

Reemplazada por una versión más reciente



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

H.245

(02/98)

SERIE H: SISTEMAS AUDIOVISUALES Y
MULTIMEDIOS

Infraestructura de los servicios audiovisuales –
Procedimientos de comunicación

**Protocolo de control para comunicación
multimedios**

Recomendación UIT-T H.245

Reemplazada por una versión más reciente

(Anteriormente Recomendación del CCITT)

Reemplazada por una versión más reciente

RECOMENDACIONES DE LA SERIE H DEL UIT-T

SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

Características de los canales de transmisión para usos distintos de los telefónicos	H.10–H.19
Utilización de circuitos de tipo telefónico para telegrafía armónica	H.20–H.29
Utilización de circuitos o cables telefónicos para transmisiones telegráficas de diversos tipos o transmisiones simultáneas	H.30–H.39
Utilización de circuitos de tipo telefónico para telegrafía facsímil	H.40–H.49
Características de las señales de datos	H.50–H.99
CARACTERÍSTICAS DE LOS SISTEMAS VIDEOTELEFÓNICOS	H.100–H.199
INFRAESTRUCTURA DE LOS SERVICIOS AUDIOVISUALES	
Generalidades	H.200–H.219
Multiplexación y sincronización en transmisión	H.220–H.229
Aspectos de los sistemas	H.230–H.239
Procedimientos de comunicación	H.240–H.259
Codificación de imágenes vídeo en movimiento	H.260–H.279
Aspectos relacionados con los sistemas	H.280–H.299
Sistemas y equipos terminales para los servicios audiovisuales	H.300–H.399

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

Reemplazada por una versión más reciente

RECOMENDACIÓN UIT-T H.245

PROTOCOLO DE CONTROL PARA COMUNICACIÓN MULTIMEDIOS

Resumen

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 transmisión y recepción así como preferencia de modos desde el extremo de recepción, señalización de canal lógico, y Control y Indicación. Se especifican procedimientos de señalización con acuse de recibo para garantizar comunicaciones fiables audiovisuales y de datos.

Orígenes

La Recomendación UIT-T H.245, ha sido revisada por la Comisión de Estudio 16 (1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 6 de febrero 1998.

Reemplazada por una versión más reciente

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 Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), 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 N.º 1 de la CMNT.

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

NOTA

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

PROPIEDAD INTELECTUAL

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

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

© UIT 1998

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

Reemplazada por una versión más reciente

ÍNDICE

	Página
1 Alcance	1
2 Referencias.....	1
3 Definiciones	5
4 Abreviaturas.....	6
5 Generalidades.....	8
5.1 Determinación director-subordinado	8
5.2 Intercambio de capacidad	9
5.3 Procedimientos de señalización de canal lógico.....	9
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.....	10
5.7 Determinación del retardo de ida y vuelta	10
5.8 Bucles de mantenimiento.....	10
5.9 Instrucciones e indicaciones	10
6 Mensajes: sintaxis	11
7 Mensajes: definiciones semánticas	63
7.1 Mensajes de determinación director-subordinado	64
7.1.1 Determinación director-subordinado.....	64
7.1.2 Acuse de recibo de determinación director-subordinado	64
7.1.3 Rechazo de determinación director-subordinado	64
7.1.4 Liberación de determinación de director-subordinado.....	64
7.2 Mensajes de capacidad de terminal	64
7.2.1 Visión general.....	64
7.2.2 Conjunto de capacidades de terminal	65
7.2.3 Acuse de recibo del conjunto de capacidades de terminal	89
7.2.4 Rechazo del conjunto de capacidades de terminal	89
7.2.5 Liberación del conjunto de capacidades de terminal.....	90
7.3 Mensajes de señalización de canal lógico	90
7.3.1 Apertura de canal lógico.....	90
7.3.2 Acuse de recibo de apertura de canal lógico	95
7.3.3 Rechazo de apertura de canal lógico	97
7.3.4 Confirmación de apertura de canal lógico.....	98
7.3.5 Cierre de canal lógico.....	98

Reemplazada por una versión más reciente

Página

7.3.6	Acuse de recibo de cierre de canal lógico	99
7.3.7	Petición de cierre de canal lógico	99
7.3.8	Acuse de recibo de petición de cierre de canal lógico.....	99
7.3.9	Rechazo de petición de cierre de canal.....	99
7.3.10	Liberación de petición de cierre de canal	99
7.4	Mensajes de señalización de tabla múltiplex	99
7.4.1	Envío de entradas múltiplex	100
7.4.2	Acuse de recibo de envío de entradas múltiplex	100
7.4.3	Rechazo de envío de entradas múltiplex	100
7.4.4	Liberación de envío de entradas múltiplex.....	101
7.5	Mensajes de señalización de petición de tabla múltiplex	101
7.5.1	Petición de entrada múltiplex	101
7.5.2	Acuse de recibo de petición de entrada múltiplex.....	101
7.5.3	Rechazo de petición de entrada múltiplex.....	101
7.5.4	Liberación de petición de entrada múltiplex	101
7.6	Mensajes de petición de modo.....	102
7.6.1	Petición de modo	102
7.6.2	Acuse de recibo de petición de modo.....	105
7.6.3	Rechazo de petición de modo.....	105
7.6.4	Liberación de petición de modo	106
7.7	Mensajes de retardo de ida y vuelta.....	106
7.7.1	Petición de retardo de ida y vuelta	106
7.7.2	Respuesta de retardo de ida y vuelta	106
7.8	Mensajes de bucle de mantenimiento	106
7.8.1	Petición de bucle de mantenimiento.....	106
7.8.2	Acuse de recibo de bucle de mantenimiento	106
7.8.3	Rechazo de bucle de mantenimiento	106
7.8.4	Instrucción de desconexión de bucle de mantenimiento	106
7.9	Mensajes de modo de comunicación	106
7.9.1	Instrucción de modo de comunicación	107
7.9.2	Petición de modo de comunicación	108
7.9.3	Respuesta a modo de comunicación.....	108
7.10	Mensajes de petición y respuesta de conferencia	108
7.10.1	Petición de lista de terminales	108
7.10.2	Respuesta a lista de terminales	108
7.10.3	Concédame la presidencia	108
7.10.4	Anulación de concédame la presidencia.....	108
7.10.5	Respuesta a concédame la presidencia	108

Reemplazada por una versión más reciente

Página

7.10.6	Separación de terminal	108
7.10.7	Rechazo de separación de terminal	108
7.10.8	Petición de identificación de terminal	108
7.10.9	Respuesta a identificación de terminal MC.....	108
7.10.10	Petición de introducción de contraseña H.243	108
7.10.11	Respuesta a contraseña	108
7.10.12	Petición de introducción de identificación de terminal H.243	109
7.10.13	Respuesta a identificación de terminal	109
7.10.14	Petición de introducción de identificación de conferencia H.243.....	109
7.10.15	Respuesta a identificación de conferencia.....	109
7.10.16	Rechazo de instrucción de vídeo	109
7.10.17	Petición de introducción de dirección de extensión	109
7.10.18	Respuesta a dirección de extensión	109
7.10.19	Petición al propietario del testigo del control por la presidencia	109
7.10.20	Respuesta del propietario del testigo de la presidencia.....	109
7.10.21	Petición de certificado de terminal	109
7.10.22	Respuesta de certificado de terminal	110
7.10.23	Difunde mi canal lógico	110
7.10.24	Respuesta a difunde mi canal lógico	110
7.10.25	Haz al terminal difundir.....	110
7.10.26	Respuesta a haz al terminal difundir	110
7.10.27	Envía esta fuente.....	110
7.10.28	Respuesta a envía esta fuente	110
7.10.29	Petición de las identificaciones de todos los terminales.....	110
7.10.30	Respuesta a petición de identificaciones de todos los terminales	110
7.10.31	Petición a MC distante.....	110
7.10.32	Respuesta de MC distante	111
7.11	Instrucciones	111
7.11.1	Envío del conjunto de capacidades de terminal	111
7.11.2	Encriptación.....	112
7.11.3	Control de flujo.....	112
7.11.4	Finalización de sesión.....	112
7.11.5	Instrucción diversos.....	113
7.11.6	Instrucción de conferencia.....	116
7.11.7	Reconfiguración de múltiplex H.223.....	116
7.12	Indicaciones	116
7.12.1	Función no comprendida	116
7.12.2	Indicación diversos	116

Reemplazada por una versión más reciente

Página

7.12.3	Indicación de fluctuación	117
7.12.4	Indicación de asimetría H.223	118
7.12.5	Indicación de nuevo canal virtual ATM	119
7.12.6	Entrada de usuario	119
7.12.7	Indicaciones de conferencia	122
7.12.8	Máxima asimetría de canal lógico H2250	122
7.12.9	Indicación de la ubicación del MC	122
7.12.10	Indicación identificación del vendedor	122
7.12.11	Función no soportada.....	122
8	Procedimientos.....	123
8.1	Introducción	123
8.1.1	Método de especificación	123
8.1.2	Comunicación entre una entidad de protocolo y un usuario de protocolo ...	123
8.1.3	Comunicación entre entidades pares	124
8.1.4	Diagramas SDL	124
8.1.5	Símbolos gráficos utilizados en los diagramas SDL	124
8.2	Procedimientos de determinación director-subordinado	125
8.2.1	Introducción.....	125
8.2.2	Comunicación entre la MSDSE y el usuario de MSDSE.....	127
8.2.3	Comunicación entre MSDSE pares	129
8.2.4	Procedimientos de la MSDSE	130
8.3	Procedimientos de intercambio de capacidad.....	136
8.3.1	Introducción.....	136
8.3.2	Comunicación entre la CESE y el usuario de CESE.....	137
8.3.3	Comunicación entre CESE pares.....	139
8.3.4	Procedimientos de CESE.....	140
8.4	Procedimientos de señalización de canal lógico unidireccional.....	144
8.4.1	Introducción.....	144
8.4.2	Comunicación entre la LCSE y el usuario de LCSE.....	146
8.4.3	Comunicación entre LCSE pares.....	148
8.4.4	Procedimientos de LCSE.....	149
8.5	Procedimientos de señalización de canal lógico bidireccional.....	156
8.5.1	Introducción.....	156
8.5.2	Comunicación entre la B-LCSE y el usuario de B-LCSE.....	158
8.5.3	Comunicaciones entre B-LCSE pares	161
8.5.4	Procedimientos de B-LCSE.....	163

Reemplazada por una versión más reciente

Página

8.6	Procedimientos de cierre de canal lógico	171
8.6.1	Introducción.....	171
8.6.2	Comunicación entre la CLCSE y el usuario de CLCSE.....	171
8.6.3	Comunicación entre CLCSE pares	173
8.6.4	Procedimientos de la CLCSE	174
8.7	Procedimientos de la tabla múltiplex H.223.....	179
8.7.1	Introducción.....	179
8.7.2	Comunicación entre la MTSE y el usuario de MTSE	181
8.7.3	Comunicación entre MTSE pares.....	183
8.7.4	Procedimientos de MTSE.....	184
8.8	Procedimientos de petición de entrada múltiplex	191
8.8.1	Introducción.....	191
8.8.2	Comunicación entre la RMESE y el usuario de RMESE.....	192
8.8.3	Comunicación entre RMESE pares	194
8.8.4	Procedimientos de RMESE	195
8.9	Procedimientos de petición de modo	199
8.9.1	Introducción.....	199
8.9.2	Comunicación entre la MRSE y el usuario de MRSE.....	201
8.9.3	Comunicación entre MRSE pares.....	203
8.9.4	Procedimientos de MRSE.....	203
8.10	Procedimientos de retardo de ida y vuelta	209
8.10.1	Introducción.....	209
8.10.2	Comunicación entre la RTDSE y el usuario de RTDSE	210
8.10.3	Comunicación entre RTDSE pares.....	211
8.10.4	Procedimientos de RTDSE.....	212
8.11	Procedimientos de bucle de mantenimiento	214
8.11.1	Introducción.....	214
8.11.2	Comunicación entre la MLSE y el usuario de MLSE	215
8.11.3	Comunicación entre MLSE pares.....	218
8.11.4	Procedimientos de la MLSE	219
	Anexo A – Asignaciones de identificadores de objetos.....	224
	Apéndice I – Visión general de la sintaxis en ASN.1	225
I.1	Introducción a la ASN.1	225
I.2	Tipos de datos ASN.1 básicos	225
I.3	Tipos de datos agregados.....	227
I.4	Tipo identificador de objeto.....	228

Reemplazada por una versión más reciente

Página

Apéndice II – Ejemplos de procedimientos H.245	229
II.1 Introducción	229
II.2 Entidad de señalización de determinación director-subordinado	229
II.3 Entidad de señalización de intercambio de capacidad.....	233
II.4 Entidad de señalización de canal lógico	235
II.5 Entidad de señalización de cierre de canal lógico	238
II.6 Entidad de señalización de tabla múltiplex	239
II.7 Entidad de señalización de petición de modo.....	241
II.8 Entidad de señalización de retardo de ida y vuelta.....	243
II.9 Entidad de señalización de canal lógico bidireccional	245
Apéndice III – Sumario de temporizadores y contadores de procedimiento	247
III.1 Temporizadores	248
III.2 Contadores	249
Apéndice IV – Procedimiento de extensión de H.245	249
Apéndice V – Procedimiento replacementFor	250
Apéndice VI – Ejemplos de fijaciones de parámetros de la estructura de capacidad H.263 ...	252
VI.1 Ejemplos de fijación de parámetros H.245 de capa de mejora.....	252
VI.2 Ejemplos de fijación de parámetros H.245 de canal de retorno de vídeo.....	254

Reemplazada por una versión más reciente

Recomendación H.245

PROTOCOLO DE CONTROL PARA COMUNICACIÓN MULTIMEDIOS

(revisada en 1998)

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 Recomendaciones 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

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.

- [1] Recomendación UIT-T E.164 (1997), *Plan internacional de numeración de telecomunicaciones públicas*.
- [2] Recomendación G.711 del CCITT (1988), *Modulación por impulsos codificados (MIC) de frecuencias vocales*.
- [3] Recomendación G.722 del CCITT (1988), *Codificación de audio de 7 kHz dentro de 64 kbit/s*.
- [4] Recomendación UIT-T G.723.1 (1996), *Códec de voz de doble velocidad para transmisión en comunicaciones multimedia a 5,3 y 6,3 kbit/s*.

Reemplazada por una versión más reciente

- [5] Recomendación G.728 del CCITT (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 (1997), *Estructura de trama para un canal de 64 a 1920 kbit/s en teleservicios audiovisuales.*
- [8] Recomendación UIT-T H.222.0 (1996) | ISO/CEI 13818-1:1996, *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 comunicaciones visuales en entornos del modo de transferencia asíncrono.*
- [10] Recomendación UIT-T H.223 (1996), *Protocolo de multiplexación para comunicación multimedia a baja velocidad binaria.*
- [11] Recomendación UIT-T H.224 (1994), *Protocolo de control en tiempo real para aplicaciones símplex 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 (1998), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicaciones multimedios basados por paquetes.*
- [13] Recomendación UIT-T H.230 (1997), *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 sistemas audiovisuales.*
- [16] Recomendación UIT-T H.235 (1998), *Seguridad y encriptación de los terminales multimedios de la serie H (H.323 y otros basados en la H.245).*
- [17] Recomendación UIT-T H.243 (1997), *Procedimiento 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 (1995) | ISO/CEI 13818-2:1996, *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 (1996), *Sistemas y terminales para comunicaciones audiovisuales de banda ancha.*
- [22] Recomendación UIT-T H.320 (1997), *Sistemas y equipos terminales videotelefónicos de banda estrecha.*
- [23] Recomendación UIT-T H.323 (1998), *Sistemas de comunicaciones multimedios basados por paquetes.*

Reemplazada por una versión más reciente

- [24] Recomendación UIT-T H.324 (1998), *Terminal para comunicación multimedia a baja velocidad binaria.*
- [25] Recomendación UIT-T I.363 (1996), *Especificación de la capa de adaptación del modo transferencia asíncrono de la RDSI de banda ancha.*
- [26] Recomendación UIT-T Q.2931 (1995), *Red digital de servicios integrados de banda ancha – Sistema de señalización digital de abonado número 2 – Especificación de la capa 3 de la interfaz usuario-red para el control de llamada/conexión básica.*
- [27] Recomendación UIT-T T.30 (1996), *Procedimientos de transmisión de documentos para facsímil por la red telefónica general conmutada.*
- [28] Recomendación T.35 del CCITT (1991), *Procedimiento para la asignación de códigos definidos por el CCITT para facilidades no normalizadas.*
- [29] Recomendación T.51 del CCITT (1992), *Juegos de caracteres codificados basados en el alfabeto latino para los servicios de telemática.*
- [30] Recomendación UIT-T T.84 (1996) | ISO/CEI 10918-3:1996, *Tecnología de la información – Compresión digital y codificación digitales de imágenes fijas de tonos continuos – Ampliaciones.*
- [31] Recomendación UIT-T T.120 (1996), *Protocolo de datos para conferencias multimedia.*
- [32] Recomendación UIT-T T.123 (1996), *Pilas de protocolos de datos para conferencias multimedios.*
- [33] Recomendación UIT-T T.434 (1996), *Formato de transferencia de ficheros binarios en los servicios de telemática.*
- [34] Recomendación UIT-T V.14 (1993), *Transmisión de caracteres arrítmicos por canales portadores síncronos.*
- [35] 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 con conmutación y en circuitos arrendados punto a punto a dos hilos de tipo telefónico.*
- [36] Recomendación UIT-T V.42 (1996), *Procedimientos de corrección de errores para los equipos de terminación del circuito de datos que utilizan la conversión de modo asíncrono a modo síncrono.*
- [37] Recomendación UIT-T X.680 (1994), *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*
- [38] Recomendación UIT-T X.680/enm. 1 (1995), *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de la notación básica – Enmienda 1: Reglas de extensibilidad.*
- [39] Recomendación UIT-T X.681/enm. 1 (1995), *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de objetos de información – Enmienda 1: Reglas de extensibilidad.*
- [40] Recomendación UIT-T X.691 (1997), *Tecnología de la información – Reglas de codificación en ASN.1 – Especificación de las reglas de codificación paquetizada.*
- [41] ISO/CEI 3309:1993, *Information Technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame Structure.*

Reemplazada por una versión más reciente

- [42] 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.*
- [43] 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.*
- [44] ISO/CEI 13818-3:1995, *Information Technology – Generic coding of moving pictures and associated audio – Part 3: Audio.*
- [45] ISO/CEI DIS 13818-6, *Information Technology – Generic coding of moving pictures and associated audio – Part 6: Extensions for DSM-CC.*
- [46] ISO/CEI TR 9577:1996, *Information Technology – Protocol identification in the network layer.*
- [47] Recomendación I-ETS 300 036 (GSM 06.10), *Full-rate speech transcoding.*
- [48] Recomendación I-ETS 300 969 (GSM 06.20), *Half-rate speech transcoding.*
- [49] Recomendación I-ETS 300 726 (GSM 06.60), *Enhanced full-rate speech transcoding.*
- [50] Recomendación I-ETS 300 038 (GSM 06.12), *Comfort noise aspects for full-rate speech traffic channels.*
- [51] Recomendación I-ETS 300 971 (GSM 06.22), *Comfort noise aspects for half-rate speech traffic channels.*
- [52] Recomendación I-ETS 300 728 (GSM 06.62), *Comfort noise aspects for enhanced full-rate speech traffic channels.*
- [53] Recomendación I-ETS 300 039 (GSM 06.31), *Discontinuous transmission for full-rate speech traffic channels.*
- [54] Recomendación I-ETS 300 972 (GSM 06.41), *Discontinuous transmission for half-rate speech traffic channels.*
- [55] Recomendación I-ETS 300 729 (GSM 06.81), *Discontinuous transmission for enhanced full-rate speech traffic channels.*
- [56] Recomendación I-ETS 300 037 (GSM 06.11), *Substitution and muting of lost frames for full-rate speech traffic channels.*
- [57] Recomendación I-ETS 300 970 (GSM 06.21), *Substitution and muting of lost frames for half-rate speech traffic channels.*
- [58] Recomendación I-ETS 300 727 (GSM 06.61), *Substitution and muting of lost frames for enhanced full-rate speech traffic channels.*
- [59] Recomendación I-ETS 300 040 (GSM 06.32), *Voice activity detection for full-rate speech traffic channels.*
- [60] Recomendación I-ETS 300 973 (GSM 06.42), *Voice activity detection for half-rate speech traffic channels.*
- [61] Recomendación I-ETS 300 730 (GSM 06.82), *Voice activity detection for enhanced full-rate speech traffic channels.*
- [62] Recomendación I-ETS 300 724 (GSM 06.53), *ANSI-C code for the GSM enhanced full-rate speech.*

Reemplazada por una versión más reciente

3 Definiciones

Para los fines de esta Recomendación se aplican las siguientes definiciones:

- 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 (de ida):** Se utiliza la expresión hacia adelante, (o de ida) 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 director (terminal amo):** Un terminal director es el terminal que ha sido determinado como el terminal director por el procedimiento de determinación director-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 multimedios:** Por comunicación multimedios ha de entenderse la transmisión y/o recepción simultáneas de señales de dos o más tipos de medios.
- 3.18 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.

Reemplazada por una versión más reciente

- 3.19 no normalizado:** No conforme con una norma nacional o internacional a que se hace referencia en esta Recomendación.
- 3.20 de salida:** Una entidad de señalización de salida es una entidad que inicia un procedimiento.
- 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 (o inversa):** 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 (terminal esclavo):** 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 llamada. 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

A los fines de esta Recomendación, se utilizan las siguientes siglas:

AAL	Capa de adaptación ATM (<i>ATM adaptation layer</i>)
AL1, 2, 3	Capas 1, 2 y 3 de adaptación 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>bi-directional 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 Recomendaciones H.261 y H.263)

Reemplazada por una versión más reciente

CLCSE	Entidad de señalización de cierre de canal lógico (<i>close logical channel signalling entity</i>)
CPCS	Subcapa de convergencia de la parte común (<i>common part convergence sublayer</i>) (de la capa 5 de adaptación ATM)
DSM-CC	Instrucción y control de medios de almacenamiento digitales (<i>digital storage media-command and control</i>)
DTMF	Multifrecuencia de doble tono (<i>dual tone multi-frequency</i>)
GOB	Grupo de bloques (<i>group of blocks</i>) (de una imagen vídeo: véanse las Recomendaciones H.261 y H.263)
HDLC	Control de alto nivel del enlace de datos (<i>high-level data link control</i>)
HRD	Decodificador de referencia hipotético (<i>hypothetical reference decoder</i>) (véanse las Recomendaciones H.261 y H.263)
IV	Vector de inicialización (<i>initialisation vector</i>) (utilizado para encriptación: véanse las Recomendaciones H.233 y H.234)
LAPM	Protocolo de acceso al enlace para módems (<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 de bucle de mantenimiento (<i>maintenance loop signalling entity</i>)
MPI	Intervalo mínimo de imagen (<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 director-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 Rec. UIT-T H.222.0 ISO/CEI 13818-1)
PID	Identificador de paquete (<i>packet identifier</i>) (véase 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 tiempo de propagación 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>)

Reemplazada por una versión más reciente

SE	Mensaje de intercambio de sesión (<i>session exchange message</i>) (utilizado para encriptación: véanse las Recomendaciones H.233 y H.234)
SQCIF	Subcuarto de CIF (<i>sub QCIF</i>)
STD	Decodificador de destino del sistema (<i>system target decoder</i>) (véase Rec. UIT-T H.222.0 ISO/CEI 13818-1)
VC	Canal virtual 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 director 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 [37] 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 director-subordinado

Se presentan conflictos cuando dos terminales que intervienen en una llamada inician simultáneamente eventos similares y solo 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 director y el otro actuará como terminal subordinado. La reglas especificarán el comportamiento del terminal director y del terminal subordinado en caso de conflicto.

El procedimiento de determinación director-subordinado permite a los terminales en una llamada determinar cuál es el director 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 directa-subordinado si no está activo localmente ningún procedimiento que dependa de su resultado.

Reemplazada por una versión más reciente

5.2 Intercambio de capacidad

Los procedimientos de intercambio de capacidad tienen por finalidad asegurar que únicamente las señales multimedios 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 este 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 puede 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.

Reemplazada por una versión más reciente

Una parte de este protocolo se aplica a la apertura de canales bidireccionales. Para evitar problemas de temporización, se define un terminal como terminal director y el otro como terminal subordinado. Únicamente el terminal director puede iniciar la apertura de canales bidireccionales: sin embargo, el terminal subordinado puede solicitar al terminal director 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 (o cuadro) 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 les transmita 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

Reemplazada por una versión más reciente

fuentes, 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.

6 Mensajes: sintaxis

En esta cláusula se especifica la sintaxis de mensajes que utilizan la notación definida en ASN.1 [37]. Los mensajes se codificarán para transmisión aplicando las reglas de codificación compactada especificadas en [38] 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 Recomendación X.691.

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,
}
```

Reemplazada por una versión más reciente

```
roundTripDelayRequest      RoundTripDelayRequest,
maintenanceLoopRequest     MaintenanceLoopRequest,
...,
communicationModeRequest   CommunicationModeRequest,
conferenceRequest          ConferenceRequest
}
```

-- A ResponseMessage is the response to a requestMessage

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

  masterSlaveDeterminationAck
  masterSlaveDeterminationReject   MasterSlaveDeterminationAck,
                                     MasterSlaveDeterminationReject,

  terminalCapabilitySetAck
  terminalCapabilitySetReject       TerminalCapabilitySetAck,
                                     TerminalCapabilitySetReject,

  openLogicalChannelAck
  openLogicalChannelReject         OpenLogicalChannelAck,
                                     OpenLogicalChannelReject,
  closeLogicalChannelAck          CloseLogicalChannelAck,

  requestChannelCloseAck
  requestChannelCloseReject       RequestChannelCloseAck,
                                     RequestChannelCloseReject,

  multiplexEntrySendAck
  multiplexEntrySendReject       MultiplexEntrySendAck,
                                     MultiplexEntrySendReject,

  requestMultiplexEntryAck
  requestMultiplexEntryReject     RequestMultiplexEntryAck,
                                     RequestMultiplexEntryReject,

  requestModeAck
  requestModeReject               RequestModeAck,
                                     RequestModeReject,

  roundTripDelayResponse         RoundTripDelayResponse,

  maintenanceLoopAck
  maintenanceLoopReject          MaintenanceLoopAck,
                                     MaintenanceLoopReject,

  ...,
  communicationModeResponse     CommunicationModeResponse,

  conferenceResponse            ConferenceResponse
}
```

-- A CommandMessage requires action, but no explicit response

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

  maintenanceLoopOffCommand   MaintenanceLoopOffCommand,

  sendTerminalCapabilitySet   SendTerminalCapabilitySet,
```

Reemplazada por una versión más reciente

```
encryptionCommand          EncryptionCommand,
flowControlCommand         FlowControlCommand,
endSessionCommand          EndSessionCommand,
miscellaneousCommand        MiscellaneousCommand,
...,
communicationModeCommand    CommunicationModeCommand,
conferenceCommand           ConferenceCommand,
h223MultiplexReconfiguration H223MultiplexReconfiguration
}

-- 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
}
```

Reemplazada por una versión más reciente

-- SequenceNumber is defined here as it is used in a number of Messages

SequenceNumber ::=INTEGER (0..255)

-- 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 T.35
 t35Extension INTEGER (0..255), -- assigned nationally
 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,
 ...
 },
 ...
}

Reemplazada por una versión más reciente

```
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) 2}

  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,
  }
}
```

Reemplazada por una versión más reciente

```
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
}

H235SecurityCapability          ::=SEQUENCE
{
  encryptionAuthenticationAndIntegrity EncryptionAuthenticationAndIntegrity,

  mediaCapability               CapabilityTableEntryNumber,
  -- Note: the mediaCapability shall refer to Capability Table Entries that do contain a transmit,
  -- receive, or receiveAndTransmit AudioCapability, VideoCapability,
  -- DataApplicationCapability, or similar capability indicated by a NonStandardParameter only
  ...
}
}
```

Reemplazada por una versión más reciente

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

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

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

VCCapability ::= SEQUENCE
{
    aal1                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
            }
        },
        ...
    },
    ...
}
```

Reemplazada por una versión más reciente

```

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
}

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

```

Reemplazada por una versión más reciente

maximumAL1MPDUSize	INTEGER (0..65535),	<i>--units octets</i>
maximumAL2MSDUSize	INTEGER (0..65535),	<i>-- units octets</i>
maximumAL3MSDUSize	INTEGER (0..65535),	<i>-- units octets</i>
...		
}		
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),	<i>-- units in milliseconds</i>
receiveMultipointCapability	MultipointCapability,	
transmitMultipointCapability	MultipointCapability,	
receiveAndTransmitMultipointCapability	MultipointCapability,	
mcCapability	SEQUENCE	
{		
centralizedConferenceMC	BOOLEAN,	
decentralizedConferenceMC	BOOLEAN,	
...		
},		
rtcpVideoControlCapability	BOOLEAN,	<i>-- FIR and NACK</i>
mediaPacketizationCapability	MediaPacketizationCapability,	
...,		
transportCapability	TransportCapability OPTIONAL,	
redundancyEncodingCapability	SEQUENCE SIZE(1..256) OF	
	RedundancyEncodingCapability OPTIONAL,	
logicalChannelSwitchingCapability	BOOLEAN,	
t120DynamicPortCapability	BOOLEAN	
}		
MediaPacketizationCapability	::=SEQUENCE	
{		
h261aVideoPacketization	BOOLEAN,	
...,		
rtpPayloadType	SEQUENCE SIZE(1..256) OF RTPPayloadType	
OPTIONAL		
}		

Reemplazada por una versión más reciente

```
RSVPPParameters ::=SEQUENCE
{
    qosMode           QOSMode OPTIONAL,
    tokenRate         INTEGER (1..4294967295) OPTIONAL,
                    -- rate in bytes/sec
    bucketSize        INTEGER (1..4294967295) OPTIONAL,      -- size in bytes
    peakRate          INTEGER (1..4294967295) OPTIONAL,
                    -- peak bandwidth bytes/sec
    minPoliced        INTEGER (1..4294967295) OPTIONAL,
    maxPktSize        INTEGER (1..4294967295) OPTIONAL,      -- size in bytes
    ...
}

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

-- Ed. Replace with ITU-T ATM Transfer Capability Types
ATMPParameters ::=SEQUENCE
{
    maxNTUSize        INTEGER(0..65535),      -- units in octets
    atmUBR             BOOLEAN,               -- unspecified bit rate
    atmrtVBR           BOOLEAN,               -- real time variable bit rate
    atmnrtVBR          BOOLEAN,               -- non real time variable bit rate
    atmABR             BOOLEAN,               -- available bit rate
    atmCBR             BOOLEAN,               -- constant bit rate
    ...
}

QOSCapability ::=SEQUENCE
{
    nonStandardData    NonStandardParameter OPTIONAL,
    rsvpParameters     RSVPPParameters OPTIONAL,
    atmParameters      ATMPParameters OPTIONAL,
    ...
}

MediaTransportType ::=CHOICE
{
    ip-UDP             NULL,
    ip-TCP             NULL,
    atm-AAL5-UNIDIR   NULL, -- virtual circuits used as uni-directional
    atm-AAL5-BIDIR    NULL, -- virtual circuits used as bi-directional
    ...
}

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

Reemplazada por una versión más reciente

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,
}	
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,
...	
}	

Reemplazada por una versión más reciente

```
MediaDistributionCapability ::=SEQUENCE
{
    centralizedControl          BOOLEAN,
    distributedControl BOOLEAN,
    centralizedAudio           BOOLEAN,
    distributedAudio           BOOLEAN,
    centralizedVideo           BOOLEAN,
    distributedVideo           BOOLEAN,
    centralizedData            SEQUENCE OF DataApplicationCapability OPTIONAL,
    distributedData            SEQUENCE OF DataApplicationCapability OPTIONAL,

    -- for further study in H.323

    ...
}
```

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

```
VideoCapability ::=CHOICE
{
    nonStandard          NonStandardParameter,
    h261VideoCapability  H261VideoCapability,
    h262VideoCapability  H262VideoCapability,
    h263VideoCapability  H263VideoCapability,
    is11172VideoCapability IS11172VideoCapability,

    ...
}
```

```
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 of H.261

    ...
}
```

```
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/sec

    ...
}
```


Reemplazada por una versión más reciente

```

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 EnhancementParameters 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
}

```

Reemplazada por una versión más reciente

slowCif16MPI	INTEGER (1..3600) OPTIONAL, -- <i>units seconds/frame</i>
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,
...	
}	
TransparencyParameters	::= SEQUENCE
{	
presentationOrder	INTEGER(1..256),
offset-x	INTEGER(-262144..262143), -- <i>1/8 pixels</i>
offset-y	INTEGER(-262144..262143), -- <i>1/8 pixels</i>
scale-x	INTEGER(1..255),
scale-y	INTEGER(1..255),
...	
}	

Reemplazada por una versión más reciente

RefPictureSelection	::=SEQUENCE	
{		
additionalPictureMemory	SEQUENCE	
{		
sqcifAdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
qcifAdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
cifAdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
cif4AdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
cif16AdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
bigCpfAdditionalPictureMemory	INTEGER (1..256) OPTIONAL,	<i>-- units frame</i>
...		
} OPTIONAL,		
videoMux	BOOLEAN,	
videoBackChannelSend	CHOICE	
{		
none	NULL,	
ackMessageOnly	NULL,	
nackMessageOnly	NULL,	
ackOrNackMessageOnly	NULL,	
ackAndNackMessage	NULL,	
...		
},		
...		
}		
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),	<i>-- units 4 pixels</i>
maxCustomPictureHeight	INTEGER(1..2048),	<i>-- units 4 pixels</i>
minCustomPictureWidth	INTEGER(1..2048),	<i>-- units 4 pixels</i>
minCustomPictureHeight	INTEGER(1..2048),	<i>-- units 4 pixels</i>
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,		
...		
},		

Reemplazada por una versión más reciente

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,
...	
}	
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/sec
...	
}	

Reemplazada por una versión más reciente

-- 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
}

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,
    ...
}
```

Reemplazada por una versión más reciente

```
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,
    ...
}
```

Reemplazada por una versión más reciente

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

```
DataApplicationCapability ::=SEQUENCE
{
  application CHOICE
  {
    nonStandard NonStandardParameter,
    t120 DataProtocolCapability,
    dsm-cc DataProtocolCapability,
    userData DataProtocolCapability,
    t84 SEQUENCE
    {
      t84Protocol DataProtocolCapability,
      t84Profile T84Profile
    },
    t434 DataProtocolCapability,
    h224 DataProtocolCapability,
    nlpid SEQUENCE
    {
      nlpidProtocol DataProtocolCapability,
      nlpidData OCTET STRING
    },
    dsvdControl NULL,
    h222DataPartitioning DataProtocolCapability,
    ...,
    t30fax DataProtocolCapability,
    t140 DataProtocolCapability
  },
  maxBitRate INTEGER (0..4294967295), -- units 100 bit/s
  ...
}

DataProtocolCapability ::=CHOICE
{
  nonStandard NonStandardParameter,
  v14buffered NULL,
  v42lapm NULL, -- may negotiate to V.42bis
  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,
    ...
  }
}
```

Reemplazada por una versión más reciente

```
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,

        ...
    }
}

-----
-- Encryption Capability Definitions:
-----

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

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


Reemplazada por una versión más reciente

```
MediaEncryptionAlgorithm ::=CHOICE
{
    nonStandard          NonStandardParameter,
    algorithm            OBJECT IDENTIFIER, -- many defined in ISO/IEC 9979
    ...
}
```

```
AuthenticationCapability ::=SEQUENCE
{
    nonStandard          NonStandardParameter 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
    ...
}
```

```
-- =====
-- Capability Exchange Definitions: Conference
-- =====
```

```
ConferenceCapability ::=SEQUENCE
{
    nonStandardData      SEQUENCE OF NonStandardParameter OPTIONAL,
    chairControlCapability BOOLEAN,
    ...
}
```

```
-- =====
-- 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 bi-directional channel request.
```

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

    forwardLogicalChannelParameters SEQUENCE
    {
        portNumber          INTEGER (0..65535) OPTIONAL,
        dataType            DataType,
    }
}
```

Reemplazada por una versión más reciente

```

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 bi-directional 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 uni-directional 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,
    ...
  } OPTIONAL,
  -- For further study in T.120
}

```

Reemplazada por una versión más reciente

<pre> networkAddress { q2931Address e164Address localAreaAddress ... }, associateConference externalReference **** t120SetupProcedure { originateCall waitForCall issueQuery ... } OPTIONAL </pre>	<pre> CHOICE Q2931Address, IA5String(SIZE(1..128)) (FROM ("0123456789#*,")), TransportAddress, BOOLEAN, OCTET STRING(SIZE(1..255)) OPTIONAL, CHOICE NULL, NULL, NULL, </pre>
<pre> Q2931Address { address { internationalNumber nsapAddress ... }, subaddress ... } </pre>	<pre> ::=SEQUENCE CHOICE NumericString(SIZE(1..16)), OCTET STRING (SIZE(1..20)), OCTET STRING (SIZE(1..20)) OPTIONAL, </pre>
<pre> V75Parameters { audioHeaderPresent ... } </pre>	<pre> ::= SEQUENCE BOOLEAN, </pre>
<pre> DataType { nonStandard nullData videoData audioData data encryptionData **** h235Control h235Media } </pre>	<pre> ::=CHOICE NonStandardParameter, NULL, VideoCapability, AudioCapability, DataApplicationCapability, EncryptionMode, NonStandardParameter, H235Media </pre>
<pre> H235Media { encryptionAuthenticationAndIntegrity </pre>	<pre> ::=SEQUENCE EncryptionAuthenticationAndIntegrity, </pre>

Reemplazada por una versión más reciente

mediaType	CHOICE
{	
nonStandard	NonStandardParameter,
videoData	VideoCapability,
audioData	AudioCapability,
data	DataApplicationCapability,
...	
},	
...	
}	
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,
al1Framed	NULL,
al1NotFramed	NULL,
al2WithoutSequenceNumbers	NULL,
al2WithSequenceNumbers	NULL,
al3	SEQUENCE
{	
controlFieldOctets	INTEGER (0..2),
sendBufferSize	INTEGER (0..16777215) <i>-- units octets</i>
},	
...,	
al1M	H223AL1MParameters,
al2M	H223AL2MParameters,
al3M	H223AL3MParameters
},	
segmentableFlag	BOOLEAN,
...	
}	
H223AL1MParameters	::=SEQUENCE
{	
transferMode	CHOICE
{	
framed	NULL,
unframed	NULL,
...	
},	
headerFEC	CHOICE
{	
sebch16-7	NULL,
golay24-12	NULL,
...	
},	

Reemplazada por una versión más reciente

<pre> crcLength { crc4bit crc12bit crc20bit crc28bit ... }, </pre>	<pre> CHOICE NULL, NULL, NULL, NULL, ... </pre>
<pre> rcpcCodeRate </pre>	<pre> INTEGER (8..32) </pre>
<pre> arqType { noArq typeIArq typeIIArq ... }, alpduInterleaving alsduSplitting ... </pre>	<pre> CHOICE NULL, H223AnnexCArqParameters, H223AnnexCArqParameters, ... BOOLEAN, BOOLEAN, ... </pre>
<pre> H223AL2MParameters { headerFEC { sebch16-5 golay24-12 ... }, alpduInterleaving ... } </pre>	<pre> ::=SEQUENCE CHOICE NULL, NULL, ... BOOLEAN, ... </pre>
<pre> H223AL3MParameters { headerFormat { sebch16-7 golay24-12 ... }, crcLength { crc4bit crc12bit crc20bit crc28bit ... }, rcpcCodeRate </pre>	<pre> ::=SEQUENCE CHOICE NULL, NULL, ... CHOICE NULL, NULL, NULL, NULL, ... INTEGER (8..32) </pre>
<pre> arqType { noArq typeIArq typeIIArq ... }, </pre>	<pre> CHOICE NULL, H223AnnexCArqParameters, H223AnnexCArqParameters, ... </pre>

Reemplazada por una versión más reciente

```

    alpduInterleaving          BOOLEAN,
    ...
}

H223AnnexCArqParameters      ::=SEQUENCE
{
    numberOfRetransmissions    CHOICE
    {
        finite                 INTEGER (1..16),
        infinite               NULL,
        ...
    },
    sendBufferSize             INTEGER (0..16777215),    -- units octets
    ...
}

V76LogicalChannelParameters  ::=SEQUENCE
{
    hdlcParameters            V76HDLCParameters,
    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,
    ...
}

V76HDLCParameters            ::=SEQUENCE
{
    crcLength                  CRCLength,
    n401                       INTEGER (1..4095),
    loopbackTestProcedure      BOOLEAN,
    ...
}

```

Reemplazada por una versión más reciente

CRCLength	::=CHOICE
{	
crc8bit	NULL,
crc16bit	NULL,
crc32bit	NULL,
...	
}	
H2250LogicalChannelParameters	::=SEQUENCE
{	
nonStandard	SEQUENCE OF NonStandardParameter OPTIONAL,
sessionID	INTEGER(0..255),
associatedSessionID	INTEGER(1..255) OPTIONAL,
mediaChannel	TransportAddress OPTIONAL,
mediaGuaranteedDelivery	BOOLEAN OPTIONAL,
mediaControlChannel	TransportAddress OPTIONAL, -- reverse RTCP channel
mediaControlGuaranteedDelivery	BOOLEAN OPTIONAL,
silenceSuppression	BOOLEAN OPTIONAL,
destination	TerminalLabel OPTIONAL,
dynamicRTPPayloadType	INTEGER(96..127) OPTIONAL,
mediaPacketization	CHOICE
{	
h261aVideoPacketization	NULL,
...,	
rtpPayloadType	RTPPayloadType
} OPTIONAL,	
...,	
transportCapability	TransportCapability OPTIONAL,
redundancyEncoding	RedundancyEncoding OPTIONAL,
source	TerminalLabel OPTIONAL}
RTPPayloadType	::= SEQUENCE
{	
payloadDescriptor	CHOICE
{	
nonStandardIdentifier	NonStandardParameter,
rfc-number	INTEGER (1..32768, ...),
oid	OBJECT IDENTIFIER,
...	
},	
payloadType	INTEGER (0..127) OPTIONAL,
...	
}	
}	
RedundancyEncoding	::=SEQUENCE
{	
redundancyEncodingMethod	RedundancyEncodingMethod,
secondaryEncoding	Data Type OPTIONAL, -- depends on method
...	
}	
TransportAddress	::=CHOICE
{	
unicastAddress	UnicastAddress,
multicastAddress	MulticastAddress,
...	
}	

Reemplazada por una versión más reciente

UnicastAddress	::=CHOICE
{	
iPAddress	SEQUENCE
{	
network	OCTET STRING (SIZE(4)),
tsapIdentifier	INTEGER(0..65535),
...	
},	
iPXAAddress	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
}	

Reemplazada por una versión más reciente

EncryptionSync	::=SEQUENCE <i>-- used to supply new key and synchronization point</i>
{	
nonStandard	NonStandardParameter OPTIONAL,
synchFlag	INTEGER(0..255) , -- may need to be larger for H.324, etc. <i>-- shall be the Dynamic Payload# for H.323</i>
h235Key	OCTET STRING (SIZE(1..65535)), -- H.235 encoded value
escrowentry	SEQUENCE SIZE(1..256) OF EscrowData 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, <i>-- H.223 parameters are never present in reverse direction</i>
...	
h2250LogicalChannelParameters	H2250LogicalChannelParameters
} OPTIONAL,	<i>-- Not present for H.223</i>
...,	
replacementFor	LogicalChannelNumber OPTIONAL
} OPTIONAL,	<i>-- Not present for uni-directional channel request</i>
...,	
separateStack	NetworkAccessParameters OPTIONAL, <i>-- for Open requester to establish the stack</i>
forwardMultiplexAckParameters	CHOICE
{	
<i>-- H.222 parameters are never present in the Ack</i>	
<i>-- H.223 parameters are never present in the Ack</i>	
<i>-- V.76 parameters are never present in the Ack</i>	
h2250LogicalChannelAckParameters	H2250LogicalChannelAckParameters,
...	
} OPTIONAL,	
encryptionSync	EncryptionSync OPTIONAL <i>-- used only by Master</i>
}	
OpenLogicalChannelReject	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
cause	CHOICE
{	
unspecified	NULL,
unsuitableReverseParameters	NULL,
}	

Reemplazada por una versión más reciente

dataTypeNotSupported	NULL,
dataTypeNotAvailable	NULL,
unknownDataType	NULL,
dataTypeALCombinationNotSupported	NULL,
....	
multicastChannelNotAllowed	NULL,
insufficientBandwidth	NULL,
separateStackEstablishmentFailed	NULL,
invalidSessionID	NULL,
masterSlaveConflict	NULL,
waitForCommunicationMode	NULL,
invalidDependentChannel	NULL,
replacementForRejected	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
dynamicRTPPayloadType	INTEGER(96..127) OPTIONAL, -- used only by the master
	-- or MC
....	
flowControlToZero	BOOLEAN
}	
CloseLogicalChannel	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
source	CHOICE
{	
user	NULL,
lcse	NULL
},	
....	
reason	CHOICE
{	
unknown	NULL,
reopen	NULL,
reservationFailure	NULL,
...	
}	
}	
CloseLogicalChannelAck	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
...	
}	

Reemplazada por una versión más reciente

```
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
    },
}
```

Reemplazada por una versión más reciente

repeatCount { finite untilClosingFlag }	CHOICE INTEGER (1..65535), NULL	 <i>-- repeats of type</i> <i>-- used for last element</i>
MultiplexTableEntryNumber	::=INTEGER (1..15)	
MultiplexEntrySendAck { sequenceNumber multiplexTableEntryNumber ... }	::=SEQUENCE SequenceNumber, SET SIZE (1..15) OF MultiplexTableEntryNumber, ...	
MultiplexEntrySendReject { sequenceNumber rejectionDescriptions ... }	::=SEQUENCE SequenceNumber, SET SIZE (1..15) OF MultiplexEntryRejectionDescriptions, ...	
MultiplexEntryRejectionDescriptions { multiplexTableEntryNumber cause { unspecifiedCause descriptorTooComplex ... }, ... }	::=SEQUENCE MultiplexTableEntryNumber, CHOICE NULL, NULL, ...	
MultiplexEntrySendRelease { multiplexTableEntryNumber ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...	
RequestMultiplexEntry { entryNumbers ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...	
RequestMultiplexEntryAck { entryNumbers ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...	
RequestMultiplexEntryReject { entryNumbers rejectionDescriptions ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, SET SIZE (1..15) OF RequestMultiplexEntryRejectionDescriptions, ...	

Reemplazada por una versión más reciente

```
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
{
    ...
}
```

Reemplazada por una versión más reciente

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

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

ModeElement ::= SEQUENCE
{
    type CHOICE
    {
        nonStandard NonStandardParameter,
        videoMode VideoMode,
        audioMode AudioMode,
        dataMode DataMode,
        encryptionMode EncryptionMode,
        ...,
        h235Mode H235Mode
    },
    h223ModeParameters H223ModeParameters OPTIONAL,
    ...,
    v76ModeParameters V76ModeParameters OPTIONAL,
    h2250ModeParameters H2250ModeParameters OPTIONAL
}

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

H223ModeParameters ::=SEQUENCE
{
    adaptationLayerType CHOICE
    {
        nonStandard NonStandardParameter,
        al1Framed NULL,
        al1NotFramed NULL,
        al2WithoutSequenceNumbers NULL,
        al2WithSequenceNumbers NULL,
        al3 SEQUENCE
        {
            controlFieldOctets INTEGER(0..2),
            sendBufferSize INTEGER(0..16777215) -- units octets
        },
        ...,
        al1M H223AL1MParameters,
        al2M H223AL2MParameters,
        al3M H223AL3MParameters
    },
}
```

Reemplazada por una versión más reciente

```
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,
    ...
}

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

Reemplazada por una versión más reciente

```

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/sec
    ...
}
H263VideoMode ::=SEQUENCE
{
    resolution CHOICE
    {
        sqcif NULL,
        qcif NULL,
        cif NULL,
        cif4 NULL,
        cif16 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/sec
    ...
}

```


Reemplazada por una versión más reciente

-- =====
-- 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
}

```

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

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

```

Reemplazada por una versión más reciente

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	
{		
maxAI-sduAudioFrames	INTEGER (1..256),	
silenceSuppression	BOOLEAN,	
g723AnnexCAudioMode	SEQUENCE	
{		
highRateMode0	INTEGER (27..78),	-- units octets
highRateMode1	INTEGER (27..78),	-- units octets
}		

Reemplazada por una versión más reciente

```

lowRateMode0          INTEGER (23..66),          -- units octets
lowRateMode1          INTEGER (23..66),          -- units octets
sidMode0              INTEGER (6..17),          -- units octets
sidMode1              INTEGER (6..17),          -- units octets
...
},
...
}

```

=====
-- 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
  },
  bitRateINTEGER (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,
  ...
}

```

Reemplazada por una versión más reciente

```
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
{
    ...
}
```

Reemplazada por una versión más reciente

```
-----  
-- 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)
```

Reemplazada por una versión más reciente

dropTerminal	TerminalLabel , -- <i>same as H.230 CCD(term->MC)</i>
requestTerminalID	TerminalLabel , -- <i>same as TCP (term->MC)</i>
enterH243Password	NULL, -- <i>same as H.230 TCS1(MC->term)</i>
enterH243TerminalID	NULL, -- <i>same as H.230 TCS2/TCI</i> -- <i>(MC->term)</i>
enterH243ConferenceID	NULL, -- <i>same as H.230 TCS3 (MC->term)</i>
enterExtensionAddress	NULL, -- <i>same as H.230 TCS4 (GW->term)</i>
requestChairTokenOwner	NULL, -- <i>same as H.230 TCA (term-</i>
<i>>MC)</i>	
requestTerminalCertificate	SEQUENCE
{	
terminalLabel	TerminalLabel OPTIONAL,
certSelectionCriteria	CertSelectionCriteria OPTIONAL,
sRandom	INTEGER (1..4294967295) OPTIONAL,
	-- <i>this is the requester's challenge</i>
...	
},	
broadcastMyLogicalChannel	LogicalChannelNumber , -- <i>similar to H.230 MCV</i>
makeTerminalBroadcaster	TerminalLabel , -- <i>similar to H.230 VCB</i>
sendThisSource	TerminalLabel , -- <i>similar to H.230 VCS</i>
requestAllTerminalIDs	NULL,
remoteMCRequest	RemoteMCRequest
}	
CertSelectionCriteria	::=SEQUENCE SIZE (1..16) OF Criteria
Criteria	::=SEQUENCE
{	
field	OBJECT IDENTIFIER , -- <i>may include certificate type</i>
value	OCTET STRING (SIZE(1..65535)) ,
...	
}	
TerminalLabel	::=SEQUENCE
{	
mcuNumber	McuNumber ,
terminalNumber	TerminalNumber ,
...	
}	
McuNumber	::=INTEGER(0..192)
TerminalNumber	::=INTEGER(0..192)
-- =====	
-- <i>Conference Response definitions</i>	
-- =====	
ConferenceResponse	::=CHOICE
{	
mCTerminalIDResponse	SEQUENCE -- <i>response to TCP(same as TIP)</i> -- <i>sent by MC only</i>
{	
terminalLabel	TerminalLabel ,
terminalID	TerminalID ,
...	
},	

Reemplazada por una versión más reciente

terminalIDResponse { terminalLabel terminalID ... },	SEQUENCE TerminalLabel, TerminalID,	-- response to TCS2 or TCI -- same as IIS -- (term->MC)
conferenceIDResponse { terminalLabel conferenceID ... },	SEQUENCE TerminalLabel, ConferenceID,	-- response to TCS3 -- same as IIS -- (term->MC)
passwordResponse { terminalLabel password ... },	SEQUENCE TerminalLabel, Password,	-- response to TCS1 -- same as IIS -- (term->MC)
terminalListResponse	SET SIZE (1..256) OF TerminalLabel,	
videoCommandReject terminalDropReject	NULL, NULL,	-- same as H.230 VCR -- same as H.230 CIR
makeMeChairResponse { grantedChairToken deniedChairToken NULL, ... }, ...,	CHOICE NULL,	-- same as H.230 CCR -- same as H.230 CIT -- same as H.230 CCR
extensionAddressResponse { extensionAddress ... },	SEQUENCE TerminalID,	-- response to TCS4 -- same as IIS (term->GW)
chairTokenOwnerResponse { terminalLabel terminalID ... },	SEQUENCE TerminalLabel, TerminalID,	-- response to TCA(same as TIR) -- sent by MC only
terminalCertificateResponse { terminalLabel certificateResponse ... },	SEQUENCE TerminalLabel OPTIONAL, OCTET STRING (SIZE(1..65535)) OPTIONAL,	
broadcastMyLogicalChannelResponse { grantedBroadcastMyLogicalChannel NULL, deniedBroadcastMyLogicalChannel NULL, ... },	CHOICE	

Reemplazada por una versión más reciente

```

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,
        ...
    },
    ...
}

```


Reemplazada por una versión más reciente

=====
-- 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
    },
    ...
}
```

Reemplazada por una versión más reciente

```
-----  
-- 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,  
        ...  
    }  
}
```

```
-----  
-- 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)),  
    ...  
}
```

Reemplazada por una versión más reciente

```

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

MiscellaneousCommand ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    type CHOICE
    {
        equalizeDelay 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),
            ...
        },
        maxH223MUXPDUsSize 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
    },
    ...
}

```

Reemplazada por una versión más reciente

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

```
EncryptionUpdateRequest                           ::=SEQUENCE
{
    keyProtectionMethod                         KeyProtectionMethod OPTIONAL,
    ...
}
```

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

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

Reemplazada por una versión más reciente

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

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

```
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  
}
```

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

```
MiscellaneousIndication ::=SEQUENCE  
{  
    logicalChannelNumber LogicalChannelNumber,
```

Reemplazada por una versión más reciente

```

type
{
  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
  ...
}

```

Reemplazada por una versión más reciente

```
=====  
-- 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,  
            }  
        }  
    }  
}
```

Reemplazada por una versión más reciente

```

        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
-- =====

```

```

UserInputIndication            ::=CHOICE
{
    nonStandard                 NonStandardParameter,
    alphanumeric                GeneralString,
    ...,
    userInputSupportIndication  CHOICE
    {
        nonStandard            NonStandardParameter,
        basicString            NULL,
        iA5String              NULL,
        generalString          NULL,
        ...
    },
    signal                       SEQUENCE
    {
        signalType             IA5String (SIZE (1) ^ FROM ("0123456789#*ABCD!")),
    }
}

```


Reemplazada por una versión más reciente

```
duration          INTEGER (1..65535) OPTIONAL, -- milliseconds
rtp               SEQUENCE

{
    timestamp      INTEGER (0..4294967295) OPTIONAL,
    expirationTime INTEGER (0..4294967295) OPTIONAL,
    logicalChannelNumber LogicalChannelNumber,
    ...
} OPTIONAL,
...
},
signalUpdate     SEQUENCE
{
    duration       INTEGER (1..65535), -- milliseconds
    rtp            SEQUENCE
    {
        logicalChannelNumber LogicalChannelNumber,
        ...
    } OPTIONAL,
    ...
}
}

END
```

7 Mensajes: definiciones semánticas

En esta cláusula se facilitan definiciones y limitaciones semánticas para los elementos de sintaxis definidos en la sección anterior.

