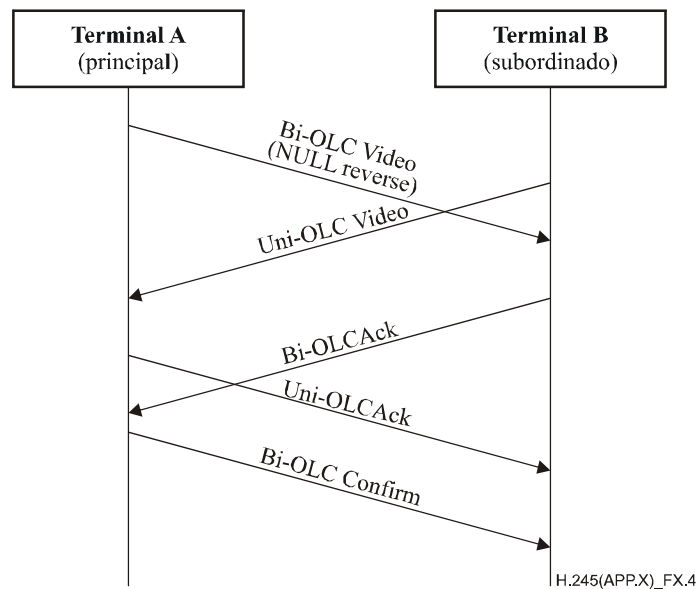


Figura X.3/H.245 – El terminal principal propone OLC unidireccional y el terminal subordinado OLC bidireccional

### X.4 El terminal principal propone OLC bidireccional con nullData y el terminal subordinado propone OLC unidireccional

En este escenario, el terminal principal propone un canal vídeo bidireccional con tipo nullData en los parámetros de canal inverso y el terminal subordinado propone un canal vídeo unidireccional, como se ilustra en la figura X.4.

En razón de que el resultado de estas propuestas sería un canal vídeo simple en cada sentido, no se detectarían conflictos y cada terminal aceptará la propuesta iniciada por el otro terminal.

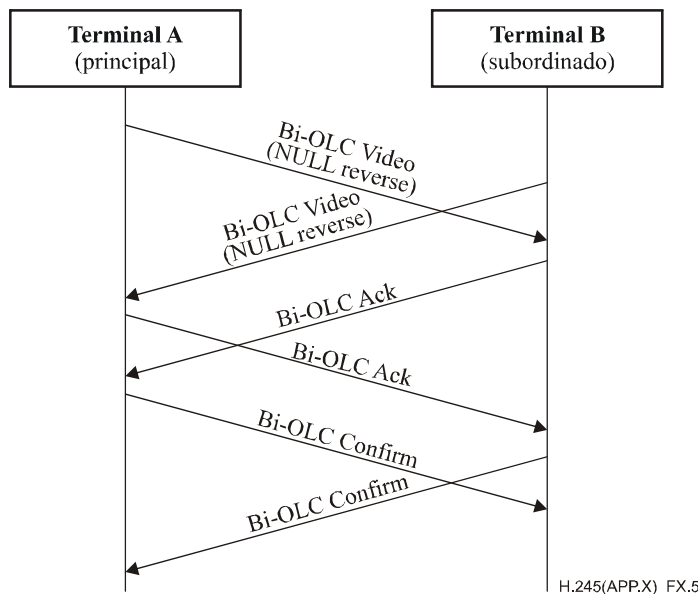


**Figura X.4/H.245 – El terminal principal propone OLC bidireccional con nullData y el terminal subordinado propone OLC unidireccional**

**X.5 Ambos terminales proponen OLC bidireccional con nullData**

En este escenario, ambos terminales proponen un canal vídeo bidireccional con el tipo nullData en los parámetros del canal inverso, como se ilustra en la figura X.5.

En razón de que el resultado de estas propuestas sería un canal vídeo simple en cada sentido, no se detectará conflicto y cada terminal aceptará la propuesta iniciada por el otro terminal.



**Figura X.5/H.245 – Ambos terminales proponen OLC bidireccional con nullData**







## SERIES DE RECOMENDACIONES DEL UIT-T

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