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

RequestMessage: Mensaje de petición que provoca una acción por parte del terminal distante y exige una respuesta inmediata del mismo. Puede utilizarse `nonStandardRequest` para enviar peticiones no normalizadas.

ResponseMessage: Mensaje de respuesta a un mensaje petición. Puede utilizarse `nonStandardResponse` para enviar respuestas no normalizadas.

CommandMessage: Mensaje de instrucción que requiere una acción pero no una respuesta explícita. Puede utilizarse `nonStandardCommand` para enviar instrucciones no normalizadas.

IndicationMessage: Contiene información que no exige ni acción ni respuesta. Puede utilizarse `nonStandardIndication` para enviar indicaciones no normalizadas.

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.

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: indicativo de país (octeto 1 como en T.35 [28]; octeto 2*), indicativo de fabricante (los dos octetos* siguientes), *= asignado nacionalmente. Los indicativos de fabricante son los mismos asignados para uso en 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

Reemplazada por una versión más reciente

mensajes h221NonStandard provienen de espacios que no se superponen y no pueden confundirse. Sin embargo, como los mensajes h221NonStandard son también utilizados por H.320, proceden del mismo espacio que los mensajes H.320, y deberán tener el mismo significado.

7.1 Mensajes de determinación director-subordinado

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

7.1.1 Determinación director-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, MCUs y cabeceras. 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 $0..2^{24}-1$.

7.1.2 Acuse de recibo de determinación director-subordinado

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

7.1.3 Rechazo de determinación director-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.

7.1.4 Liberación de determinación de director-subordinado

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

7.2 Mensajes de capacidad de terminal

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

7.2.1 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}}

Reemplazada por una versión más reciente

significa que el terminal puede funcionar con los dos canales vídeo y un canal de audio simultáneamente: un canal de vídeo según la Recomendación H.261, otro canal de vídeo según las Recomendaciones 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 sustentar 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`.

7.2.2 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 (o instancias) 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 A, 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.

7.2.2.1 Tabla (o cuadro) 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`s o carecer de ellas. En el arranque no se definen inscripciones del cuadro. 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`.

Reemplazada por una versión más reciente

7.2.2.2 Descriptores de capacidad

Se utilizan los CapabilityDescriptors para expresar 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 CapabilityDescriptors. Si un terminal tiene preferencia sobre el modo que desearía transmitir o recibir y quiere expresar tal preferencia cuando transmite sus capacidades, puede hacerlo proporcionando CapabilityDescriptors 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 CapabilityTableEntrys que se hayan definido, aunque es posible definir CapabilityTableEntrys 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 AlternativeCapabilitySets 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 sustentar 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 AlternativeCapabilitySets 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.

7.2.2.3 Capacidad

Las elecciones receiveVideoCapability, receiveAudioCapability, receiveDataApplicationCapability y receiveUserInputCapability indican la capacidad de recepción según la VideoCapability, AudioCapability, DataApplicationCapability y UserInputCapability respectivas.

Las elecciones transmitVideoCapability, transmitAudioCapability, transmitDataApplicationCapability y transmitUserInputCapability indican la capacidad de transmitir según la VideoCapability, AudioCapability, DataApplicationCapability y UserInputCapability respectivas.

Las elecciones receiveAndTransmitVideoCapability, receiveAndTransmitAudioCapability, receiveAndTransmitDataApplicationCapability y receiveAndTransmitUserInputCapability indican la capacidad para recibir y transmitir según las VideoCapability, AudioCapability, DataApplicationCapability y UserInputCapability respectivas. Estos puntos de código pueden ser útiles para expresar que las capacidades de recepción y transmisión no son independientes.

La variable booleana h233EncryptionTransmitCapability, cuando toma el valor verdadero, indica que el terminal sustenta encriptación, de conformidad con las Recomendaciones H.233 y H.234 [14], [15].

Reemplazada por una versión más reciente

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.

La variable booleana `h235SecurityCapability` indica las capacidades que sustenta el terminal de conformidad con la Recomendación 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.

`EncryptionAuthenticationAndIntegrity` indica las capacidades de encriptación, autenticación e integridad que se sustentan para la `mediaCapability` señalada. La `mediaCapability` define el audio, el vídeo o los algoritmos de datos sustentados 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.

7.2.2.4 Capacidades de múltiplex

`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 Recomendación H.222.1 [9].

`numberOfVCs` indica el número de canales virtuales (VC) ATM simultáneos que puede sustentar 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 la Recomendación I.363 [25] sustentadas. En el cuadro 1 se definen los puntos de código.

Reemplazada por una versión más reciente

Cuadro 1/H.245 – Puntos de código de la capa 1 de adaptación de 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 sustenta la corrección de errores.
LongInterleaver	Se sustenta el método de corrección intrínseca de errores para el transporte de señales sensibles a las pérdidas.
ShortInterleaver	Se sustenta el método de corrección intrínseca de errores para el transporte de señales sensibles al retardo.
ErrorCorrectionOnly	Se sustenta el método de corrección intrínseca de errores sin entrelazado de células.
StructuredDataTransfer	Se sustenta la transferencia de datos estructurados.
PartiallyFilledCells	Se sustentan células rellenas parcialmente.

La secuencia aal5, cuando está presente, indica las opciones sustentadas de la capa 5 de adaptación ATM, especificada en la Recomendación I.363 [25]. forwardMaximumSDUSize y backwardMaximumSDUSize indican el tamaño CPCS-SDU máximo en los sentidos hacia delante y hacia detrás, medido en octetos.

Las variables booleanas transportStream y programStream, cuando toman el valor verdadero, expresan la capacidad para sustentar 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 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.

H223Capability: Indica capacidades específicas del múltiplex 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 Recomendación V.42 [36].

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 maximumA12SDUSize y maximumA13SDUSize expresan 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

Reemplazada por una versión más reciente

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 múltiplex.

La indicación básico (*basic*) expresa que el múltiplex únicamente puede recibir los MultiplexEntryDescriptors básicos definidos en la Recomendación H.223 [10].

La indicación intensificado (*enhanced*) expresa que el múltiplex puede recibir MultiplexEntryDescriptors intensificados con los parámetros adicionales definidos seguidamente.

La profundidad maximumNestingDepth indica la profundidad máxima de anidamiento de los campos subElementList invocados recursivamente. Los MultiplexEntryDescriptors 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 expresa 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 múltiplex 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 de recibir y procesar las AL-PDU descritas en el anexo C/H.223, con la condición siguiente.

Las variables booleanas videoWithAL1M, videoWithAL2M, videoWithAL3M, audioWithAL1M, audioWithAL2M, audioWithAL3M, dataWithAL1M, dataWithAL2M y dataWithAL3M, 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 (AL1M, AL2M o AL3M).

alpdInterleaving, cuando toma el valor verdadero, indica la capacidad de recibir y procesar las AL-PDU para las que se aplica el entrelazado.

El entero maximumAL1MPDUSize expresa el número máximo de octetos de cada PDU que puede recibir el terminal cuando utilice la capa de adaptación AL1M.

Reemplazada por una versión más reciente

Los enteros `maximumAL2MSDUSize` y `maximumAL3MSDUSize` expresan el número máximo de octetos de cada SDU que puede recibir el terminal cuando utilice las capas de adaptación AL2M y AL3M, respectivamente.

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 de la V.76.

`numOfDLCs` indica el número de DLCs 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 Recomendación V.76.

`n401Capability` indica el valor máximo de N401 descrito en la Recomendación 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.

Reemplazada por una versión más reciente

h261aVideoPacketizationCapability indica que se está utilizando el formato de cabida útil RTP alternativo de H261, descrito en H.225.0.

rtpPayloadType indica los esquemas de paquetización de la cabida útil RTP admitidos 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 de Ingeniería de 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.

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 estadísticamente, 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 que se ha de utilizar. Si se utiliza un tipo de cabida útil dinámico, los valores del campo payloadType y el campo dynamicPayloadType deberán concordar.

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 el anexo N/H.263+ o que deberá abrirse un canal lógico que 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 admite 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 admitidos 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).

Reemplazada por una versión más reciente

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 presente Recomendación 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 [31] a una dirección de transporte dinámica en vez de a la dirección normalizada de puerto bien conocida definida en la Recomendación T.123 [32].

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 admite el soporte del canal de medio por una capa de red IP y una capa de transporte TCP. atm-AAL5-UNIDIR indica que el punto extremo admite el transporte del canal de medio por un circuito virtual unidireccional AAL5 del ATM. atm-AAL5-BIDIR indica que el punto extremo admite 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.

Reemplazada por una versión más reciente

7.2.2.5 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 (o instancias) de VideoCapability en AlternativeCapabilitySets diferentes dentro de un único CapabilityDescriptor.

H261VideoCapability: Expresa capacidades H.261 [18].

Si está presente, qcifMPI expresa 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 expresa 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.

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 2.

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

Punto de código ASN.1	Unidades del parámetro referenciado
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: Expresa capacidades H.263 [20].

Si está presente, sqcifMPI expresa 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.

Reemplazada por una versión más reciente

Si está presente, `qcifMPI` expresa 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` expresa 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` expresa 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` expresa 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, expresan la capacidad de transmitir y/o recibir esos modos facultativos definidos en los anexos de la Recomendación 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 utilizada en ciertas aplicaciones de muy baja 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.

Reemplazada por una versión más reciente

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, `slowCif6MPI` indica que 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 expresar 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 `videoNotDecodedMBs` 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) de la Recomendación 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 Recomendación H.263.

EnhancementLayerInfo: indica capacidad para el modo escalabilidad de la Recomendación 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.

Reemplazada por una versión más reciente

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 Recomendación 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 admitir una gama de vectores de movimiento sin límite si se indica también el modo vector de movimiento sin restricción del 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

Reemplazada por una versión más reciente

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 admitir 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 admitir 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 admitir al 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 admitir 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`.

`independentSegmentDecoding`, cuando toma el valor verdadero, indica la capacidad de un codificador o de un decodificador de admitir 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 admitir 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 admitir 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 admitir 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 admitir 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 admitir el modo códigos de longitud variable (VLC) Inter alternativos del anexo S/H.263.

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

Reemplazada por una versión más reciente

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

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 ó 2: si es 0, indica que se admite el tipo de soporte de transparencia segundo plano de la imagen de referencia (RPB) 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 píxels, 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 admite 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 admitir el modo canal lógico separado: ninguna otra capacidad de vídeo se indicará en la misma `H.263VideoCapability`, 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 `H.263VideoMode` 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.

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 admita 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 `H.263VideoMode`, 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

Reemplazada por una versión más reciente

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`, `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 `H.263VideoMode`, indica que se prefiere la recepción de mensajes de canal de retorno de vídeo en modo múltiplex de vídeo. Cuando se utiliza en `H.263VideoMode`, `VideoMux` (múltiplex de vídeo) y `separateVideoBackChannel` no tomarán el valor verdadero.

Reemplazada por una versión más reciente

videoBackChannelSend indica el tipo de mensaje de canal de retorno de vídeo que soporta el terminal. Si se señala en H.263VideoMode, 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.

CustomPictureClockFrequency: indica la capacidad de admitir 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 frequency*) 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 H.263VideoCapability), 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 Recomendación 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 expresa 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 expresa 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 expresa 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.

Reemplazada por una versión más reciente

Si está presente, cif4MPI expresa 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 expresa 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 admitir 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 que un codificador o un decodificador pueden admitir, 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 Recomendación 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 expresa 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 admitir 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 admitir la relación de aspecto de píxel indicada por el código PAR de la Recomendación H.263.

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

H263VideoModeCombos

Cuando está presente, se utiliza h263VideoModeCombos para indicar dependencias entre modos opcionales de la Recomendación 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.

Reemplazada por una versión más reciente

`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 conjunto cualquiera 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 Recomendación 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 en `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`, `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`.

Reemplazada por una versión más reciente

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 Recomendación 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 a menos que esté presente refPicturesSelection en el mismo mensaje H263Options.

IS11172 VideoCapability: indica capacidades IS11172 [42].

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 3.

Cuadro 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 2.4.3.2 de IS11172-2
luminanceSampleRate	muestras por segundo

7.2.2.6 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 instancias de AudioCapability en AlternativeCapabilitySets 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 multiplex H.222.1, esos números se refieren al tamaño de la memoria tampón STD disponible, en unidades de 256 octetos. Cuando se emplee un multiplex H.223, tales números hacen referencia al número máximo de tramas de audio por cada AL-SDU. Cuando se utiliza el multiplex H.225.0 estos números indican el número máximo de tramas de audio por paquete. El significado exacto de los puntos de código se indica en el cuadro 4.

Reemplazada por una versión más reciente

Cuadro 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 μ
g711Ulaw56k	Audio G.711 a 56 kbit/s, ley μ , 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.7231 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 G.729AnnexA 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 G.729AnnexA a 8 kbit/s con supresión de periodos de silencio como en el anexo B
g7231AnnexCCapability	G.723.1 con anexo C
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)

G7231: indica la aptitud para procesar audio con la codificación/ decodificación G723.1. maxA1-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.

G7231AnnexCCapability: indica la aptitud para procesar audio con la codificación/decodificación G723.1, con su anexo C. maxA1-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 G.723.1 anexo C que se utilizarán en el canal lógico.

IS11172AudioCapability: Expresa la aptitud para el procesamiento de audio codificado según la ISO/CEI 11172-3 [43].

Las variables booleanas que tengan el valor de verdadero expresan 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

Reemplazada por una versión más reciente

pueden procesarse, respectivamente. Las variables booleanas `singleChannel` y `twoChannels` expresan 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 kbits por segundo.

IS13818AudioCapability: Expresa la aptitud para el procesamiento de audio codificado según la ISO/CEI 13818-3 [44].

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 expresan la capacidad de funcionamiento en modos concretos, como se indica en el cuadro 5.

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

Punto de código ASN.1	Significación semántica del punto de código
<code>singleChannel</code>	Un canal, utilizando la configuración 1/0. Modo canal único (como en la ISO/CEI 11172-3)
<code>twoChannels</code>	Dos canales, utilizando la configuración 2/0. Modo canal estéreo o dual (como en la ISO/CEI 11172-3)
<code>threeChannels2-1</code>	Tres canales, utilizando la configuración 2/1. Canal izquierdo, derecho y canal ambiente único
<code>threeChannels3-0</code>	Tres canales, utilizando la configuración 3/0. Canales izquierdo, central y derecho, sin canal ambiente
<code>fourChannels2-0-2-0</code>	Cuatro canales, utilizando la configuración 2/0 + 2/0. Canales izquierdo y derecho del primer programa e izquierdo y derecho del segundo programa
<code>fourChannels2-2</code>	Cuatro canales, utilizando la configuración 2/2. Canales izquierdo, derecho, izquierdo ambiente y derecho ambiente
<code>fourChannels3-1</code>	Cuatro canales, utilizando la configuración 3/1. Canales izquierdo, central, derecho y un único canal ambiente
<code>fiveChannels3-0-2-0</code>	Cinco canales, utilizando la configuración 3/0 + 2/0. Izquierdo, central y derecho del primer programa e izquierdo y derecho del segundo programa
<code>fiveChannels3-2</code>	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 sustentar hasta siete canales multilingües. Cuando toma el valor falso expresa que no se sustenta ningún canal multilingüe.

El entero `bitRate` indica la máxima capacidad de velocidad de bits de audio medida en unidades kbits por segundo.

Reemplazada por una versión más reciente

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 admitir 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 **scrambled** (aleatorizado), cuando toma el valor verdadero indica la capacidad de admitir 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).

7.2.2.7 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 casos de **DataApplicationCapability** en **AlternativeCapabilitySets** diferentes dentro de un único **CapabilityDescriptor**.

Las Recomendaciones que empleen esta Recomendación pueden imponer restricciones acerca de cuales 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 sustentarse mediante uno o más **DataProtocolCapabilities**.

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 expresa la capacidad de sustentar el protocolo T.120 [31].

dsm-cc expresa la capacidad de sustentar el protocolo DSM-CC [45].

userData expresa la capacidad de sustentar datos de usuario no especificados procedentes de puertos de datos externos.

t84 expresa la capacidad de sustentar la transferencia de imágenes de tipo T.84 [30] (JPEG, JBIG, Facsímil Gr.3/4).

t434 expresa la capacidad de sustentar la transferencia de ficheros binarios telemáticos T.434 [33].

h224 expresa la capacidad de sustentar el protocolo H.224 [11] de control de dispositivos simples en tiempo real.

nlpid expresa la capacidad de sustentar el protocolo de capa de red especificado mediante **nlpidData**, definido en la ISO/CEI TR9 577 [46]. Estos protocolos incluyen, entre otros, el protocolo Internet (IP) y el protocolo de punto a punto IETF (PPP).

NOTA – El empleo del NLPID se describe con detalle en IETC RFC1490 “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 de H.262, tal como se especifican en H.222.1, en las cuales los datos de intensificación se transmiten como un canal de datos soportado por la **DataProtocolCapability** indicada.

Reemplazada por una versión más reciente

t30fax: este punto de código expresa la capacidad de utilizar el modo analógico del anexo C/T.30 (G3V), especificado en la Recomendación T.39 para los modos DSVF/MSVF.

t140: este punto de código expresa la capacidad de utilizar el protocolo de conversación de texto T.140, especificado en la Recomendación T.140.

DataProtocolCapability: Contiene una lista de protocolos de datos.

v14buffered expresa la capacidad de sustentar una aplicación de datos especificada empleando la Recomendación V.14 con memoria tampón [34].

v42lapm expresa la capacidad de sustentar una aplicación de datos especificada que utilice el protocolo LAPM definido en la Recomendación V.42 [36].

hdlcFrameTunneling expresa la capacidad de sustentar una aplicación de datos especificada que utilice el efecto túnel de tramas HDLC. Véase 4.5.2 de la ISO/CEI 3309 [41].

h310SeparateVCStack indica la capacidad para soportar una aplicación de datos especificada utilizando la pila de protocolos definida en 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 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 para ulterior estudio en 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, y 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 admite 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.

Reemplazada por una versión más reciente

v76wCompression indica la capacidad para soportar la compresión de datos en un canal de datos V.76.

T84Profile: Expresa el tipo de perfil de imagen fija que puede sustentar el terminal.

t84Unrestricted no proporciona ninguna indicación del tipo de imagen fija T.84 que puede sustentar 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 sustentar 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 (200x100ppi).

g3FacsMH200x200 indica el soporte de una imagen facsímil secuencial de grupo 3 MH (Modified Huffman) codificada en dos niveles con alta resolución (200x200ppi).

g4FacsMMR200x100 indica el soporte de una imagen facsímil secuencial de grupo 4 codificada con dos niveles MMR (Modified Modified Reed) con una resolución normal (200x100ppi).

g4FacsMMR200x200 indica el soporte de una imagen facsímil secuencial de grupo 4 codificada con dos niveles MMR (Modified Modified Reed) con alta resolución (200x200ppi).

jbig200x200Seq expresa el soporte de una imagen secuencial de tipo JBIG de dos niveles, codificada en dos niveles con una resolución de 200x200ppi.

jbig200x200Prog expresa el soporte de una imagen progresiva tipo JBIG de dos niveles, codificada en dos niveles con una resolución de 200x200ppi.

jbig300x300Seq expresa el soporte de una imagen secuencial de tipo JBIG de dos niveles, codificada con dos niveles con una resolución de 300x300ppi.

jbig300x300Prog expresa el soporte de una imagen progresiva de tipo JBIG de dos niveles, codificada con dos niveles con una resolución de 300x300ppi.

digPhotoLow expresa el soporte de una imagen secuencial de color codificada según JPEG con un tamaño de imagen de hasta 720x576.

digPhotoMedSeq expresa el soporte de una imagen de color secuencial codificada según JPEG con un tamaño de imagen de hasta 1440x1152.

digPhotoMedProg expresa el soporte de una imagen de color progresiva codificada según JPEG con un tamaño de imagen de hasta 1440x1152.

digPhotoHighSeq expresa el soporte de una imagen de color secuencial codificada según JPEG con un tamaño de imagen de hasta 2880x2304.

digPhotoHighProg expresa el soporte de una imagen de color progresiva codificada según JPEG con un tamaño de imagen de hasta 2880x2304.

Reemplazada por una versión más reciente

7.2.2.8 Capacidades de encriptación, autenticación e integridad

EncryptionCapabilities, si está presente, indica las capacidades de encriptación de un terminal para cada tipo de medio en el que están presentes las capacidades. El alcance de la encriptación 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 encriptación.

AuthenticationCapabilities, si está presente, indica que el terminal admite los componentes de autenticación de la Recomendación H.235 [16].

IntegrityCapabilities, si está presente, indica que el terminal admite los componentes de integridad de la Recomendación H.235 [16].

7.2.2.9 Capacidades de conferencia

ConferenceCapability indica capacidades de conferencia tales como la aptitud para soportar el control de la presidencia que se describe en la Recomendación H.243.

7.2.2.10 Capacidades de entrada de usuario

UserInputCapabilities indica los parámetros del mensaje UserInputIndication que admite el terminal. BasicString indica que el terminal admite la opción basicString de userInputSupportIndication, iA5String indica que el terminal admite la opción iA5String de userInputSupportIndication, y generalString indica que el terminal soporta la opción generalString de userInputSupportIndication. Dtmf indica que el terminal admite dtmf utilizando la señal y los componentes de actualización de señal del mensaje userInputIndication. Hookflash indica que el terminal admite hookflash (señal de enganchado) utilizando la señal y los componentes de actualización de la señal del mensaje userInputIndication.

7.2.3 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.

7.2.4 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 6 se indican los motivos de envío de este mensaje.

Reemplazada por una versión más reciente

Cuadro 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.

7.2.5 Liberación del conjunto de capacidades de terminal

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

7.3 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.

7.3.1 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

Reemplazada por una versión más reciente

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].

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 ulteriormente asociación o información relativa a la `separateStack`.

Si es del tipo `VideoCapability`, `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 una instancia 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 encriptación según se haya especificado.

`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

Reemplazada por una versión más reciente

nuevo no se utilizarán simultáneamente y, por consiguiente, no se rebasará la capacidad de decodificación del receptor.

El campo `encryptionSync` será utilizado por el director cuando acuse la apertura de un canal por un subordinado. Se utiliza para proporcionar el valor de la clave de encriptación y el punto de sincronización en el que deberá utilizarse la clave. En el caso de la Recomendación 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 Recomendación 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.

`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` expresa 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 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 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 Recomendación H.223 [10]. Deberá estar presente en `forwardLogicalChannelParameters` y `reverseLogicalChannelParameters`.

`adaptationLayerType` expresa 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), `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 con los parámetros especificados), `al2M` (AL2M se define en el anexo C con los parámetros especificados) o `al3M` (AL3M se define en el anexo C 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, *forward error correction*) 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, 12, 20 ó 28 bits.

`rcpcCodeRate` indica la proporción de código RCPC como 8/8, 8/9, ..., 8/32.

Reemplazada por una versión más reciente

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.

H233AL2MParameters: 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 alsduSplitting 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 (splitting) 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 Recomendación 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 V.76. suspendResumewAddress indica que el canal de suspensión/reanudación utilizará el campo de dirección definido en 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 en V.76.

uNERM indica que el canal lógico funcionará en el modo de ausencia de recuperación tras error definido en V.76.

Para la definición de n401, windowSize y loopbackTestProcedure, véase 12.2.1/V.42, y sus subcláusulas. Para los fines de 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 la CRC de 16 bits, y crc32bit indica que se utilizará la CRC de 32 bits definida en V.76.

recovery es un parámetro facultativo que indica los procedimientos de recuperación tras error definidos en 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

Reemplazada por una versión más reciente

selectivo y mSREJ indica que se utilizará el procedimiento de rechazo de trama selectivo múltiple definido en V.76.

uIH indica la utilización de tramas UIH de V.76.

rej indica la utilización del procedimiento de rechazo de V.76.

V75Parameters se utiliza para indicar parámetros específicos de la utilización de la V.75. audioHeaderPresent indica la presencia del encabezamiento de audio de la V.75.

H2250LogicalChannelParameters: se utiliza para indicar parámetros específicos de la utilización de la Recomendación 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 director 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 director 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 cuando el transporte es de unidifusión. Si la transportAddress es de multidifusión, el director 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 director 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.

Reemplazada por una versión más reciente

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 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.

`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` (fuente) se utiliza para identificar el número de terminal del emisor del mensaje `OpenLogicalChannel`.

`h235Key`: 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 Recomendación 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 encriptación 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 Recomendación 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 Recomendación T.124).

7.3.2 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.

Reemplazada por una versión más reciente

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 utiliza 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 de la Recomendación Q.2931, o de la 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 director cuando acuse la apertura de un canal por un subordinado. Se utiliza para proporcionar el valor de la clave de encriptación y el punto de sincronización en el que deberá utilizarse la clave. En el caso de la Recomendación 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 Recomendación H.225.0.

sessionID es un identificador de sesión RTP único en la conferencia, que sólo puede ser creado por el director. Este 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. Si la transportAddress es de multidifusión, el director 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 director 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 describirla dirección de transporte para el canal lógico. Las direcciones IPv4 e IPv6 se codificarán comenzando por el octeto más significativo de la dirección,

Reemplazada por una versión más reciente

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.

dynamicRTPPayloadType 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.

NOTA – Los parámetros H.223 no se han incluido en reverseLogicalChannelParameters, pues sus valores se especificaron en el mensaje de petición OpenLogicalChannel.

7.3.3 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 7.

Cuadro 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 H.223LogicalChannelParameters.
MulticastChannelNotAllowed	No pudo abrirse el canal de multidifusión.

Reemplazada por una versión más reciente

Cuadro 7/H.245 – Motivos para rechazar un OpenLogicalChannel (*fin*)

Punto de código ASN.1	Causa
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 director.
MasterSlaveConflict	El subordinado ha tratado de abrir un canal lógico en el que el director ha determinado que puede producirse un conflicto. (Véanse 8.4.1.3 y 8.5.1.3).
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.

7.3.4 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.

7.3.5 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 8.

Cuadro 8/H.245 – Fuentes de liberación del canal lógico

Punto de código ASN.1	Causa
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.

Reemplazada por una versión más reciente

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.

7.3.6 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.

7.3.7 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 (motivo) 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 (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 separan de una conferencia puntos extremos.

7.3.8 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.

7.3.9 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.

7.3.10 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.

7.4 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.

Reemplazada por una versión más reciente

7.4.1 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 `MultiplexEntryDescriptors`.

MultiplexEntryDescriptor: Describe una única entrada de tabla múltiplex. Comprende el `MultiplexTableEntryNumber` y una lista de `MultiplexElements`. 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 `MultiplexElements` 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 `MultiplexElements` 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 `MultiplexEntryDescriptors`, cada uno de los cuales describe una sola entrada de tabla múltiplex. Las entradas múltiplex pueden enviarse en cualquier orden.

7.4.2 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.

7.4.3 Rechazo de envío de entradas múltiplex

Se utiliza para rechazar uno o más `multiplexEntryDescriptors` 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 9 se indican los motivos de rechazo.

Reemplazada por una versión más reciente

Cuadro 9/H.245 – Motivos de rechazo de un MultiplexEntrySend

Punto de código ASN.1	Causa
unspecified	No se han especificado causas del rechazo.
DescriptorTooComplex	El MultiplexEntryDescriptor rebasaba la capacidad del terminal de recepción.

7.4.4 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 que se ha aplicado la temporización.

7.5 Mensajes de señalización de petición de tabla múltiplex

Este conjunto de mensajes se utiliza para la petición segura de retransmisión de uno o más MultiplexEntryDescriptors del transmisor al receptor.

7.5.1 Petición de entrada múltiplex

Se utiliza para solicitar la retransmisión de uno o más MultiplexEntryDescriptors.

entryNumbers es una lista de los MultiplexTableEntryNumbers de los MultiplexEntryDescriptors cuya retransmisión se ha pedido.

7.5.2 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 MultiplexTableEntryNumbers de los MultiplexEntryDescriptors que se transmitirán.

7.5.3 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 MultiplexTableEntryNumbers de los MultiplexEntryDescriptors 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 10.

Cuadro 10/H.245 – Motivos para el rechazo de un MultiplexEntrySend

Punto de código ASN.1	Causa
unspecified	No se especifica causa de rechazo.

7.5.4 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 MultiplexTableEntryNumbers de los MultiplexEntryDescriptors para los que ha expirado el periodo de temporización.

Reemplazada por una versión más reciente

7.6 Mensajes de petición de modo

Un terminal receptor utiliza este conjunto de mensajes para solicitar del terminal transmisor modos de transmisión determinados.

7.6.1 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 instancias de *RequestMode* de forma que pueda identificarse la respuesta correspondiente.

ModeDescription: Conjunto de uno o más *ModeElements*.

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 encriptados.

h223ModeParameters: Se emplea para expresar los parámetros propios de la utilización de H.223 [10].

adaptationLayerType expresa 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* (AL1M se define en el anexo C/H.223 con parámetros especificados), *al2M* (AL2M se define en el anexo C/H.223 con parámetros especificados), y *al3M* (AL3M 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 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*.

7.6.1.1 Modo vídeo

Se trata de una elección de *VideoModes*.

H261VideoMode: Expresa la resolución de imagen (QCIF o CIF), velocidad binaria, en unidades de 100 bit/s, y transmisión de imágenes fijas solicitadas.

H262VideoMode: Expresa 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 enteros números con las unidades definidas en el cuadro 2.

H263VideoMode: Expresa la resolución de imagen (SQCIF, QCIF, CIF, 4 veces CIF y 16 veces CIF o algún formato de imagen personalizado), velocidad binaria, en unidades de 100 bit/s, solicitadas.

Reemplazada por una versión más reciente

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 Recomendación 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 Recomendación 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 3.

7.6.1.2 Modo audio

Se trata de una elección de `AudioModes`.

En el cuadro 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 `silenceSuppression`, 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 G.723.1 y anexo C/G.723.1.

IS11172AudioMode: Se utiliza para solicitar audio codificado según la ISO/CEI 11172-3 [43].

`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` expresa 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` expresa la velocidad de bits de audio solicitada, medida en unidades de kbit/s.

IS13818AudioMode: Se utiliza para solicitar audio codificado según la ISO/CEI 13818-3 [44].

`audioLayer` expresa 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` expresa el modo multicanal solicitado según se especifica en el cuadro 11.

Reemplazada por una versión más reciente

Cuadro 11/H.245 – Códigos de punto multicanal de ISO/CEI 13818-3

Punto de código ASN.1	Significado semántico de punto de código
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 y 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 7 canales multilingües.

`bitRate` expresa la velocidad de bits de audio solicitada, medida en unidades de kbit/s.

7.6.1.3 Modo datos

Se trata de una elección de aplicaciones de datos y velocidades binarias.

`bitRate` expresa la velocidad binaria solicitada en unidades de 100 bit/s.

`t120` solicita la utilización del protocolo T.120 [31].

`dsm-cc` solicita la utilización del protocolo DSM-CC [45].

`userData` solicita la utilización de datos de usuario no especificados procedentes de puertos de datos externos.

`t84` solicita la utilización de T.84 [30] para la transferencia de esas imágenes (JPEG, JBIG, Facsímil Gr.3/4).

`t434` solicita el uso de T.434 [33] 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.

Reemplazada por una versión más reciente

h222DataPartitioning solicita la utilización modificada y restringida de las particiones de datos de H.262, como se especifica en 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 Recomendación T.39 para los modos DSVF/MSVF.

T140 solicita la utilización del protocolo de conversación de texto, especificado en la Recomendación T.140.

7.6.1.4 Modo encriptación

Se trata de una elección de modos de cifrado.

h233Encryption solicita el uso de encriptación de conformidad con las Recomendaciones H.233 [14] y H.234 [15].

7.6.2 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 12 se indican los posibles valores de respuesta.

Cuadro 12/H.245 – Respuestas de confirmación a petición de modo

Punto de código ASN.1	Respuesta
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.

7.6.3 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 13 se indican los valores de causa.

Cuadro 13/H.245 – Respuestas de rechazo a petición de modo

Punto de código ASN.1	Respuesta
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.

Reemplazada por una versión más reciente

7.6.4 Liberación de petición de modo

Se utiliza la MRSE de salida en caso de expiración de un periodo de temporización.

7.7 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.

7.7.1 Petición de retardo de ida y vuelta

Se envía desde la RTDSE de salida a la RTDSE de entrada.

Se utiliza sequenceNumber para etiquetar instancias de RoundTripDelayRequest de forma que pueda identificarse la respuesta correspondiente.

7.7.2 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.

7.8 Mensajes de bucle de mantenimiento

Un terminal utiliza este conjunto de mensajes para ejecutar funciones de bucle de mantenimiento.

7.8.1 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.

7.8.2 Acuse de recibo de bucle de mantenimiento

Se utiliza para confirmar que el terminal establecerá el bucle en la forma pedida.

7.8.3 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.

7.8.4 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.

7.9 Mensajes de modo de comunicación

Este conjunto de mensajes lo utiliza la H.323 MC para indicar el modo de comunicación de una conferencia H.323.

Reemplazada por una versión más reciente

7.9.1 Instrucción de modo de comunicación

La **CommunicationModeCommand** la envía el MC de la H.323 para especificar el modo de comunicación para cada tipo de medio, a saber, unificación o multifusión. 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 el 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 del MC. Los puntos extremos deben esperar a 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, el MC incluirá la **terminalLabel** de ese punto extremo.

El MC fija **sessionDependency** para indicar cuándo una sesión depende de otra sesión a efectos de una decodificación significativa de los datos.

El campo **destination** (destino) 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.

Reemplazada por una versión más reciente

7.9.2 Petición de modo de comunicación

La envía el MC para solicitar el modo de comunicación de la conferencia actual.

7.9.3 Respuesta a modo de comunicación

La envía el MC en respuesta a una CommunicationModeRequest, para especificar el modo de comunicación de una conferencia.

7.10 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 cabecera H.323, este campo será truncado a 32 octetos.

7.10.1 Petición de lista de terminales

Esta petición es equivalente a la TCU de la H.230, descrita en la Recomendación H.243.

7.10.2 Respuesta a lista de terminales

Esta respuesta es equivalente a una secuencia de terminalNumbers, descrita en la Recomendación H.230.

7.10.3 Concédame la presidencia

Esta petición es equivalente a CCA, descrita en la Recomendación H.230.

7.10.4 Anulación de concédame la presidencia

Esta petición es equivalente a CIS, descrita en la Recomendación H.230.

7.10.5 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.

7.10.6 Separación de terminal

Esta petición es equivalente a CCD, descrita en la Recomendación H.230.

7.10.7 Rechazo de separación de terminal

Esta respuesta es equivalente a CIR, descrita en la Recomendación H.230.

7.10.8 Petición de identificación de terminal

Esta petición es equivalente a TCP, descrita en la Recomendación H.230.

7.10.9 Respuesta a identificación de terminal MC

Esta respuesta es equivalente a TIP, descrita en la Recomendación H.230.

7.10.10 Petición de introducción de contraseña H.243

Esta petición es equivalente a TCS1, descrita en la Recomendación H.230.

7.10.11 Respuesta a contraseña

Esta respuesta es equivalente a IIS, descrita en la Recomendación H.230.

Reemplazada por una versión más reciente

7.10.12 Petición de introducción de identificación de terminal H.243

Esta petición es equivalente a TCS2/TCI, descrita en la Recomendación H.230.

7.10.13 Respuesta a identificación de terminal

Esta respuesta es equivalente a IIS, descrita en la Recomendación H.230.

7.10.14 Petición de introducción de identificación de conferencia H.243

Esta petición es equivalente a TCS3, descrita en la Recomendación H.230.

7.10.15 Respuesta a identificación de conferencia

Esta respuesta es equivalente a IIS, descrita en la Recomendación H.230.

7.10.16 Rechazo de instrucción de vídeo

Esta petición es equivalente a VCR descrita en la Recomendación H.230.

7.10.17 Petición de introducción de dirección de extensión

Esta petición es equivalente a TCS4, descrita en la Recomendación H.230.

7.10.18 Respuesta a dirección de extensión

Esta respuesta es equivalente a IIS, descrita en la Recomendación H.230.

7.10.19 Petición al propietario del testigo del control por la presidencia

Esta petición es equivalente a TCA, descrita en la Recomendación H.230 para el testigo de control por la presidencia.

7.10.20 Respuesta del propietario del testigo de la presidencia

Esta respuesta es equivalente a TIR, descrita en la Recomendación H.230 para el testigo de control por la presidencia.

7.10.21 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 encriptada con la clave privada.

CertSelectionCriteria define un conjunto de certificados que son aceptables para el peticionario. El respondedor (el 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), el MC respondedor deberá devolver un certificado asociado al punto extremo solicitado (contenido en la estructura certificateResponse).

Reemplazada por una versión más reciente

- 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 firmas 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.

7.10.22 Respuesta de certificado de terminal

Esta respuesta devuelve el certificado digital y una `responseString` está encriptada con la clave privada de un terminal específico.

7.10.23 Difunde mi canal lógico

Esta petición es similar a MCV de H.230, pero se refiere a un solo canal lógico y tiene el mensaje de respuesta `broadcastMyLogicalChannelResponse` que acusa recibo de la petición.

7.10.24 Respuesta a difunde mi canal lógico

Da una respuesta de concedido o denegado a la petición `BroadcastMyLogicalChannel` (difundir mi canal lógico).

7.10.25 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.

7.10.26 Respuesta a haz al terminal difundir

Da una respuesta de concedido o denegado a la petición `MakeTerminalBroadcaster` (haz al terminal difundir).

7.10.27 Envía esta fuente

Esta petición es similar a VCS de H.230 y tiene como mensaje de respuesta `SendThisSourceResponse` que acusa de recibo de la petición.

7.10.28 Respuesta a envía esta fuente

Da una respuesta de concedido o denegado a la petición `SendThisSourceResponse` (envía esta fuente).

7.10.29 Petición de las identificaciones de todos los terminales

Enviado por un punto extremo al MC de una conferencia para obtener las etiquetas y las identificaciones de todos los terminales de los participantes en la conferencia.

7.10.30 Respuesta a petición de identificaciones de todos los terminales

Respuesta a `RequestAllTerminalIDs` (petición de las identificaciones de todos los terminales) que contiene una lista de todos los puntos extremos de la conferencia por `terminalLabel` y `terminalID`.

7.10.31 Petición a MC distante

Esta petición es enviada por un MC activo a otro MC para activarlo/desactivarlo. La petición de MC distante con elección de `masterActivate` o `slaveActivate` puede ser enviada por un MC activo a un

Reemplazada por una versión más reciente

MC inactivo para activarlo como director o subordinado, respectivamente, de una conexión en cascada. La petición de MC distante con elección de `deActivate` puede ser enviada por un MC director a un MC subordinado ya activo para desactivarlo.

7.10.32 Respuesta de MC distante

La respuesta de MC distante se envía para indicar aceptación o rechazo de la petición a 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) de H.225 o el receptor de la petición inicia esta llamada con un `conferenceGoal` de `JOIN` (incorporación) en el mensaje `Setup` (establecimiento) de 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) de H.225.

Elección = `deActivate`

El receptor es un MC activo.

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 admiten la configuración en cascada.

7.11 Instrucciones

Un mensaje de instrucción requiere una acción pero no una respuesta explícita.

7.11.1 Envío del conjunto de capacidades de terminal

Esta instrucción ordena al terminal del extremo distante que indique sus capacidades de transmisión y recepción enviando uno o más `TerminalCapabilitySets` 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 `CapabilityTableEntries` que el terminal solicita que se transmitan.

`capabilityDescriptorNumbers` es un conjunto de los `CapabilityDescriptorNumbers` que indican los `CapabilityDescriptors` cuya transmisión solicita el terminal.

`genericRequest` ordena al terminal del extremo distante que envíe su conjunto completo de capacidades de terminal.

Reemplazada por una versión más reciente

7.11.2 Encriptación

Se utiliza esta instrucción para intercambiar capacidades de encriptación y ordenar la transmisión de un vector de inicialización (IV); véanse las Recomendaciones H.233 y H.234 [14], [15].

encryptionSE es un mensaje de intercambio de sesión (SE) H.233 con la salvedad de que no se aplicarán los bits de protección contra errores descritos en la Recomendación H.233.

encryptionIVRequest ordena al encriptador 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 encriptación no normalizado associatedAlgorithm.

7.11.3 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 virtualChannelID 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.

7.11.4 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 14 se indican las posibles opciones.

Reemplazada por una versión más reciente

Cuadro 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.8bis.
v34v76	El terminal conservará la conexión del módem V.34, pero la utilizará para sustentar V.DSVD.
v34DuplexFAX	El terminal conservará la conexión del módem V.34, pero la utilizará para sustentar T.30 FAX [27].
v34H324	El terminal conservará la conexión del módem V.34, pero la utilizará para sustentar 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 15 se indican las posibles opciones.

Cuadro 15/H.245 – Opciones después de la 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.
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.

7.11.5 Instrucción diversos

Se usa para diversas instrucciones, algunas de las cuales se indican en las Recomendaciones 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, y videoSendSyncEveryGOBCancel. 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.

Reemplazada por una versión más reciente

`equaliseDelay` y `zeroDelay` tendrán el mismo significado que en las instrucciones ACE y ACZ definidas en la Recomendación 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.

`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 Recomendación 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 Recomendación 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 y se refiere únicamente a H.263, `firstMB` indica el número del primer MB que habrá de actualizarse y se refiere únicamente a H.261, y `numberOfMBs` indica el número de MBs que habrá que actualizar. Sólo se utilizará con algoritmos de compresión vídeo que definen MBs, por ejemplo, H.261 y H.263. Los terminales pueden responder a esta instrucción con una actualización de GOB que incluya los MB solicitados.

`maxH223MUXPDUsize` ordena al transmisor que limite el tamaño de las MUX-PDU de la 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 encriptación 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.

Reemplazada por una versión más reciente

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.

Reemplazada por una versión más reciente

7.11.6 Instrucción de conferencia

BroadcastMyLogicalChannel será similar a MCV de la H.230, pero se referirá solamente a un canal lógico único.

CancelBroadcastMyLogicalChannel será similar a Cancel-MCV de la H.230 pero se referirá a un canal lógico único.

MakeTerminalBroadcaster se definirá como VCB de la H.230.

CancelMakeTerminalBroadcaster se definirá como Cancel-VCB.

SendThisSource se definirá como VCS de la H.230.

CancelSendThisSource se definirá como Cancel-VCS de la H.230.

DropConference se definirá como CCK de la H.230.

La instrucción sustitución de CID permite a un MC activo cambiar el identificador de conferencia (CID), 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.

7.11.7 Reconfiguración de multiplex H.223

h223ModeChange ordena al transmisor que cambie el nivel del modo multiplex, 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.

7.12 Indicaciones

Una indicación contiene informaciones que no requieren acción ni respuesta.

7.12.1 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 la Recomendación H.245. 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.

7.12.2 Indicación diversos

Se utiliza para una diversidad de indicaciones, algunas de las cuales están presentes en las Recomendaciones H.221 [4] 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 TemporalSpatialTradeOff. 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.

Reemplazada por una versión más reciente

Se utiliza `logicalChannelInactive` para expresar que el contenido del canal lógico no representa una señal normal. Es análogo a AIM y VIS definidas en la Recomendación H.230.

`logicalChannelActive` es complementario de `logicalChannelInactive`. Es análogo a AIA y VIA definidas en la Recomendación H.230. `MultipointZeroComm`, `cancelMultipointZeroComm`, `multipointSecondaryStatus`, y `cancel multipointSecondaryStatus` tendrán el mismo significado que MIZ, `cancelMIZ`, MIS y `cancelMIS` respectivamente, definidos en la Recomendación 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 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 Recomendación 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 expresa una alta resolución espacial y el valor 31 indica una alta velocidad de trama. Los valores comprendidos entre 0 y 31 expresan, 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 acuerdo con H.263. 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.

7.12.3 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 (*underflow*) o por encima [desbordamiento (*overflow*)] 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.

Reemplazada por una versión más reciente

`estimatedReceivedJitterMantissa` expresa la mantisa de la estimación de fluctuación como se indica en el cuadro 16.

Cuadro 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 17.

Cuadro 17/H.245 – Exponente de `estimatedReceivedJitterExponent` en `JitterIndication`

<code>estimatedReceivedJitterExponent</code>	Exponente
0	fuera de gama
1	1 μ s
2	10 μ s
3	100 μ 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` H.262 [19].

7.12.4 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.

Reemplazada por una versión más reciente

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.

7.12.5 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 (o velocidad de bits), 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 I.363 [25], están soportadas. Los puntos de código se definen en el cuadro 1.

La secuencia aal5 indica cuáles de las opciones para la capa de adaptación ATM 5, entre las especificadas en 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 de la H.222.0), tren de transporte H.222.0 y tren de programa H.222.0.

7.12.6 Entrada de usuario

Se utiliza para los mensajes de entrada de usuario.

alphanumeric es una cadena de caracteres codificada según la Recomendación T.51 [29]. 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.

nonStandard es un NonStandardParameter que indica una utilización no normalizada del mensaje de indicación userInput.

Reemplazada por una versión más reciente

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.

NOTA 2 – todos los datos contenidos en esta Recomendación, incluidos los mensajes de entrada de usuario, no estarán encriptados.

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 cabecera de línea de red telefónica pública conmutada (RTPC), que se detectó en el tren de audio cuando se envió desde una cabecera de línea de RTPC, o que se ha de señalar entre otras combinaciones de puntos extremos. Cuando es recibida por una cabecera a la RTPC, **signal** hace que la cabecera inyecte el elemento de señalización especificado en el canal RTPC; cuando sea recibida por una cabecera de línea a otro terminal de la serie H, **signal** será traducida al mensaje apropiado del protocolo del terminal conectado. Las cabeceras 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 3 – 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 cabecera 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 `LogicalChannelNumbers` 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

Reemplazada por una versión más reciente

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**.

En una utilización típica, la cabecera 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 cabecera, el elemento **signal** contendrá la duración total del tono que ha de ser producido por la cabecera. 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 cabeceras 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,

Reemplazada por una versión más reciente

y utilizando **signalUpdate** para enviar estimaciones actualizadas en tanto en cuanto la entrada continúe estando activada y para indicar la duración total cuando se desactive la entrada.

7.12.7 Indicaciones de conferencia

sbeNumber se definirá como número SBE de la H.230.

terminalNumberAssign se definirá como TIA de la H.230.

terminalJoinedConference se definirá como TIN de la H.230.

terminalLeftConference se definirá como TID de la H.230.

seenByAtLeastOneOther se definirá como MIV de la H.230.

cancelSeenByAtLeastOneOther se definirá como cancel-MIV de la H.230.

seenByAll se definirá como MIV de la H.230.

cancelSeenByAll se definirá como MIV de la H.230.

terminalYouAreSeeing se definirá como VIN de la H.230.

requestForFloor se definirá como TIF de la H.230.

requestForFLoor se definirá como TIF de la H.230 y se enviará desde un terminal al MC.

WithdrawChairToken se definirá como CCR de H.230 y se enviará al poseedor del testigo para la presidencia.

FloorRequested se definirá como TIF de la H.230 cuando se envíe del MC al poseedor del testigo para la presidencia. Esta petición incluye la TerminalLabel del terminal solicitante.

7.12.8 Máxima asimetría de canal lógico H2250

H2250MaximumSkewIndication indica la asimetría máxima entre canales lógicos.

La asimetría (skew) 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 (timestamps).

7.12.9 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.

7.12.10 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.

7.12.11 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.

Reemplazada por una versión más reciente

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.

8 Procedimientos

8.1 Introducción

En esta cláusula se definen procedimientos genéricos de control de sistemas multimedios 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 esta cláusula se describen procedimientos para la ejecución de las siguientes funciones:

- determinación director-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;

8.1.1 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.

8.1.2 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.

Reemplazada por una versión más reciente

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.

8.1.3 Comunicación entre entidades pares

La información de protocolo se transfiere a la entidad de protocolo par mediante mensajes apropiados definidos en la cláusula 6. 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.

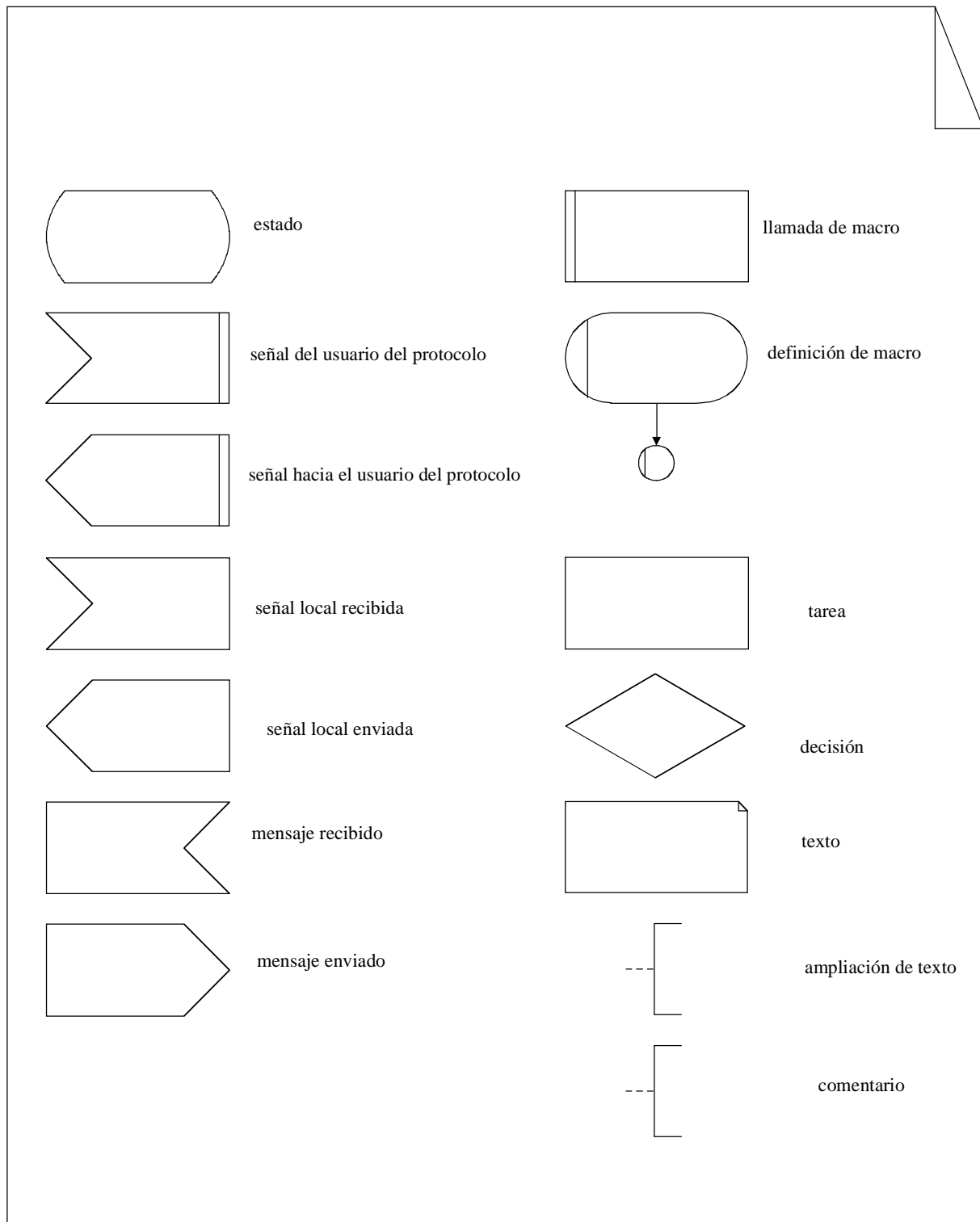
8.1.4 Diagramas SDL

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.

8.1.5 Símbolos gráficos utilizados en los diagramas SDL

En la figura 1 se muestran los símbolos gráficos utilizados en los diagramas SDL.

Reemplazada por una versión más reciente



T1519120-95

Figura 1/H.245 – Símbolos gráficos utilizados en los diagramas SDL

8.2 Procedimientos de determinación director-subordinado

8.2.1 Introducción

Pueden presentarse situaciones de conflicto cuando 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

Reemplazada por una versión más reciente

terminal puede actuar como el director 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 director y cuál es, o cuáles son los subordinados.

El protocolo descrito se designa por entidad de señalización de determinación de director-subordinado (MSDSE, *master slave determination signalling entity*). Hay una instancia de la MSDSE en cada terminal que participa en una llamada.

Cualquiera de los terminales puede iniciar el procedimiento de la determinación del director 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 director-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 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 de la instancia 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.

8.2.1.1 Visión general del protocolo – Iniciación por el usuario local

Se inicia un procedimiento director-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 director-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 director-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.

8.2.1.2 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 director-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

Reemplazada por una versión más reciente

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 director-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.

8.2.1.3 Visión general del protocolo – Iniciación simultánea

Cuando se recibe un mensaje MasterSlaveDetermination en la MSDSE 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.

8.2.1.4 Procedimiento de determinación de categoría

Se utiliza el siguiente procedimiento para determinar qué terminal es el director a partir de los valores de terminalType y statusDeterminationNumber. En primer lugar, se comparan los valores de terminalType y se determina como director el terminal con el número de tipo de terminal más grande. Si los números de tipo de terminal son los mismos, se comparan los statusDeterminationNumbers utilizando aritmética de módulo para determinar cuál es el director.

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.

8.2.2 Comunicación entre la MSDSE y el usuario de MSDSE

8.2.2.1 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 18.

Cuadro 18/H.245 – Primitivas y parámetros

nombre genérico	tipo			
	petición	indicación	respuesta	confirmación
DETERMINACIÓN	– (nota 1)	TYPE	no definida (nota 2)	TYPE
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 definida" indica que esta primitiva no está definida.				

Reemplazada por una versión más reciente

8.2.2.2 Definiciones de las primitivas

La definiciones de estas primitivas son:

- a) La primitiva DETERMINACIÓN se utiliza para iniciar el procedimiento de determinación de director-subordinado y para devolver el resultado de dicho procedimiento.

La primitiva DETERMINACIÓN.petición se utiliza para iniciar el procedimiento de determinación de director-subordinado.

La primitiva DETERMINACIÓN.indicación se utiliza para indicar el resultado del procedimiento de determinación de director-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 director-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 director-subordinado ha fracasado.
- c) La primitiva ERROR informa errores de la MSDSE a una entidad de gestión.

8.2.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas se indican en el cuadro 18 y son las siguientes:

- a) El parámetro TYPE indica la categoría (status) del terminal. Toma el valor de "MASTER" o "SLAVE".
- b) El valor ERRCODE indica el tipo del error de la MSDSE. El cuadro 22 indica los valores que puede tomar el parámetro ERRCODE.

8.2.2.4 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 director-subordinado.

Estado 1: ESPERA DE RESPUESTA EN SALIDA

El usuario de la MSDSE local ha pedido un procedimiento de determinación director-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 director-subordinado en la MSDSE local. Se envió un acuse de recibo a la MSDSE distante y se espera una respuesta de la MSDSE distante.

8.2.2.5 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 2.

Reemplazada por una versión más reciente

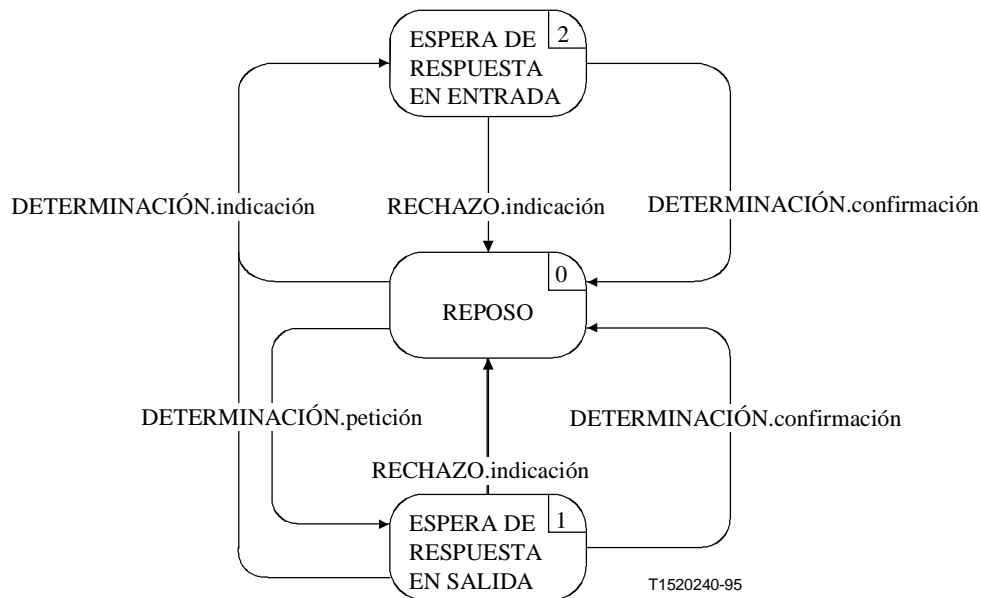


Figura 2/H.245 – Diagrama de transición de estados para la secuencia de las primitivas en la MSDSE

8.2.3 Comunicación entre MSDSE pares

8.2.3.1 Mensajes de MSDSE

En el cuadro 19 se indican los mensajes y campos de la MSDSE, definidos en la cláusula 6, que son apropiados para el protocolo de la MSDSE.

Cuadro 19/H.245 – Nombres y campos de los mensajes de la MSDSE

función	mensaje	campo
determinación	MasterSlaveDetermination	terminalType statusDeterminationNumber
	MasterSlaveDeterminationAck	decisión
	MasterSlaveDeterminationReject	causa
recuperación tras error	MasterSlaveDeterminationRelease	–

8.2.3.2 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 director-subordinado. Puede tomar los valores de "master", "slave", y "indeterminate".

Reemplazada por una versión más reciente

– 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.

8.2.3.3 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.

8.2.3.4 Contadores de la MSDSE

Se especifica el siguiente parámetro para la MSDSE:

– N100

Este parámetro especifica el valor máximo de sv_NCOUNT.

8.2.4 Procedimientos de la MSDSE

8.2.4.1 Introducción

En la figura 3 se resumen las primitivas de la MSDSE y sus parámetros, y los mensajes.

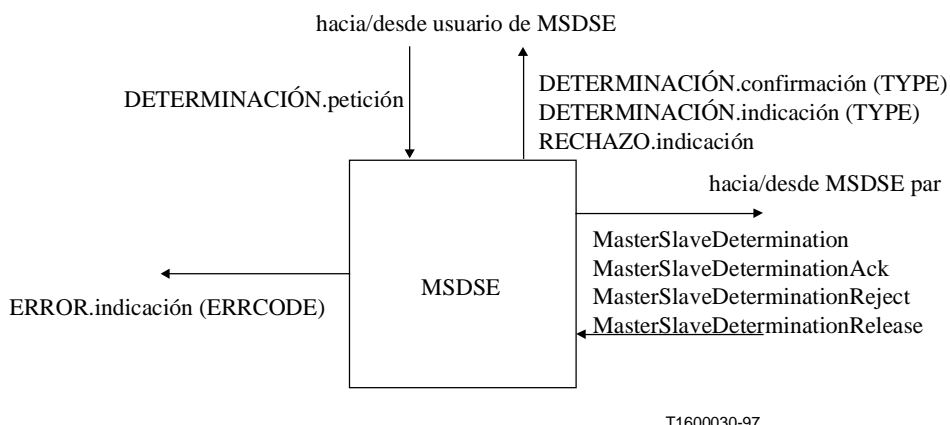


Figura 3/H.245 –Primitivas y mensajes en la MSDSE

8.2.4.2 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 20.

Cuadro 20/H.245 – Valores supletorios de los parámetros de las primitivas

primitiva	parámetro	valor supletorio
DETERMINACIÓN.confirmación	TIPO	MasterSlaveDeterminationAck.decision
DETERMINACIÓN.indicación	TIPO	sv_STATUS

Reemplazada por una versión más reciente

8.2.4.3 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 21.

Cuadro 21/H.245 – Valores supletorios de los campos de mensajes

mensaje	campo	valor supletorio
MasterSlave Determination	terminalType statusDetermination Number	sv_TT sv_SDNUM
MasterSlave DeterminationAck	decision	Opposite of sv_STATUS i.e. if (sv_STATUS == master) decision = slave if (sv_STATUS == slave) decision = master
MasterSlave DeterminationReject	cause	identicalNumbers

8.2.4.4 Valores del parámetro ERRCODE

El cuadro 22 muestra los valores que el parámetro ERRCODE de la primitiva ERROR.indicación puede tomar para la MSDSE.

Cuadro 22/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

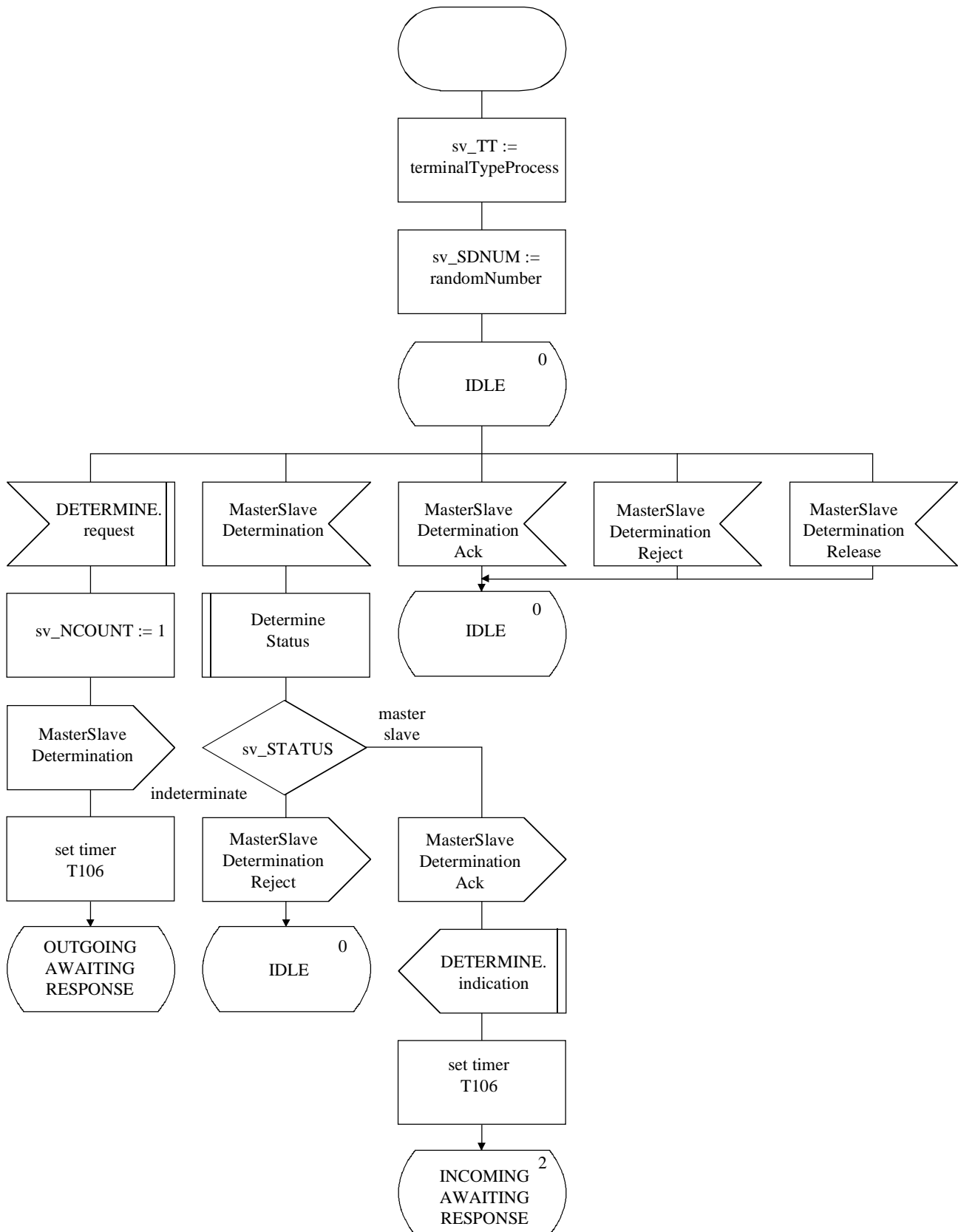
8.2.4.5 Diagramas SDL

Los procedimientos de la MSDSE se expresan en forma de diagramas SDL en la figura 4.

terminalTypeProcess es un proceso que devuelve un número que identifica los diferentes tipos de terminal, tales como terminales, MCUs y cabeceras.

randomNumber es un proceso que devuelve un número aleatorio comprendido en la gama $0..2^{24}-1$.

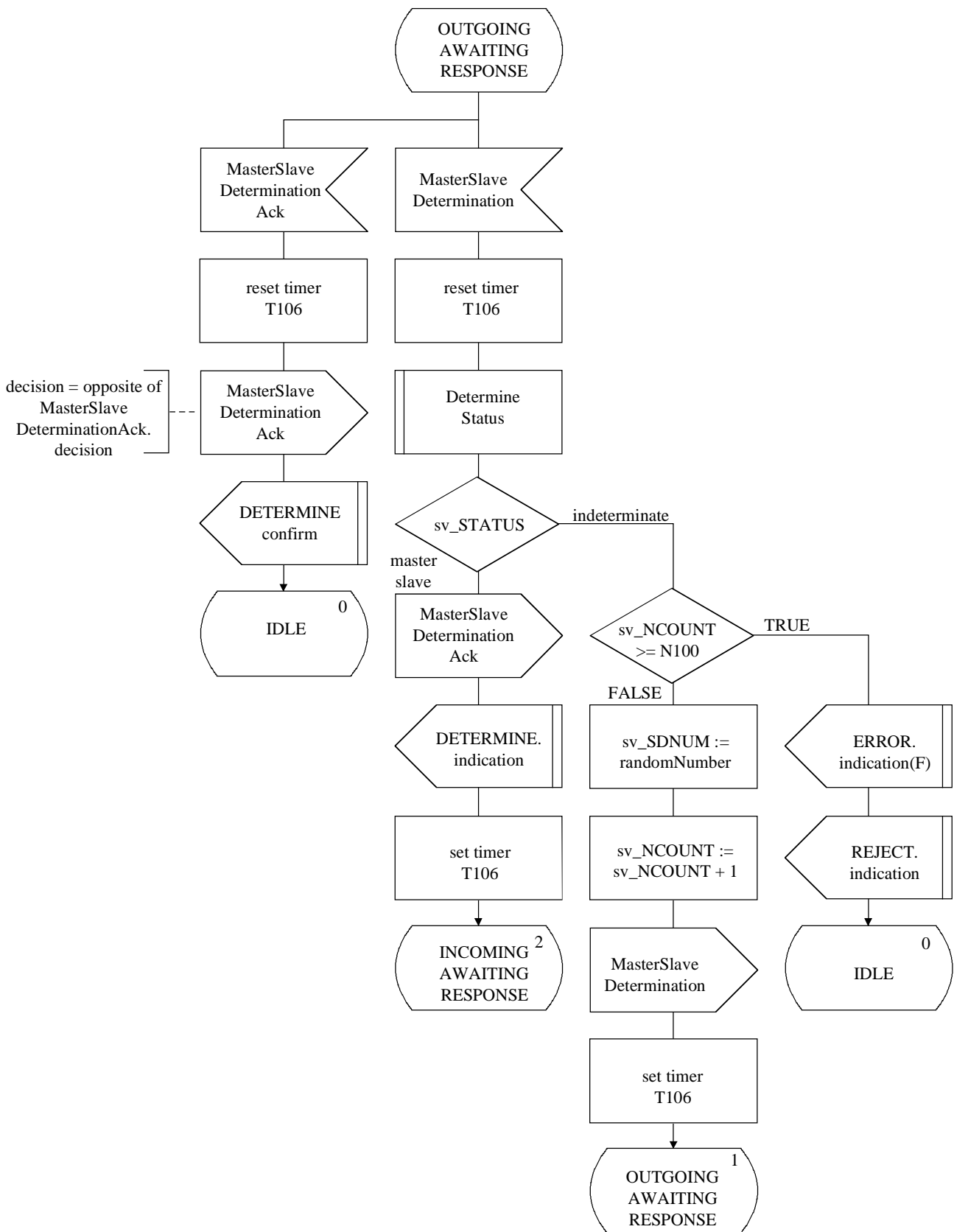
Reemplazada por una versión más reciente



T1600040-97

Figura 4.i/H.245 – Diagrama SDL de la MSDSE

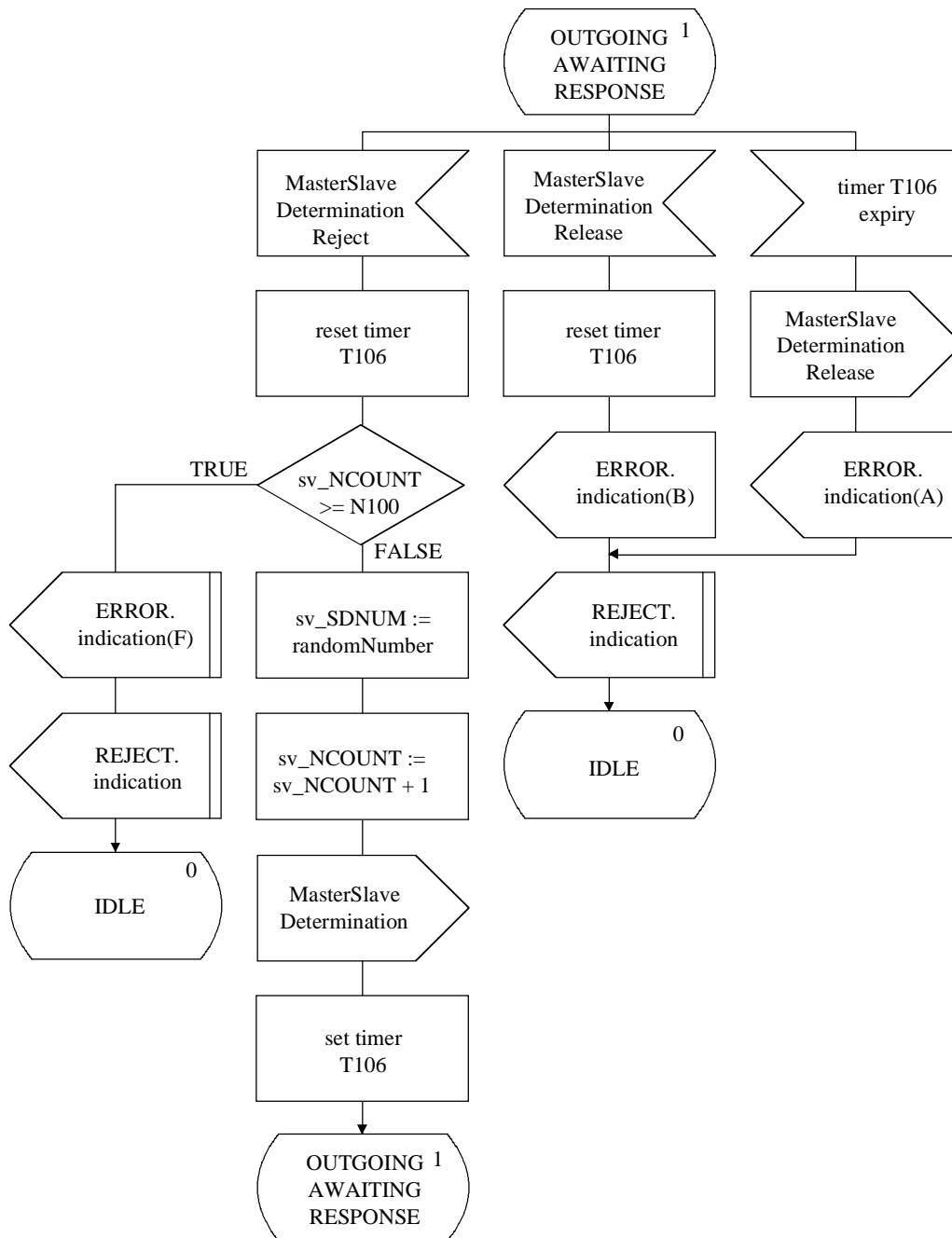
Reemplazada por una versión más reciente



T1520260-95

Figura 4.ii/H.245 – Diagrama SDL de la MSDSE (continuación)

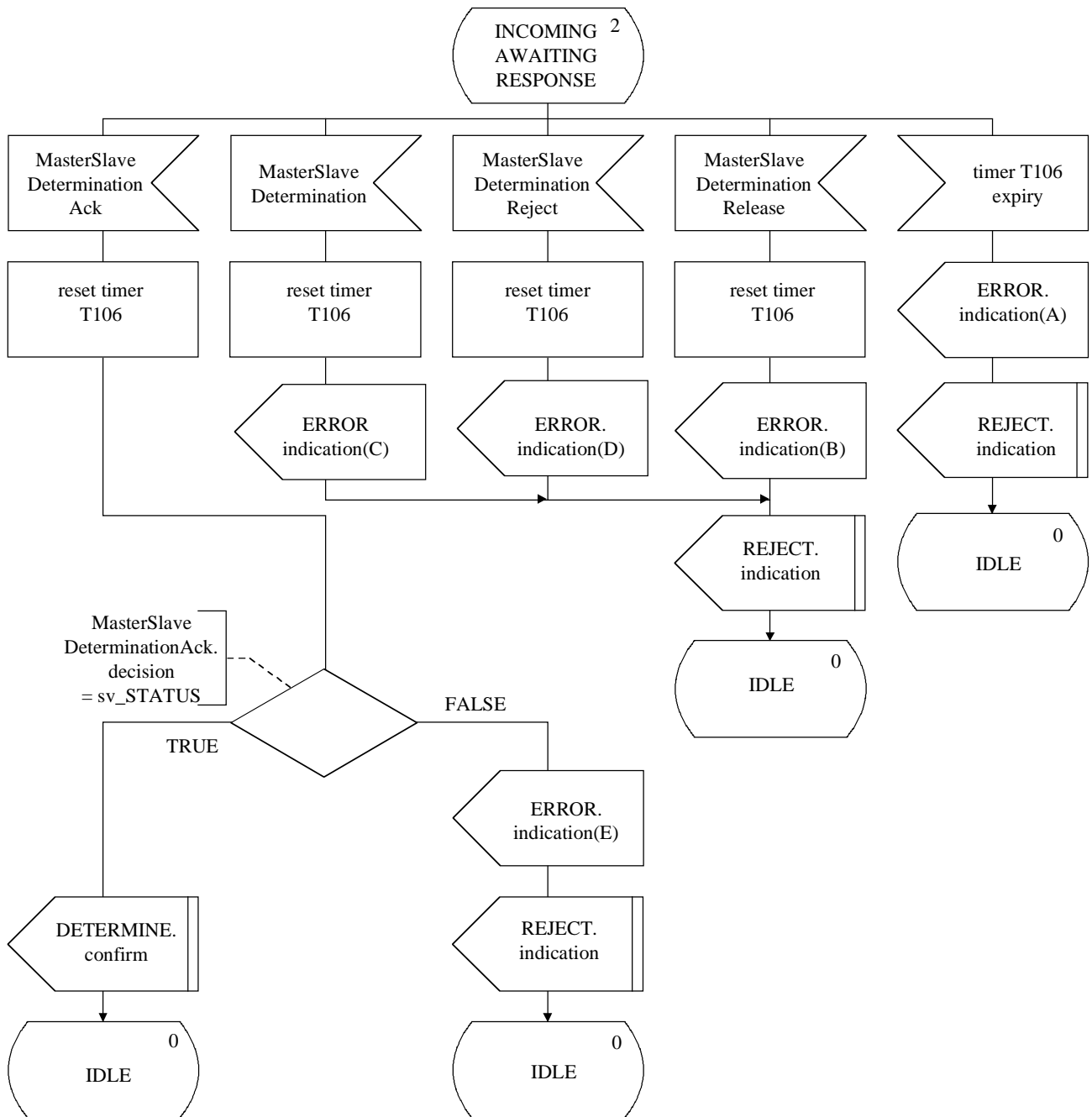
Reemplazada por una versión más reciente



T1520270-95

Figura 4.iii/H.245 – Diagrama SDL de la MSDSE (continuación)

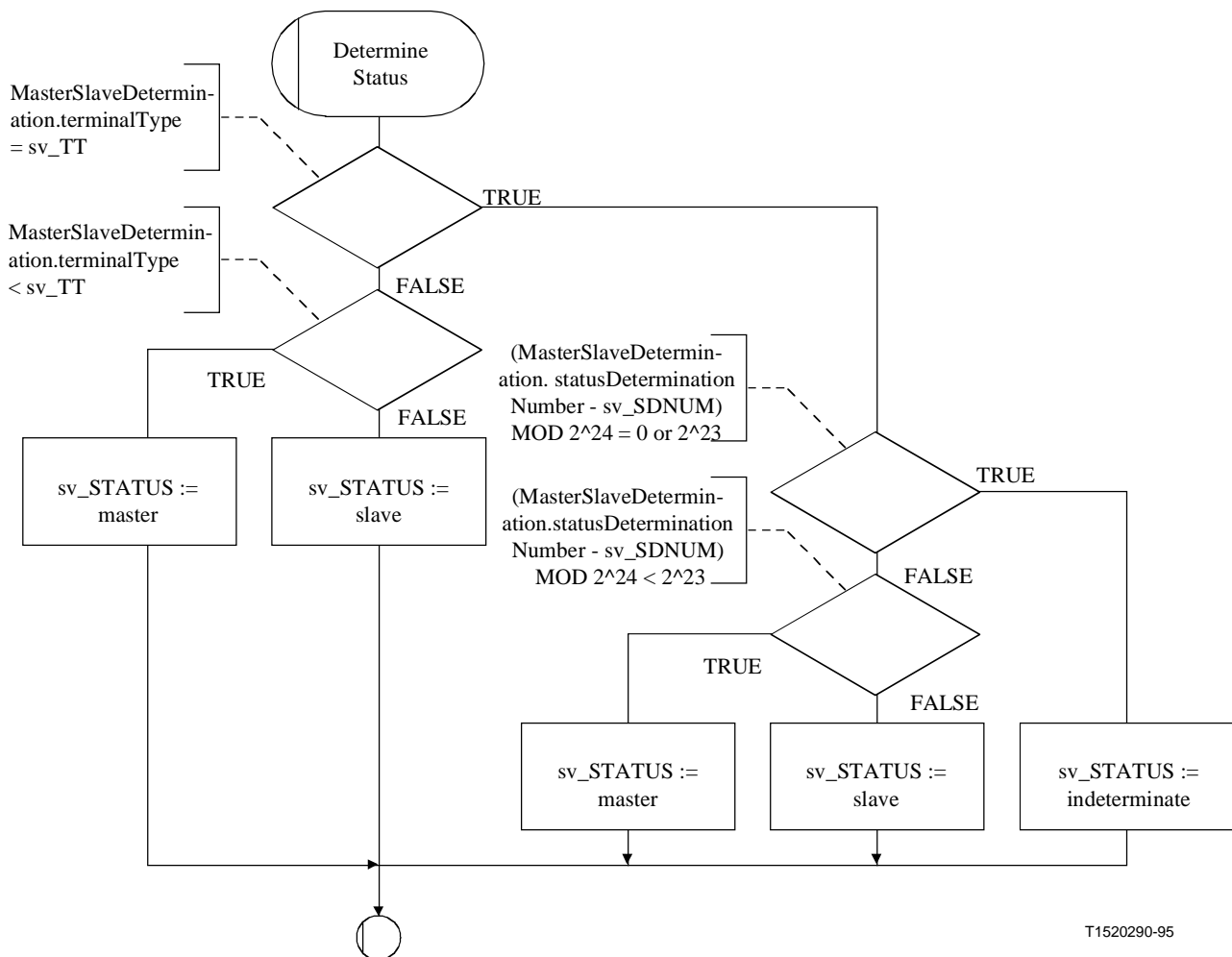
Reemplazada por una versión más reciente



T1520280-95

Figura 4.iv/H.245 – Diagrama SDL de la MSDSE (continuación)

Reemplazada por una versión más reciente



T1520290-95

Figura 4.v/H.245 – Diagrama SDL de la MSDSE (fin)

8.3 Procedimientos de intercambio de capacidad

8.3.1 Introducción

Para comunicarse sus capacidades, los terminales utilizan estos procedimientos que constituyen la entidad de señalización de intercambio de capacidad (CESE). 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 conducto de los mensajes apropiados definidos en la cláusula 6. En cada extremo, de entrada y de salida, hay una instancia 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.

Reemplazada por una versión más reciente

8.3.1.1 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 de 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.

8.3.1.2 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.

8.3.2 Comunicación entre la CESE y el usuario de CESE

8.3.2.1 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 23.

Cuadro 23/H.245 – Primitivas y parámetros

nombre genérico	tipo			
	petición	indicación	respuesta	confirmación
TRANSFERENCIA (TRANSFER)	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	– (nota 1)	–
RECHAZO (REJECT)	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.				

8.3.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- 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.

Reemplazada por una versión más reciente

8.3.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 23 son:

- 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".

8.3.2.4 Estados de CESE

Para especificar la secuencia admisible 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.

8.3.2.5 Diagrama de transición de estados

A continuación se define la secuencia admisible 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 admisibles se especifican de forma separada para cada CESE de salida y cada CESE de entrada, como se representa en las figuras 5 y 6, respectivamente.

Reemplazada por una versión más reciente

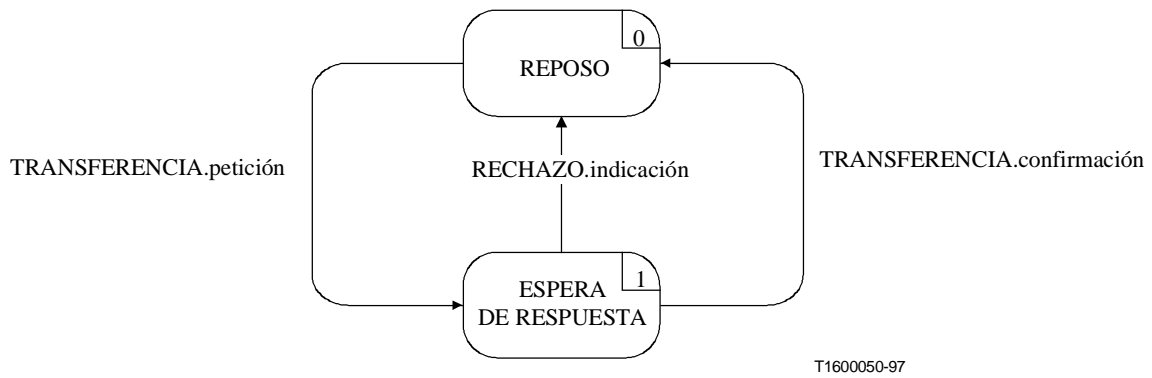


Figura 5/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CESE de salida

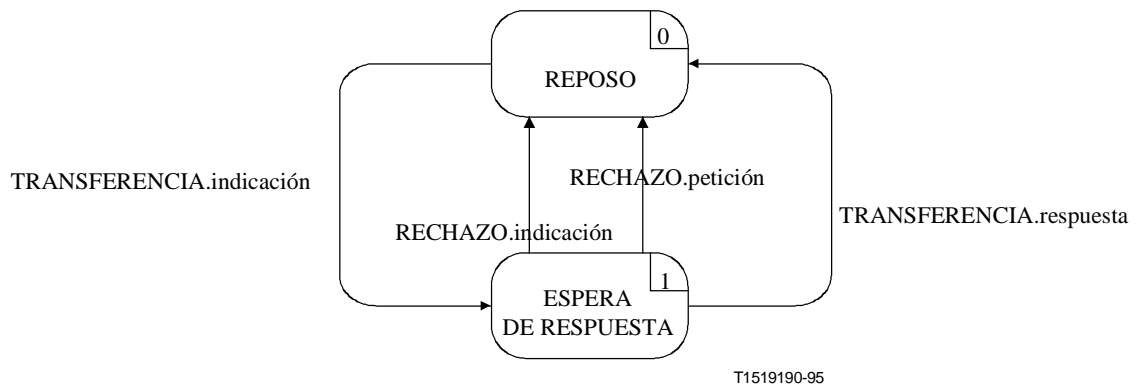


Figura 6/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CESE de entrada

8.3.3 Comunicación entre CESE pares

8.3.3.1 Mensajes

En el cuadro 24 se muestran los mensajes y campos de la CESE, definidos en la cláusula 6, apropiados para el protocolo de la CESE.

Reemplazada por una versión más reciente

Cuadro 24/H.245 – Nombres de mensajes y campos de la CESE

función	mensaje	sentido	campo
transfer	TerminalCapabilitySet	o -> i (nota)	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors
	TerminalCapabilitySetAck	O <- I	sequenceNumber
reject	TerminalCapabilitySetReject	O <- I	sequenceNumber cause
reset	TerminalCapabilitySetRelease	O -> I	–

NOTA – Sentido O de salida, I de entrada.

8.3.3.2 Variables de estado de la CESE

En la CESE de salida se definen las siguientes variables 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 1 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 definen las siguientes variables 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.

8.3.3.3 Temporizadores de la CESE

Para la CESE de salida, se especifica el siguiente temporizador:

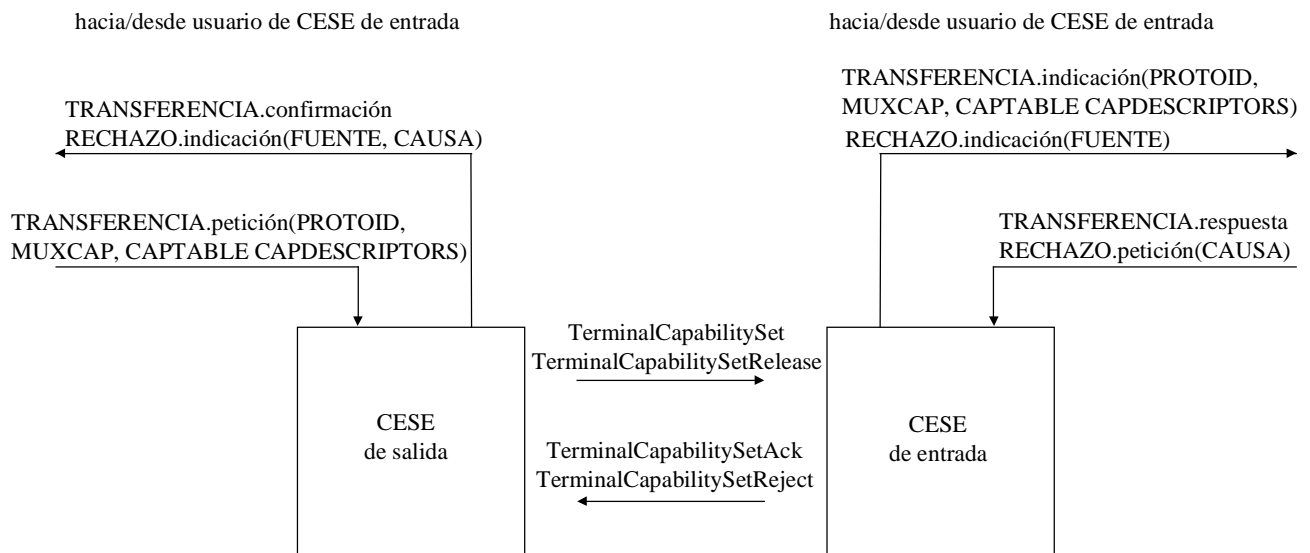
T.101

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.

8.3.4 Procedimientos de CESE

En la figura 7 se resumen las primitivas de la CESE, sus parámetros y mensajes para las CESE de salida y de entrada.

Reemplazada por una versión más reciente



T1519200-95

Figura 7/H.245 – Primitivas y mensajes en la entidad de señalización de intercambio de capacidad

8.3.4.1 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 25.

Cuadro 25/H.245 – Valores supletorios de los parámetros de las primitivas

primitiva	parámetro	valor supletorio
TRANSFERENCIA.indicación	PROTOID	TerminalCapabilitySet.protocolIdentifier
	MUXCAP	TerminalCapabilitySet.multiplexCapability
	CAPTABLE	TerminalCapabilitySet.capabilityTable
	CAPDESCRIPTORS	TerminalCapabilitySet.capabilityDescriptors
RECHAZO.indicación	SOURCE	USER
	CAUSE	null

8.3.4.2 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 26.

Reemplazada por una versión más reciente

Cuadro 26/H.245 –Valores supletorios de los campos de mensajes

mensaje	campo	valor supletorio (nota)
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.

8.3.4.3 Diagramas de SDL

Los procedimientos en la CESE de salida y en la CESE de entrada se expresan mediante diagramas SDL en las figuras 8 y 9, respectivamente.

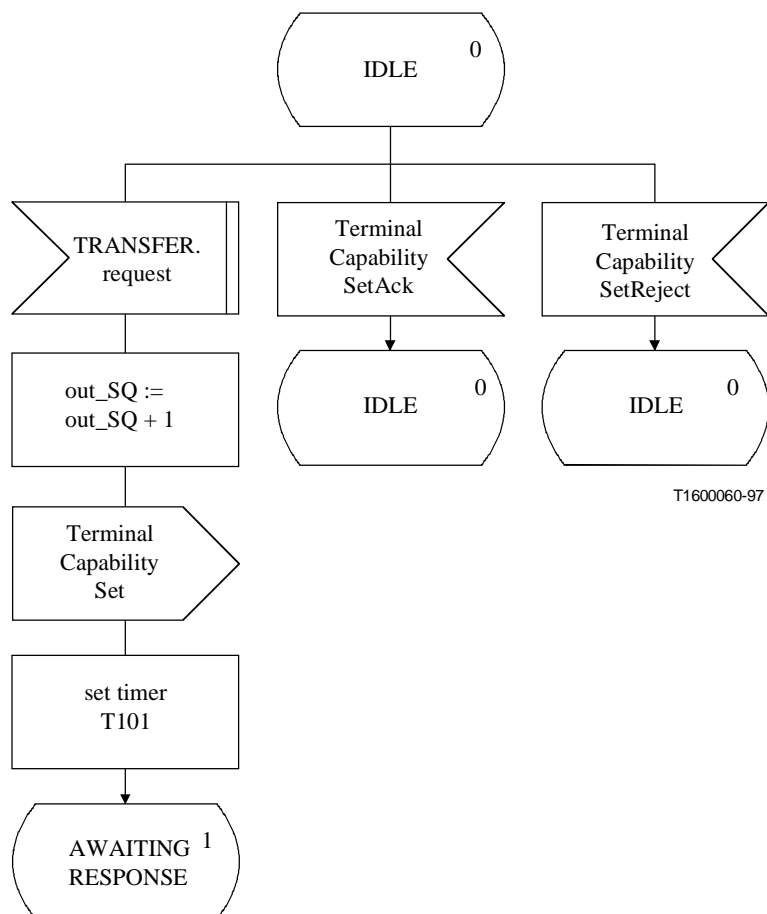
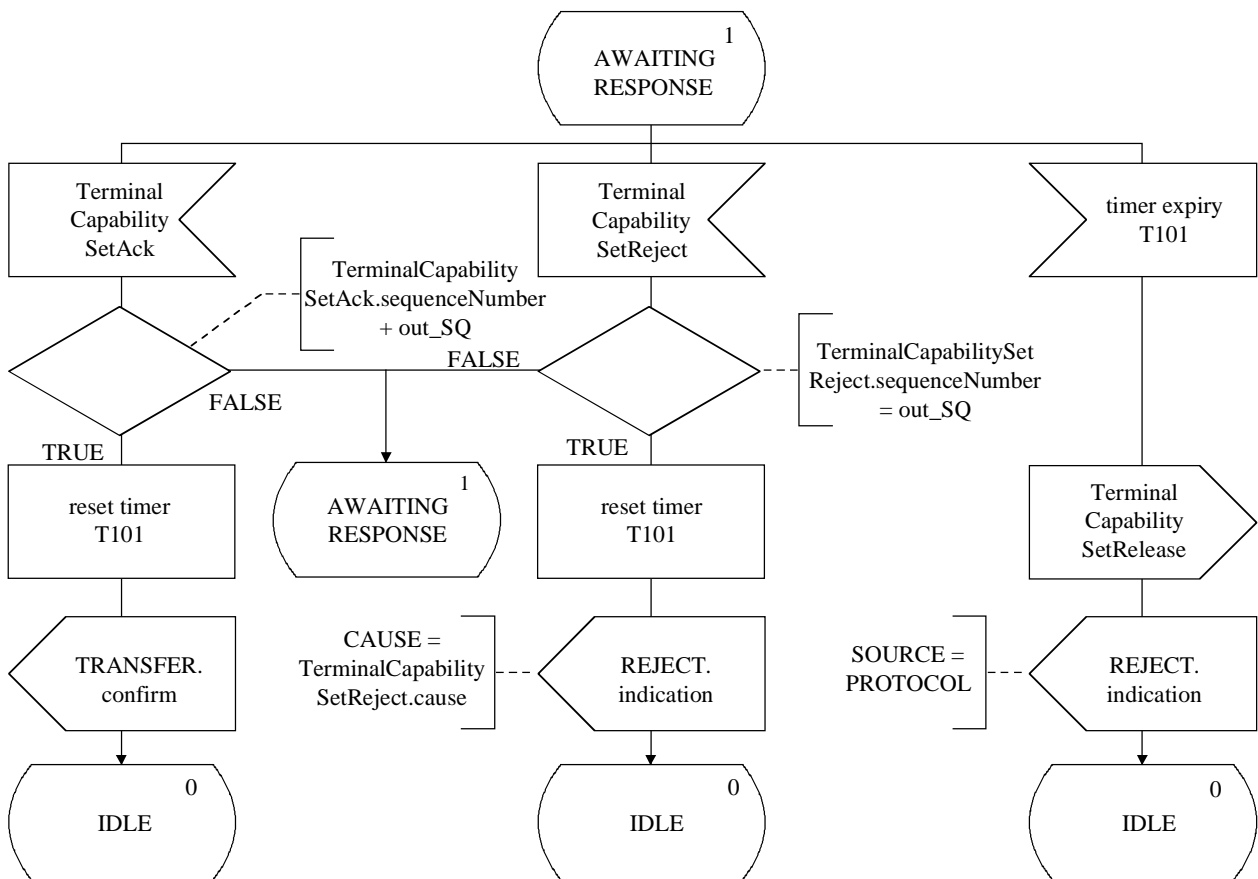


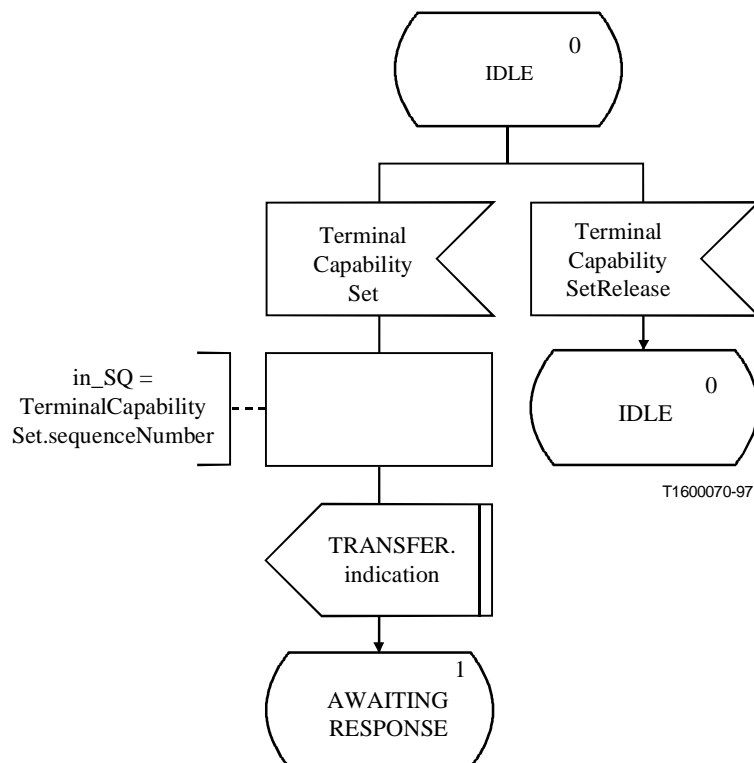
Figura 8.i/H.245 – Diagrama SDL de la CESE de salida

Reemplazada por una versión más reciente



T1520300-95

Figura 8.ii/H.245 – Diagrama SDL de la CESE de salida (*fin*)



T1600070-97

Figura 9.i/H.245 – Diagrama SDL de la CESE de entrada

Reemplazada por una versión más reciente

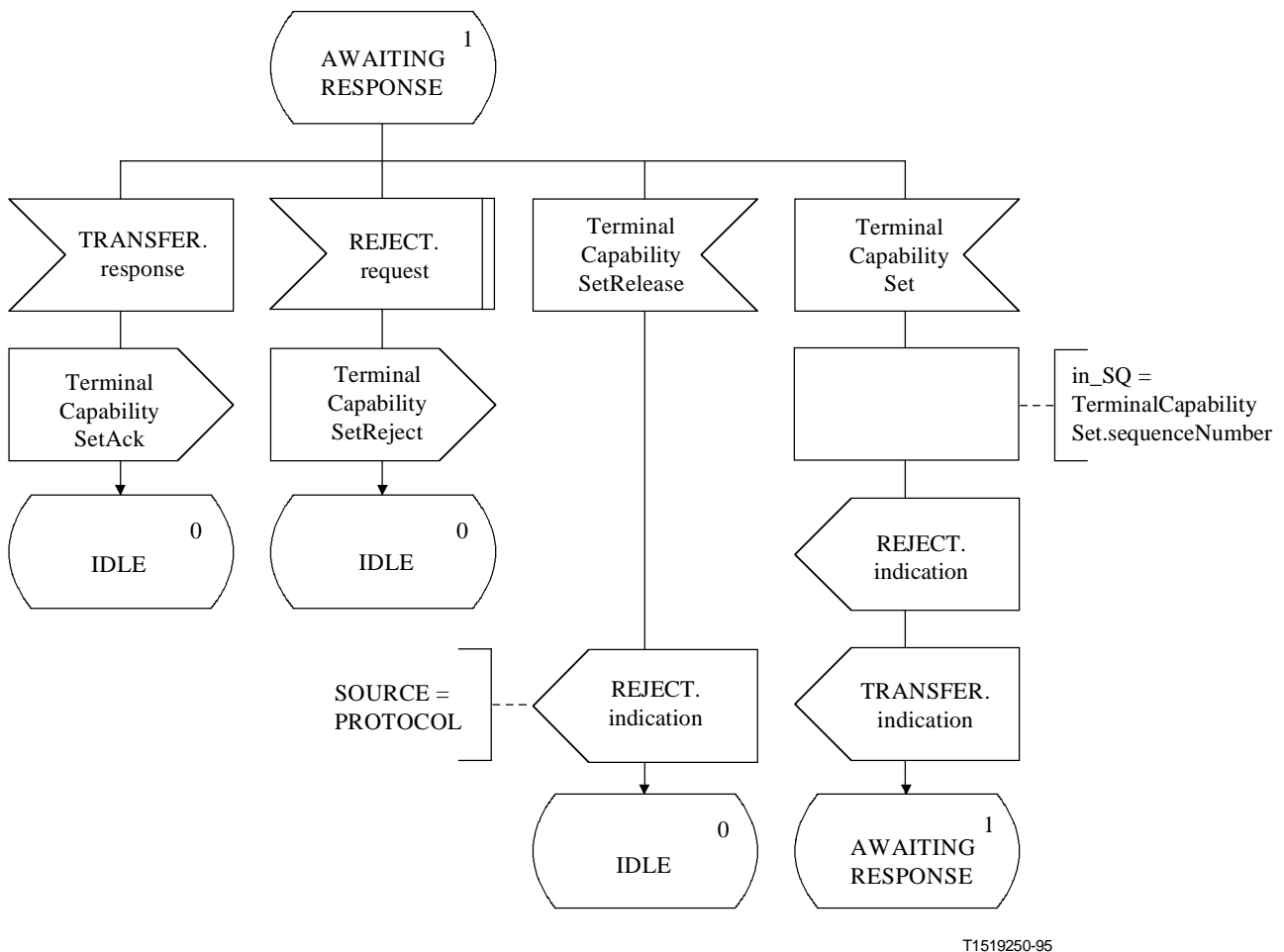


Figura 9.ii/H.245 – Diagrama SDL de la CESE de entrada (*fin*)

8.4 Procedimientos de señalización de canal lógico unidireccional

8.4.1 Introducción

El protocolo aquí especificado proporciona la apertura y el cierre de canales lógicos unidireccionales mediante procedimientos que incluyen acuse de recibo.

El protocolo especificado en esta sección 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 la cláusula 6.

Hay una LCSE de salida y una LCSE de entrada. Tanto en el lado de salida como en el lado de entrada hay una instancia 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.

Reemplazada por una versión más reciente

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.

8.4.1.1 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.

8.4.1.2 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.

Reemplazada por una versión más reciente

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.

8.4.1.3 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 de 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 director y el otro el subordinado. El protocolo definido en 8.2 proporciona un medio para tomar esta decisión. El terminal director 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 director, 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 sustentar cierto número de algoritmos de audio, pero que sólo puede decodificar el mismo algoritmo que está codificando.

8.4.2 Comunicación entre la LCSE y el usuario de LCSE

8.4.2.1 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 27.

Cuadro 27/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.

8.4.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- 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.

Reemplazada por una versión más reciente

8.4.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 27 son:

- a) 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.
- b) 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.
- c) 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".
- d) El parámetro ERRCODE indica el tipo de error de la LCSE. En el cuadro 31 se indican los valores admisibles del parámetro ERRCODE.

8.4.2.4 Estados de la LCSE

Se utilizan los estados que siguen para especificar la secuencia admisible 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.

Reemplazada por una versión más reciente

8.4.2.5 Diagrama de transición de estados

Se define aquí la secuencia admisible 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 admisibles se especifican por separado para cada LCSE de salida y LCSE de entrada, como se representa en las figuras 10 y 11 respectivamente.

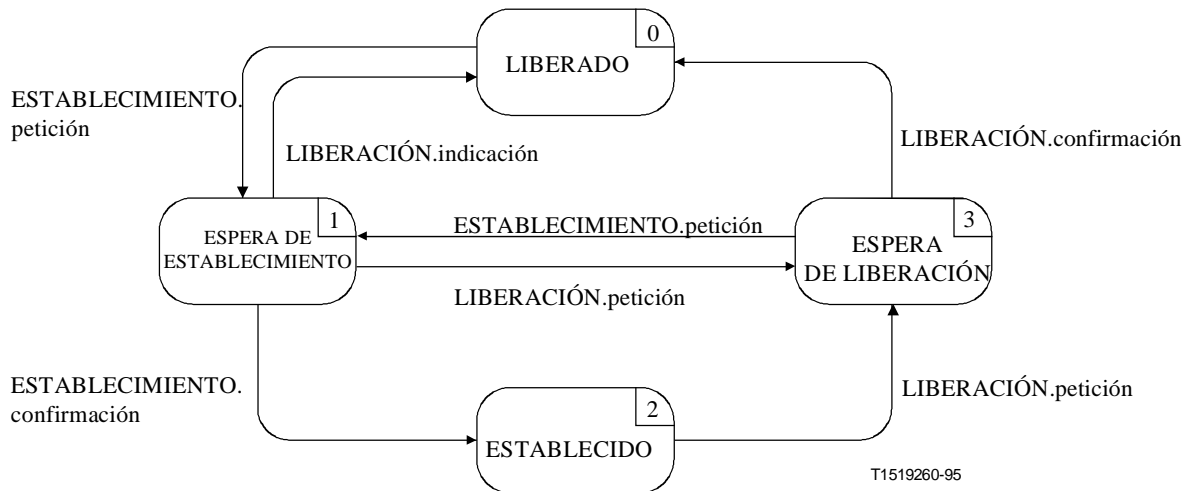


Figura 10/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una LCSE de salida

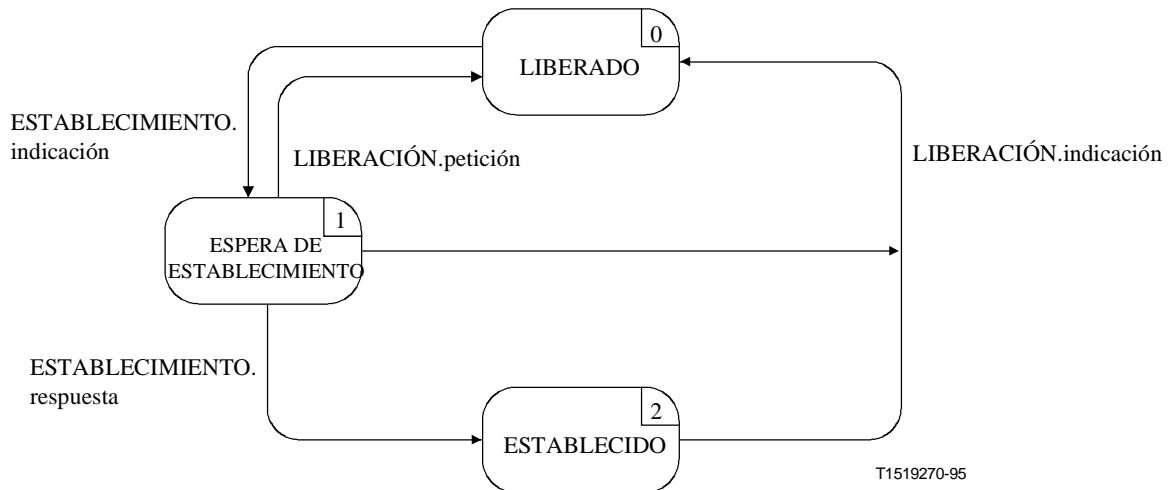


Figura 11/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una LCSE de entrada

8.4.3 Comunicación entre LCSE pares

8.4.3.1 Mensajes de LCSE

En el cuadro 28 se muestran los mensajes y campos de la LCSE, definidos en la cláusula 6, que son apropiados para el protocolo de la LCSE.

Reemplazada por una versión más reciente

Cuadro 28/H.245 – Nombres de mensajes y campos de la LCSE

función	mensajes	sentido	campo
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.

8.4.3.2 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.

8.4.3.3 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 admisible durante el cual no se reciben mensajes de OpenLogicalChannelAck, OpenLogicalChannelReject, ni CloseLogicalChannelAck.

8.4.4 Procedimientos de LCSE

8.4.4.1 Introducción

En la figura 12 se resumen las primitivas, sus parámetros y los mensajes para cada una de las LCSE de salida y de entrada.

Reemplazada por una versión más reciente

hacia/desde usuario de LCSE de salida

hacia/desde usuario de LCSE de entrada

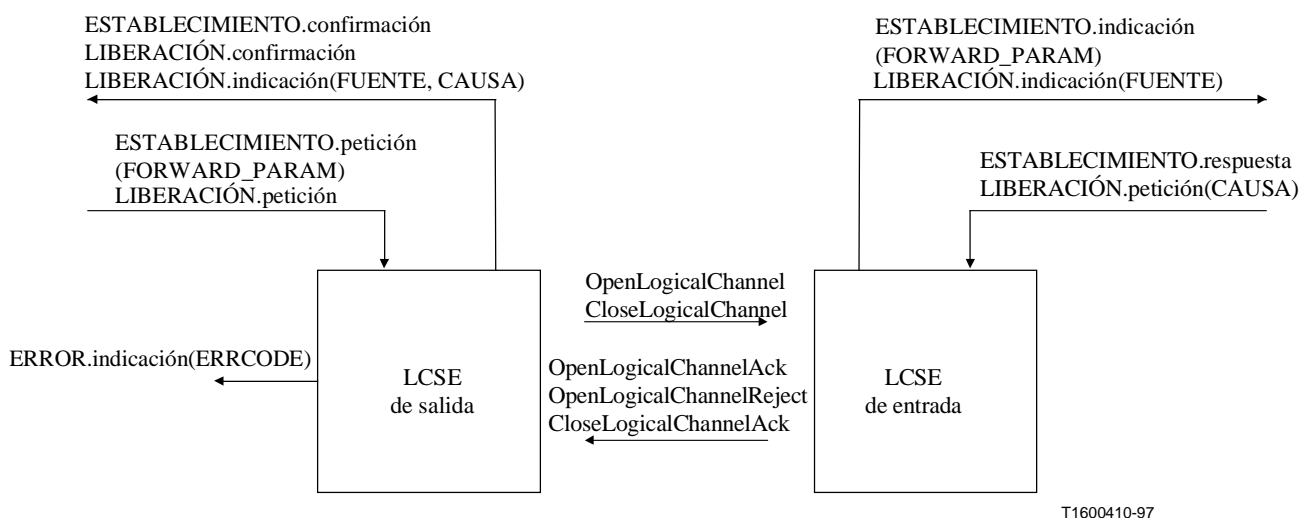


Figura 12/H.245 – Primitivas y mensajes en la entidad de señalización de canal lógico

8.4.4.2 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 29.

Cuadro 29/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.		

8.4.4.3 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 30.

Reemplazada por una versión más reciente

Cuadro 30/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>		

8.4.4.4 Valores del parámetro ERRCODE

El parámetro ERRCODE de la primitiva ERROR.indicación señala una condición de error determinada. En el cuadro 31 se muestran los valores que puede tomar el parámetro ERRCODE en la LCSE de salida. No hay ninguna primitiva indicación ERROR asociada con la LCSE de entrada.

Cuadro 31/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

8.4.4.5 Diagramas de SDL

En las figuras 13 y 14 se representan, en SDL, los procedimientos en la LCSE de salida y en la LCSE de entrada, respectivamente.

Reemplazada por una versión más reciente

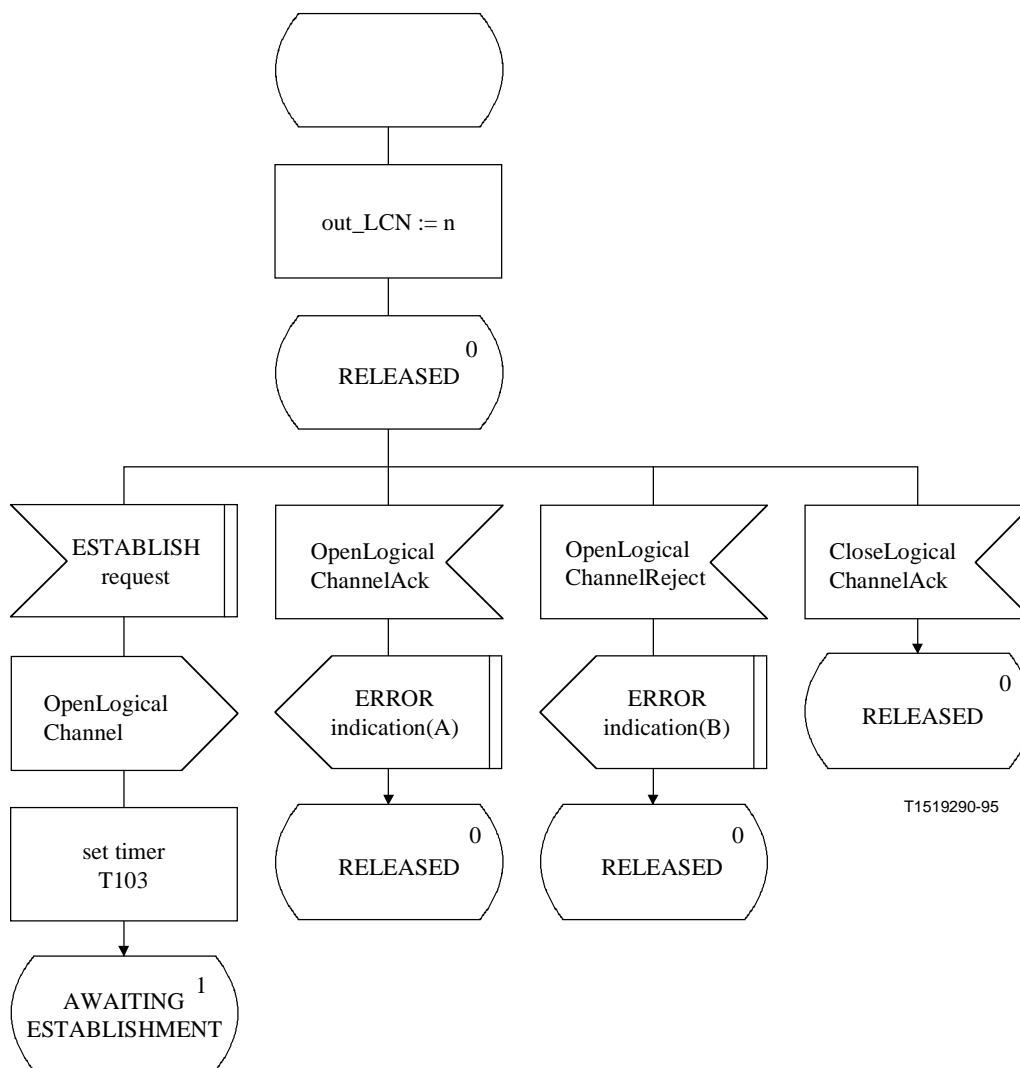


Figura 13.i/H.245 – Diagrama SDL de la LCSE de salida

Reemplazada por una versión más reciente

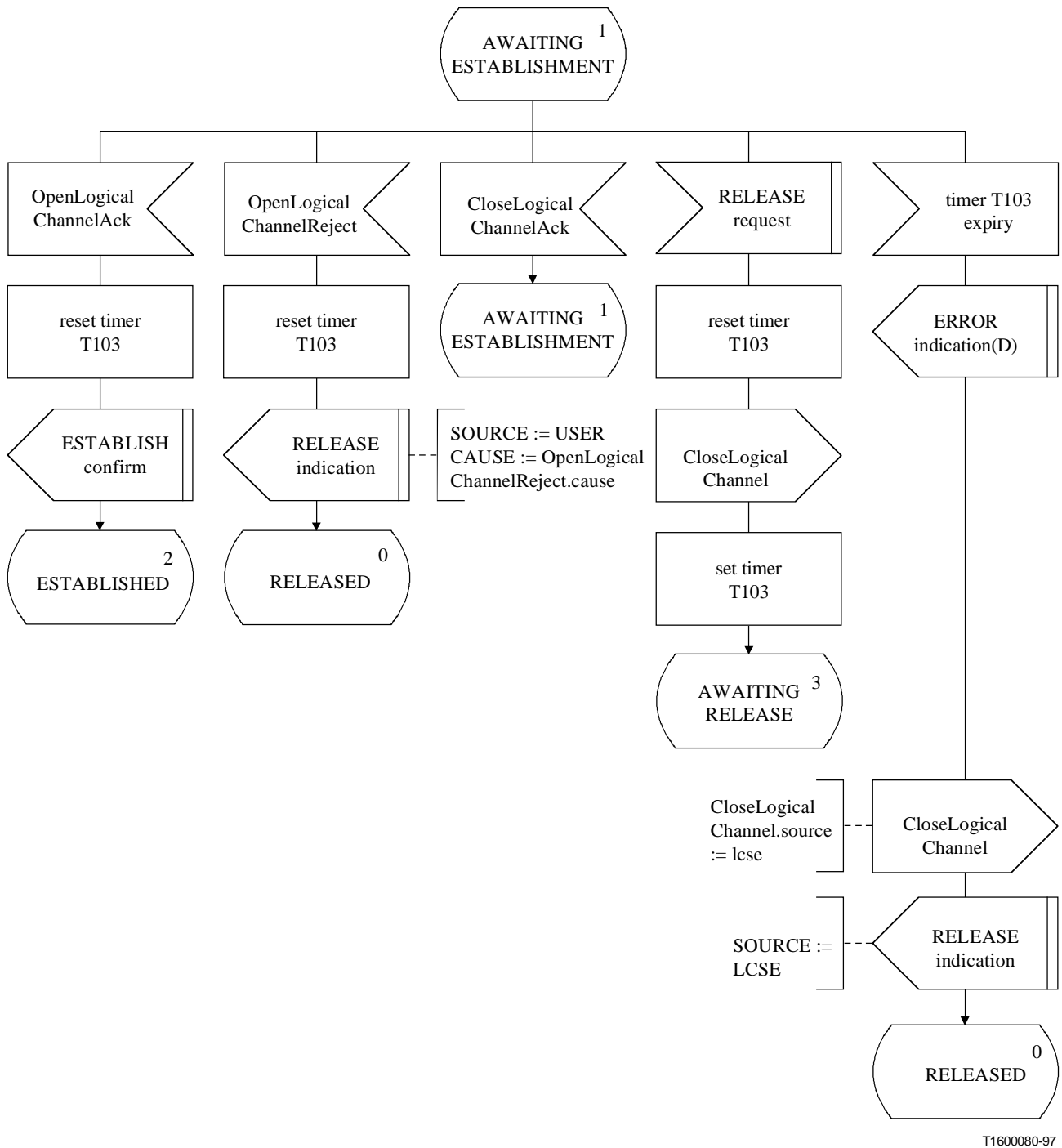


Figura 13.ii/H.245 – Diagrama SDL de la LCSE de salida (continuación)

Reemplazada por una versión más reciente

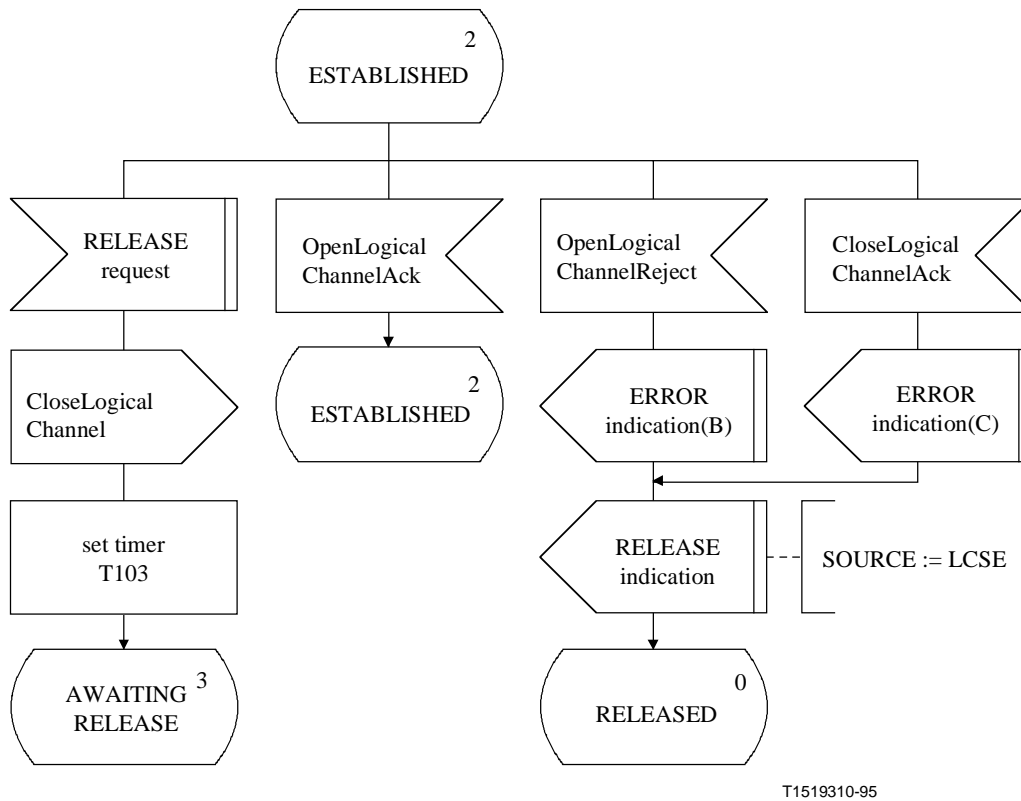


Figura 13.iii/H.245 – Diagrama SDL de la LCSE de salida (continuación)

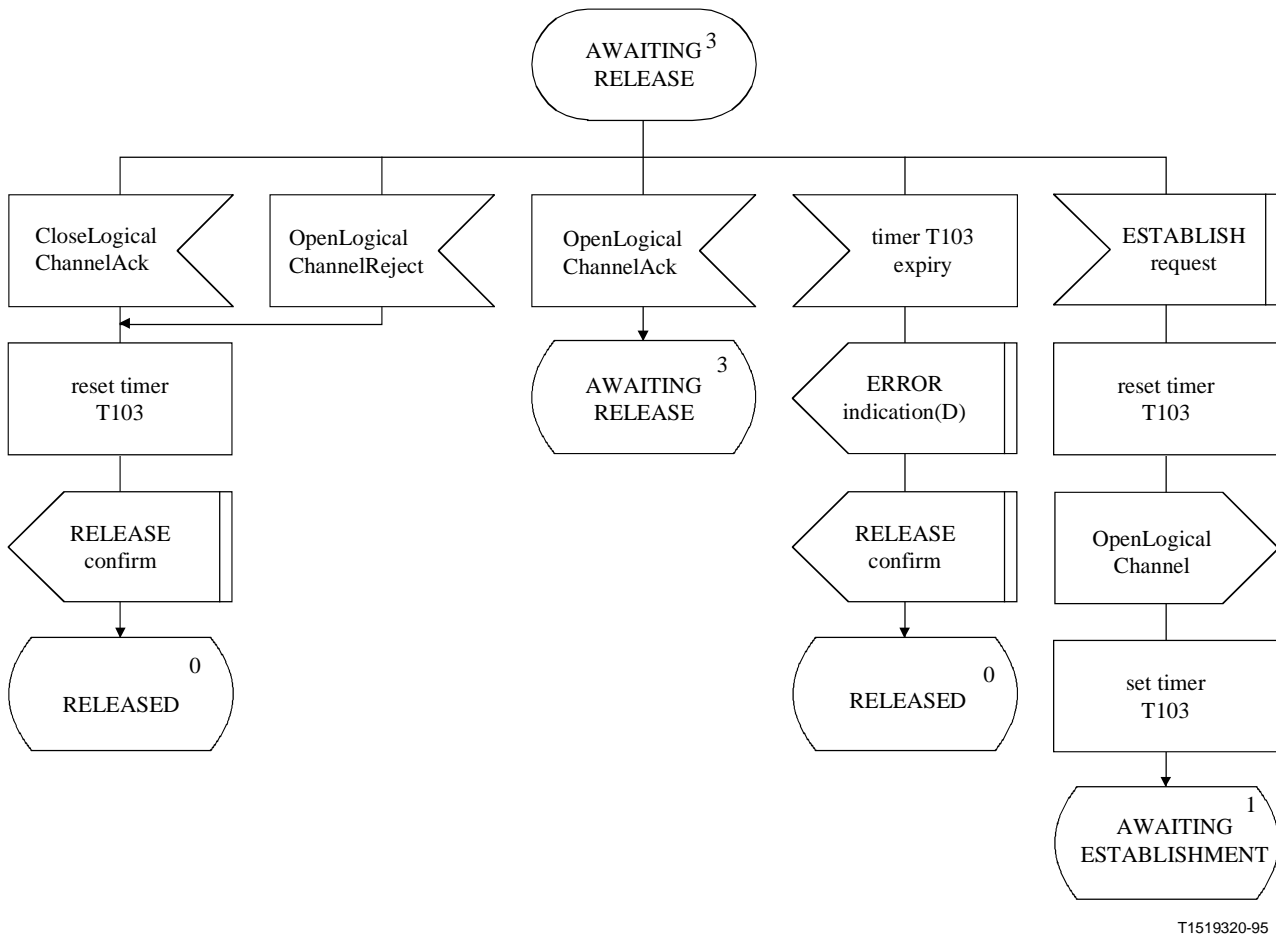
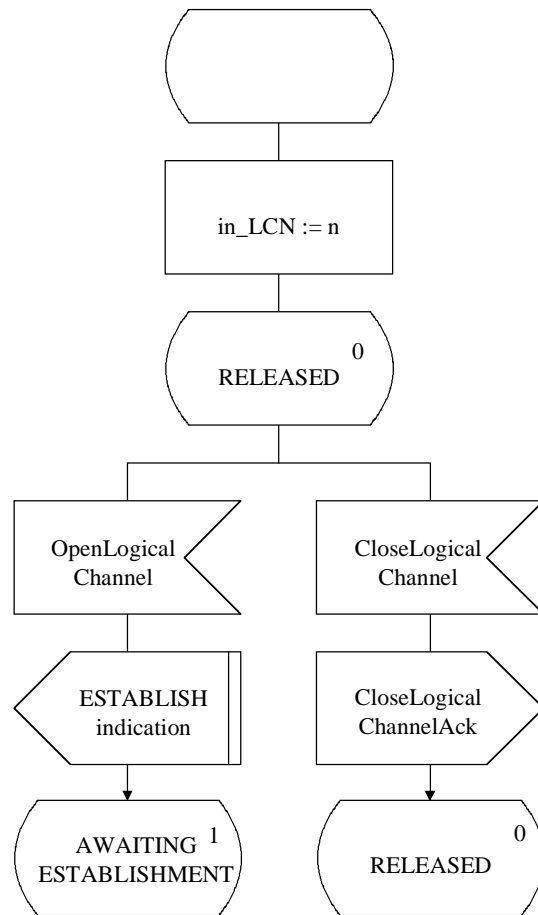


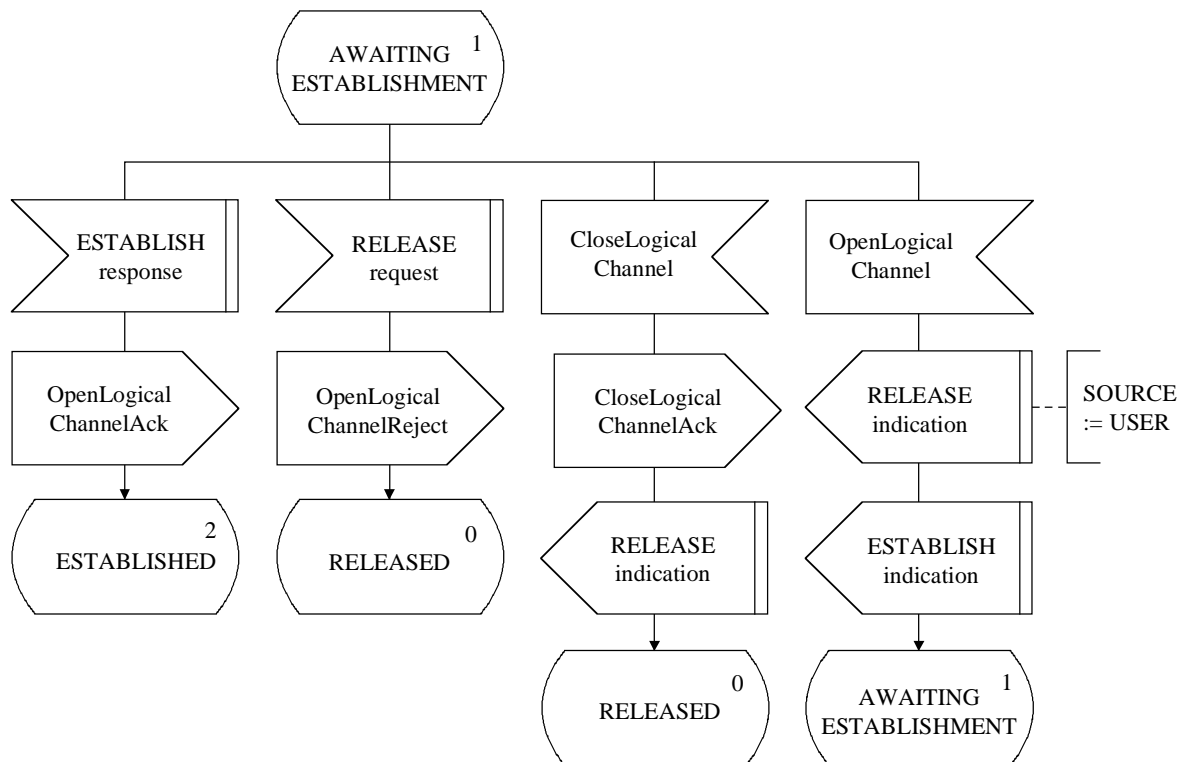
Figura 13.iv/H.245 – Diagrama SDL de la LCSE de salida (fin)

Reemplazada por una versión más reciente



T1519330-95

Figura 14.i/H.245 – Diagrama SDL de la LCSE de entrada



T1600090-97

Figura 14.ii/H.245 – Diagrama SDL de la LCSE de entrada (continuación)

Reemplazada por una versión más reciente

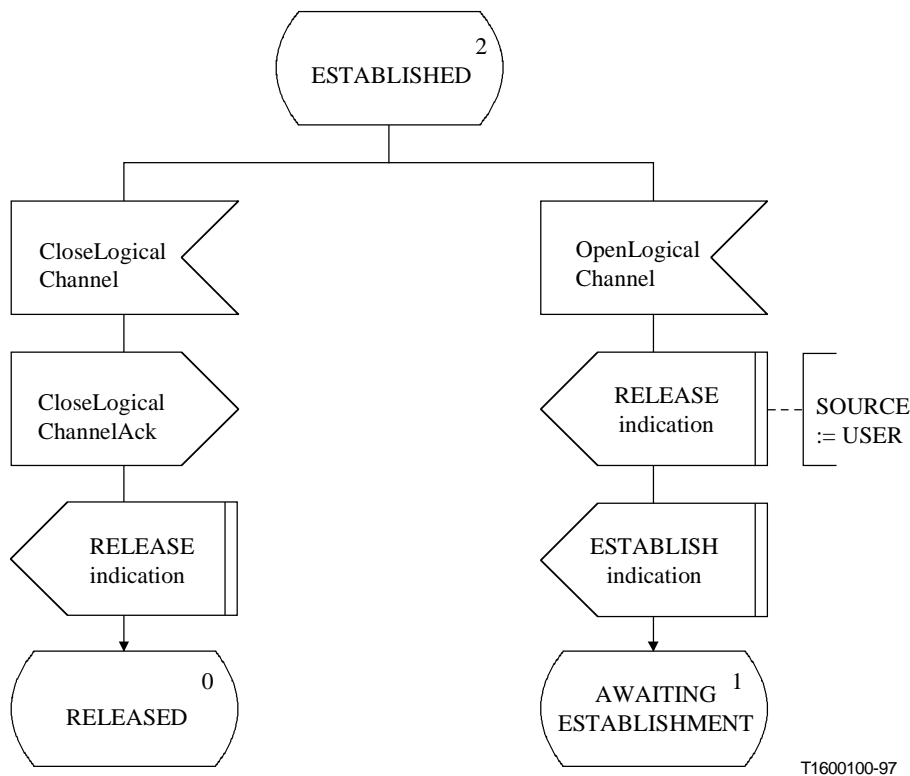


Figura 14.iii/H.245 – Diagrama SDL de la LCSE de entrada (*fin*)

8.5 Procedimientos de señalización de canal lógico bidireccional

8.5.1 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 la cláusula 6.

Hay una B-LCSE salida y una B-LCSE de entrada. Tanto en el lado de salida como en el entrada hay una instancia 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' ('forward') (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' ('reverse') (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 CONFIRMACION. Los datos que se reciban en otros estados que no sean los de ESTABLECIDO y ESPERA DE CONFIRMACION se descartarán, sin que por ello deba considerarse que se ha producido un fallo.

Reemplazada por una versión más reciente

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.

8.5.1.1 Visión general del protocolo – en la B-LCSE de salida

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 `LIBERACION.indicación` que el usuario de la B-LCSE par ha rehusado el establecimiento del canal lógico.

Si el temporizador T103 expira en este período, se informa de esto al usuario con la primitiva `LIBERACION.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 `LIBERACION.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 `LIBERACION.confirmación`, que el canal lógico ha sido correctamente cerrado.

Si el temporizador T103 expira en este período, se informa de esto al usuario con la primitiva `LIBERACION.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 `LIBERACION.petición`.

Reemplazada por una versión más reciente

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`.

8.5.1.2 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 `LIBERACION.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 `LIBERACION.indicación`, y se envía el mensaje `CloseLogicalChannelAck` a la B-LCSE de salida par.

8.5.1.3 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 pueda abrir canales lógicos, es preciso determinar que uno de los dos terminales es el terminal director, y el otro el subordinado. El protocolo definido en 8.2 proporciona un medio para decidir esta situación. El terminal director 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 director, 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.

8.5.2 Comunicación entre la B-LCSE y el usuario de B-LCSE

8.5.2.1 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 32.

Reemplazada por una versión más reciente

Cuadro 32/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.				

8.5.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- 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.

8.5.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 32 son:

- 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 B-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 B-LCSE par.
- 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.
- 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.
- 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 usuario de la B-LCSE.
- El parámetro ERRCODE indica el tipo de error B-LCSE. En el cuadro 36 se indican los valores del parámetro ERRCODE.

Reemplazada por una versión más reciente

8.5.2.4 Estados de la B-LCSE

Los siguientes estados se utilizan para especificar la secuencia admisible 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 LIBERACION

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 CONFIRMACION

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.

8.5.2.5 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 15 y 16.

Reemplazada por una versión más reciente

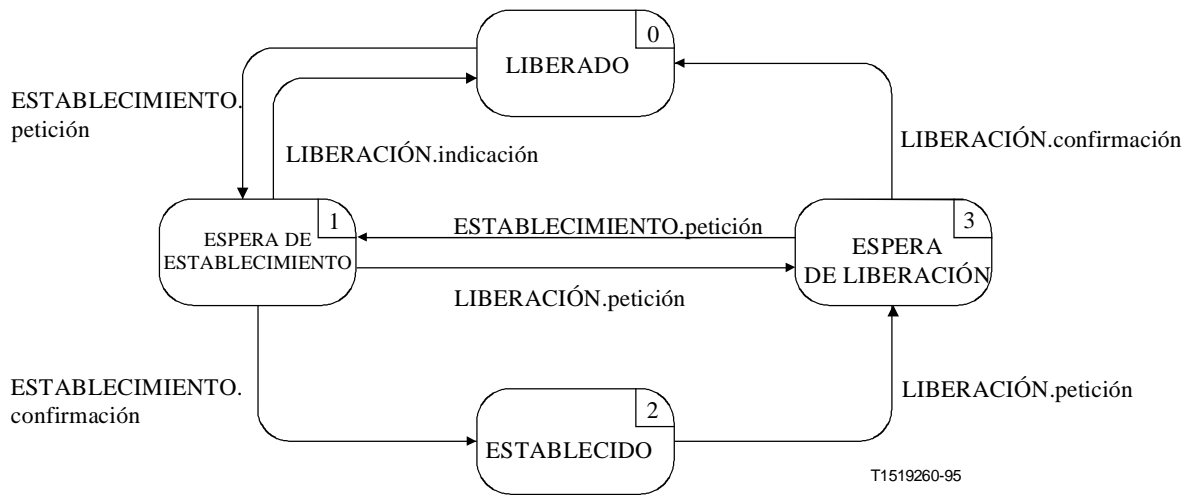


Figura 15/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una B-LCSE de salida

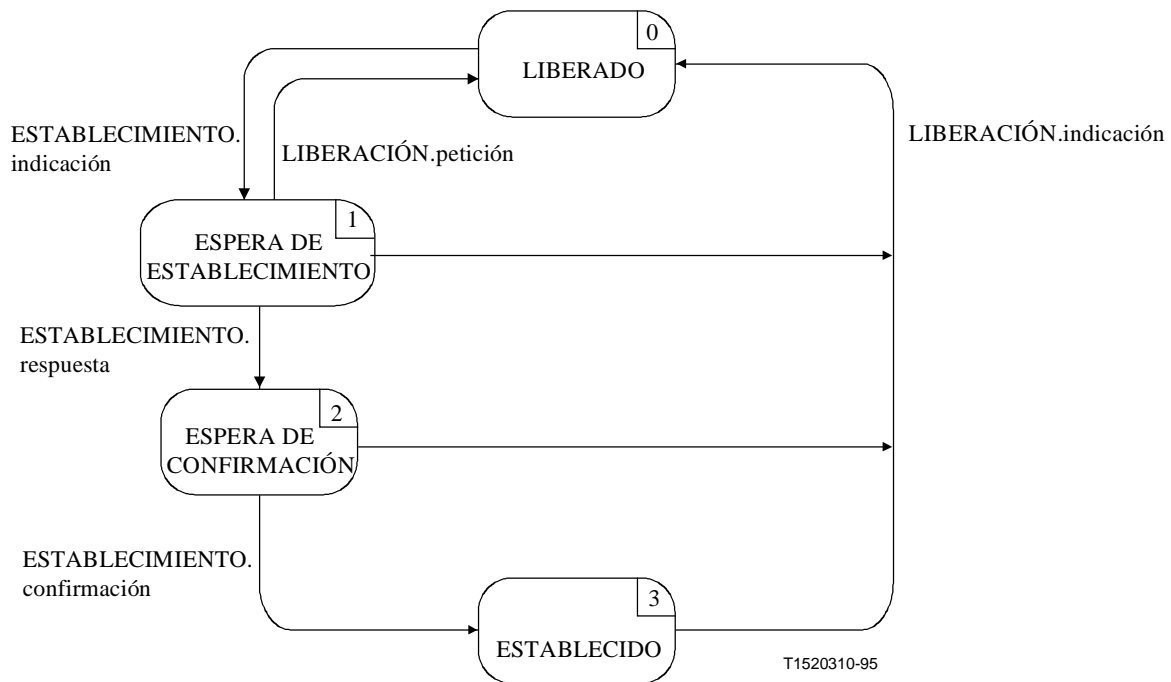


Figura 16/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una B-LCSE de entrada

8.5.3 Comunicaciones entre B-LCSE pares

8.5.3.1 Mensajes de la B-LCSE

En el cuadro 33 se indican los mensajes y campos de la B-LCSE, definidos en la cláusula 6, que son apropiados para el protocolo B-LCSE.

Reemplazada por una versión más reciente

Cuadro 33/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
	CloseLogicalChannelAck	O <- I	source forwardLogicalChannelNumber

NOTA – Sentido: O - salida, I - entrada.

8.5.3.2 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.

8.5.3.3 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 LIBERACION. 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.

Reemplazada por una versión más reciente

8.5.4 Procedimientos de B-LCSE

8.5.4.1 Introducción

La figura 17 resume las primitivas de la B-LCSE y sus parámetros, y los mensajes, para la B-LCSE de salida y la de entrada.

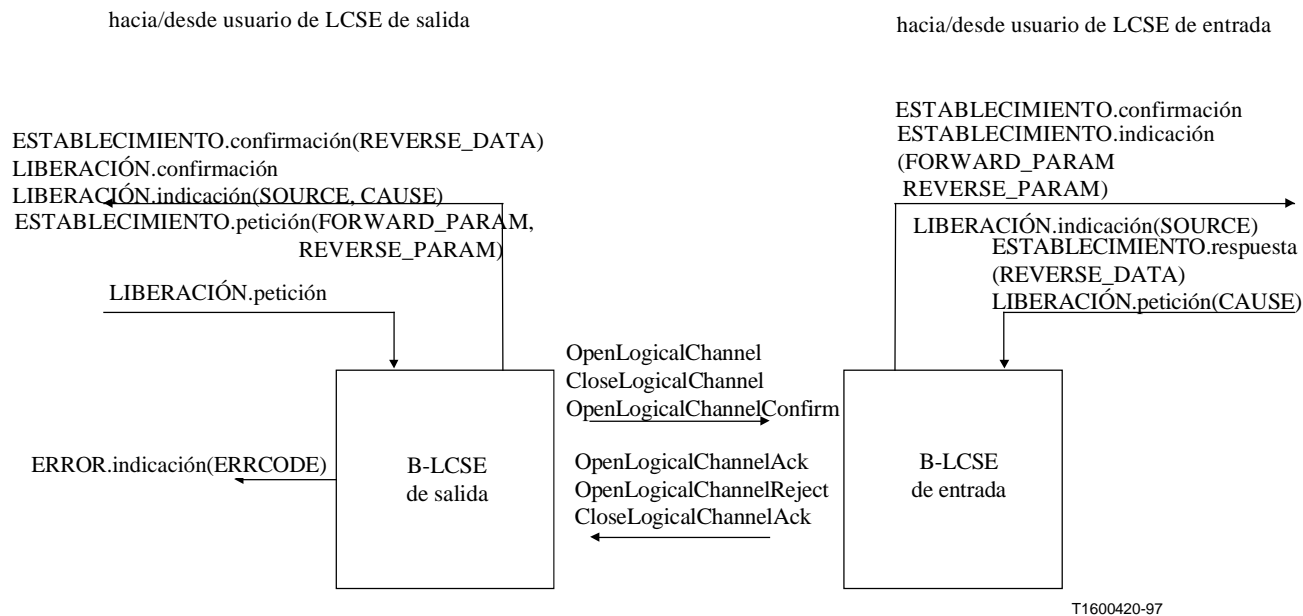


Figura 17/H.245 – Primitivas y mensajes en la entidad de señalización de canal lógico bidireccional

8.5.4.2 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 34.

Cuadro 34/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.confirmación	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.

Reemplazada por una versión más reciente

8.5.4.3 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 35.

Cuadro 35/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.		

8.5.4.4 Valores del parámetro ERRCODE

El parámetro ERRCODE de la primitiva ERROR.indicación indica una determinada condición de error. El cuadro 36 muestra los valores que puede tomar el parámetro ERRCODE en la B-LCSE de salida y el cuadro 37 muestra los valores que puede tomar el parámetro ERRCODE en la B-LCSE de entrada.

Cuadro 36/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

Reemplazada por una versión más reciente

Cuadro 37/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

8.5.4.5 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 18 y 19, respectivamente.

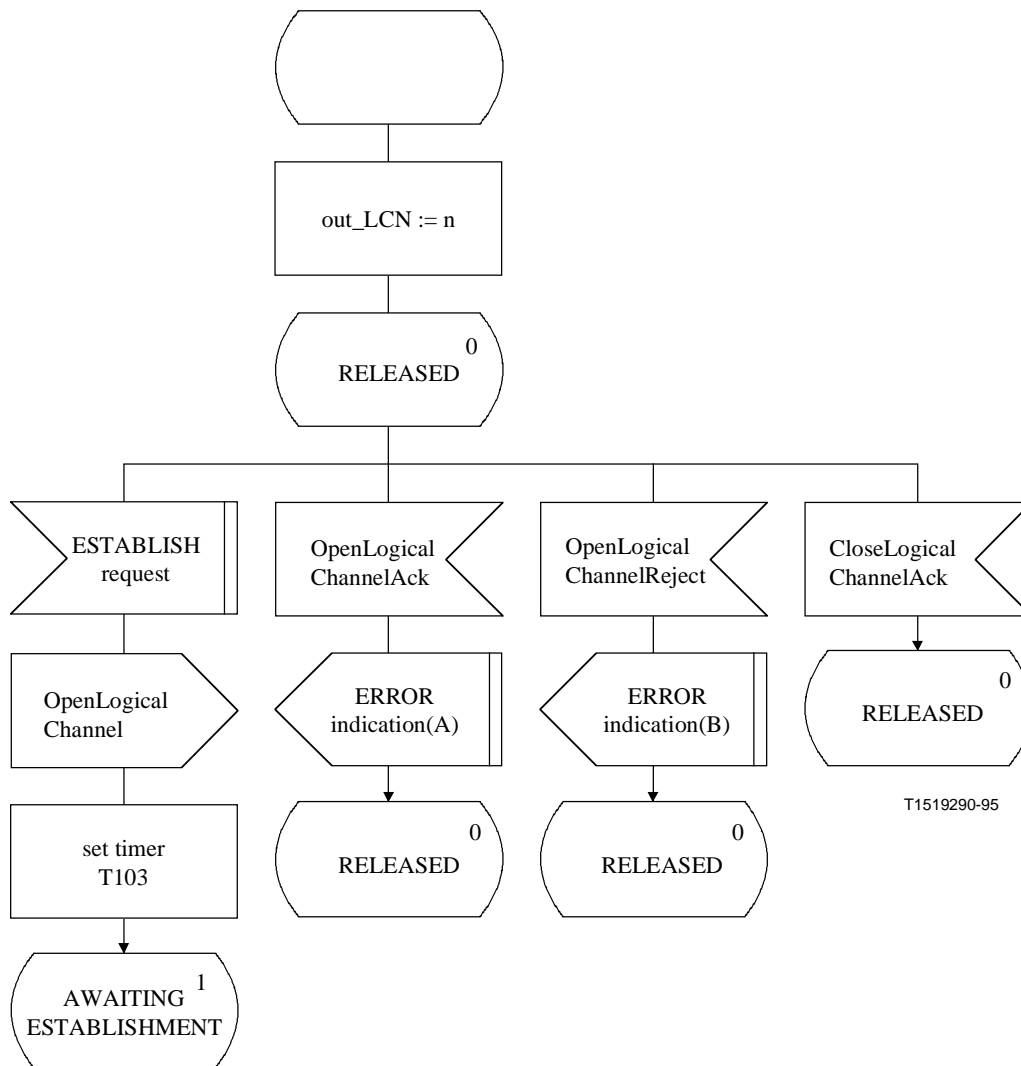
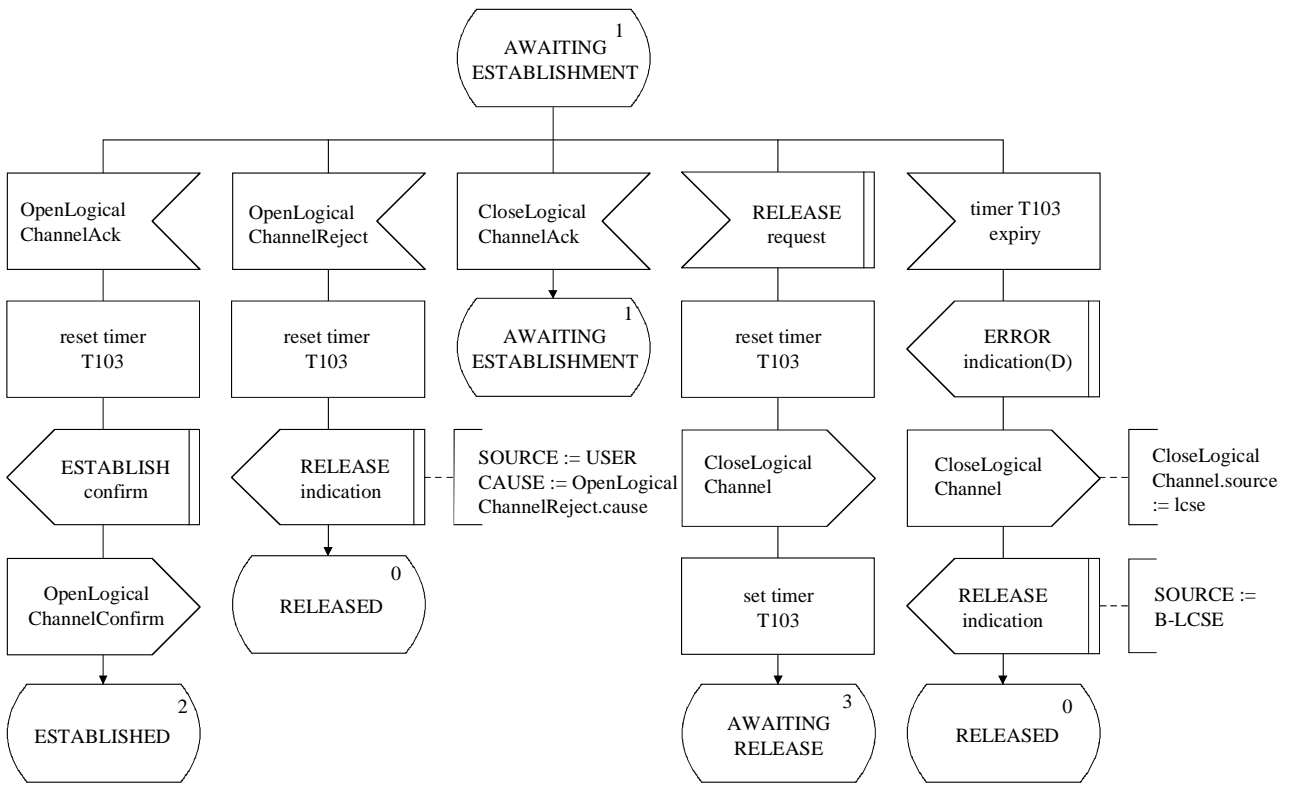


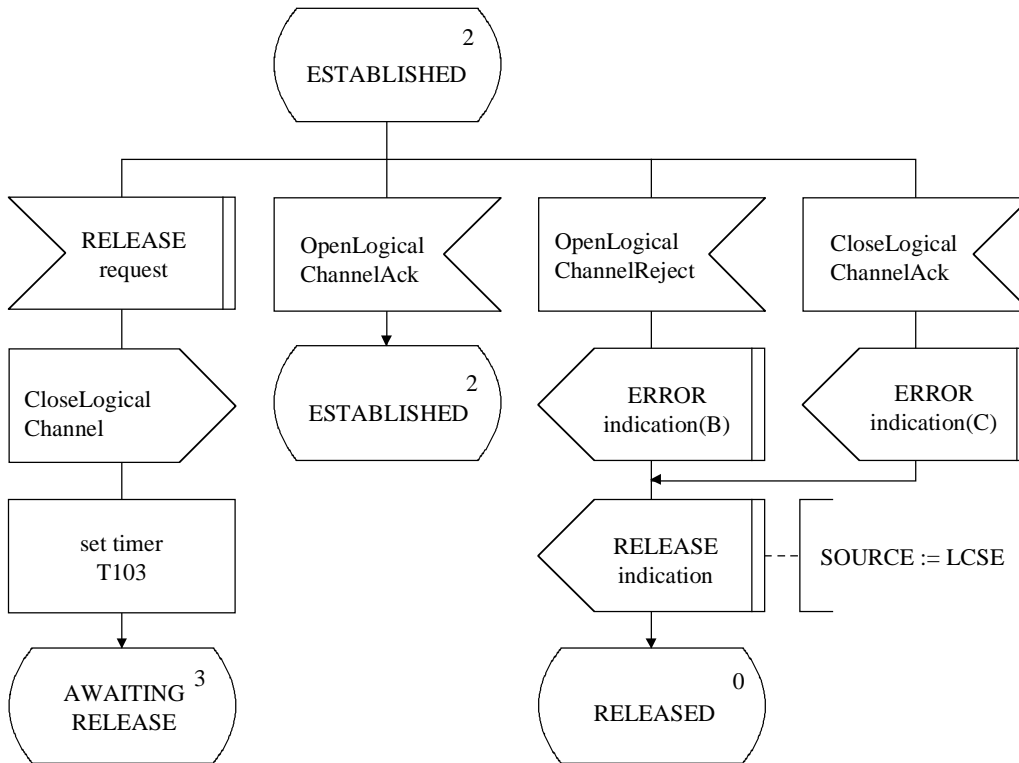
Figura 18.i/H.245 – Diagrama SDL de la B-LCSE de salida

Reemplazada por una versión más reciente



T1600110-97

Figura 18.ii/H.245 – Diagrama SDL de la B-LCSE de salida (continuación)



T1519310-95

Figura 18.iii/H.245 – Diagrama SDL de la B-LCSE de salida (continuación)

Reemplazada por una versión más reciente

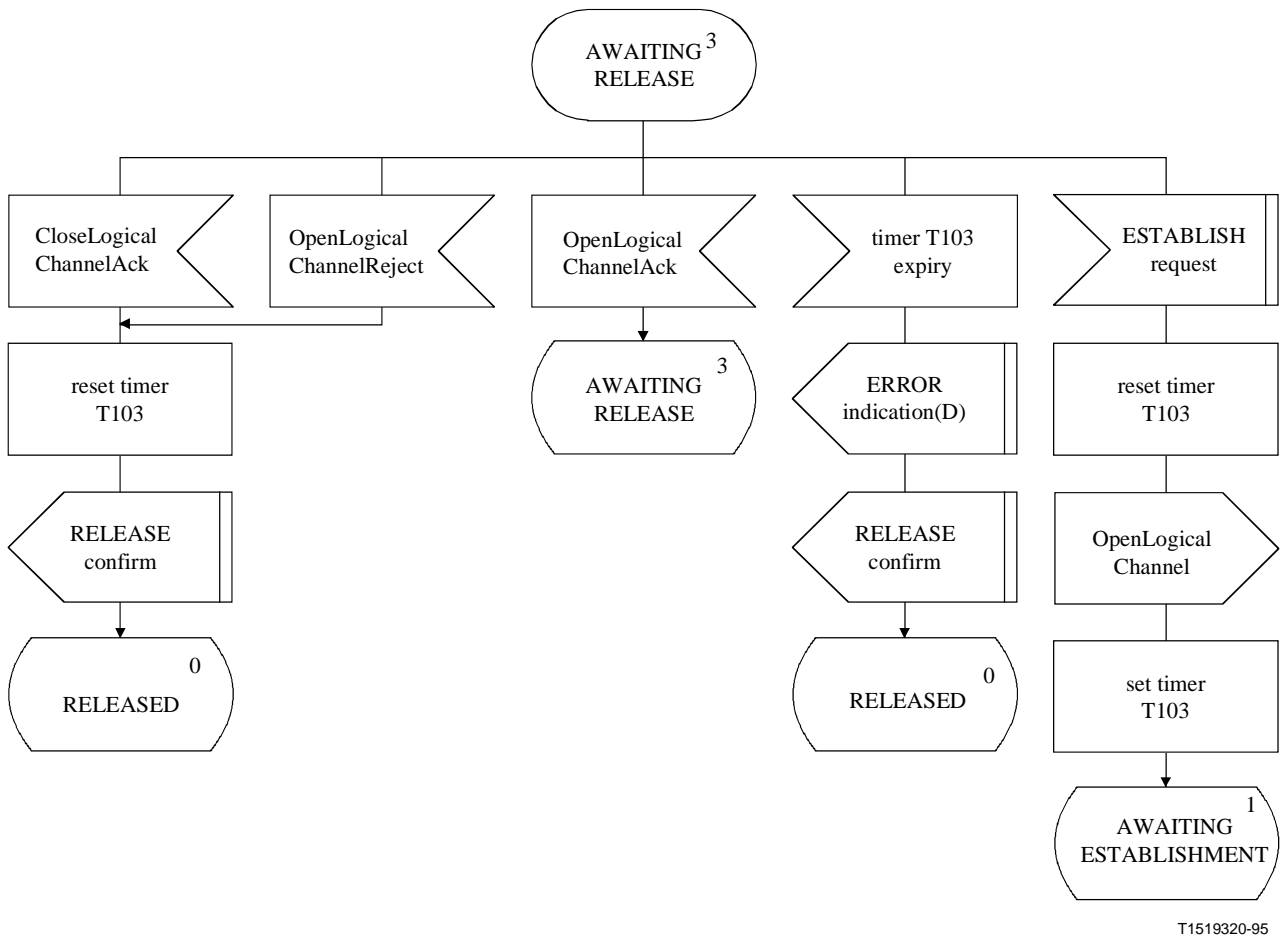


Figura 18.iv/H.245 – Diagrama SDL de la B-LCSE de salida (*fin*)

Reemplazada por una versión más reciente

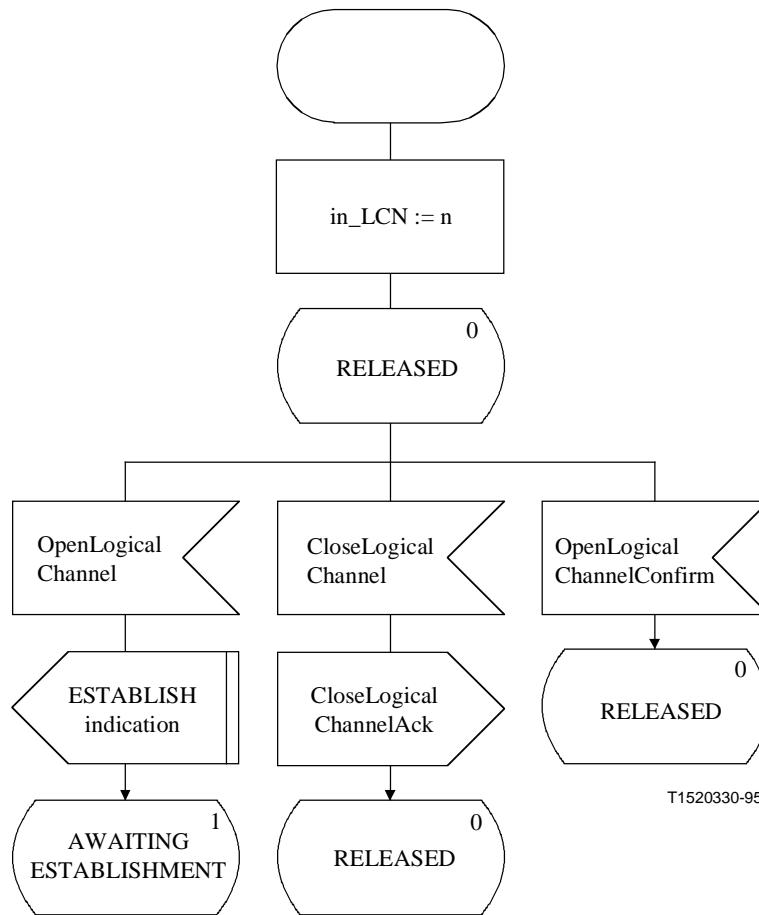


Figura 19.i/H.245 – Diagrama SDL de la B-LCSE de entrada

Reemplazada por una versión más reciente

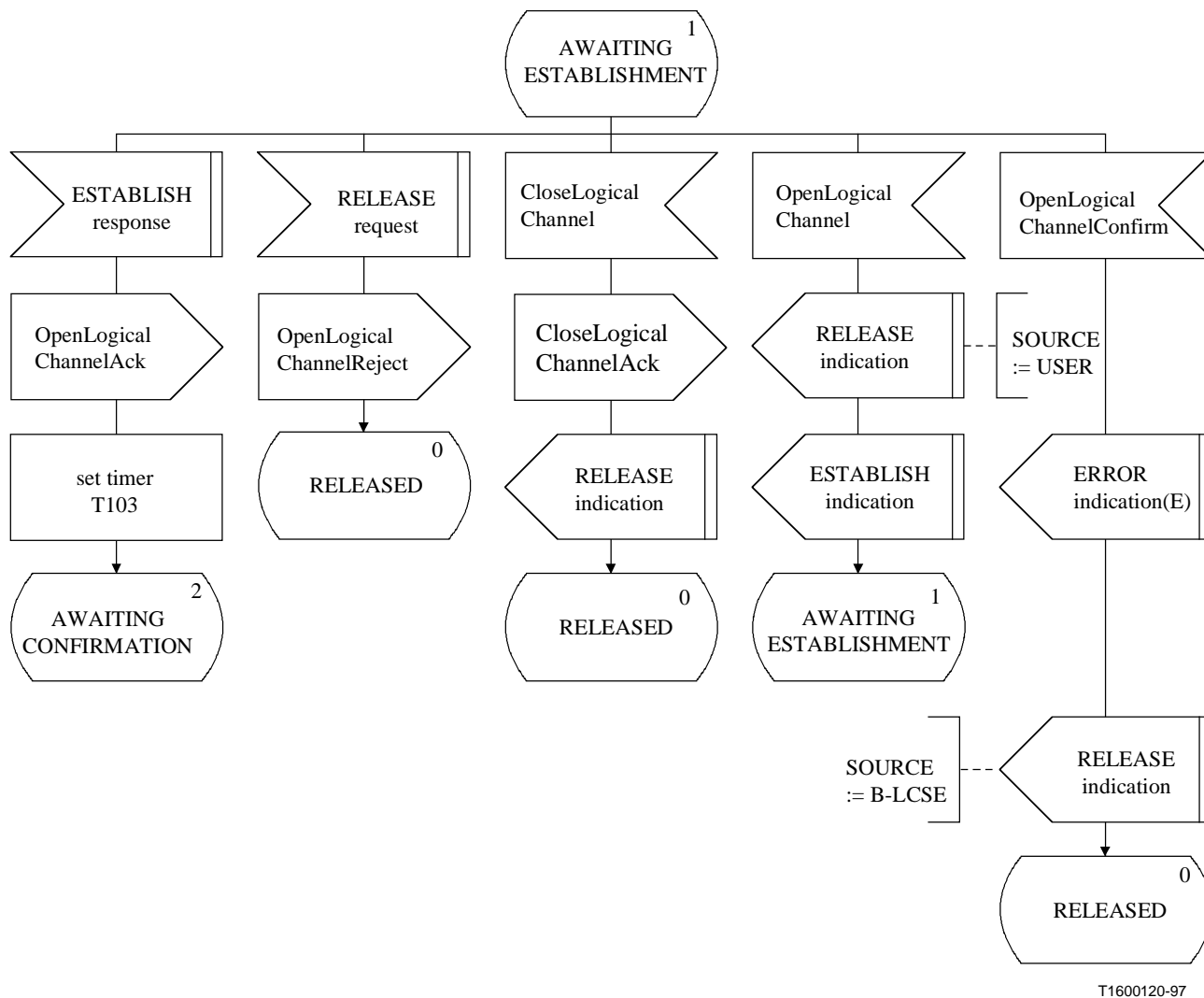


Figura 19.ii/H.245 – Diagrama SDL de la B-LCSE de entrada (continuación)

Reemplazada por una versión más reciente

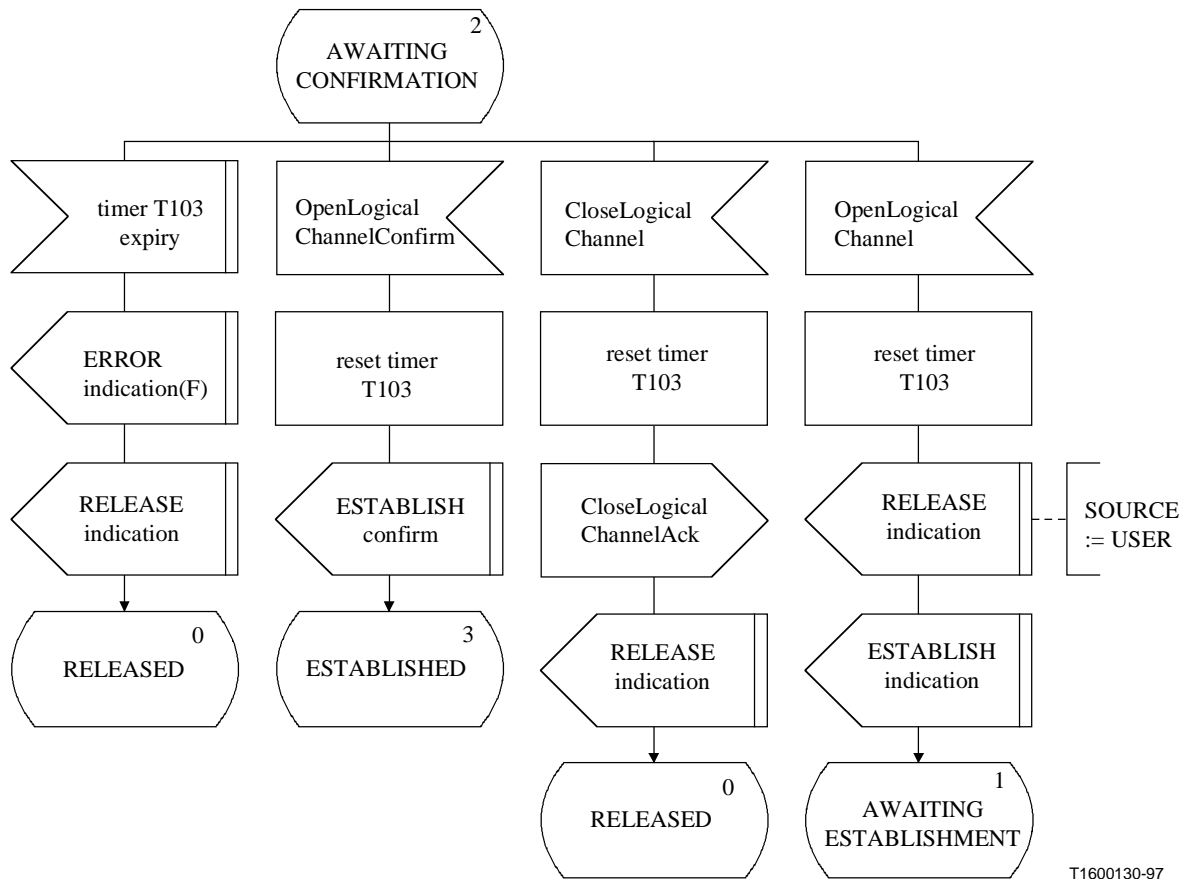


Figura 19.iii/H.245 – Diagrama SDL de la B-LCSE de entrada (continuación)

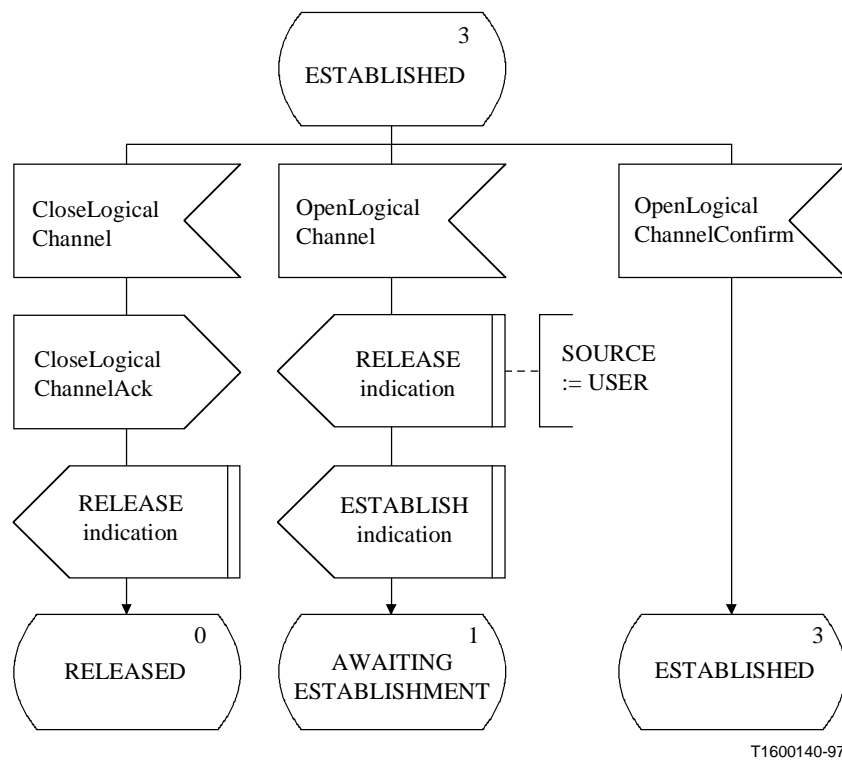


Figura 19.iv/H.245 – Diagrama SDL de la B-LCSE de entrada (fin)

Reemplazada por una versión más reciente

8.6 Procedimientos de cierre de canal lógico

8.6.1 Introducción

Estos procedimientos los utiliza un terminal para pedir al terminal distante que cierre un canal lógico. Obsérvese que éstos son únicamente procedimientos 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 par por medio de mensajes apropiados definidos en la cláusula 6. Hay una CLCSE de salida y una CLCSE de entrada. En cada uno de los extremos de salida y de entrada hay una instancia de la 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 LIBERACION.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.

8.6.1.1 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 par, 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.

8.6.1.2 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.

8.6.2 Comunicación entre la CLCSE y el usuario de CLCSE

8.6.2.1 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 38.

Reemplazada por una versión más reciente

Cuadro 38/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.

8.6.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- Se utilizan las primitivas CIERRE para la petición de cierre de un canal lógico.
- Se utilizan las primitivas RECHAZO para el rechazo del cierre de un canal lógico.

8.6.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 38 son:

- 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.
- 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".

8.6.2.4 Estados de la CLCSE

Para especificar la secuencia admisible 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.

8.6.2.5 Diagramas de transición de estados

Se define aquí la secuencia admisible de primitivas entre la CLCSE y el usuario de la CLCSE. Se especifican por separado las secuencias admitidas para una CLCSE de salida y una CLCSE de entrada, como se muestra en las figuras 20 y 21, respectivamente.

Reemplazada por una versión más reciente

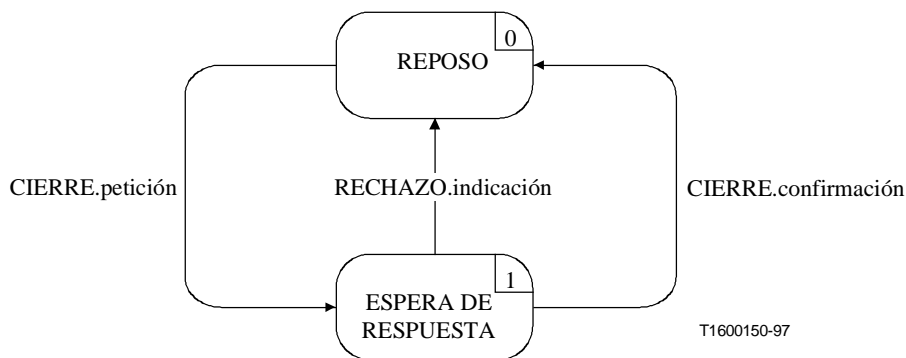


Figura 20/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CLCSE de salida

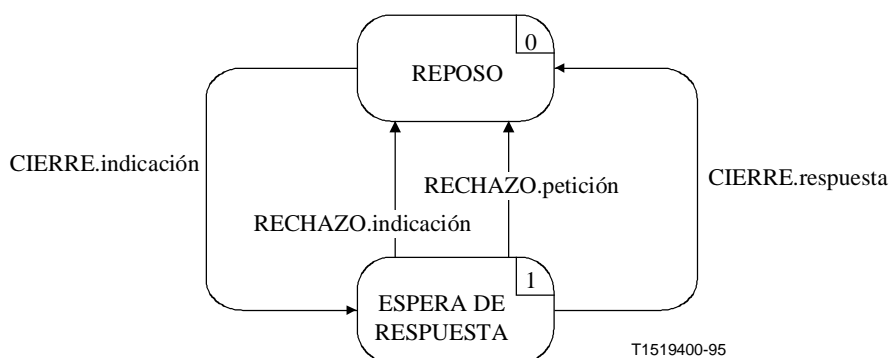


Figura 21/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una CLCSE de entrada

8.6.3 Comunicación entre CLCSE pares

8.6.3.1 Mensajes

En el cuadro 39 se muestran los mensajes y campos de la CLCSE, definidos en la cláusula 6, que son apropiados para el protocolo de la CLCSE.

Cuadro 39/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.			

Reemplazada por una versión más reciente

8.6.3.2 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.

8.6.3.3 Temporizadores de la CLCSE

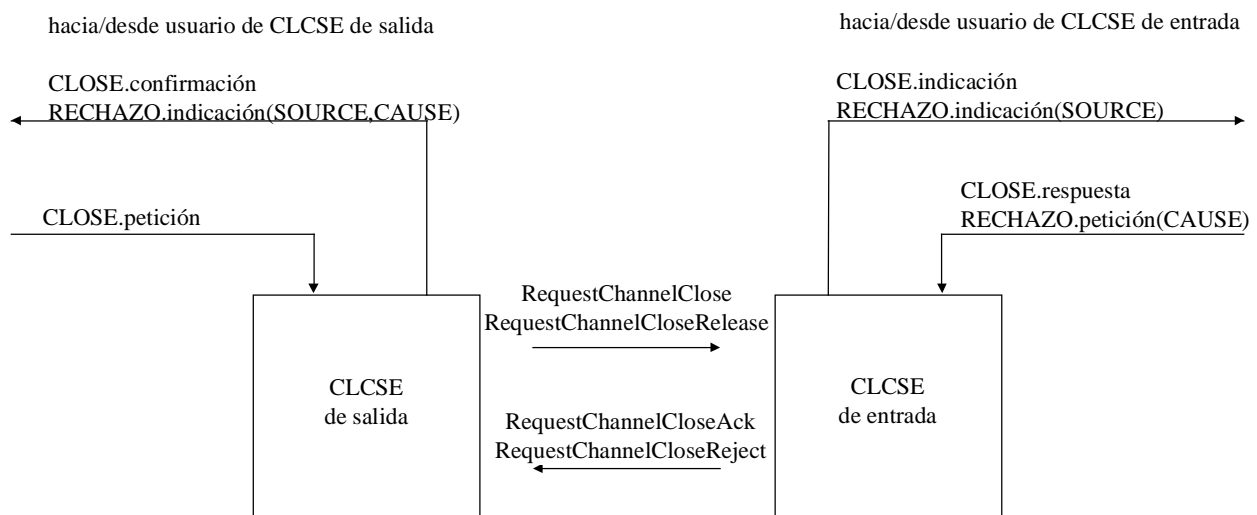
Para la CLCSE de salida se especifica el siguiente temporizador:

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.

8.6.4 Procedimientos de la CLCSE

En la figura 22 se resumen las primitivas de la CLCSE, sus parámetros y mensajes para cada CLCSE de salida y de entrada.



T1519410-95

Figura 22/H.245 – Primitivas y mensajes en la entidad señalización cierre de canal lógico

Reemplazada por una versión más reciente

8.6.4.1 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 40.

Cuadro 40/H.245 – Valores supletorios de los parámetros de las primitivas

primitiva	parámetro	valor supletorio
RECHAZO.indicación	SOURCE CAUSE	USER nulo

8.6.4.2 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 41.

Cuadro 41/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 REJECT.request(CAUSE)
RequestChannelCloseRelease	forwardLogicalChannelNumber	out_LCN

8.6.4.3 Diagramas de SDL

Los procedimientos en la CLCSE de salida y en la CLCSE de entrada, expresados en SDL, se indican en las figuras 23 y 24, respectivamente.

Reemplazada por una versión más reciente

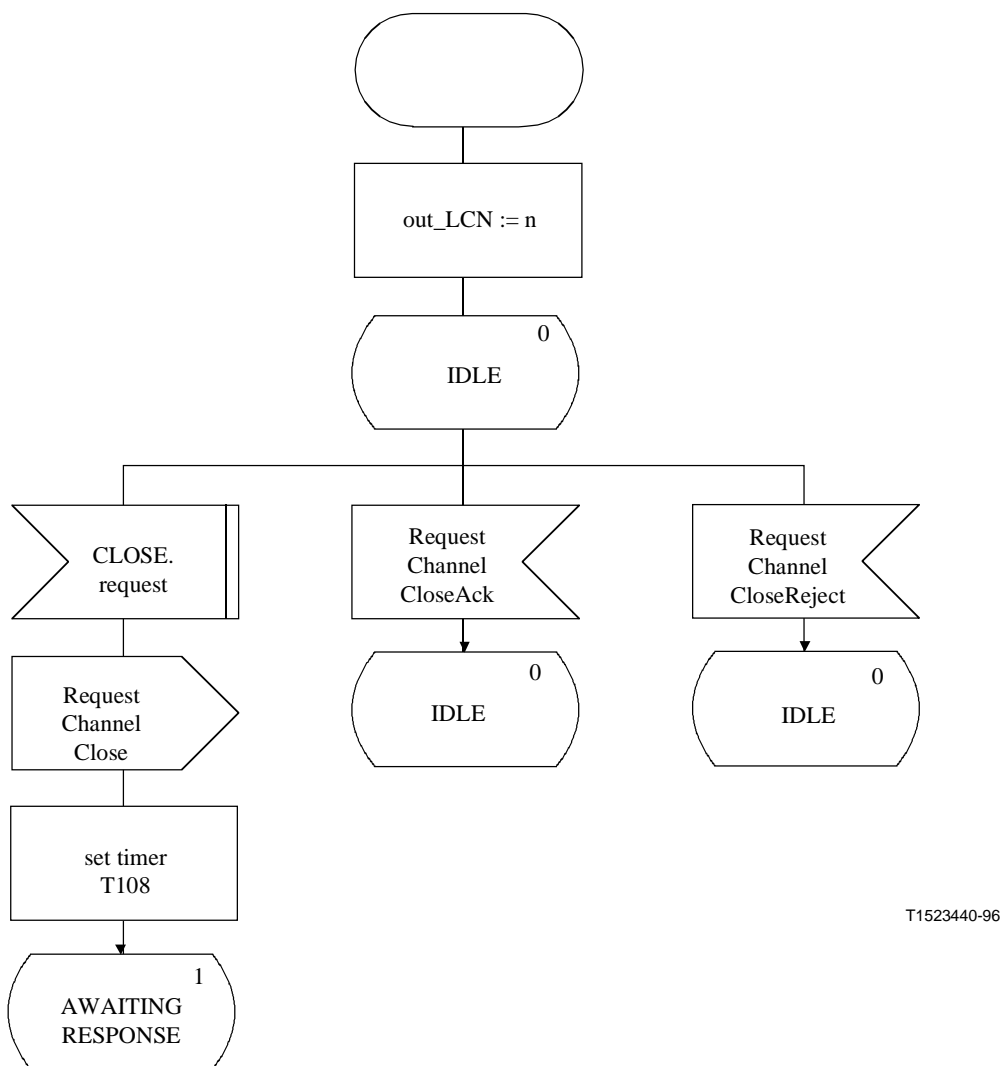
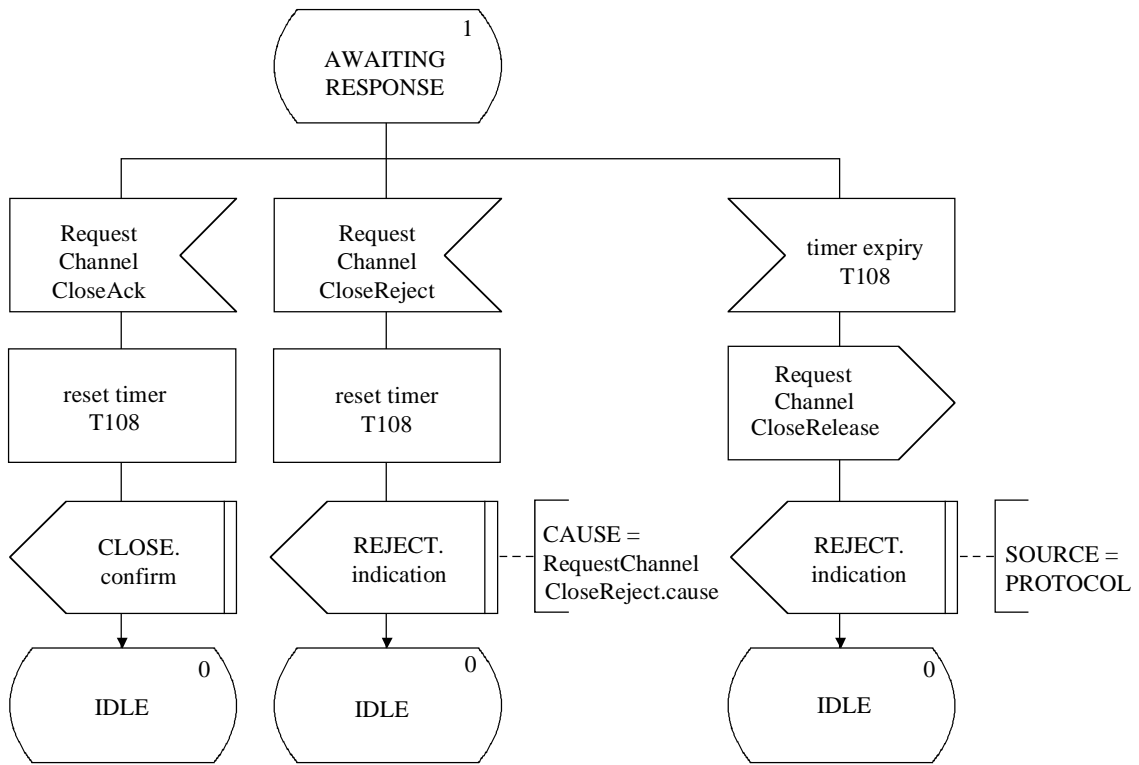


Figura 23.i/H.245 – Diagrama SDL de la CLCSE de salida

Reemplazada por una versión más reciente



T1523450-96

Figura 23.ii/H.245 – Diagrama SDL de la CLCSE de salida (*fin*)

Reemplazada por una versión más reciente

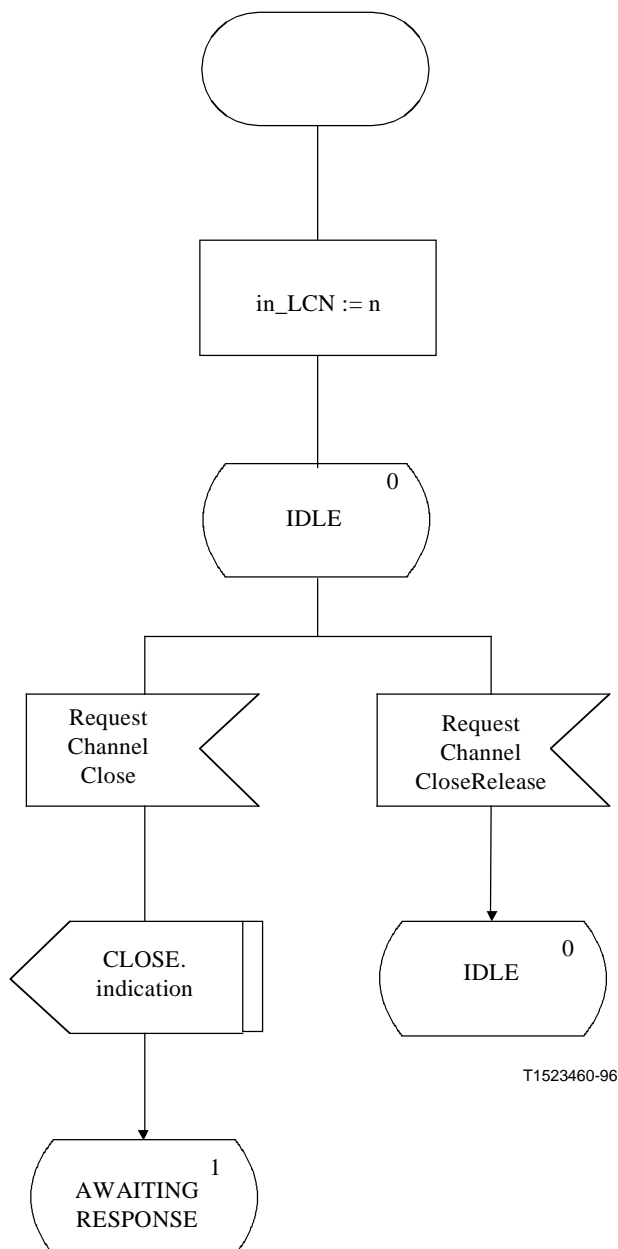


Figura 24.i/H.245 – Diagrama SDL de la CLCSE de entrada

Reemplazada por una versión más reciente

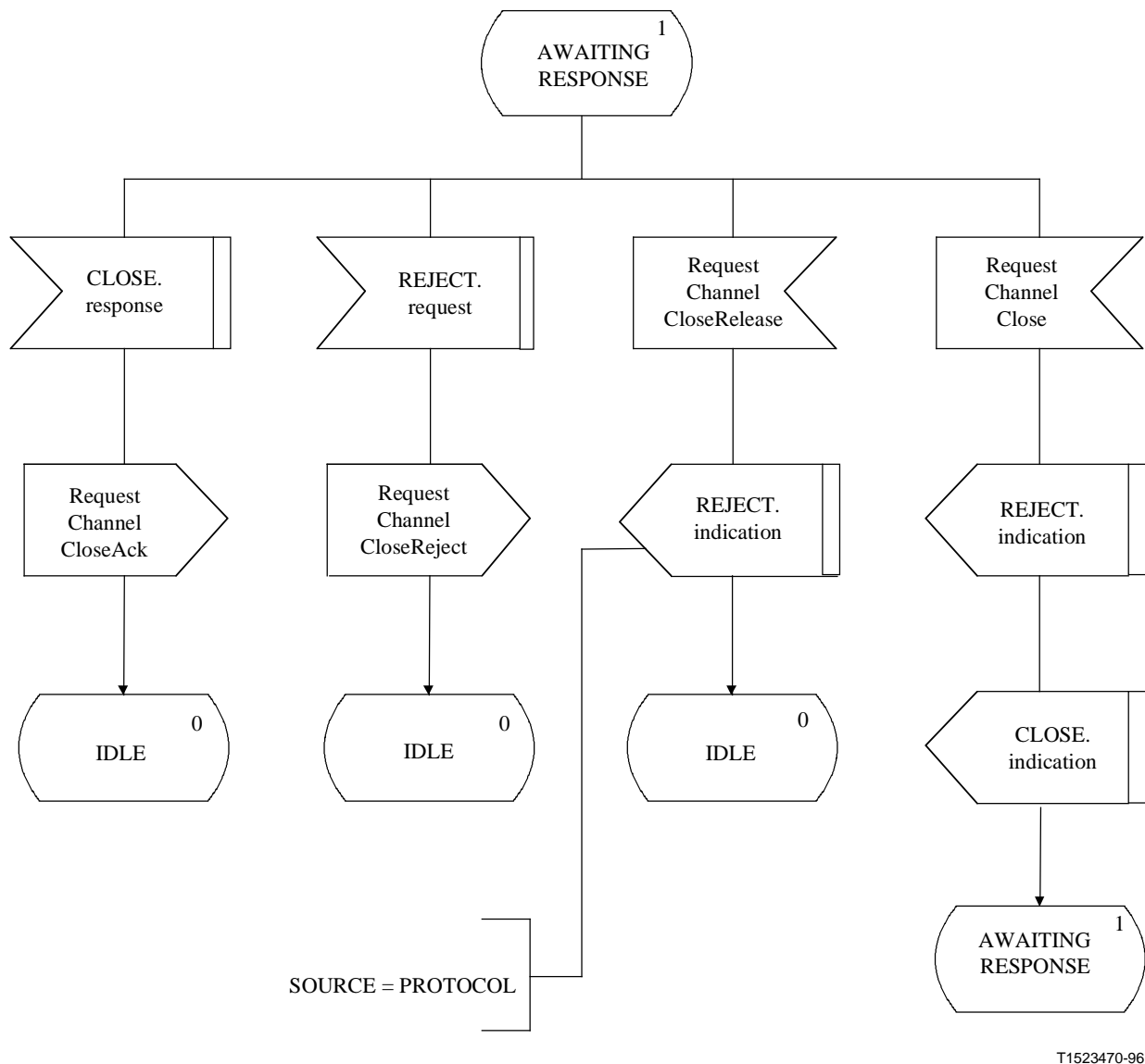


Figura 24.ii/H.245 – Diagrama SDL de la CLCSE de entrada (*fin*)

8.7 Procedimientos de la tabla múltiplex H.223

8.7.1 Introducción

La tabla múltiplex se emplea para asociar cada octeto dentro de una MUX-PDU de la Recomendación 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 la cláusula 6.

Hay una MTSE de salida y una MTSE de entrada. Para cada entrada de tabla múltiplex hay una instancia de la MTSE.

Reemplazada por una versión más reciente

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 a 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 reemplazará 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.

8.7.1.1 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.

Reemplazada por una versión más reciente

8.7.1.2 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 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.

8.7.2 Comunicación entre la MTSE y el usuario de MTSE

8.7.2.1 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 42.

Cuadro 42/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.

8.7.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- 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.

8.7.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 42 son:

- 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.

Reemplazada por una versión más reciente

- b) 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.
- c) 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”.

8.7.2.4 Estados de la MTSE

Los siguientes estados se utilizan para especificar la secuencia admisible 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.

8.7.2.5 Diagramas de transición de estados

Se define aquí la secuencia admisible de primitivas entre la MTSE y el usuario de la MTSE. Las secuencias admisibles se especifican separadamente para la MTSE de salida y la MTSE de entrada, como se indica en las figuras 25 y 26, respectivamente.

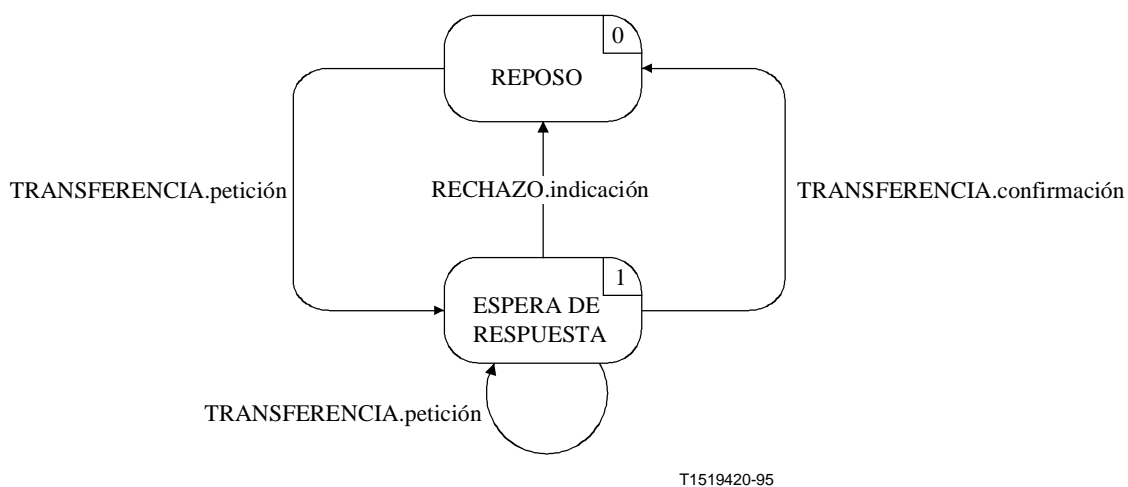


Figura 25/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MTSE de salida

Reemplazada por una versión más reciente

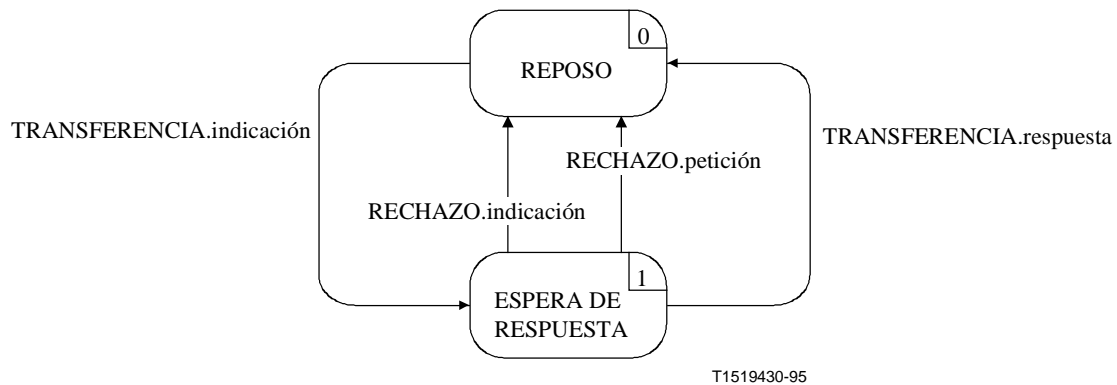


Figura 26/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MTSE de entrada

8.7.3 Comunicación entre MTSE pares

8.7.3.1 Mensajes

En el cuadro 43 se indican los mensajes y campos de la MTSE, definidos en la cláusula 6, apropiados para el protocolo de la MTSE.

Cuadro 43/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.

8.7.3.2 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.

Reemplazada por una versión más reciente

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 modulo 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.

8.7.3.3 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.

8.7.4 Procedimientos de MTSE

8.7.4.1 Introducción

En la figura 27 se resumen las primitivas de la MTSE, sus parámetros y mensajes para las MTSE de salida y de entrada.

Reemplazada por una versión más reciente

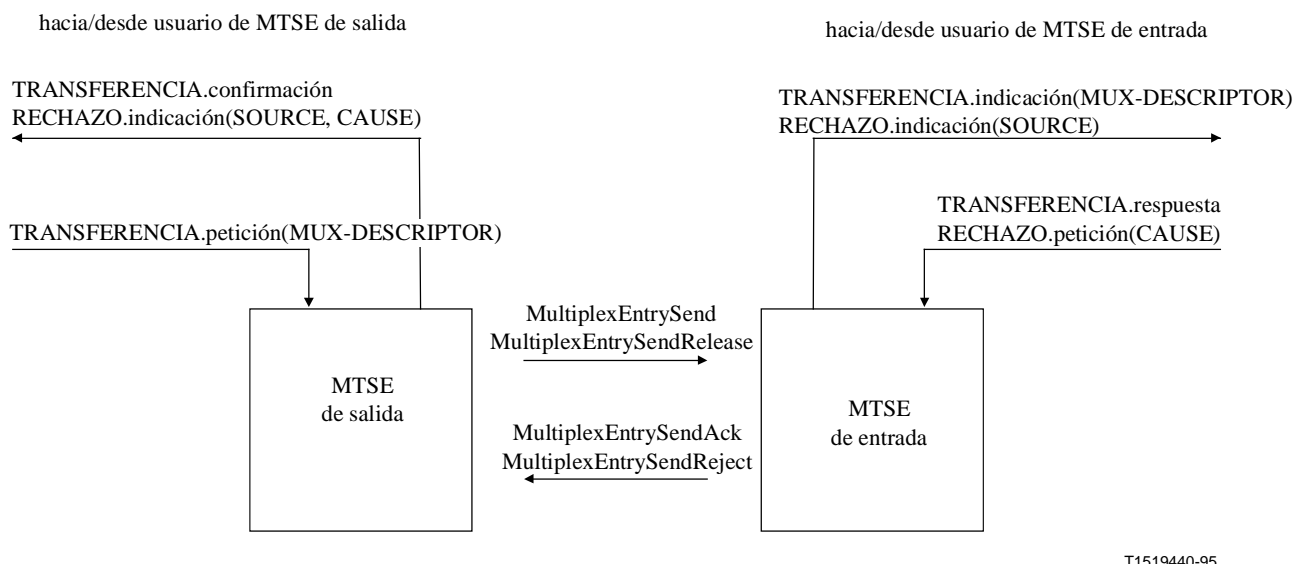


Figura 27/H.245 – Primitivas y mensajes en la entidad de señalización de tabla múltiplex

8.7.4.2 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 44.

Cuadro 44/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

8.7.4.3 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 45.

Reemplazada por una versión más reciente

Cuadro 45/H.245 – Valores supletorios de los campos de mensajes

mensaje	campo	valor supletorio (nota)
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.		

8.7.4.4 Diagrama SDL

Los procedimientos de la MTSE de salida y de la MTSE de entrada se expresan en forma SDL en las figuras 28 y 29 respectivamente.

Reemplazada por una versión más reciente

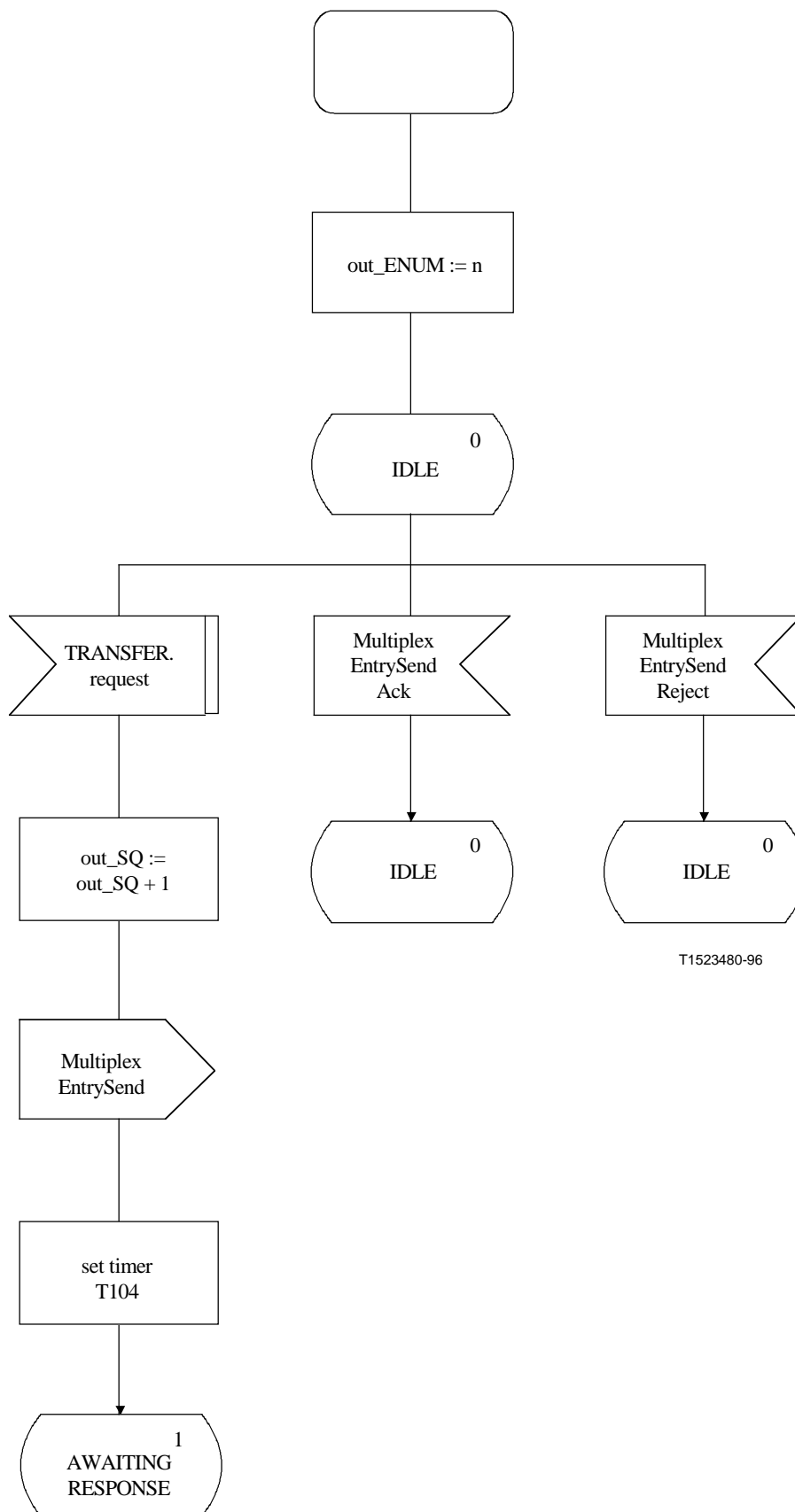


Figura 28.i/H.245 – Diagrama SDL de la MTSE de salida

Reemplazada por una versión más reciente

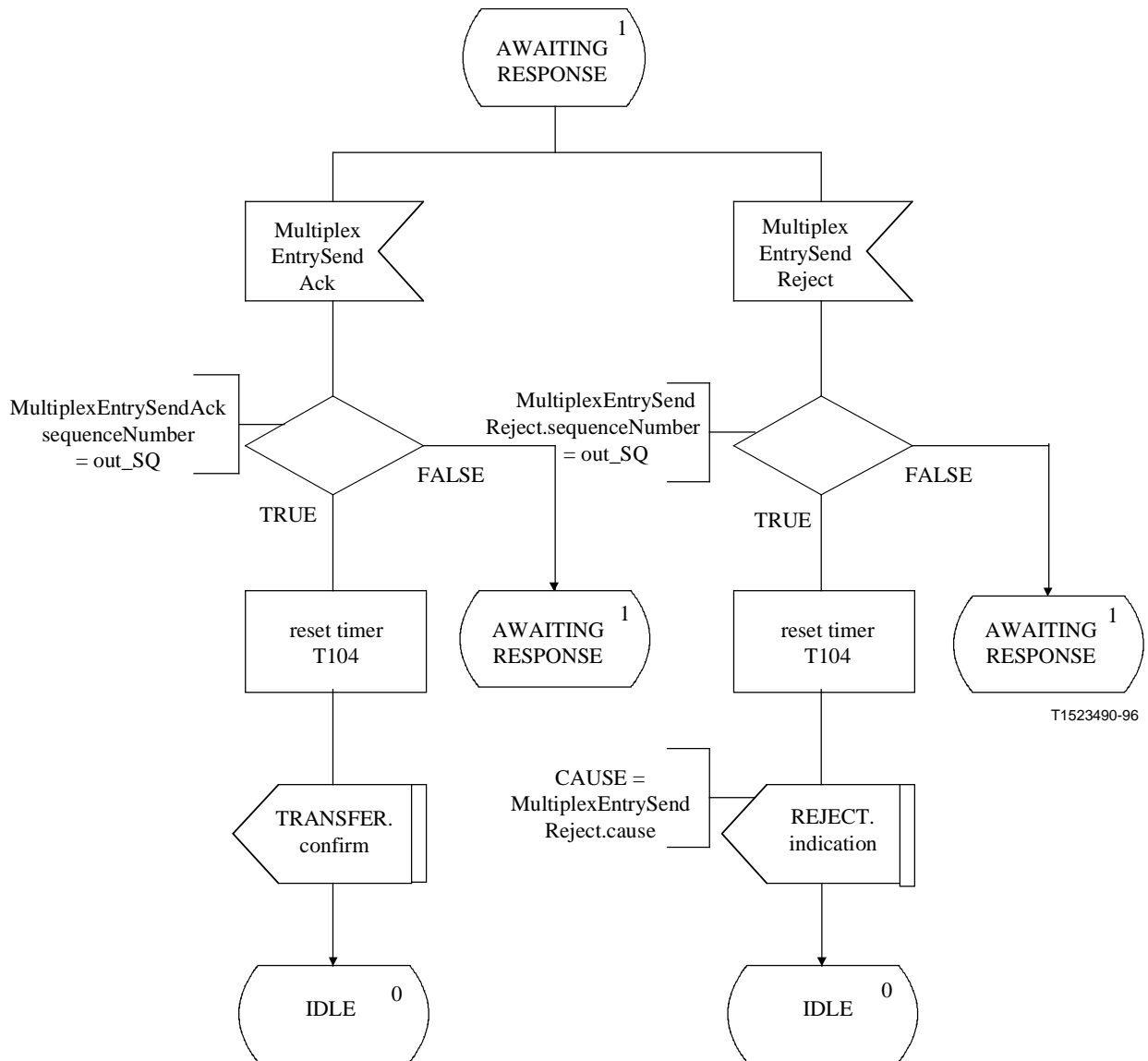


Figura 28.ii/H.245 – Diagrama SDL de la MTSE de salida (continuación)

Reemplazada por una versión más reciente

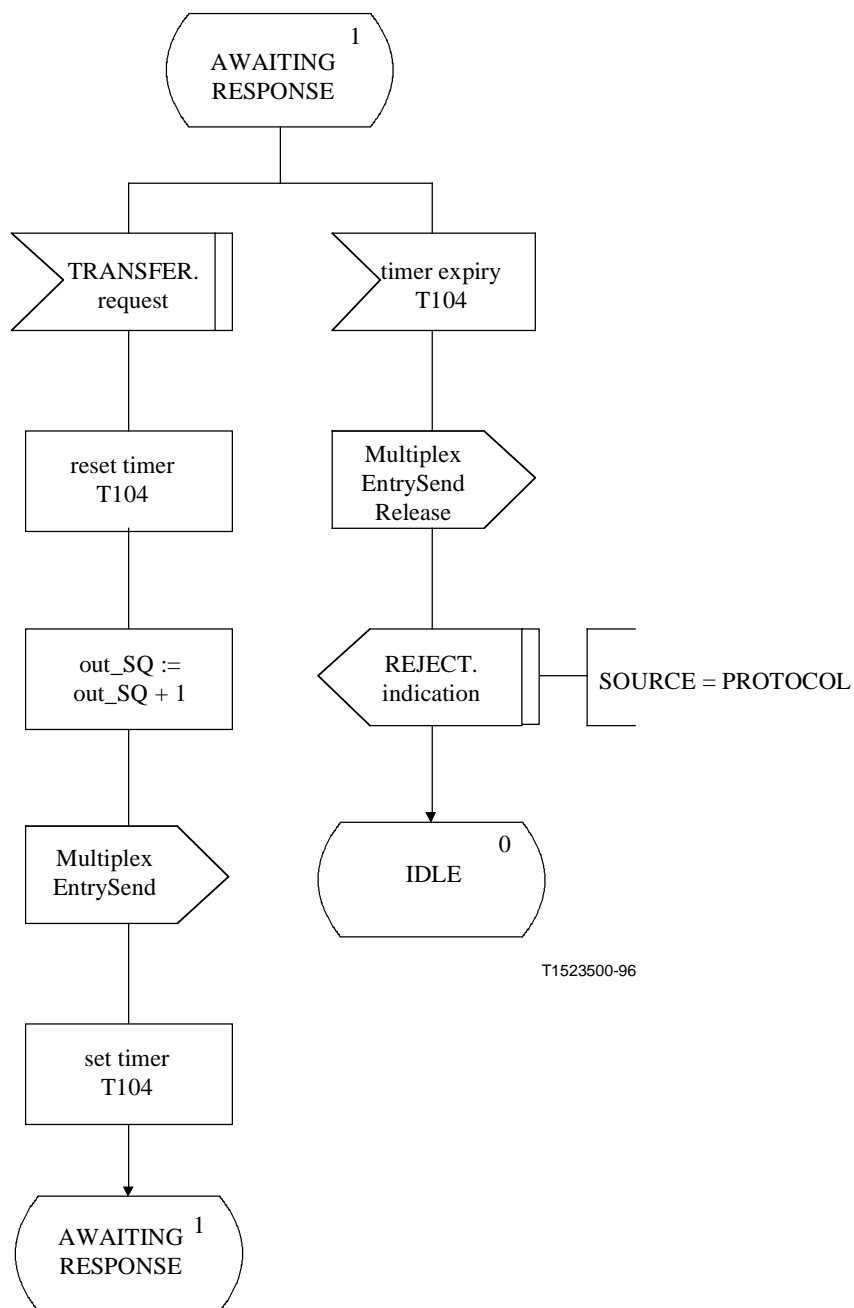


Figura 28.iii/H.245 – Diagrama SDL de la MTSE de salida (*fin*)

Reemplazada por una versión más reciente

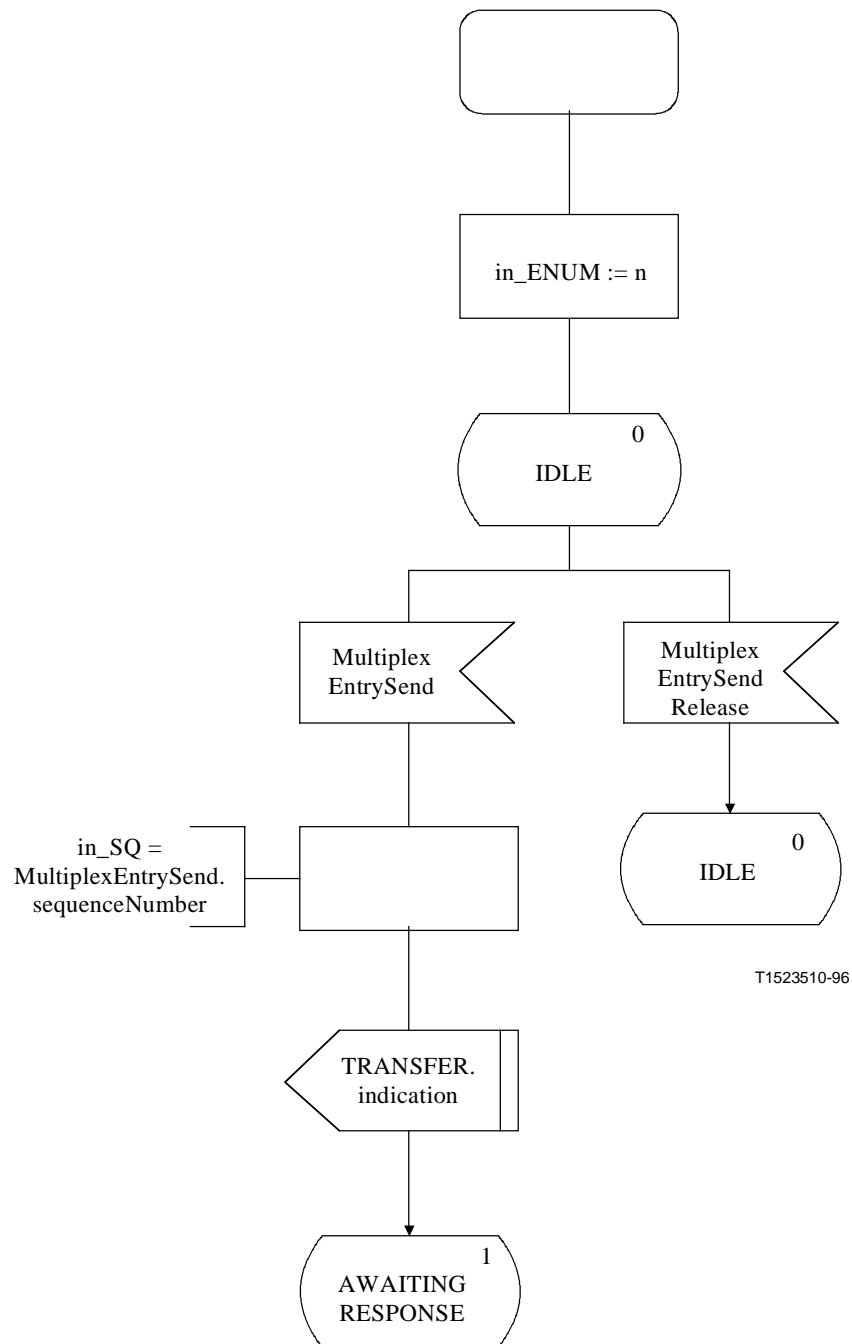


Figura 29.i/H.245 – Diagrama SDL de la MTSE de entrada

Reemplazada por una versión más reciente

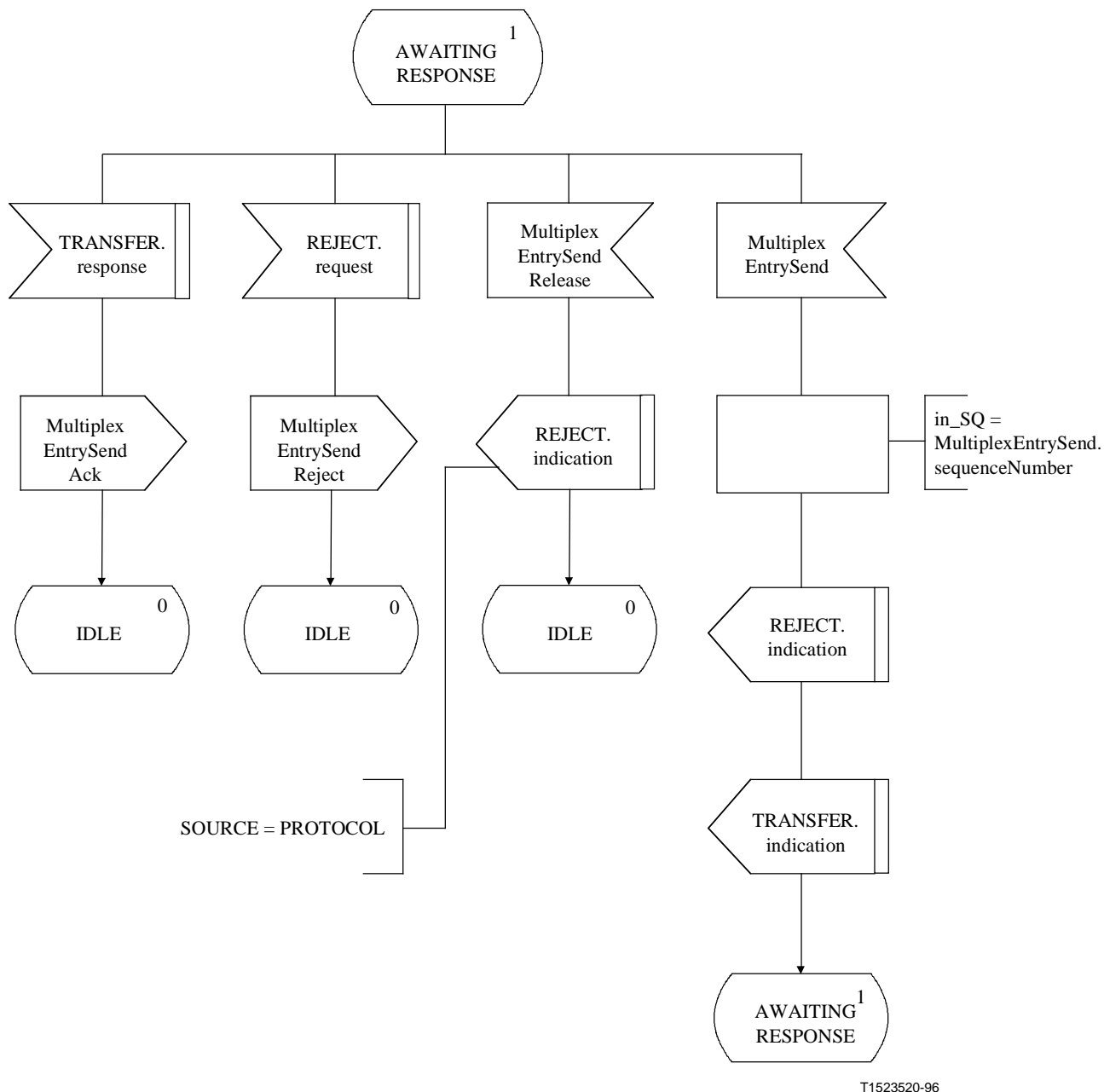


Figura 29.ii/H.245 – Diagrama SDL de la MTSE de entrada (*fin*)

8.8 Procedimientos de petición de entrada múltiplex

8.8.1 Introducción

Estos procedimientos los utiliza un terminal para solicitar la retransmisión de uno o más MultiplexEntryDescriptors. A estos procedimientos se hace referencia 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 la cláusula 6. Hay una RMESE de salida y una RMESE de entrada. Para cada entrada de tabla múltiplex hay una instancia de la RMESE.

Reemplazada por una versión más reciente

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.

8.8.1.1 Visión general del protocolo – RMESE de salida

Se inicia un procedimiento de petición de entrada múltiplex cuando en usuario en la RMESE de salida emite una primitiva ENVIO.petición. En 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.

8.8.1.2 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.

8.8.2 Comunicación entre la RMESE y el usuario de RMESE

8.8.2.1 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 46.

Reemplazada por una versión más reciente

Cuadro 46/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.

8.8.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- 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.

8.8.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 46 son:

- 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.
- El parámetro CAUSE indica el motivo por el cual se rehusa a enviar una entrada de tabla múltiplex. El parámetro CAUSE no está presente cuando el parámetro SOURCE indica "PROTOCOL".

8.8.2.4 Estados de la RMESE

Se utilizan los estados que siguen para especificar la secuencia admisible 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.

8.8.2.5 Diagramas de transición de estados

Se define aquí la secuencia admisible de primitivas entre la RMESE y el usuario de la RMESE. La secuencias admisibles se especifican separadamente para la RMESE de salida y la RMESE de entrada como se indica en las figuras 30 y 31, respectivamente.

Reemplazada por una versión más reciente

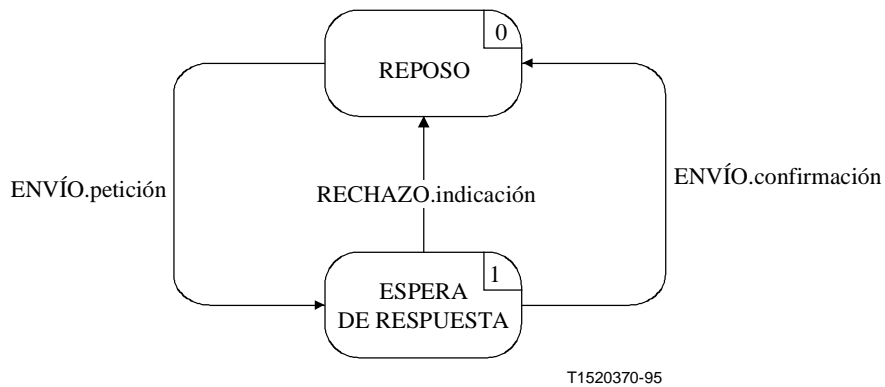


Figura 30/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una RMESE de salida

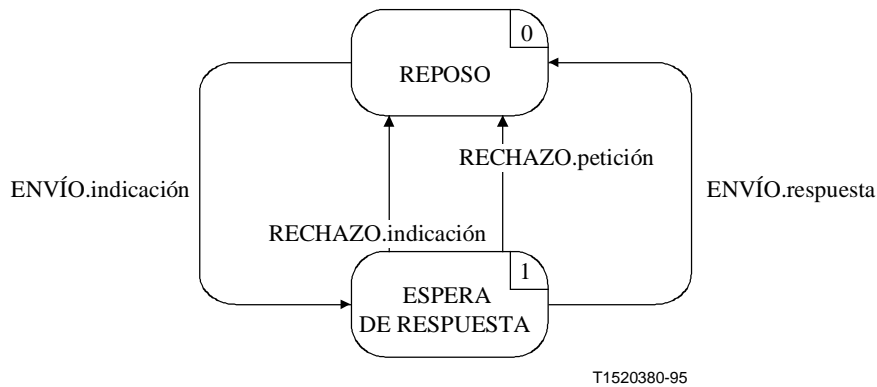


Figura 31/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una RMESE de entrada

8.8.3 Comunicación entre RMESE pares

8.8.3.1 Mensajes

En el cuadro 47 se representan los mensajes y campos de la RMESE, definidos en la cláusula 6 que son apropiados para el protocolo de la RMESE.

Reemplazada por una versión más reciente

Cuadro 47/H.245 – Nombres de mensajes y campos de la RMESE

función	mensaje	sentido	campo
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.			

8.8.3.2 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.

8.8.3.3 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 intervalo de tiempo que puede transcurrir que se reciba un mensaje RequestMultiplexEntryAck o RequestMultiplexEntryReject.

8.8.4 Procedimientos de RMESE

La figura 32 resume las primitivas de la RMESE y sus parámetros, y los mensajes, para la RMESE de salida y la de entrada.

Reemplazada por una versión más reciente

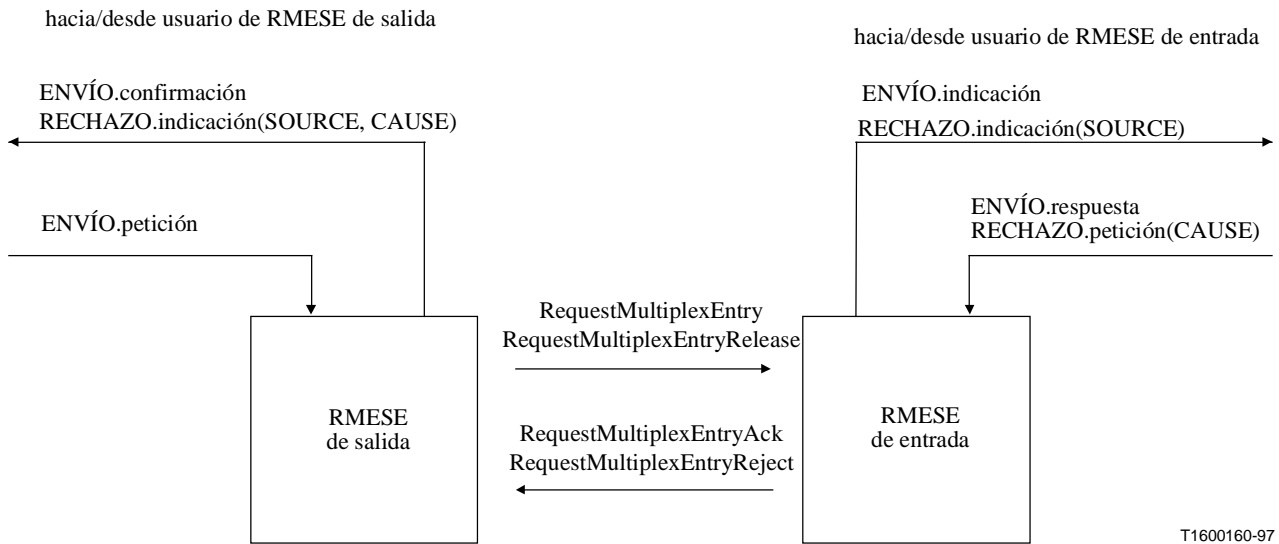


Figura 32/H.245 – Primitivas y mensajes en entidades de señalización de petición de entrada múltiplex

8.8.4.1 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 48.

Cuadro 48/H.245 – Valores supletorios de los parámetros de las primitivas

primitiva	parámetro	valor supletorio
RECHAZO.indicación	SOURCE	USER
	CAUSE	null

8.8.4.2 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 49.

Cuadro 49/H.245 – Valores supletorios de los campos de mensajes

mensajes	campo	valor supletorio
RequestMultiplexEntry	multiplexTableEntryNumber	out_ENUM
RequestMultiplexEntryAck	multiplexTableEntryNumber	in_ENUM
RequestMultiplexEntryReject	multiplexTableEntryNumber	in_ENUM
	cause	RECHAZO.petición(CAUSE)
RequestMultiplexEntryRelease	multiplexTableEntryNumber	out_ENUM

Reemplazada por una versión más reciente

8.8.4.3 Diagramas SDL

Los procedimientos de la RMESE de salida y la RMESE de entrada se expresan en forma de diagramas SDL en las figuras 33 y 34, respectivamente.

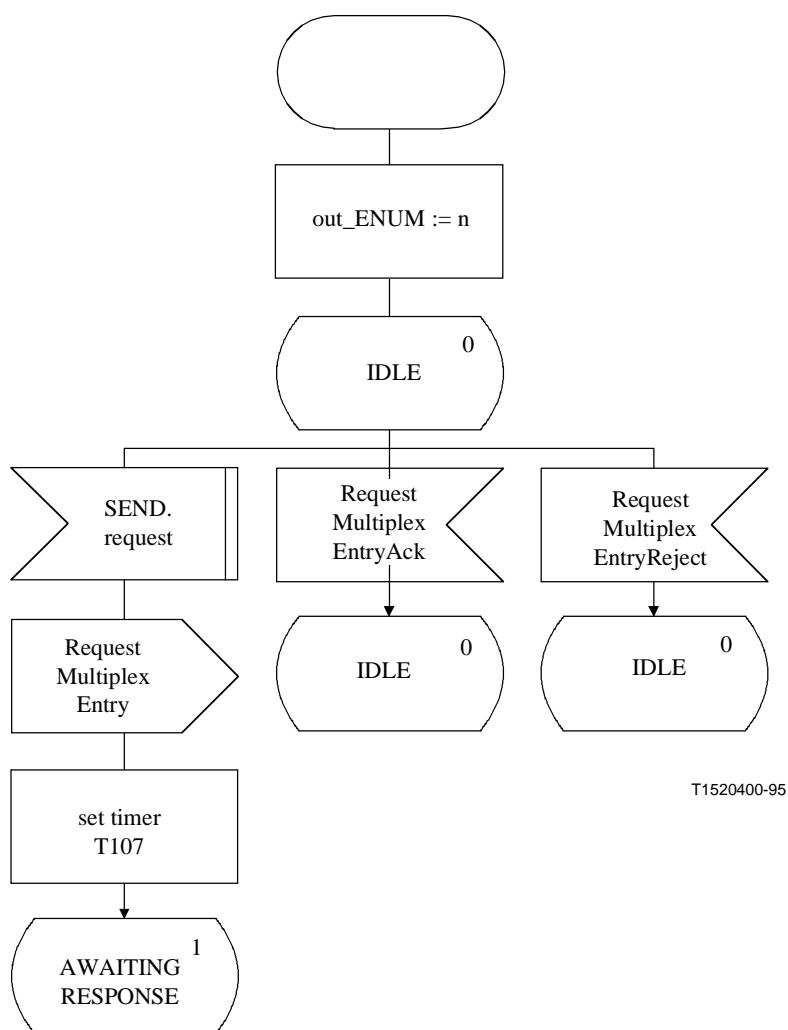


Figura 33.i/H.245 – Diagrama SDL de la RMESE de salida

Reemplazada por una versión más reciente

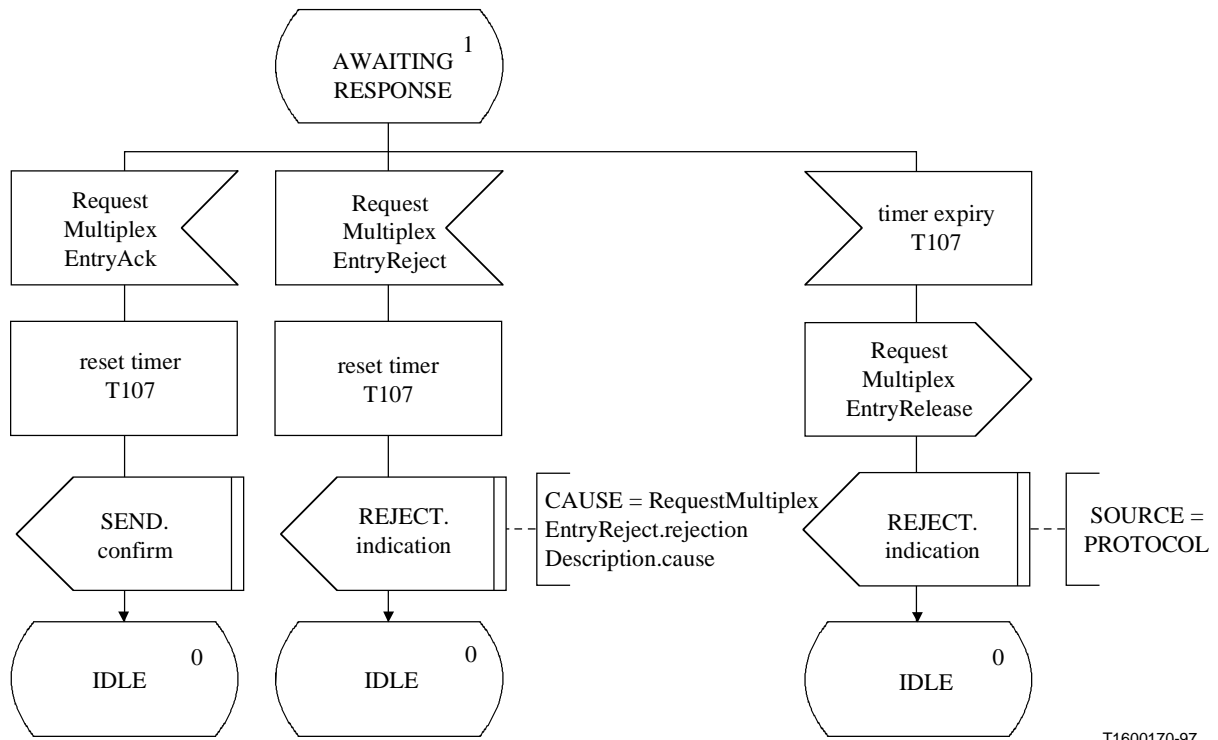


Figura 33.ii/H.245 – Diagrama SDL de la RMESE de salida (*fin*)

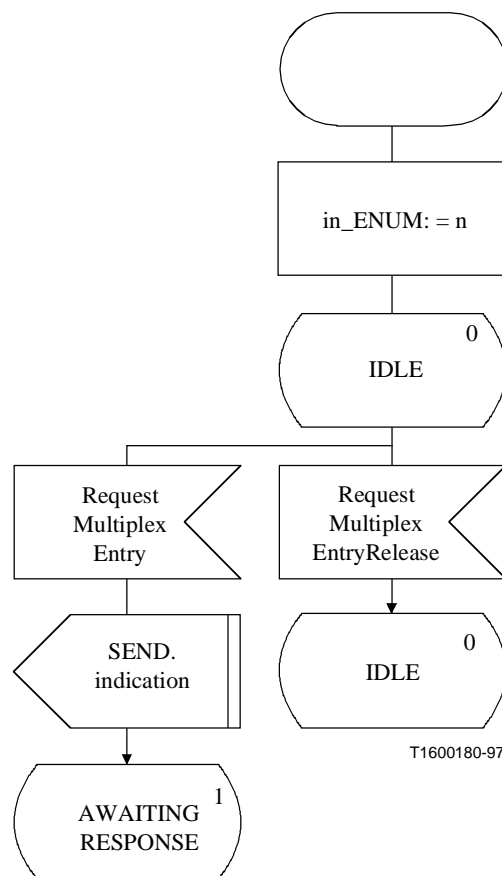


Figura 34.i/H.245 – Diagrama SDL de la RMESE de entrada

Reemplazada por una versión más reciente

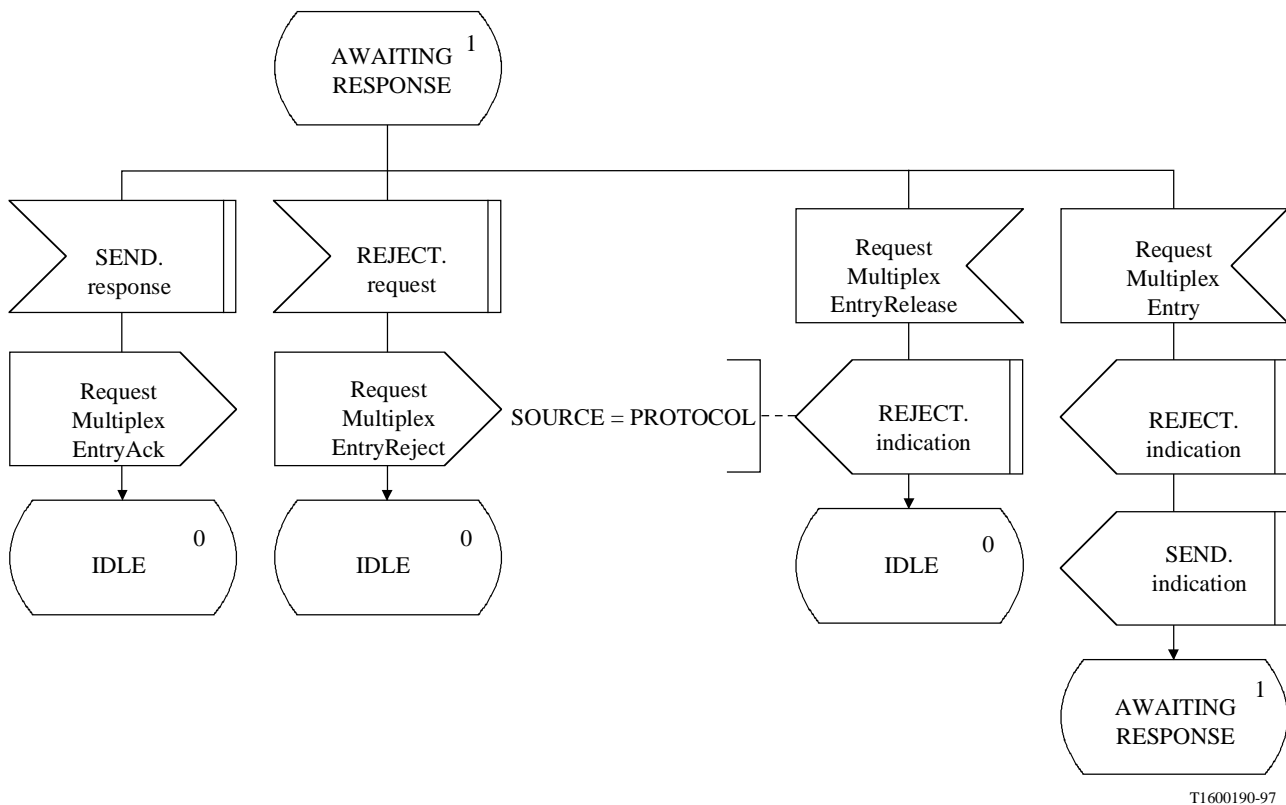


Figura 34.ii/H.245 – Diagrama SDL de la RMESE de entrada (fin)

8.9 Procedimientos de petición de modo

8.9.1 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. Se transfiere la información de protocolo a la MRSE por mediante los mensajes apropiados definidos en la cláusula 6. Hay una MRSE de salida y una MRSE de entrada. En el extremo de salida y en el extremo de entrada hay una instancia (o 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

Reemplazada por una versión más reciente

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.

NOTA – La petición de descripción de modo especifica un modo completo. 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.

8.9.1.1 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`.

8.9.1.2 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 múltiple.

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.

Reemplazada por una versión más reciente

8.9.2 Comunicación entre la MRSE y el usuario de MRSE

8.9.2.1 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 50.

Cuadro 50/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.

8.9.2.2 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.

8.9.2.3 Definiciones de los parámetros

Las definiciones de los parámetros de las primitivas del cuadro 50 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-ELEMENTS asociados con las primitivas TRANSFERENCIA.
- El parámetro MODE-PREF informa al usuario sobre si se utilizará o no el modo más 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".

8.9.2.4 Estados de la MRSE

Se utilizan los siguientes estados para especificar la secuencia admisible 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.

Reemplazada por una versión más reciente

Para la MRSE de entrada los estados son:

Estado 0: REPOSO

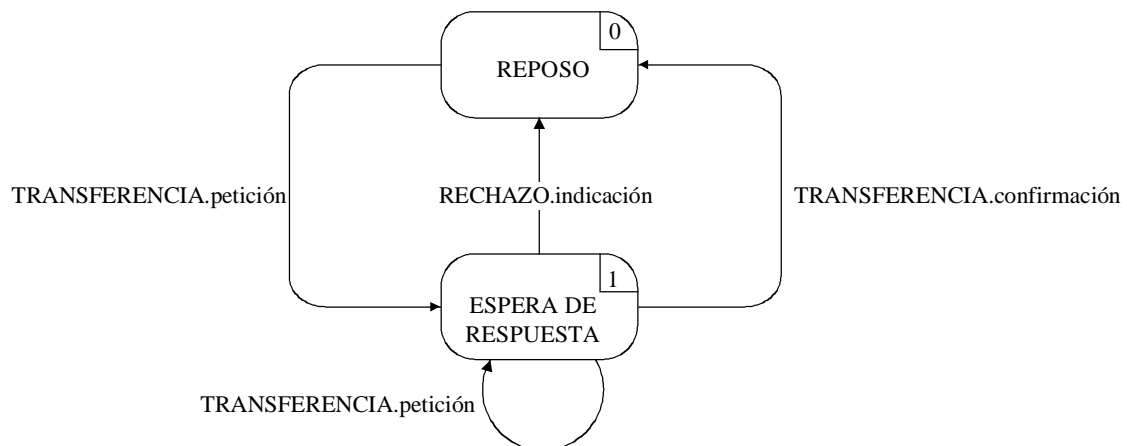
La MRSE está en reposo.

Estado 1: ESPERA DE RESPUESTA

La MRSE espera una respuesta del usuario de la MRSE.

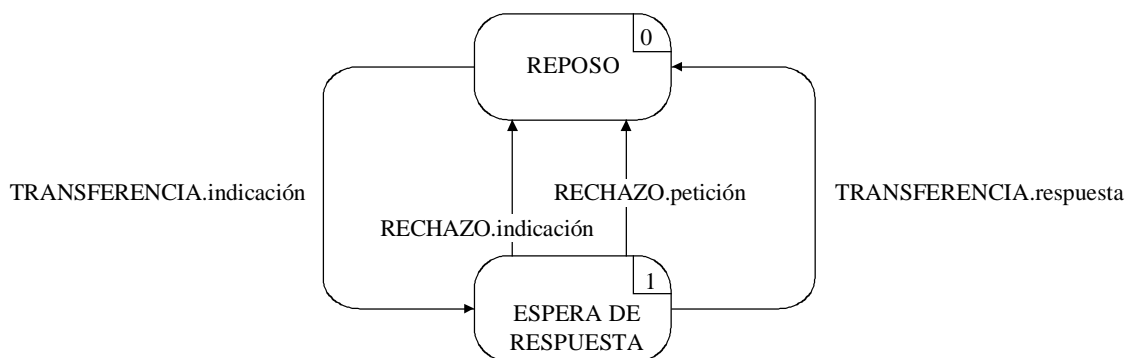
8.9.2.5 Diagramas de transición de estados

Se define aquí la secuencia admisible de primitivas entre la MRSE y el usuario de la MRSE. Las secuencias admisibles se especifican separadamente para la MRSE de salida y la MRSE de entrada como se indica en las figuras 35 y 36, respectivamente.



T1519480-95

Figura 35/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MRSE de salida



T1519490-95

Figura 36/H.245 – Diagrama de transición de estados para la secuencia de primitivas en una MRSE de entrada

Reemplazada por una versión más reciente

8.9.3 Comunicación entre MRSE pares

8.9.3.1 Mensajes

En el cuadro 51 se muestran los mensajes y campos de la MRSE, definidos en la cláusula 6 que son apropiados para el protocolo de la MRSE.

Cuadro 51/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.

8.9.3.2 Variables de estado de la MRSE

En la MRSE de salida se definen las siguiente variables 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.

8.9.3.3 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.

8.9.4 Procedimientos de MRSE

En la figura 37 se resumen las primitivas de la MRSE, sus parámetros y los mensajes para las MRSE de salida y de entrada.

Reemplazada por una versión más reciente

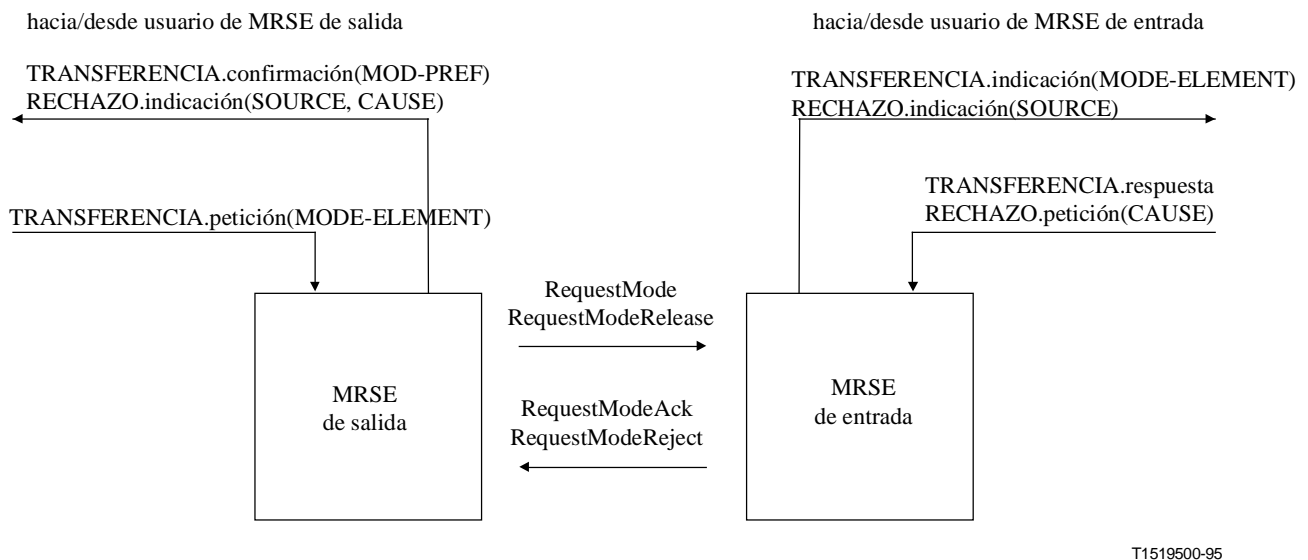


Figura 37/H.245 – Primitivas y mensajes en la entidad de señalización petición de modo

8.9.4.1 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 52.

Cuadro 52/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	USER
	CAUSE	nulo

8.9.4.2 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 53.

Cuadro 53/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	–	–

Reemplazada por una versión más reciente

8.9.4.3 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 38 y 39 respectivamente.

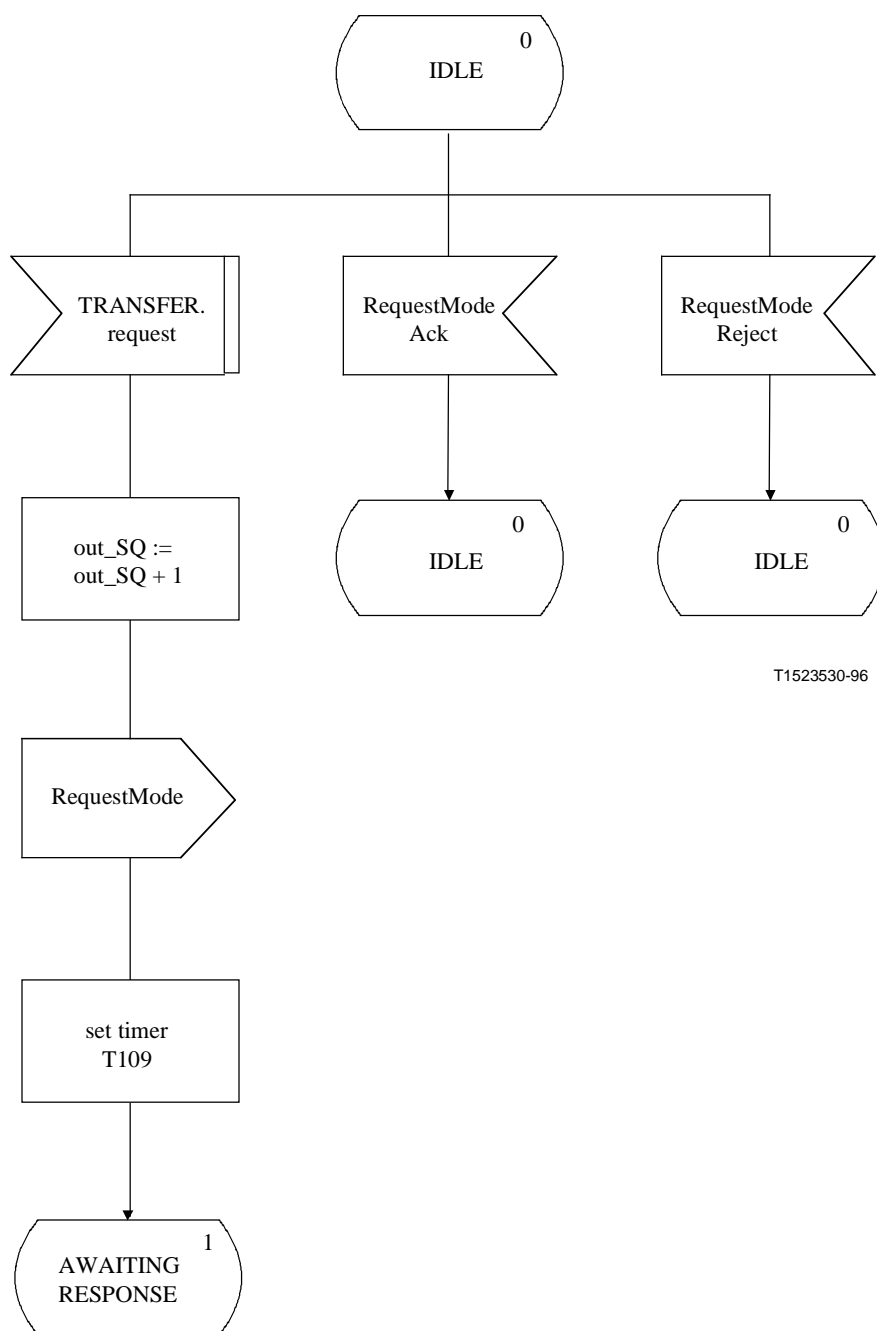


Figura 38.i/H.245 – Diagrama SDL de la MRSE de salida

Reemplazada por una versión más reciente

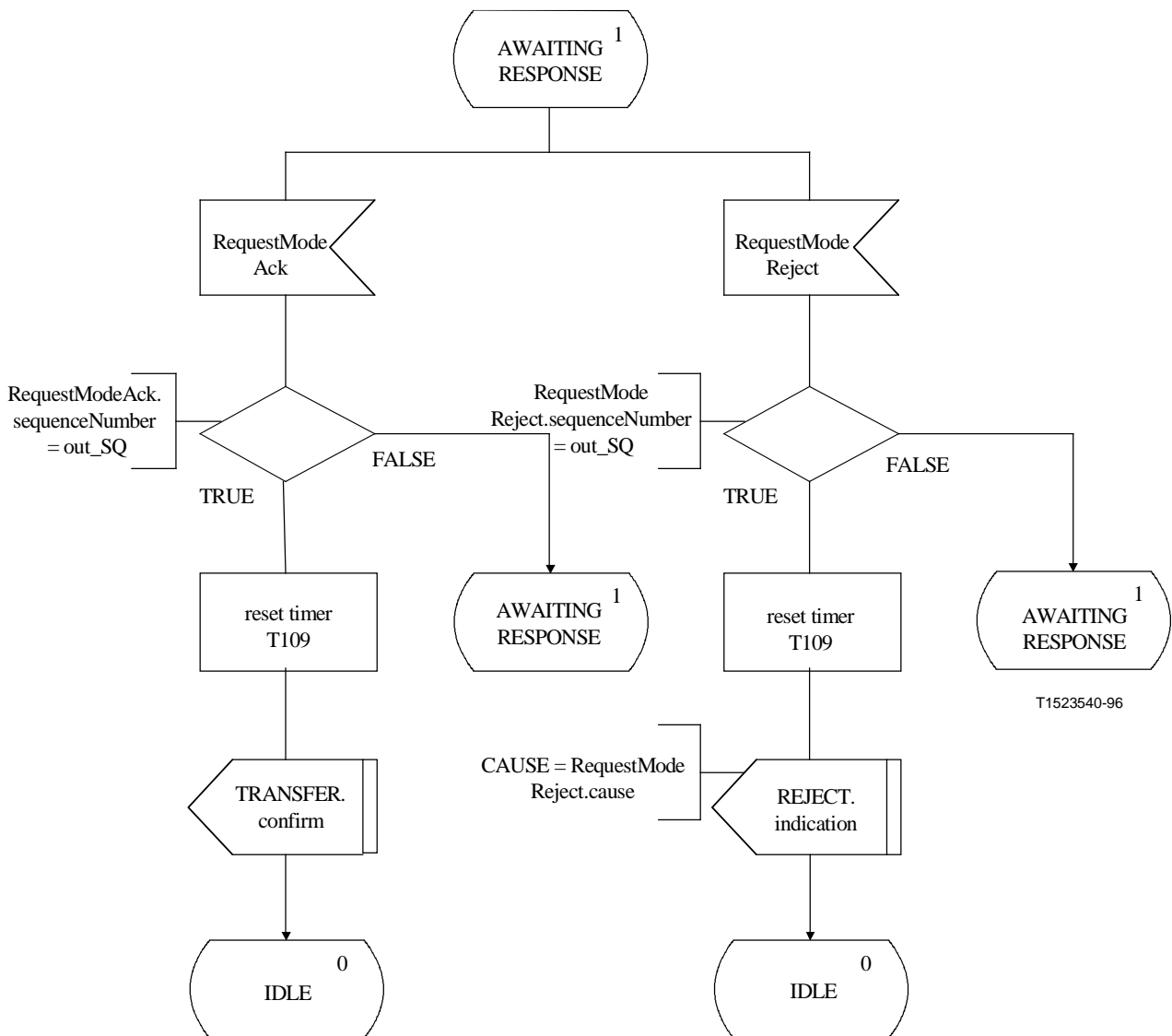
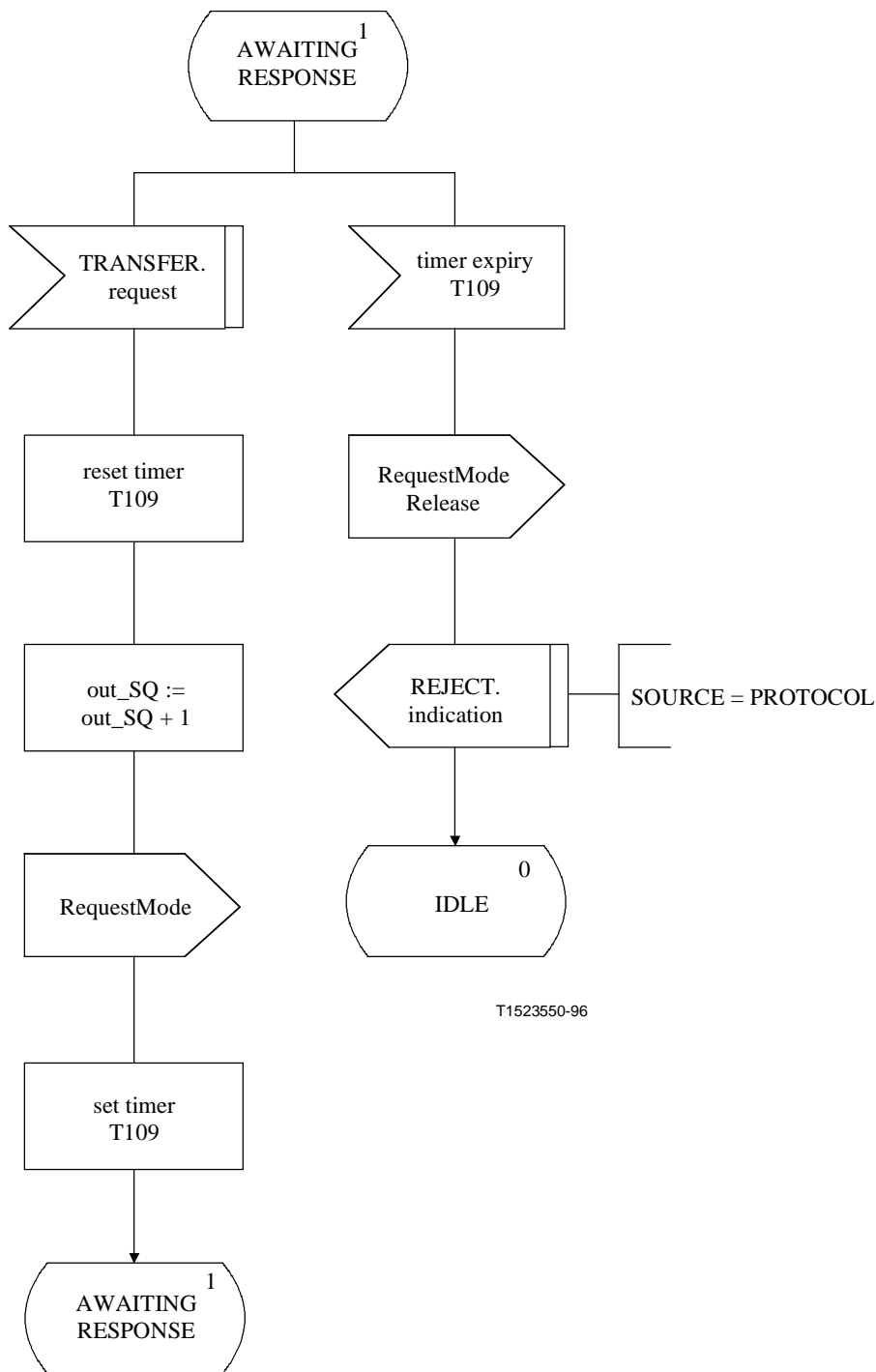


Figura 38.ii/H.245 – Diagrama SDL de la MRSE de salida (continuación)

Reemplazada por una versión más reciente



T1523550-96

Figura 38.iii/H.245 – Diagrama SDL de la MRSE de salida (fin)

Reemplazada por una versión más reciente

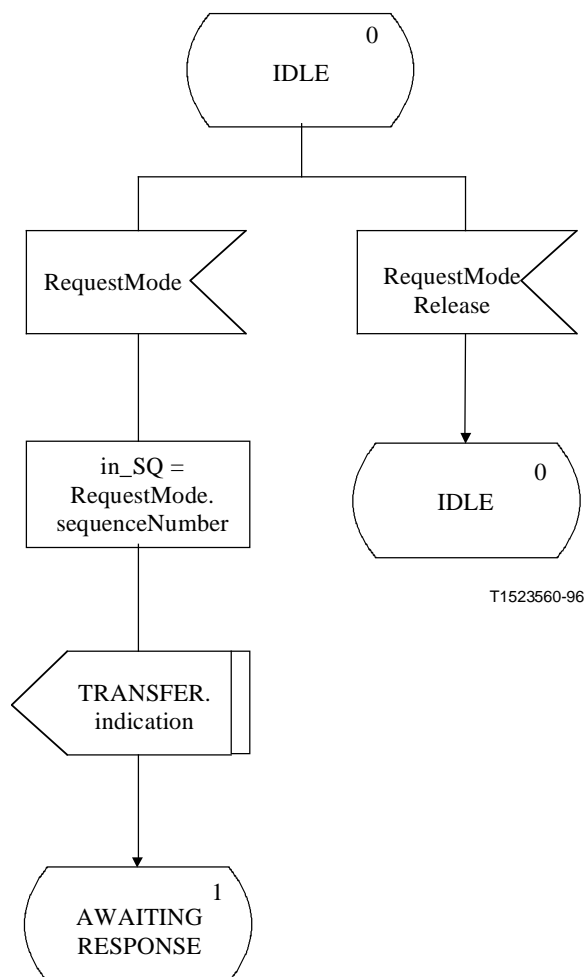


Figura 39.i/H.245 – Diagrama SDL de la MRSE de entrada

Reemplazada por una versión más reciente

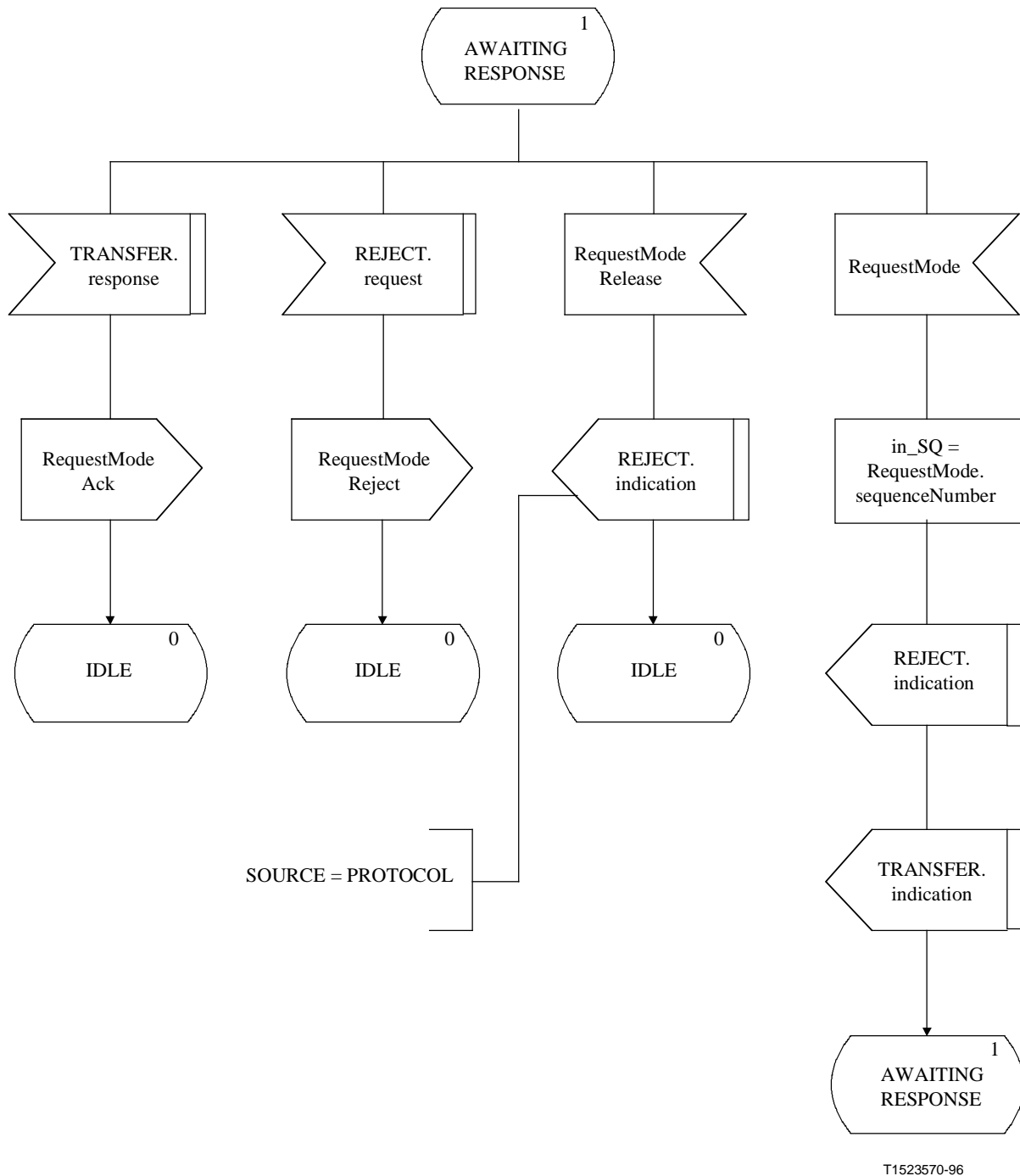


Figura 39.ii/H.245 – Diagrama SDL de la MRSE de entrada (fin)

8.10 Procedimientos de retardo de ida y vuelta

8.10.1 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

Reemplazada por una versión más reciente

en la interfaz entre la RTDSE y el usuario de la RTDSE. En cada terminal hay una instancia 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.

8.10.1.1 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 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.

8.10.2 Comunicación entre la RTDSE y el usuario de RTDSE

8.10.2.1 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 54. Estas primitivas tienen como finalidad la definición de procedimientos en la RTDSE y no se pretende que especifiquen o limiten la implementación.

Cuadro 54/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.

8.10.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- Se utiliza la primitiva TRANSFERENCIA para solicitar la determinación del retardo de ida y vuelta y notificar este retardo.
- La primitiva EXPIRACIÓN indica que no se ha recibido respuesta desde el terminal par.

Reemplazada por una versión más reciente

8.10.2.3 Definiciones de los parámetros

La definición de los parámetros de las primitivas del cuadro 54 es la siguiente:

- El parámetro DELAY proporciona el valor medido del retardo de ida y vuelta.

8.10.2.4 Estados de la RTDSE

Para especificar la secuencia admisible 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.

8.10.2.5 Diagramas de transición de estados

Se define aquí la secuencia admisible de primitivas entre la RTDSE y el usuario de la RTDSE. En la figura 40 se muestra la secuencia admisible.

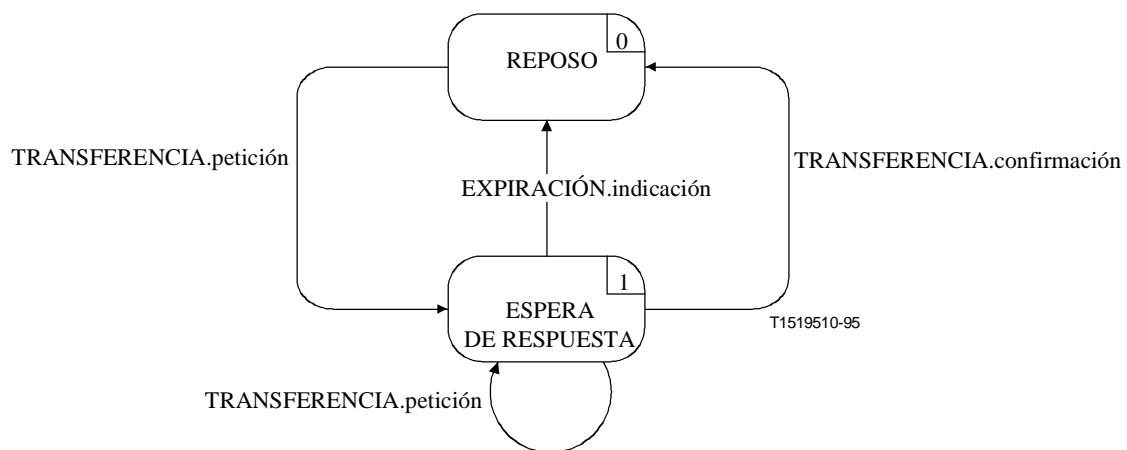


Figura 40/H.245 – Diagrama de transición de estados para la secuencia de primitivas en la RTDSE

8.10.3 Comunicación entre RTDSE pares

8.10.3.1 Mensajes

En el cuadro 55 se muestran los mensajes y campos de la RTDSE, definidos en la cláusula 6, que son apropiados para el protocolo de la RTDSE.

Cuadro 55/H.245 – Nombres de mensajes y campos de la RTDSE

función	mensaje	campo
transferencia	RoundTripDelayRequest	sequenceNumber
	RoundTripDelayResponse	sequenceNumber

Reemplazada por una versión más reciente

8.10.3.2 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.

8.10.3.3 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.

8.10.4 Procedimientos de RTDSE

8.10.4.1 Introducción

En la figura 41 se resumen las primitivas de la RTDSE y sus parámetros, y mensajes.

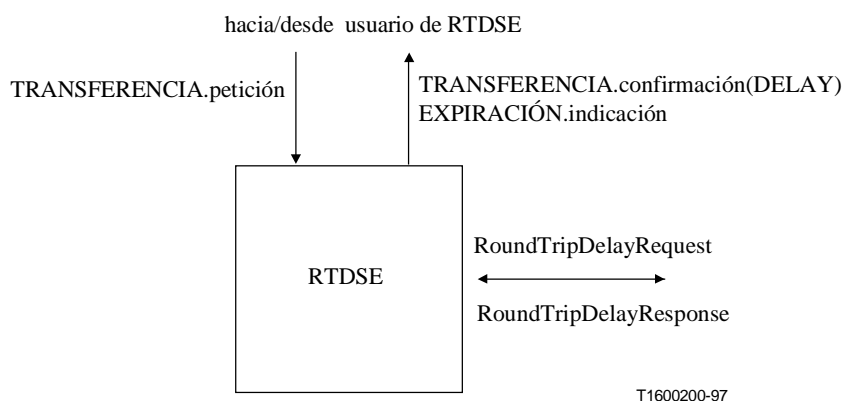


Figura 41/H.245 – Primitivas y mensajes en la RTDSE

8.10.4.2 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 56.

Cuadro 56/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 que 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.

Reemplazada por una versión más reciente

Reemplazada por una versión más reciente

8.10.4.3 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 57.

Cuadro 57/H.245 – Valores supletorios de los campos de los mensajes

mensaje	campo	valor supletorio
RoundTripDelayRequest	sequenceNumber	out_SQ
RoundTripDelayResponse	sequenceNumber	RoundTripDelayRequest.sequenceNumber

8.10.4.4 Diagramas de SDL

En la figura 42 se representan, mediante diagramas de SDL, los procedimientos en la RTDSE.

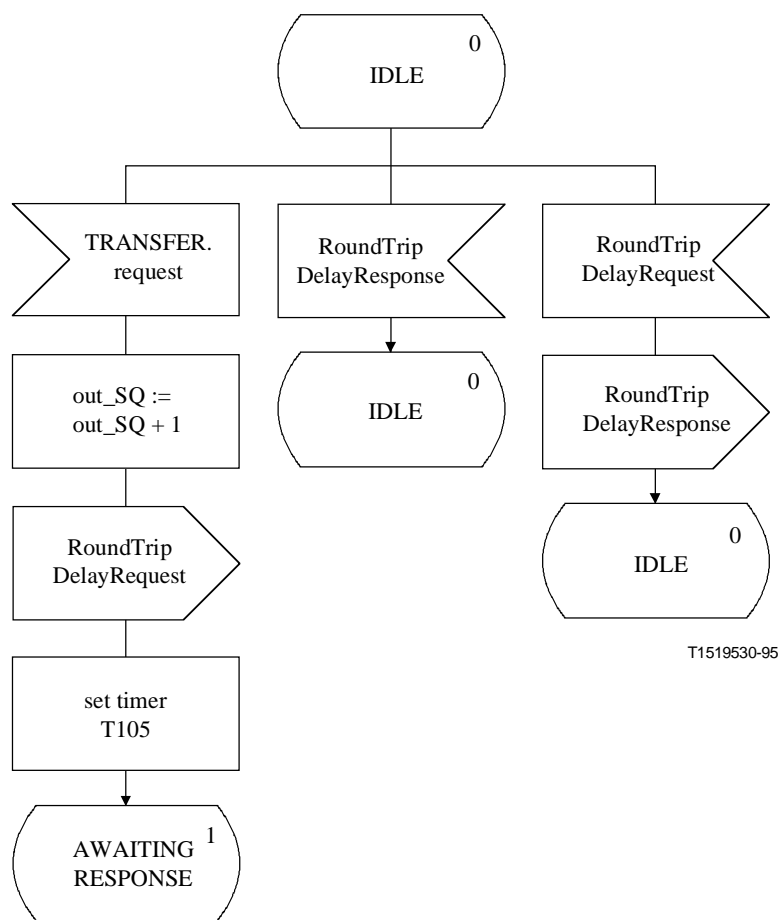


Figura 42.i/H.245 – Diagrama SDL de la RTDSE

Reemplazada por una versión más reciente

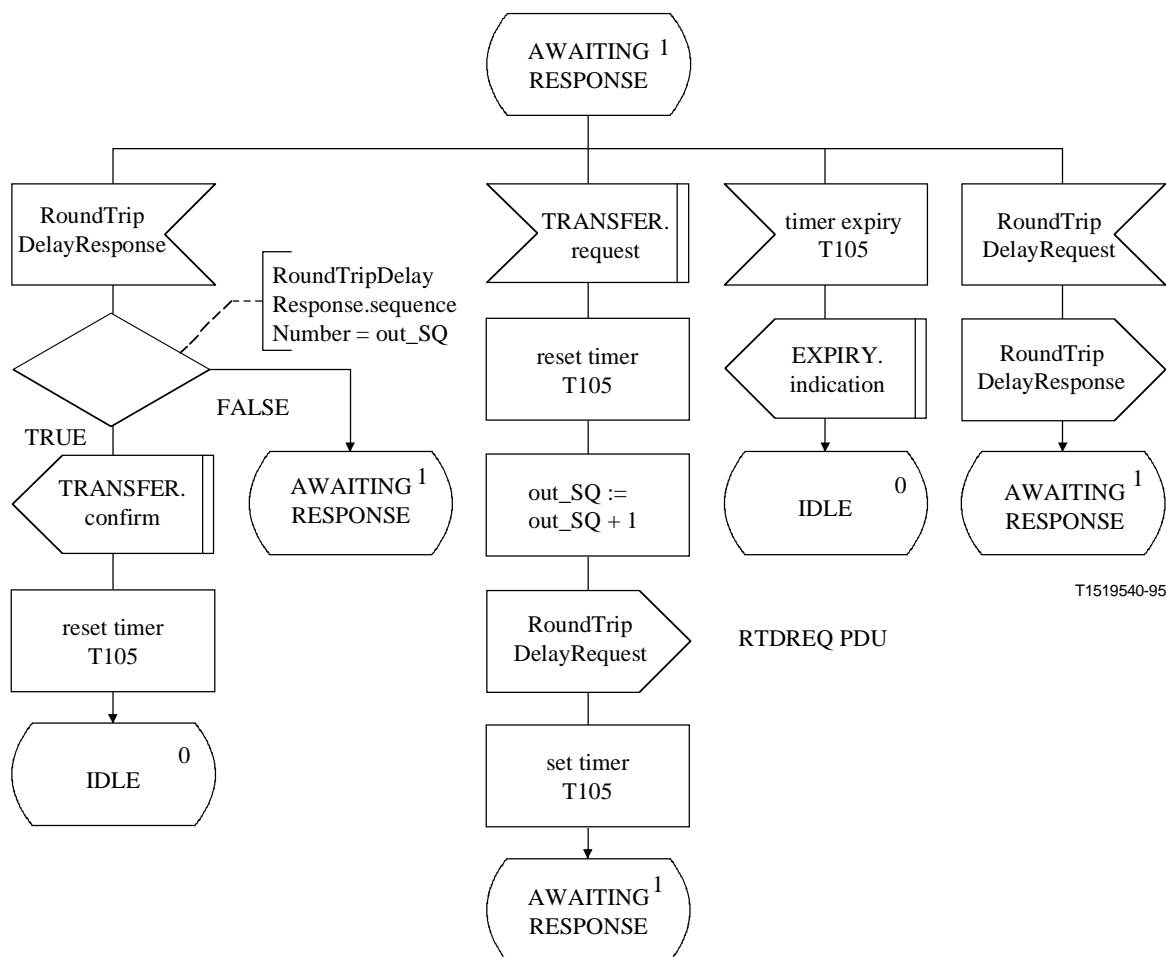


Figura 42.ii/H.245 – Diagrama de SDL de la RTDSE (fin)

8.11 Procedimientos de bucle de mantenimiento

8.11.1 Introducción

El procedimiento 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 por medio de mensajes apropiados definidos en la cláusula 6.

Hay una MLSE de salida y una MLSE de entrada. En el lado de salida y en el lado de entrada hay una instancia 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.

Reemplazada por una versión más reciente

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.

8.11.1.1 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 BUCLE.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 LIBERACION.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 LIBERACION.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 LIBERACION.petición.

8.11.1.2 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 LIBERACION.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 LIBERACION.indicación.

8.11.2 Comunicación entre la MLSE y el usuario de MLSE

8.11.2.1 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 58.

Reemplazada por una versión más reciente

Cuadro 58/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.				

8.11.2.2 Definiciones de las primitivas

Las definiciones de estas primitivas son:

- Las primitivas BUCLE se utilizan para establecer un bucle de mantenimiento.
- Las primitivas LIBERACION se utilizan para anular un bucle de mantenimiento.
- La primitiva ERROR informa de errores de la MLSE a una entidad de gestión.

8.11.2.3 Definiciones de los parámetros

Las definiciones de los parámetros indicados en el cuadro 58 son:

- 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.
- 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.
- 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".
- El parámetro ERRCODE indica el tipo de error de la MLSE. En el cuadro 62 se indican los valores permitidos del parámetro ERRCODE.

8.11.2.4 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.

Reemplazada por una versión más reciente

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.

8.11.2.5 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 43 y 44, respectivamente.

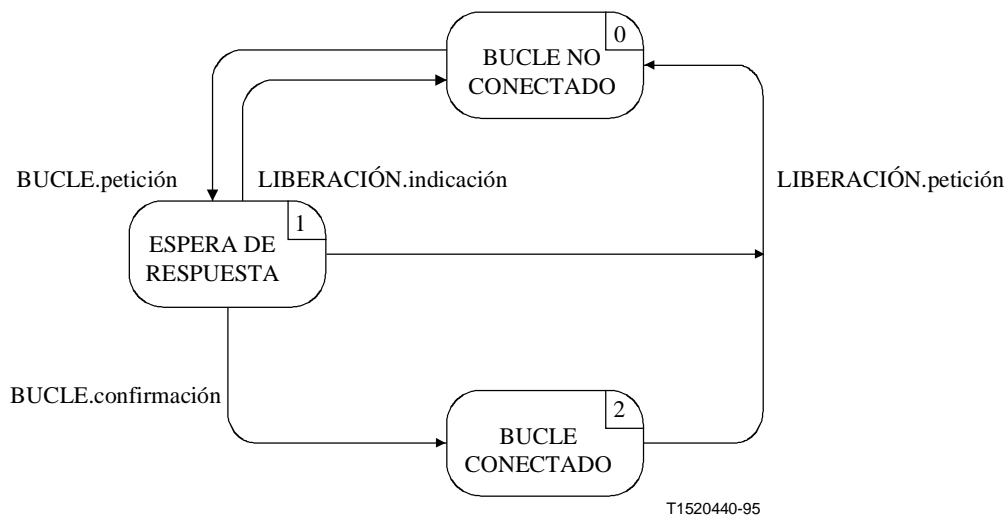


Figura 43/H.245 –Diagrama de transición de estados para la secuencia de primitivas en la MLSE de salida

Reemplazada por una versión más reciente

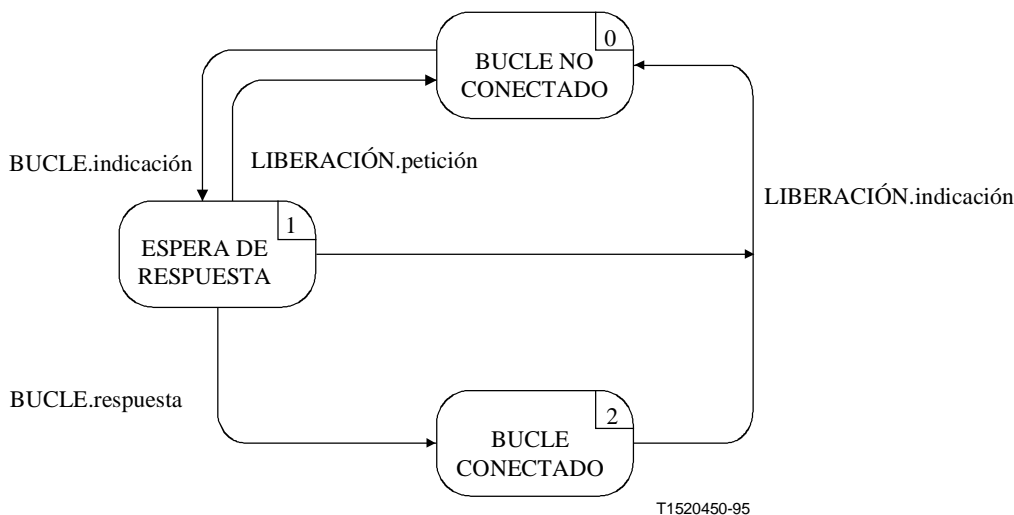


Figura 44/H.245 – Diagrama de transición de estados para la secuencia de primitivas en la MLSE de entrada

8.11.3 Comunicación entre MLSE pares

8.11.3.1 Mensajes de las MLSE

El cuadro 59 muestra los mensajes y campos, definidos en la cláusula 6, que son apropiados para el protocolo MLSE.

Cuadro 59/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.			

8.11.3.2 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.

Reemplazada por una versión más reciente

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.

8.11.3.3 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.

8.11.4 Procedimientos de la MLSE

8.11.4.1 Introducción

En la figura 45 se resumen las primitivas y sus parámetros, y los mensajes, para la MLSE de salida y la MLSE de entrada.

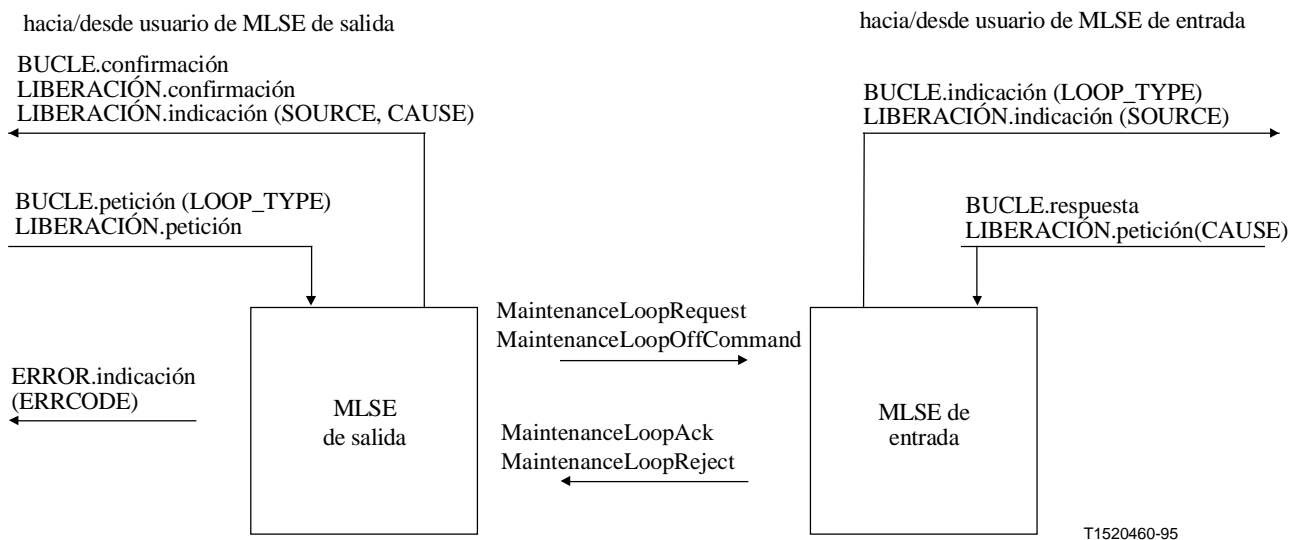


Figura 45/H.245 – Primitivas y mensajes en la entidad de señalización de bucle de mantenimiento

8.11.4.2 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 60.

Reemplazada por una versión más reciente

Cuadro 60/H.245 – Valores supletorios de los parámetros de las primitivas

primitiva	parámetro	valor supletorio (nota)
BUCLE.indicación	LOOP_TYPE	MaintenanceLoopRequest.type
LIBERACIÓN.indicación	SOURCE	USER
	CAUSE	MaintenanceLoopReject.cause
NOTA – Un parámetro de una primitiva se codificará como nulo si un campo de mensaje indicado no está presente en el mensaje.		

8.11.4.3 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 61.

Cuadro 61/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 y in_MLN (nota 3)
MaintenanceLoopReject	type	in_LOOP y in_MLN (nota 3)
	cause	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.		

8.11.4.4 Valores del parámetro ERRCODE

El parámetro ERRCODE de la primitiva ERROR.indicación indica una condición de error dada. En el cuadro 62 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 62/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

Reemplazada por una versión más reciente

8.11.4.5 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 46 y 47, respectivamente.

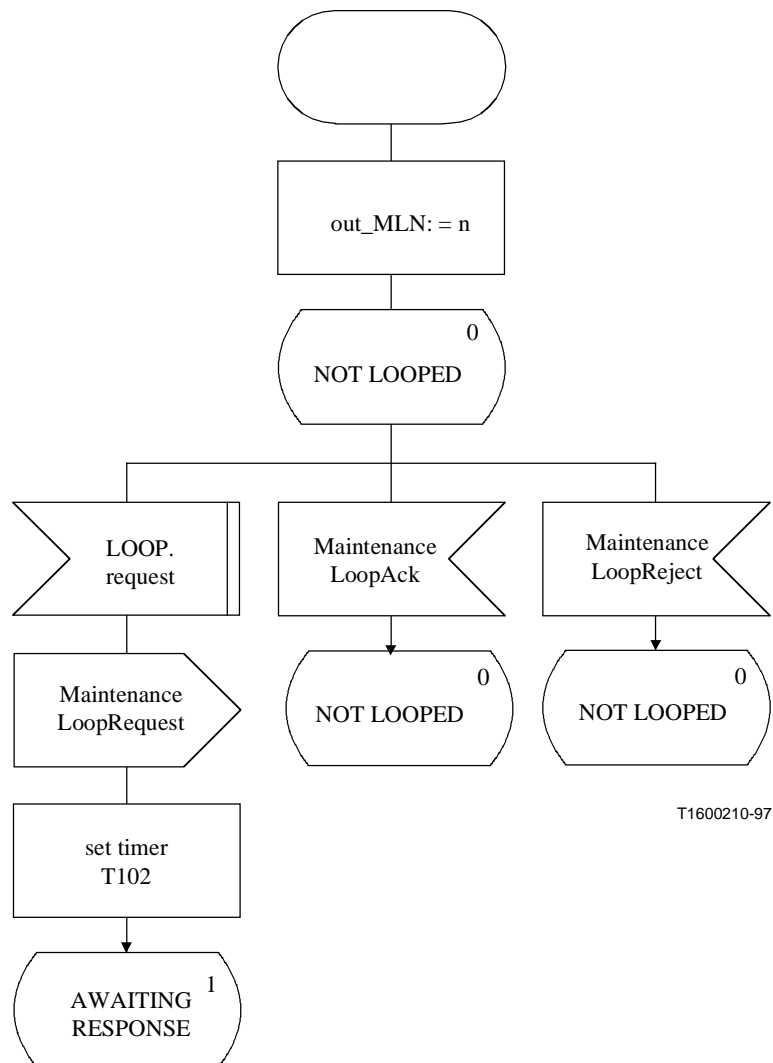


Figura 46.i/H.245 – Diagrama SDL de la MLSE de salida

Reemplazada por una versión más reciente

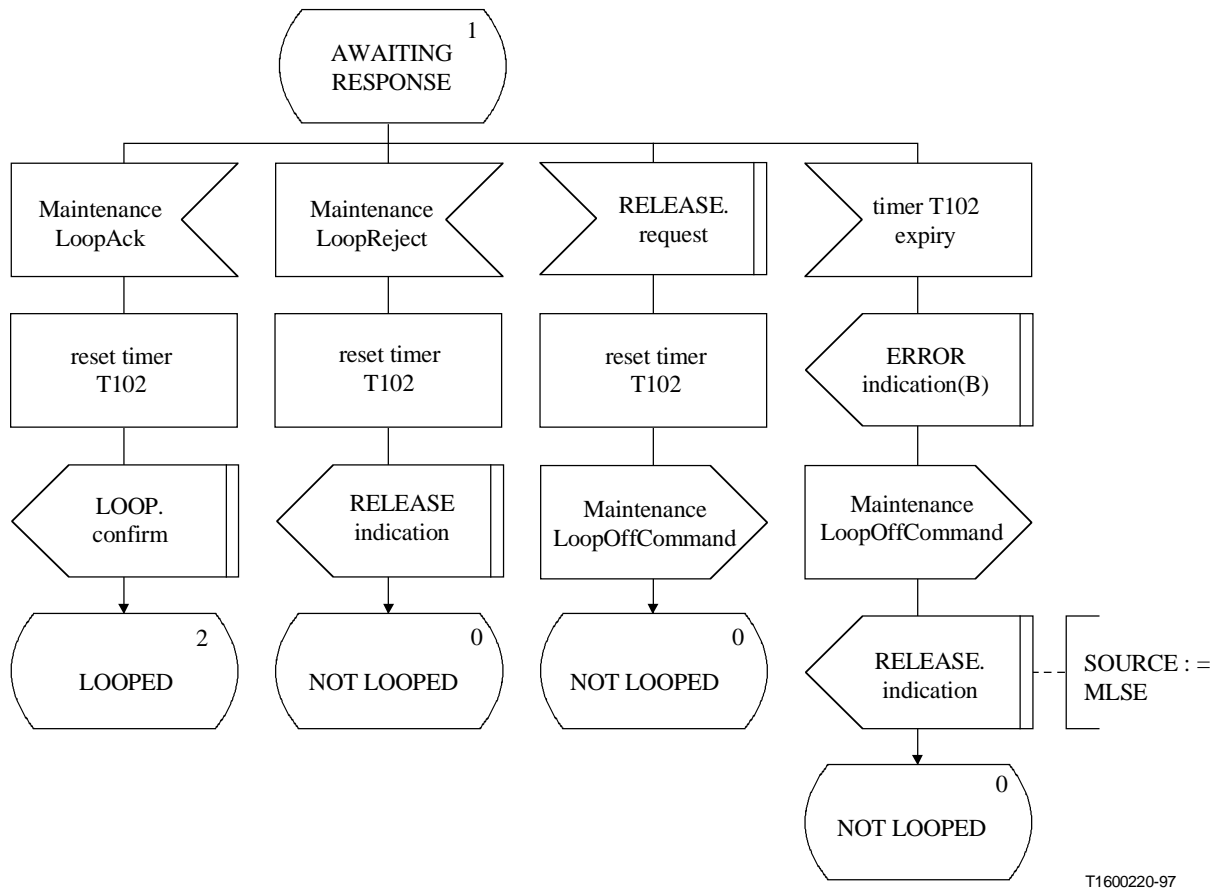


Figura 46.ii/H.245 – Diagrama SDL de la MLSE de salida (continuación)

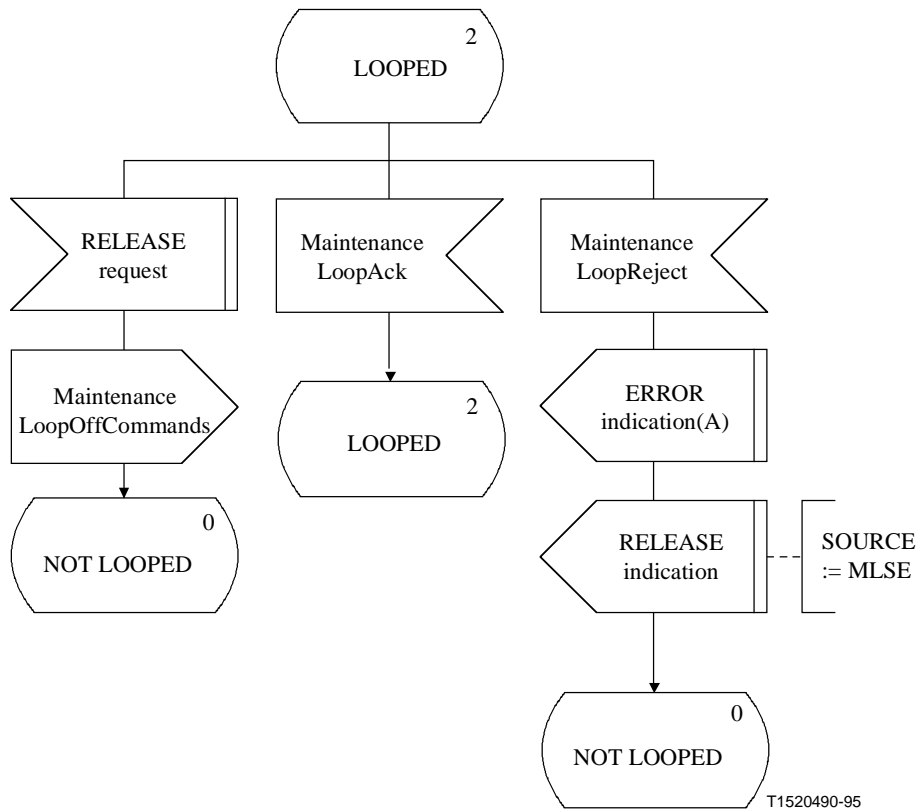


Figura 46.iii/H.245 – Diagrama SDL de la MLSE de salida (fin)

Reemplazada por una versión más reciente

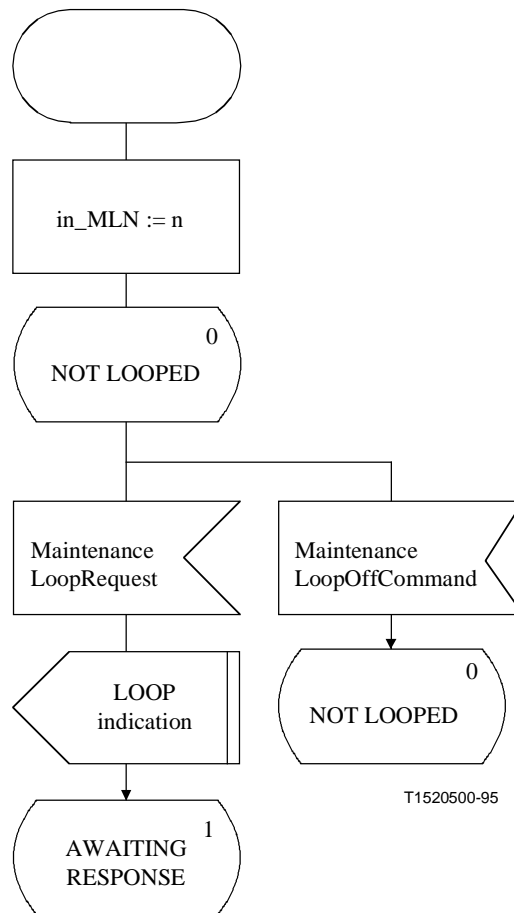


Figura 47.i/H.245 – Diagrama SDL de la MLSE de entrada

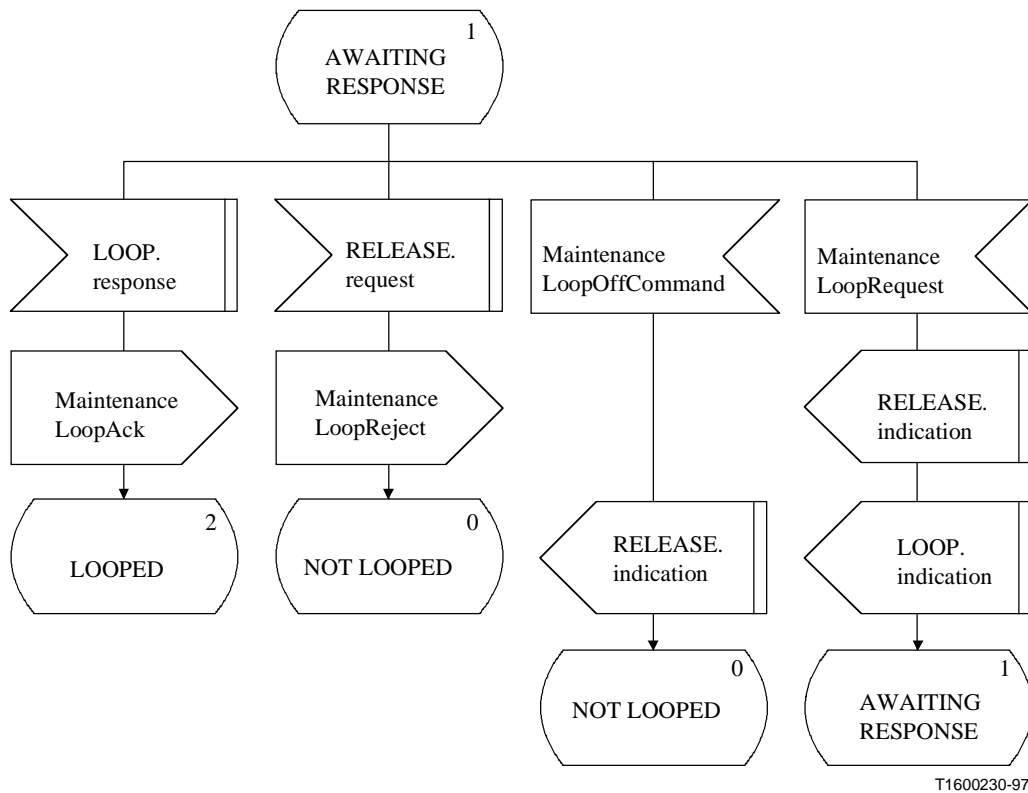


Figura 47.ii/H.245 – Diagrama SDL de la MLSE de entrada (continuación)

Reemplazada por una versión más reciente

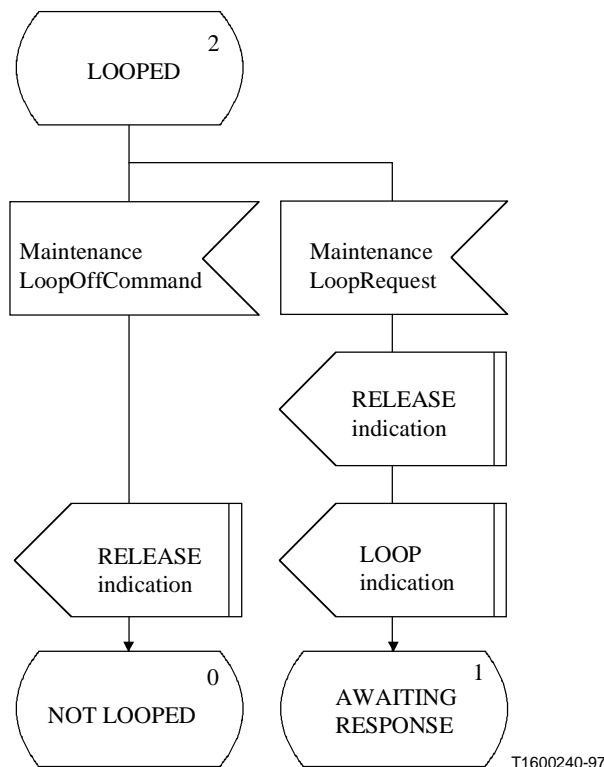


Figura 47.iii/H.245 – Diagrama SDL de la MLSE de entrada (*fin*)

ANEXO A

Asignaciones de identificadores de objetos

El cuadro A.1 indica los identificadores de objetos definidos para uso en esta Recomendación.

Cuadro A.1/H.245

Valor de identificador de objeto	Descripción
{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 un 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 un protocolo de control de sistema multimedia. En la actualidad hay tres 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 un protocolo de control de sistema multimedia. En la actualidad hay tres versiones normalizadas definidas. Este valor indica la tercera versión de esta Recomendación.

Reemplazada por una versión más reciente

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 Recomendación X.409. Este lenguaje evolucionó a la forma en que figura en la Recomendación X.208, y más recientemente en la X.680. La ASN.1 permite una especificación inequívoca de estructuras de datos complejas incluidas las que incluyen campos de longitud variable, campos facultativas 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 Recomendación 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 FALSO y 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 más 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)`

Reemplazada por una versión más reciente

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	campo 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 0 a 7 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	pad 00000011
254	pad 11111110

skew

INTEGER (0..4095)

Valor	Codificación
3	pad 00000000 00000011
4095	pad 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

Reemplazada por una versión más reciente

La notación ASN.1 sustenta 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
{
  nonStandardNonStandardParameter,
  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.

Reemplazada por una versión más reciente

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 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 $40 * \text{primero} + \text{segundo}$, en este caso $40 * 0 + 0 = 0$. 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 0100,

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.

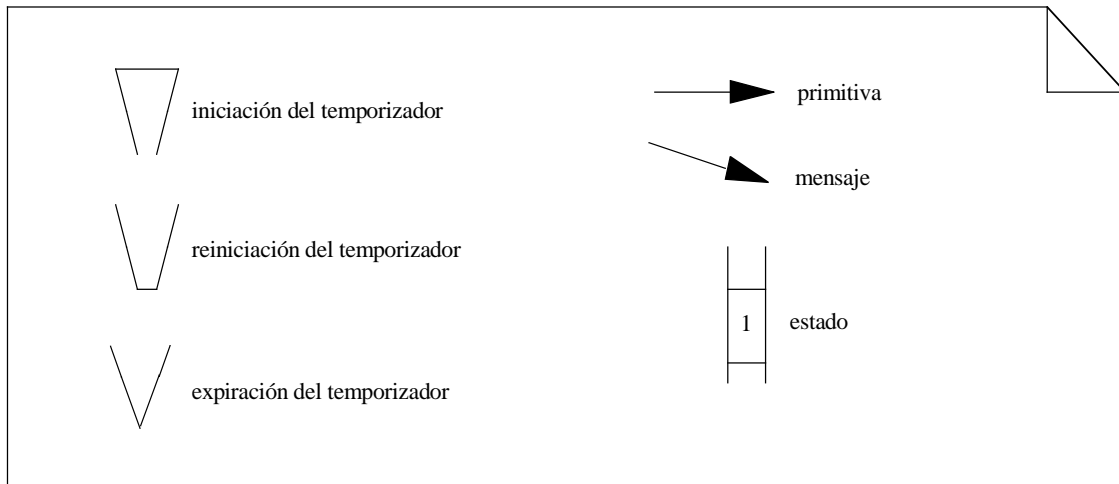
Reemplazada por una versión más reciente

APÉNDICE II

Ejemplos de procedimientos H.245

II.1 Introducción

Este apéndice ilustra ejemplos de los procedimientos definidos en la cláusula 8. La figura II.1-1 muestra los símbolos gráficos utilizados en los diagramas de este apéndice.



T1523580-96

Figura II.1-1/H.245 – Clave de las figuras

II.2 Entidad de señalización de determinación director-subordinado

En las figuras que siguen 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 director-subordinado

mensaje	nombre en los ejemplos
MasterSlaveDetermination	MSD
MasterSlaveDeterminationAck	MSDAck
MasterSlaveDeterminationReject	MSDReject
MasterSlaveDeterminationRelease	MSDRelease

En las figuras que siguen, 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.

Reemplazada por una versión más reciente

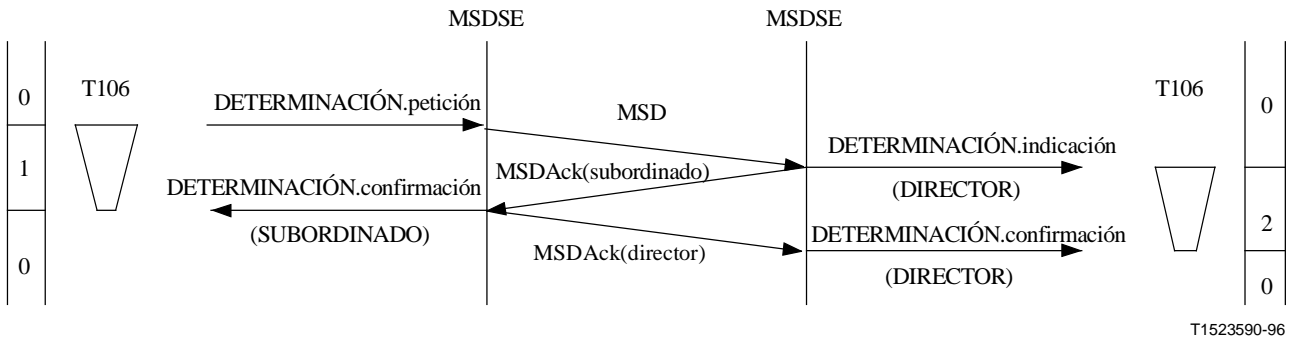


Figura II.2-1/H.245 – Determinación director-subordinado – Director en la MSDSE distante

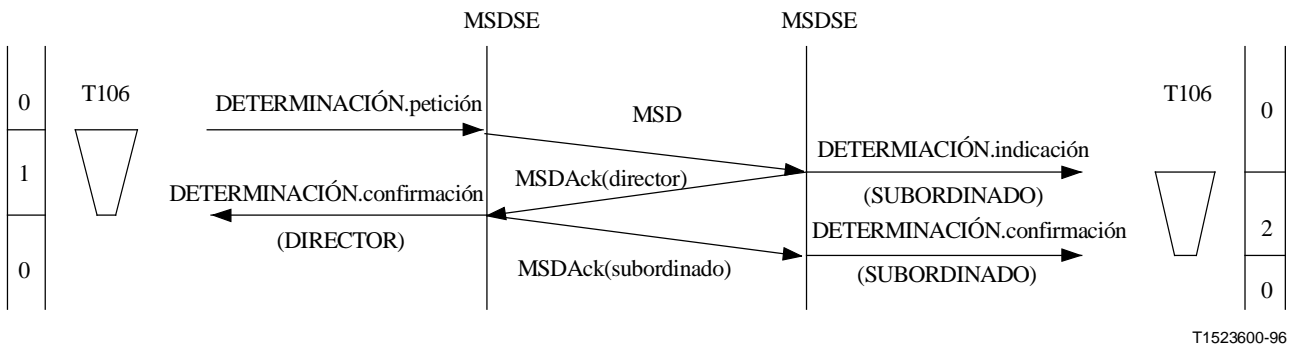


Figura II.2-2/H.245 – Determinación director-subordinado – Subordinado en la MSDSE distante

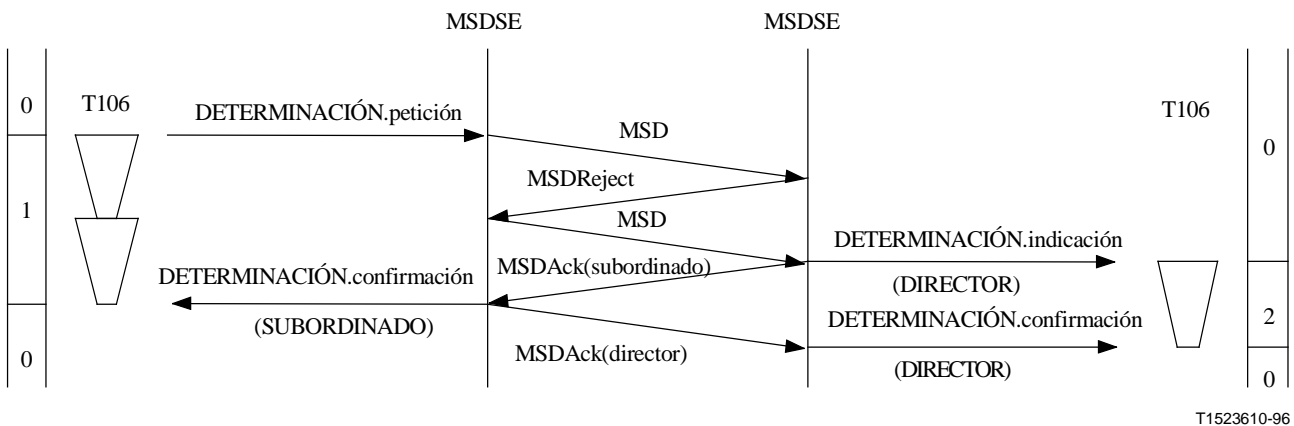


Figura II.2-3/H.245 – Determinación director-subordinado – El primer intento produjo un resultado indeterminado. El segundo intento tuvo éxito

Reemplazada por una versión más reciente

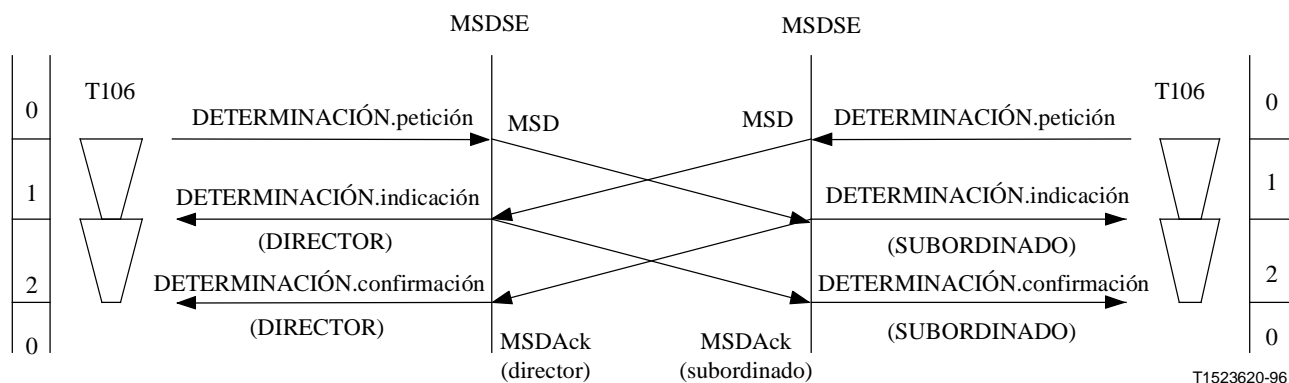


Figura II.2-4/H.245 – Determinación director-subordinado – Determinación simultánea

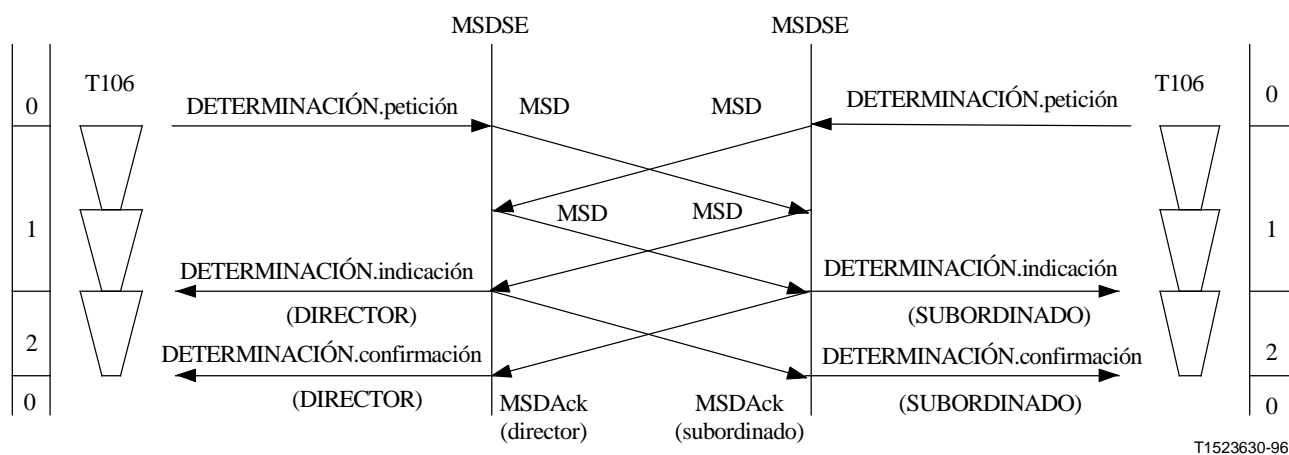


Figura II.2-5/H.245 – Determinación director-subordinado – Determinación simultánea, pero el primer intento produce un resultado indeterminado

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.

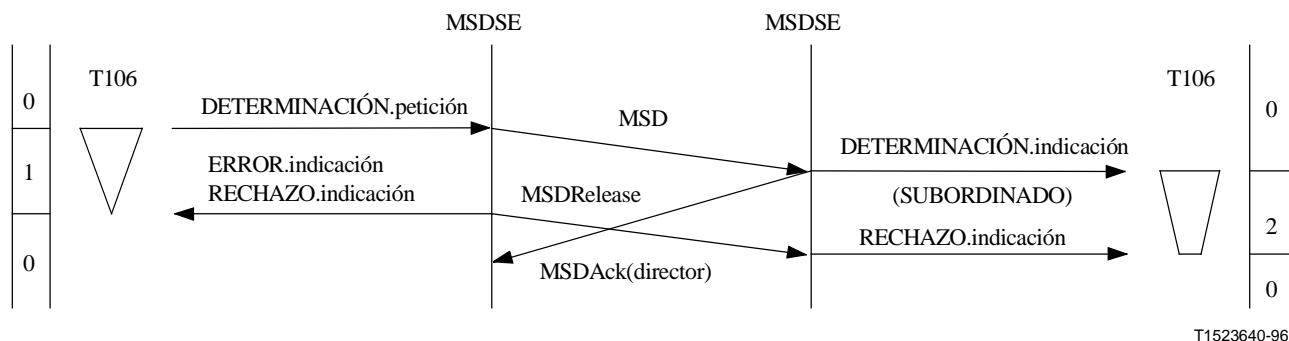


Figura II.2-6/H.245 – Determinación director-subordinado – Expiración del temporizador local T106 con subordinado en el extremo distante

Reemplazada por una versión más reciente

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.

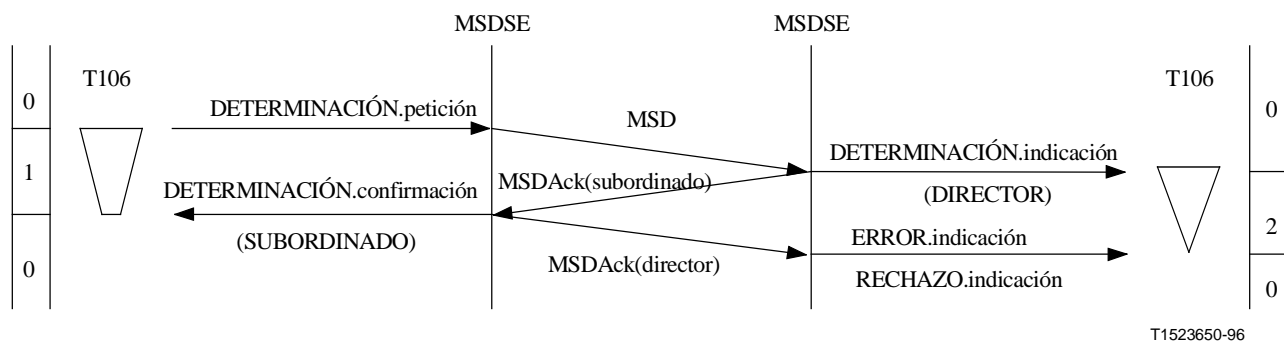


Figura II.2-7/H.245 – Determinación director-subordinado – Expiración del temporizador distante T106 con director en el extremo distante

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.

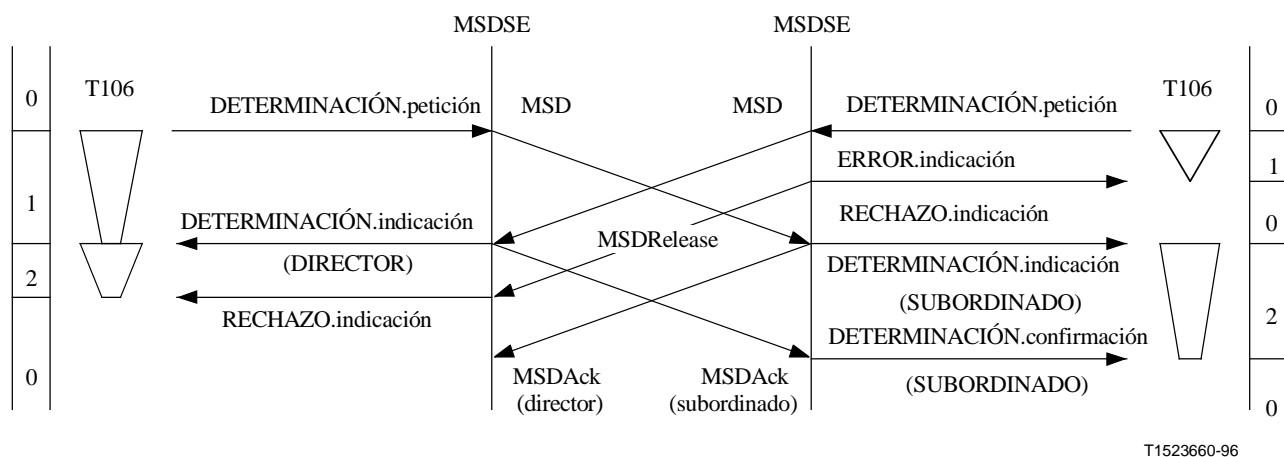


Figura II.2-8/H.245 – Determinación director-subordinado – Procedimientos de determinación simultánea con expiración del temporizador T106 en el subordinado

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

Reemplazada por una versión más reciente

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.

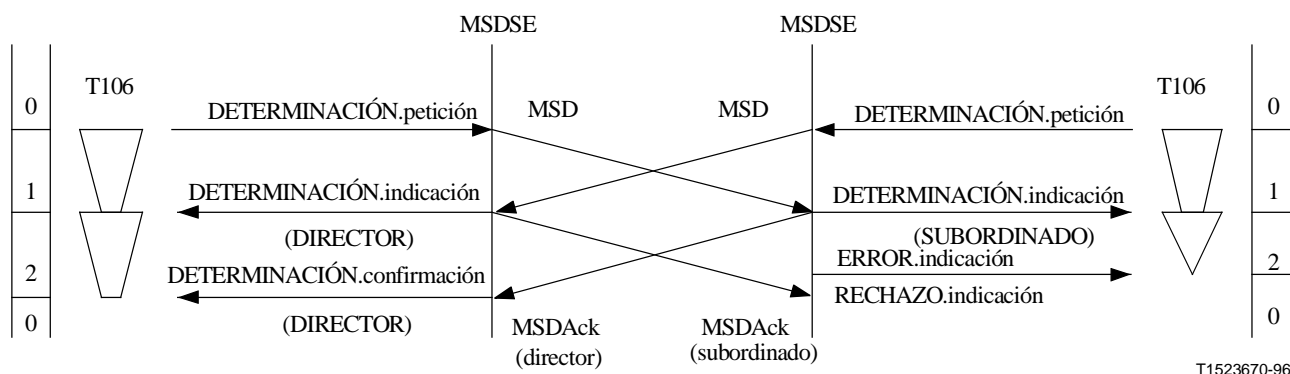


Figura II.2-9/H.245 – Determinación director-subordinado – Procedimientos de determinación simultánea con expiración del temporizador T106 durante el estado ESPERA DE ACUSE DE RECIBO EN ENTRADA

En la figura II.2-10 se obtuvo un resultado indeterminado N100 veces. En este caso N100 = 3.

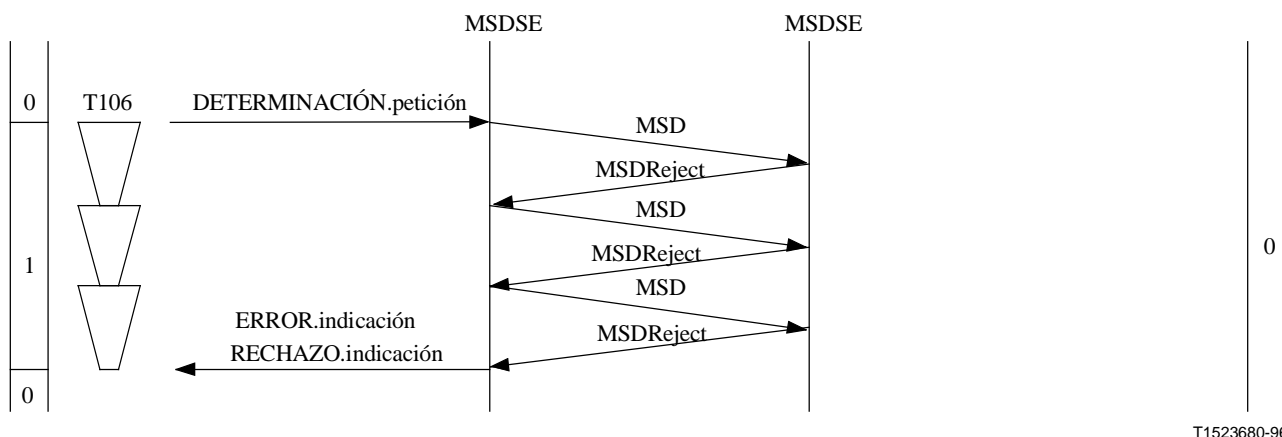


Figura II.2-10/H.245 – Determinación director-subordinado – Resultado indeterminado con N100 = 3

II.3 Entidad de señalización de intercambio de capacidad

Las figuras siguientes ilustran los procedimientos de la CESE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1" respectivamente.

Reemplazada por una versión más reciente

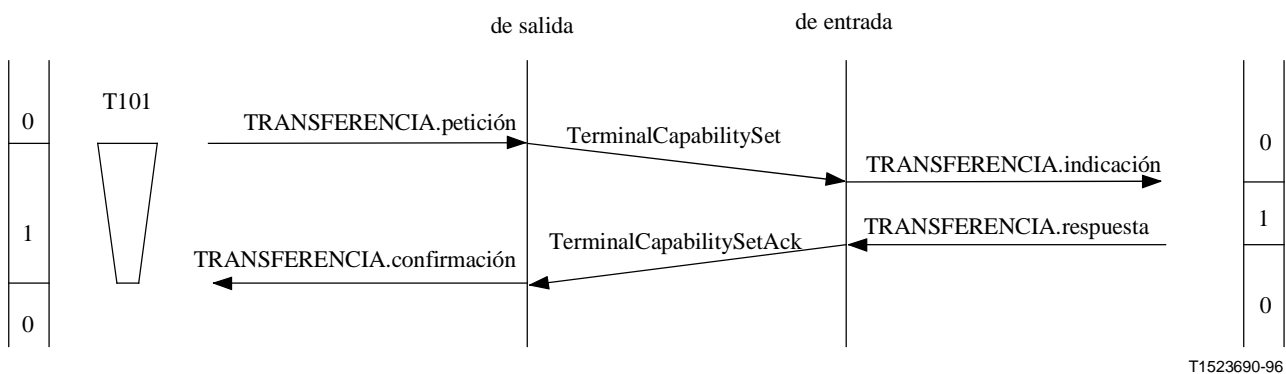


Figura II.3-1/H.245 – Intercambio de capacidades con aceptación por el usuario de la CESE de entrada par

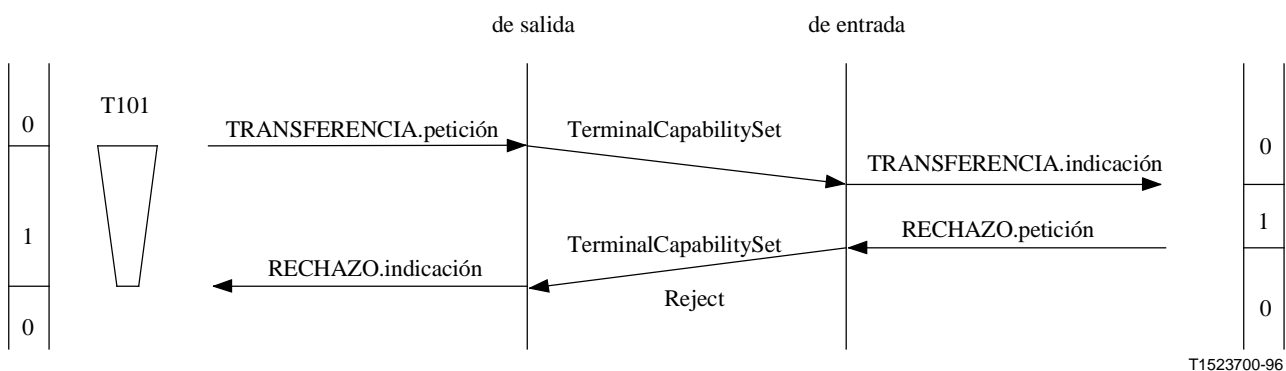


Figura II.3-2/H.245 – Intercambio de capacidades con rechazo por el usuario de la CESE de entrada par

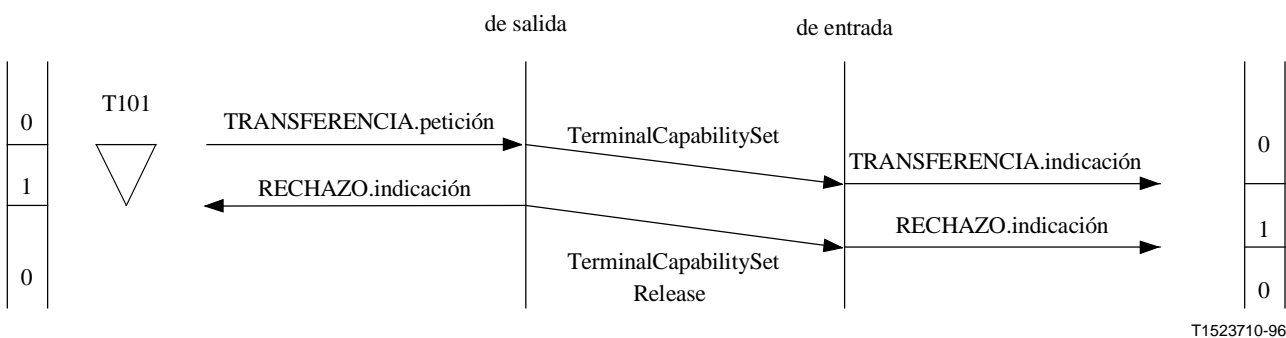
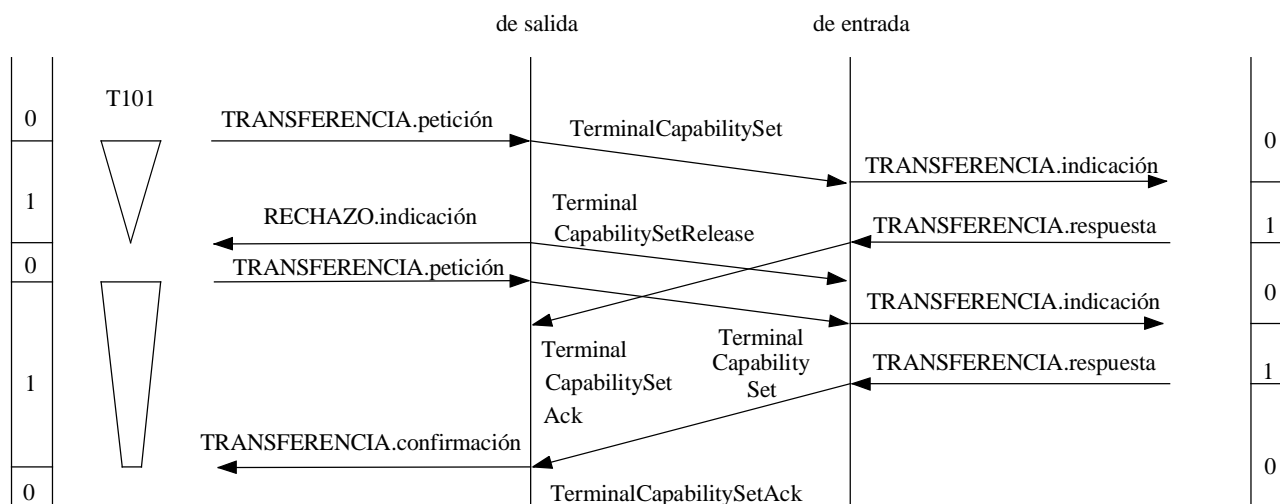


Figura II.3-3/H.245 – Intercambio de capacidades con expiración del temporizador T101. El mensaje TerminalCapabilitySetRelease llega a la CESE de entrada antes de la respuesta del usuario de la CESE de entrada

Reemplazada por una versión más reciente

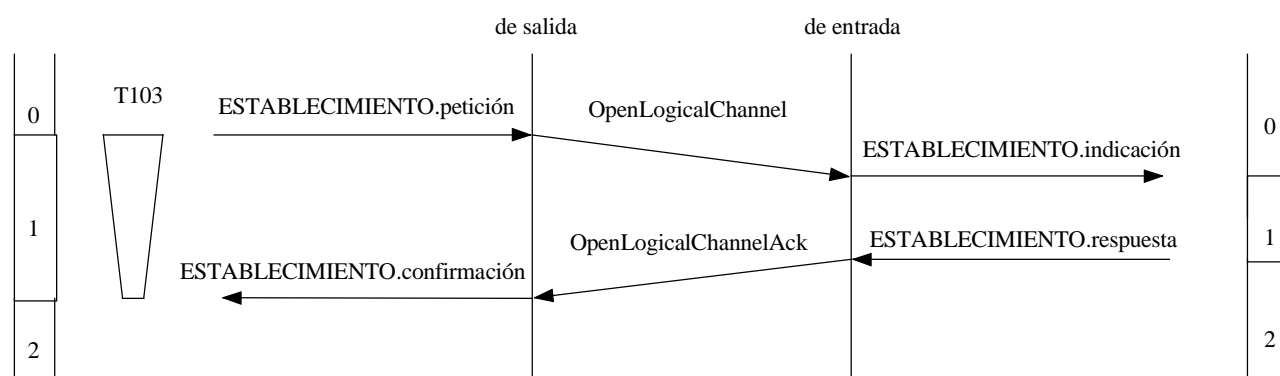


T1523720-96

Figura II.3-4/H.245 – Intercambio de capacidades con expiración del temporizador T101 seguido de un segundo intercambio de capacidad. 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

II.4 Entidad de señalización de canal lógico

Las figuras siguientes 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.



T1523730-96

Figura II.4-1/H.245 – Establecimiento de canal lógico

Reemplazada por una versión más reciente

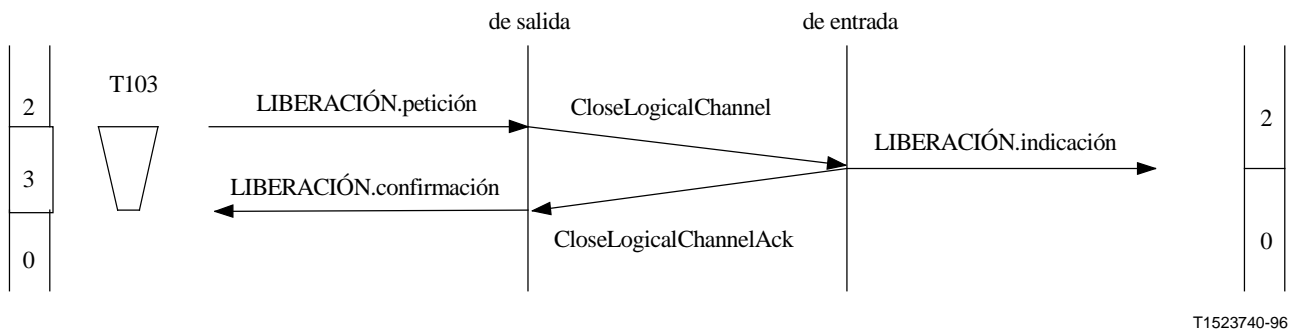


Figura II.4-2/H.245 – Liberación de canal lógico

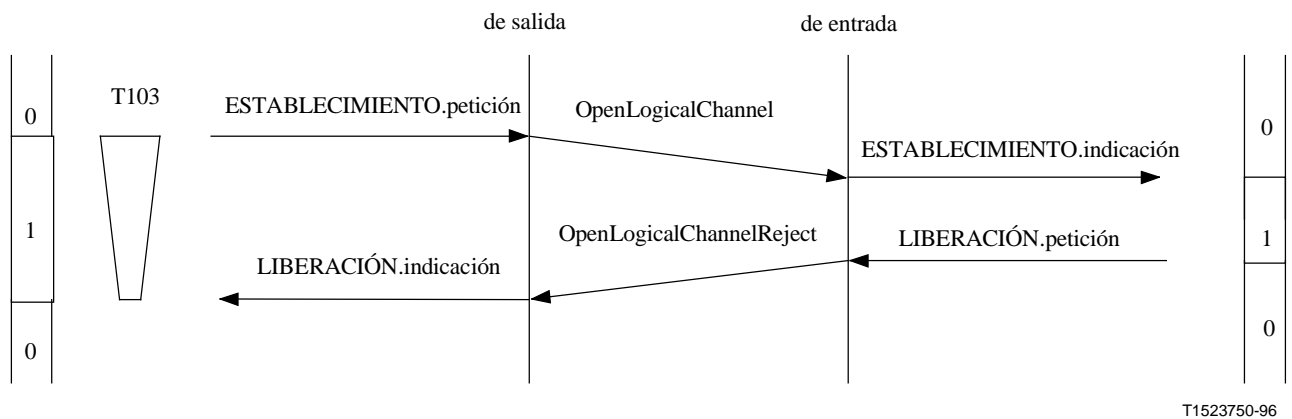


Figura II.4-3/H.245 – Rechazo de establecimiento de canal lógico por el usuario de la LCSE par

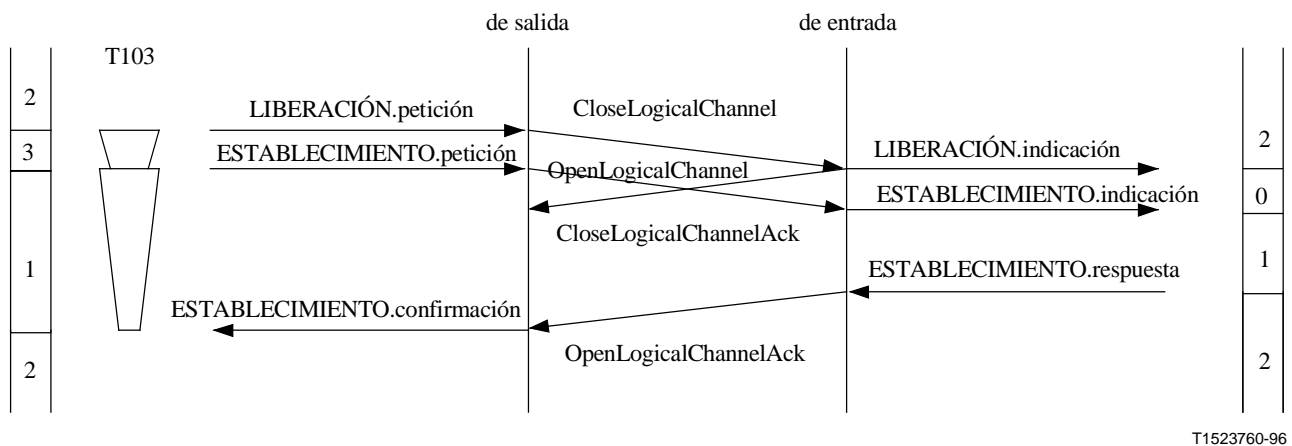
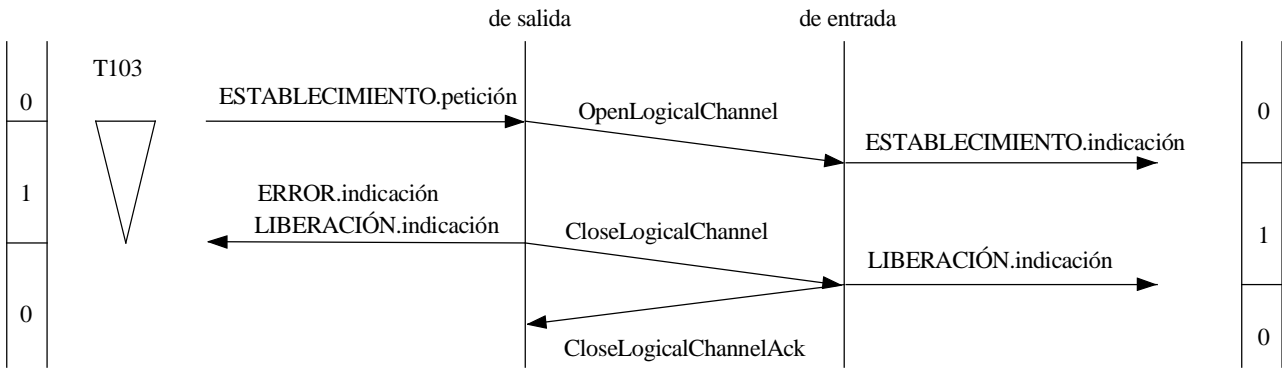


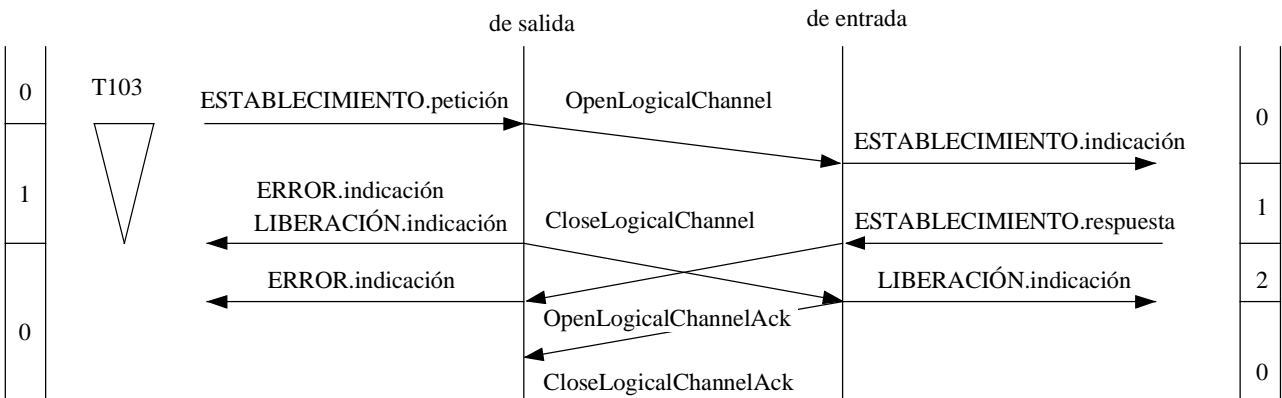
Figura II.4-4/H.245 – Liberación de canal lógico seguida de nuevo establecimiento inmediato

Reemplazada por una versión más reciente



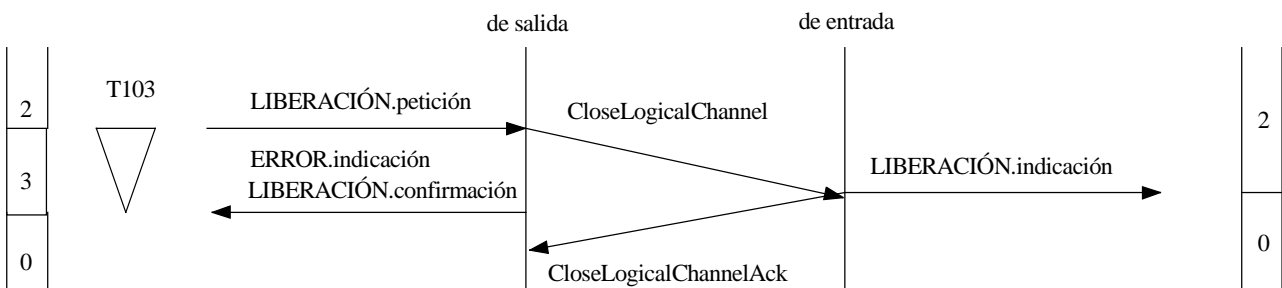
T1523770-96

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



T1523780-96

Figura II.4-6/H.245 – Petición de establecimiento de canal lógico con expiración del temporizador T103. 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



T1523790-96

Figura II.4-7/H.245 – Petición de liberación de canal lógico con expiración del temporizador T103

Reemplazada por una versión más reciente

II.5 Entidad de señalización de cierre de canal lógico

Las siguientes figuras ilustran los procedimientos de la CLCSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1" respectivamente.

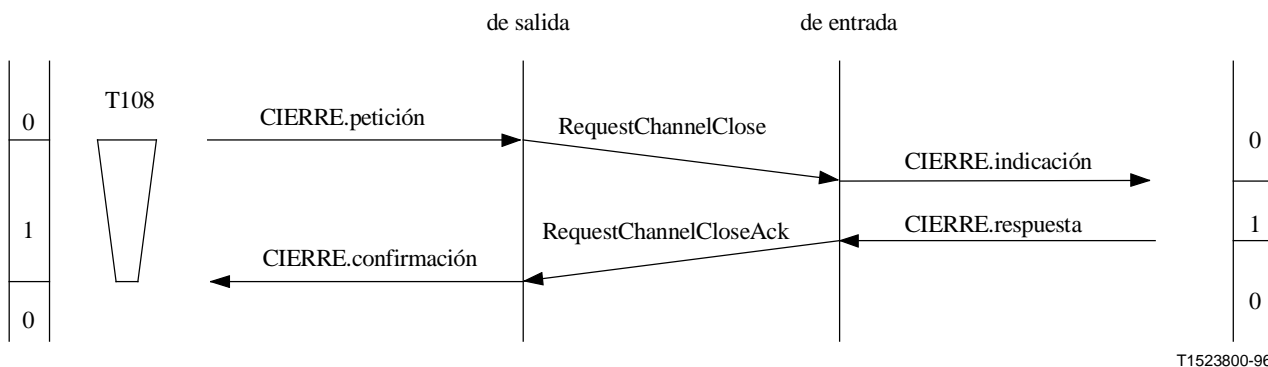


Figura II.5-1/H.245 – Petición de cierre de canal lógico

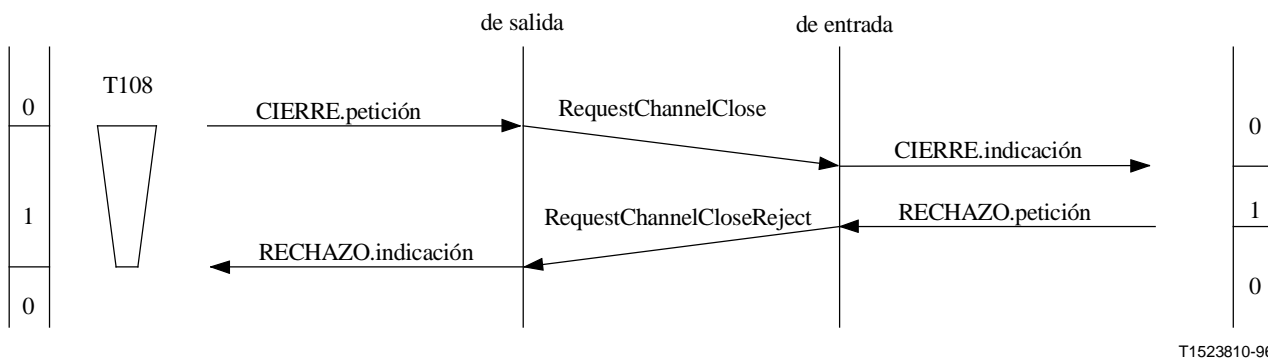


Figura II.5-2/H.245 – Petición de cierre de canal lógico con rechazo por el usuario de la CLCSE de entrada par

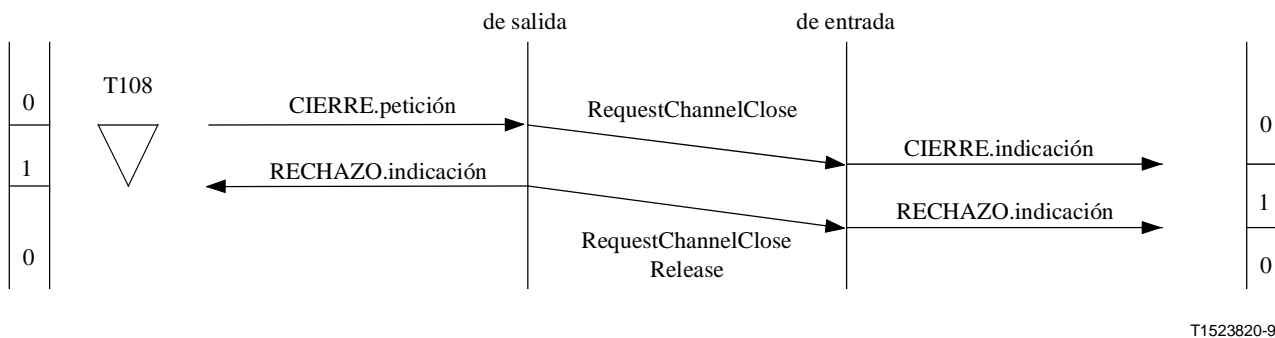
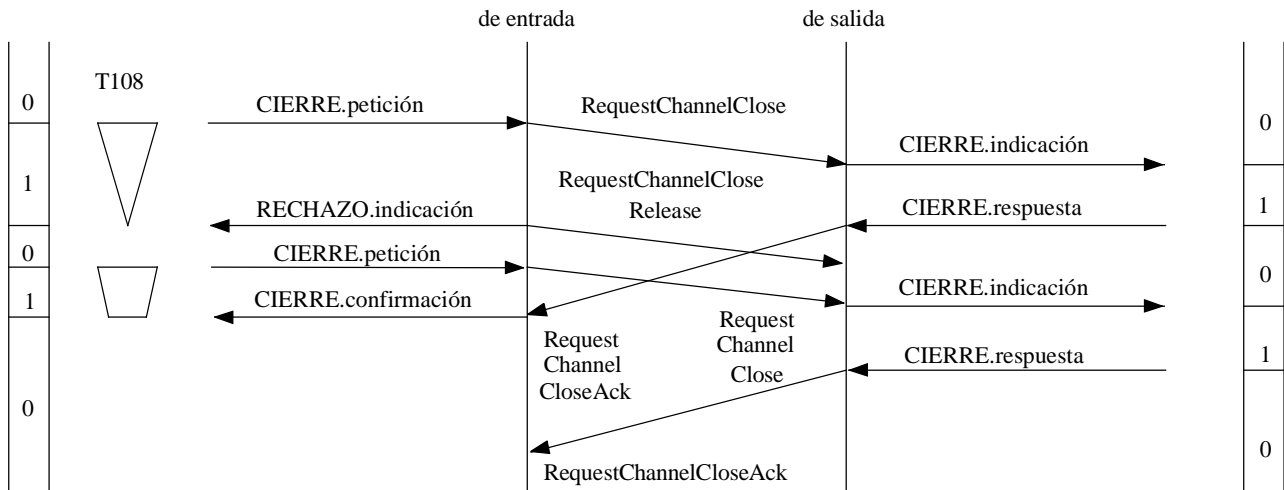


Figura II.5-3/H.245 – Petición de cierre de canal lógico con expiración del temporizador T108. El mensaje RequestChannelCloseRelease llega a la CLCSE de entrada antes que la respuesta del usuario de la CLCSE de entrada

Reemplazada por una versión más reciente

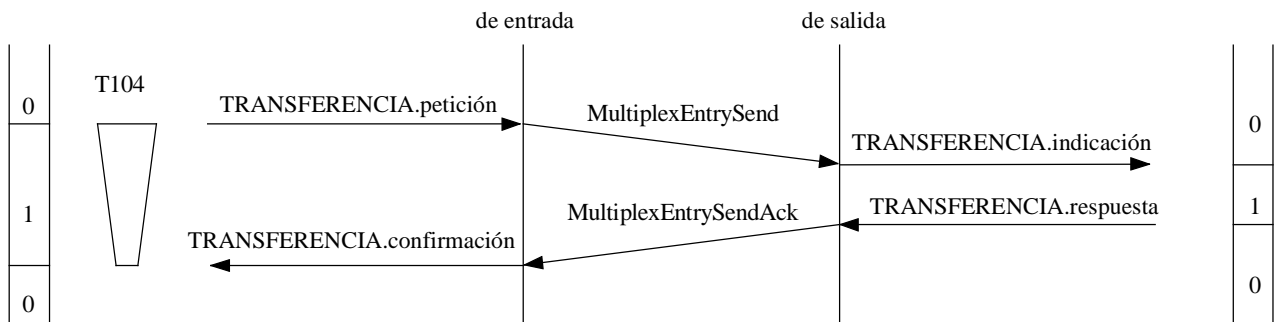


T1523830-96

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. La petición de cierre de canal es confirmada al recibo del primer mensaje RequestChannelClose

II.6 Entidad de señalización de tabla múltiplex

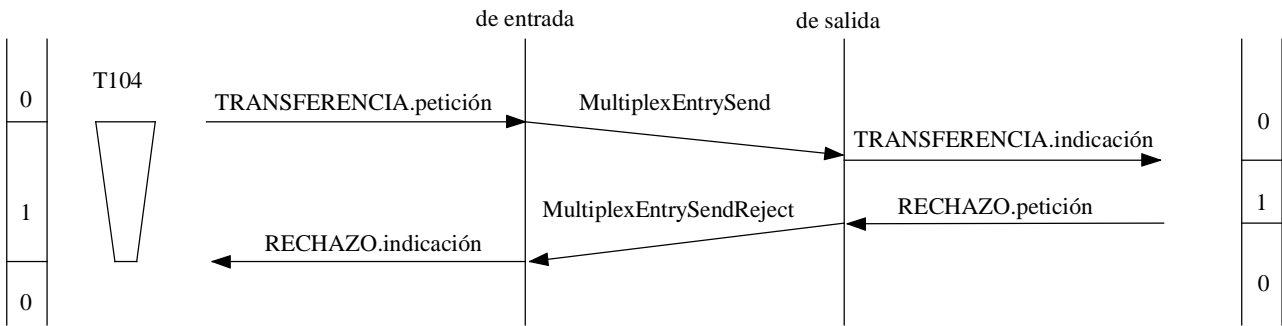
Las figuras siguientes ilustran los procedimientos de la MTSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1" respectivamente.



T1523840-96

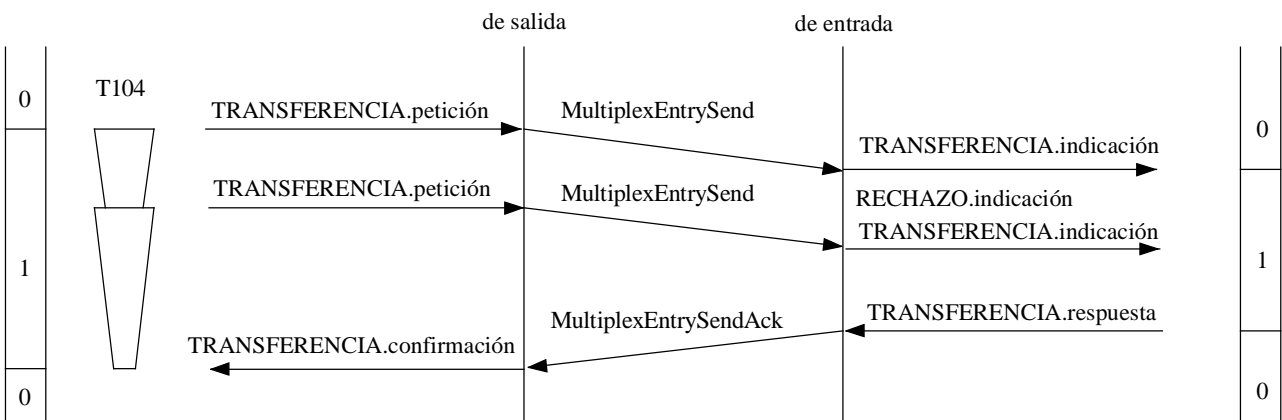
Figura II.6-1/H.245 – Petición de envío de tabla múltiplex, con éxito

Reemplazada por una versión más reciente



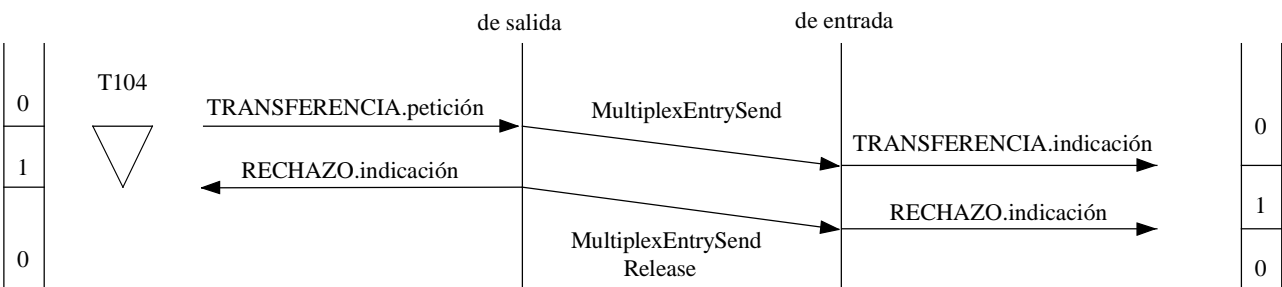
T1523850-96

Figura II.6-2/H.245 – Petición de envío de tabla múltiplex con rechazo por el usuario de la MTSE par



T1523860-96

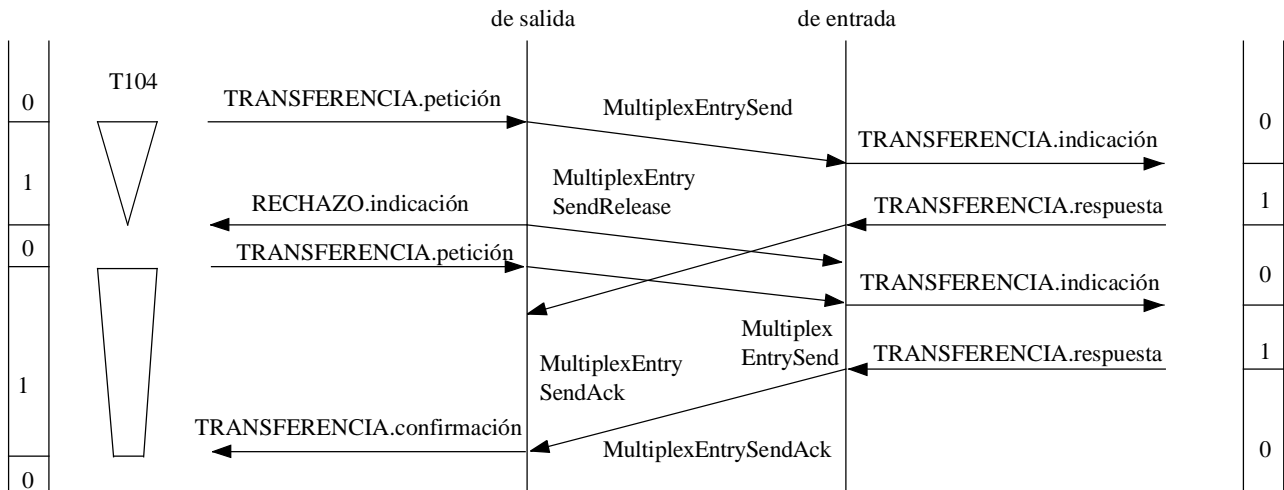
Figura II.6-3/H.245 – Petición de envío de tabla múltiplex con una segunda petición de envío de tabla múltiplex antes del acuse de recibo de la primera petición. La primera petición fracasó



T1523870-96

Figura II.6-4/H.245 – Petición de envío de tabla múltiplex con expiración del temporizador T104

Reemplazada por una versión más reciente

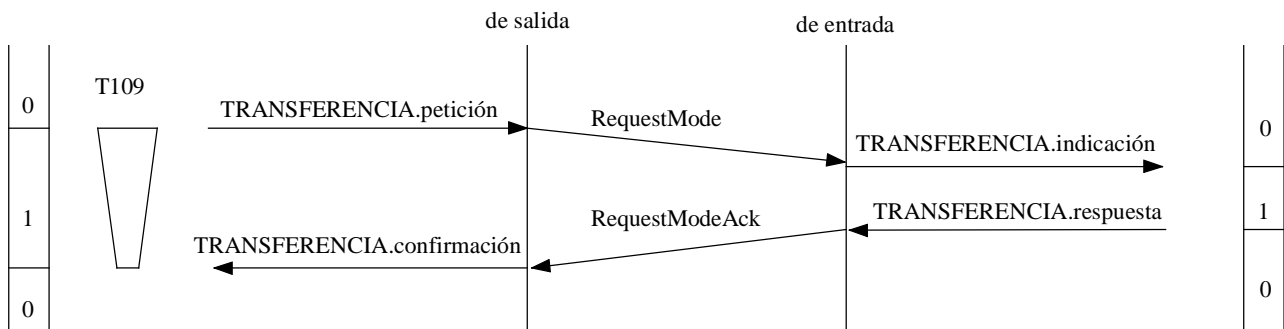


T1523880-96

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. El primer mensaje MultiplexEntrySendAck es ignorado en la MTSE desalida. Sólo tuvo éxito la segunda petición

II.7 Entidad de señalización de petición de modo

Las figuras siguientes ilustran los intercambios de la MRSE. Los estados REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1" respectivamente.



T1523890-96

Figura II.7-1/H.245 – Petición de modo con éxito

Reemplazada por una versión más reciente

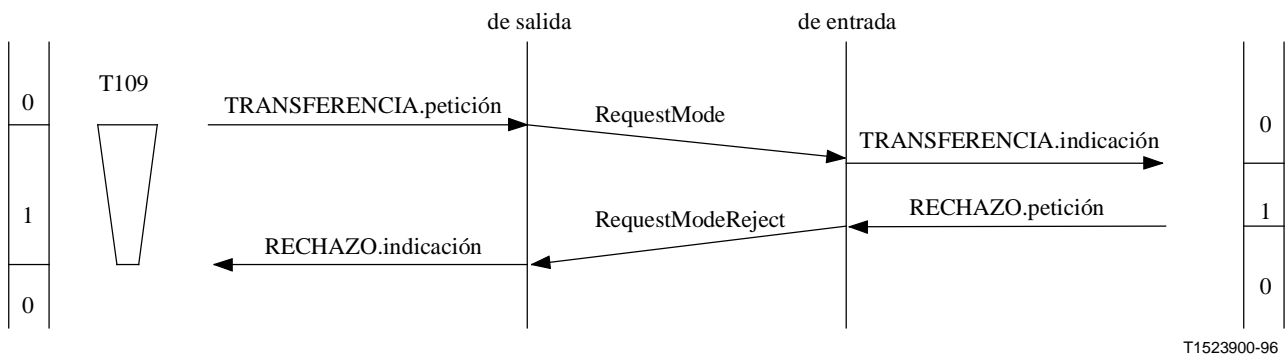


Figura II.7-2/H.245 – Petición de modo con rechazo por el usuario de la MRSE par

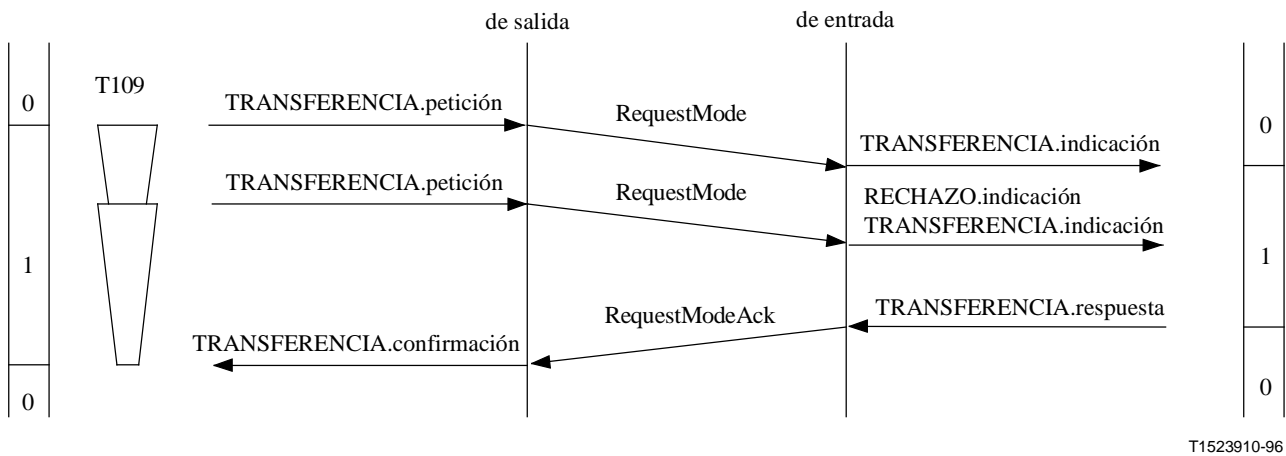


Figura II.7-3/H.245 – Petición de modo con una segunda petición de modo antes del acuse de recibo de la primera petición. La primera petición fracasó

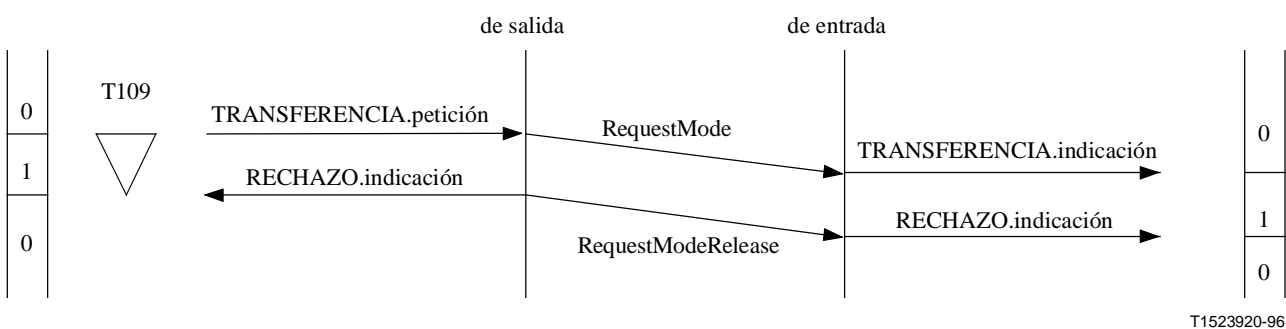


Figura II.7-4/H.245 – Petición de modo con expiración del temporizador T109. La petición de modo fracasó

Reemplazada por una versión más reciente

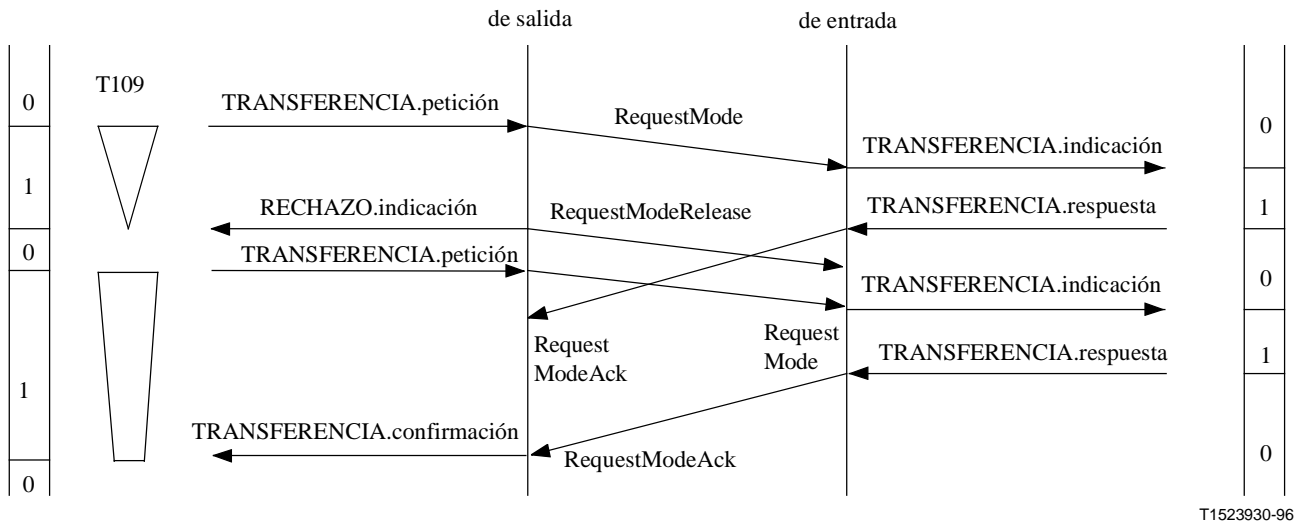


Figura II.7-5/H.245 – Petición de modo con expiración del temporizador T109 seguida de una segunda petición de modo. El primer mensaje RequestModeAck es ignorado en la MRSE de salida. Sólo tuvo éxito la segunda petición

II.8 Entidad de señalización de retardo de ida y vuelta

Las figuras siguientes ilustran los procedimientos de la RTDSE. Los estados de la RTDSE REPOSO y ESPERA DE RESPUESTA se designan por "0" y "1" respectivamente.

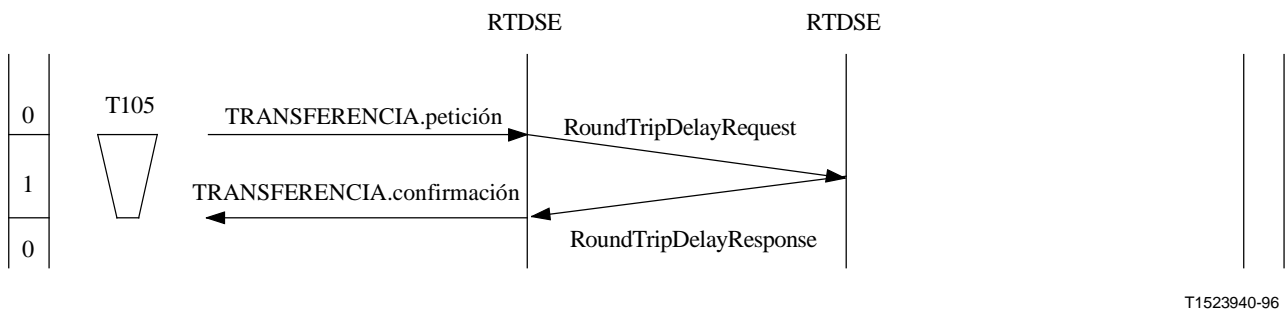


Figura II.8-1/H.245 – Procedimiento de determinación de retardo de ida y vuelta

Reemplazada por una versión más reciente

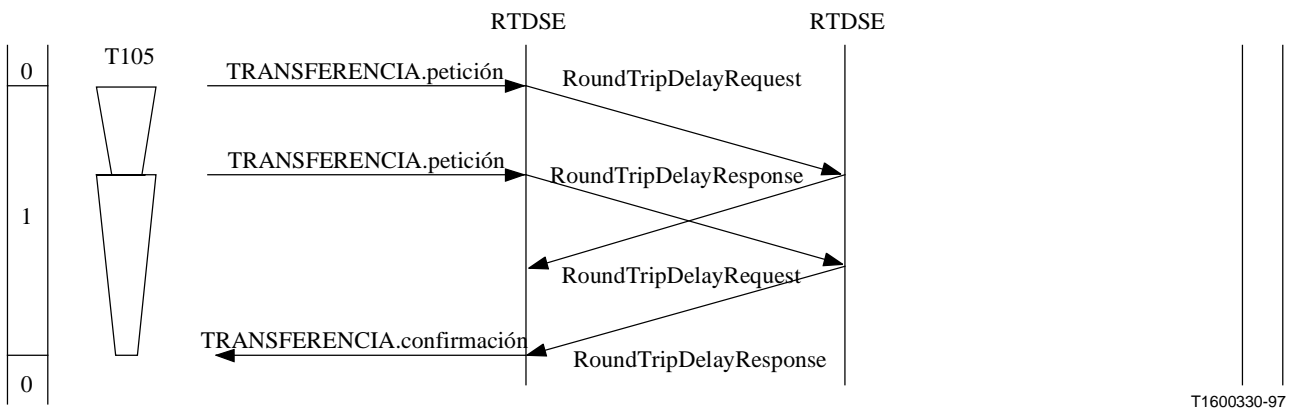


Figura II.8-2/H.245 – Procedimiento de determinación de retardo de ida y vuelta, estando pendiente un procedimiento anterior del cual no se ha acusado recibo

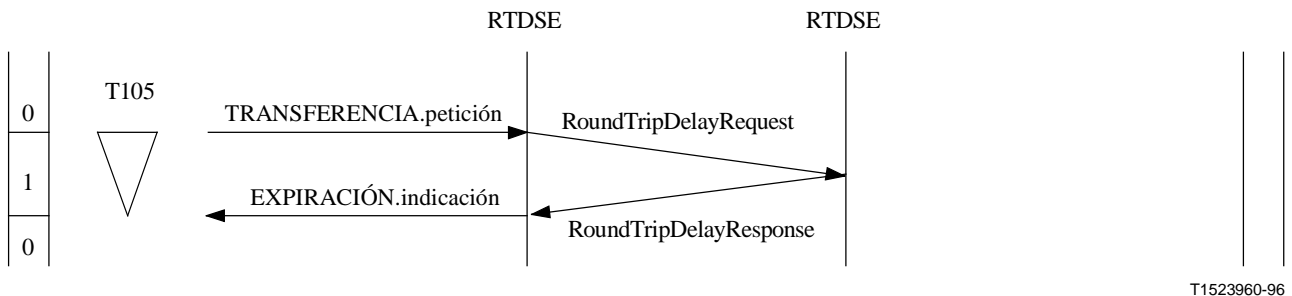


Figura II.8-3/H.245 – Procedimiento de determinación de retardo de ida y vuelta con expiración del temporizador T105

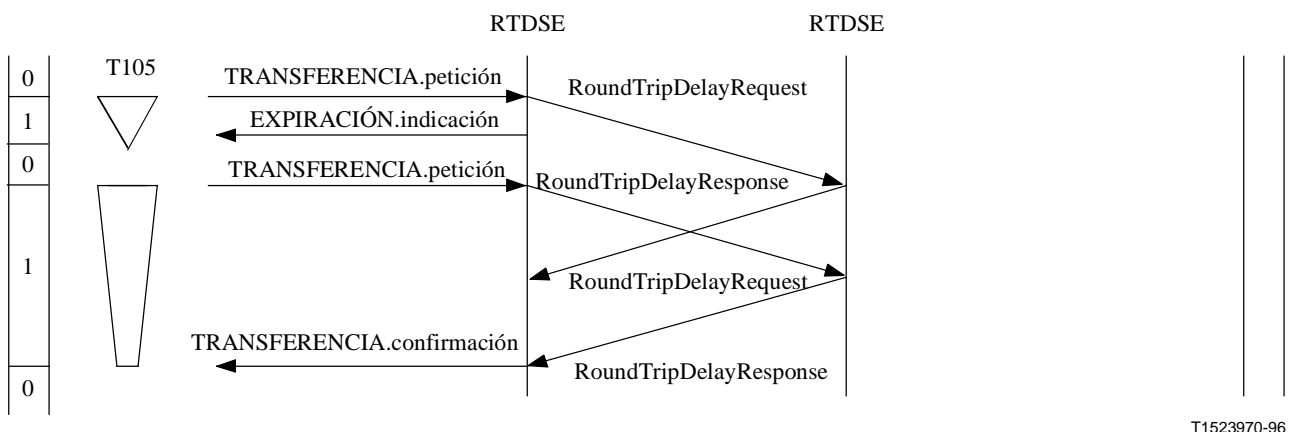


Figura II.8-4/H.245 – Procedimiento de determinación de retardo de ida y vuelta con expiración del temporizador T105, seguido de un segundo procedimiento de determinación de retardo de ida y vuelta. El mensaje RoundTripDelayResponse procedente del primer procedimiento llega estando en curso el segundo procedimiento y es ignorado

Reemplazada por una versión más reciente

II.9 Entidad de señalización de canal lógico bidireccional

Las siguientes figuras 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.

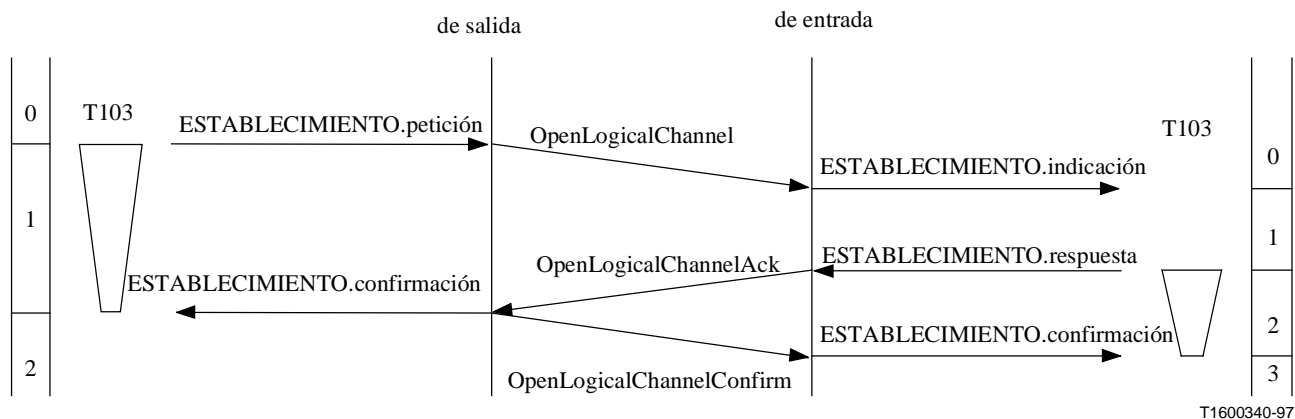


Figura II.9-1/H.245 – Establecimiento de canal lógico bidireccional

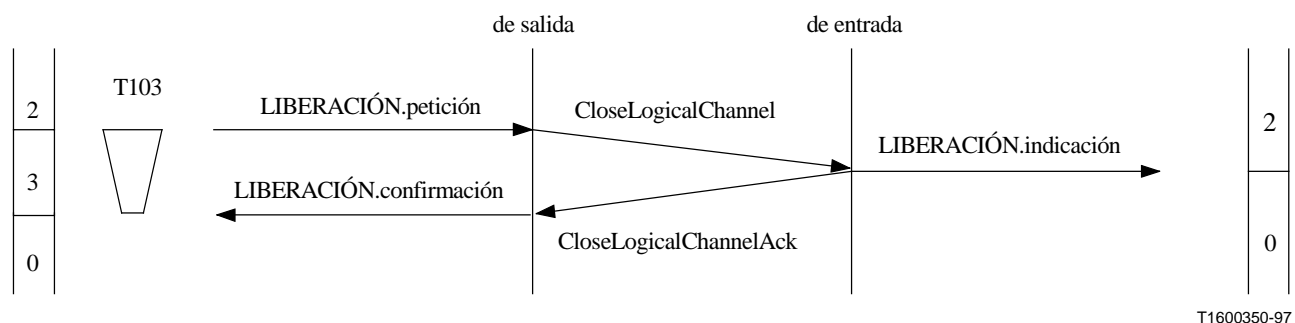


Figura II.9-2/H.245 – Liberación de canal lógico bidireccional

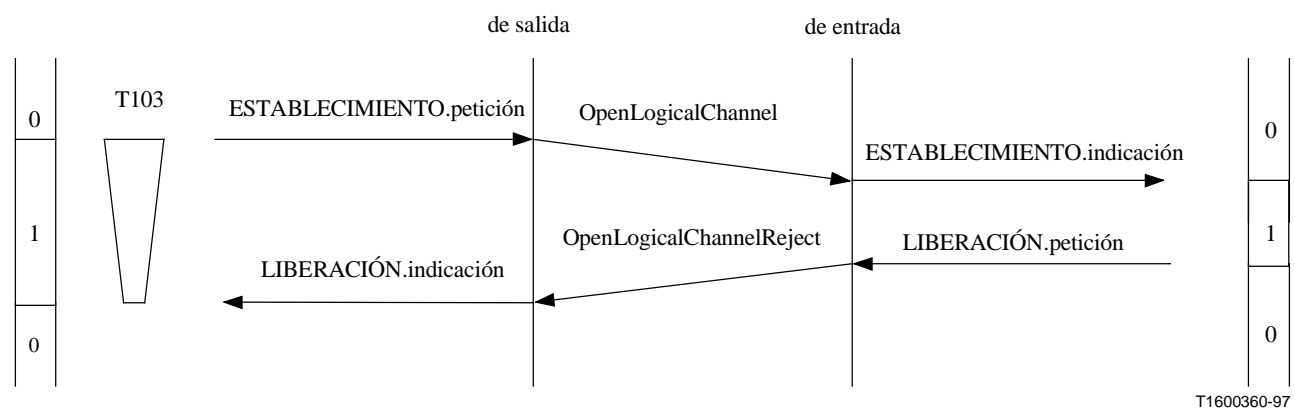


Figura II.9-3/H.245 – Establecimiento de canal lógico bidireccional con rechazo por el usuario de la B-LCSE par

Reemplazada por una versión más reciente

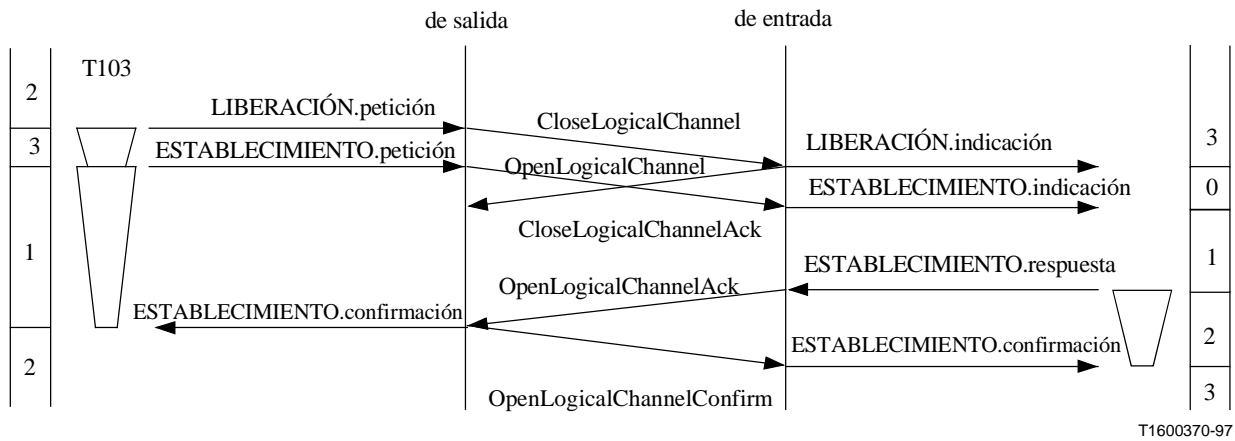


Figura II.9-4/H.245 – Liberación de canal lógico bidireccional seguida por un nuevo establecimiento inmediato

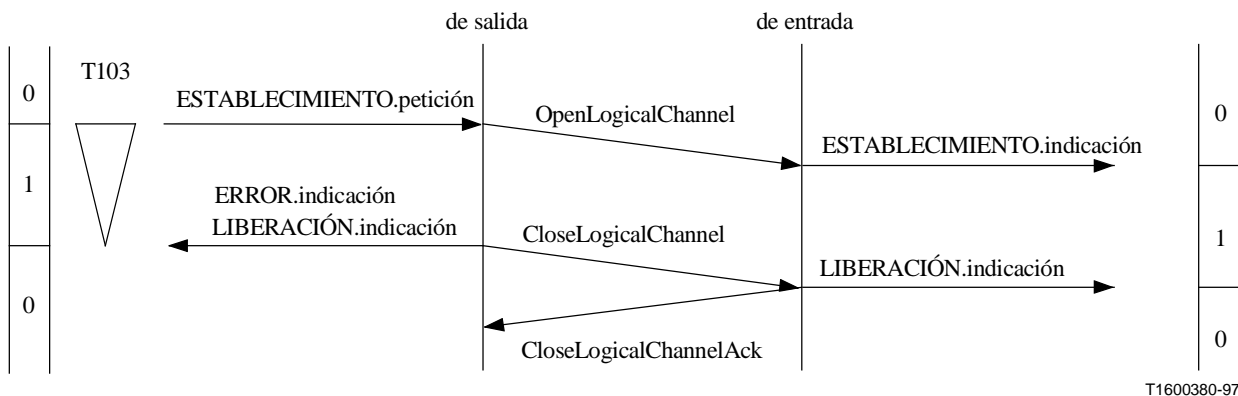
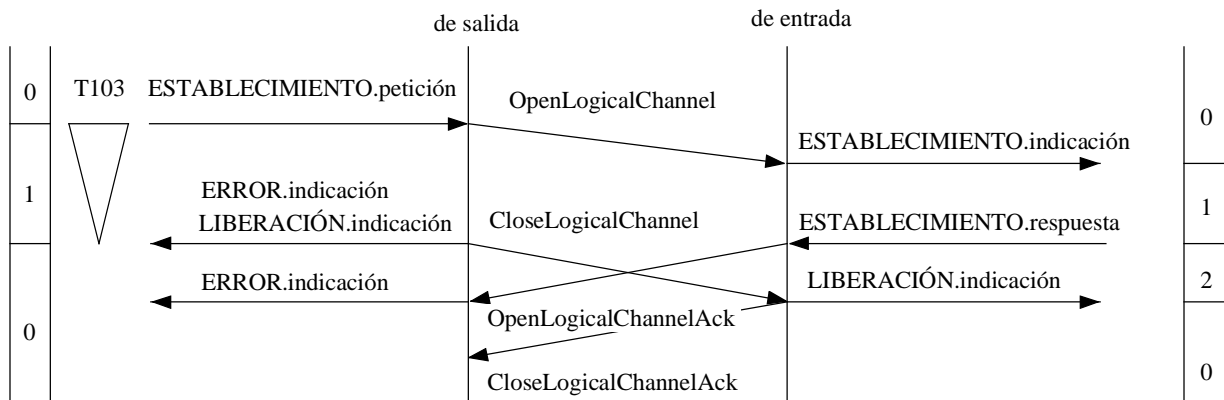


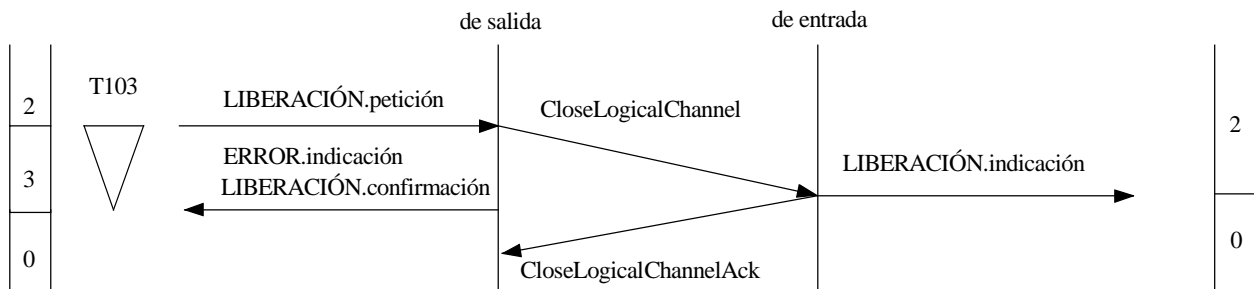
Figura II.9-5/H.245 – Petición de establecimiento de canal lógico bidireccional con expiración del temporizador T103 en el lado de salida debido a una respuesta lenta del usuario de la B-LCSE de entrada

Reemplazada por una versión más reciente



T1600390-97

Figura II.9-6/H.245 – Petición de establecimiento de canal lógico bidireccional con expiración del temporizador T103 en el lado de salida. El temporizador T103 en el lado de 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



T1600400-97

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 la cláusula 8.

Esta Recomendación no define los valores cargados en los temporizadores. Estos valores pueden definirse en otras Recomendaciones, como H.310, H.323 y H.324.

Reemplazada por una versión más reciente

III.1 Temporizadores

El cuadro III.1 muestra los temporizadores especificados en esta Recomendación.

Cuadro III.1/H.245 – Temporizadores de procedimiento

Temporizador	Procedimiento	Definición
T106	Determinación director-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.

Reemplazada por una versión más reciente

III.2 Contadores

El cuadro III.2 muestra los contadores especificados en esta Recomendación.

Cuadro III.2/H.245 – Contadores de procedimiento

Temporizador	Procedimiento	Definición
N100	Determinación director-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

La Recomendación es un "documento viviente" utilizado por muchas Recomendaciones de sistemas incluidas H.310, H.323, H.324, y V.70, que se espera será ampliado en forma retrocompatible, probablemente en cada reunión de la Comisión de Estudio 16 del UIT-T. Este apéndice explica el procedimiento que debe utilizarse para añadir extensiones a H.245.

En un momento dado cualquiera hay solamente una sintaxis H.245 en vigor. Ninguna otra Recomendación UIT-T deberá incluir otras variantes de la sintaxis H.245 en sus Recomendaciones, de una manera normativa.

Las peticiones de extensiones (o ampliaciones) de la 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 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 H.245, y,
- 2) la semántica propuesta para la cláusula 7, y,
- 3) los procedimientos propuestos para la cláusula 8 si se solicitan nuevos procedimientos.

Todas las extensiones de H.245 serán retrocompatibles con todas las versiones anteriores de H.245. 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 de la 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.

Reemplazada por una versión más reciente

Deberán evitarse las abreviaturas y los acrónimos, sobre todo si una palabra o frase no están abreviadas o expresadas 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 mediante el procedimiento de la Resolución 1. Tras la aprobación de cada nueva versión de H.245, 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** de la Recomendación 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**.

Reemplazada por una versión más reciente

Puesto que la apertura y cierre de canales lógicos H.245 no están sincronizadas 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 de la Recomendación 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 la Recomendación 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 de la Recomendación 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.

Reemplazada por una versión más reciente

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

Parámetro de H263Capability	Ejemplos								
	1	2	3	4			5		
sqcifMPI	3	NP	NP	NP			NP		
qcifMPI	NP	NP	NP	NP			3		
cifMPI	NP	NP	NP	NP			NP		
cif4MPI	NP	NP	NP	NP			NP		
cif16MPI	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		

Reemplazada por una versión más reciente

Cuadro VI.1/H.245 – Ejemplos de fijación de parámetros H.245 de capa de mejora (*fin*)

Parámetro de H263Capability		Ejemplos								
		1	2	3	4		5			
temporalSpatialTradeOffCap		F	F	F	F			F		
hrd-B		NP	NP	NP	NP			NP		
bppMaxKb		NP	NP	NP	NP			NP		
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*) =		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*) =		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*) =		NP	NP	NP	NP	NP	NP			
bframeEnhancement	sqcifMPI									
	qcifMPI									
	cifMPI									
	cif4MPI									
	cif16MPI									
	maxbitrate									
NP = No presente, T = TRUE (verdadero), F = FALSE (falso)										
* No se muestran otras opciones por debajo de "maxbitrate" en la estructura EnhancementOptions										

Reemplazada por una versión más reciente

VI.2 Ejemplos de fijación de parámetros H.245 de canal de retorno de vídeo

Esta subcláusula 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**.

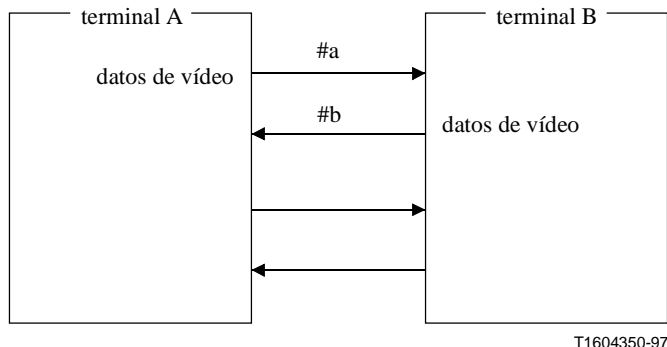


Figura VI.1/H.245 – Canales lógicos para datos de vídeo

- 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**.

Reemplazada por una versión más reciente

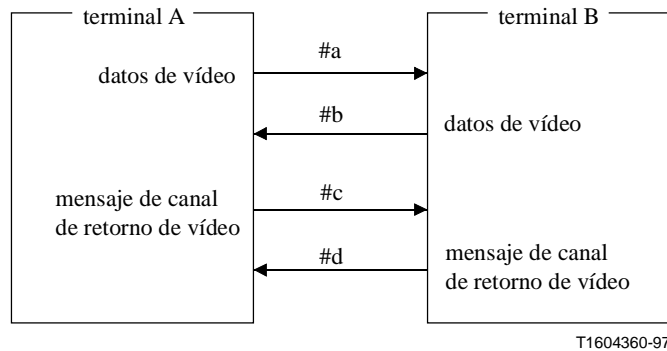


Figura VI.2/H.245 – Canales lógicos para modo canal lógico separado

- 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 de H263Capability	H263Capability en mensaje OpenLogicalChannel		
	#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			

Reemplazada por una versión más reciente

Ejemplo 2: Modo *multiplex* 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 *multiplex* 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.

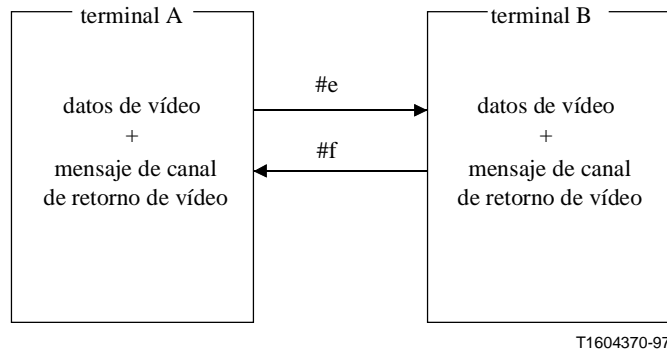


Figura VI.3/H.245 – Canales lógicos para modo *multiplex* de vídeo

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 (figura VI.4). En el cuadro VI.3 se muestran las fijaciones de los parámetros de los canales lógicos #g y #h.

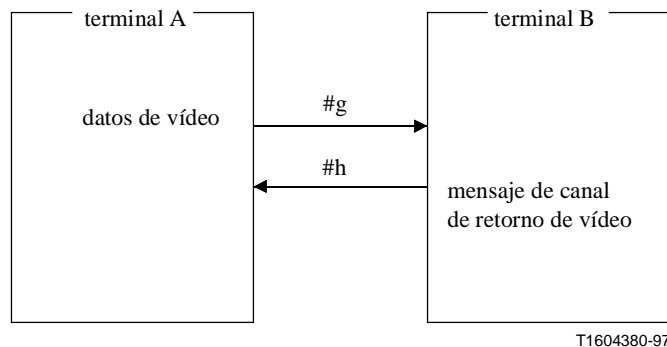
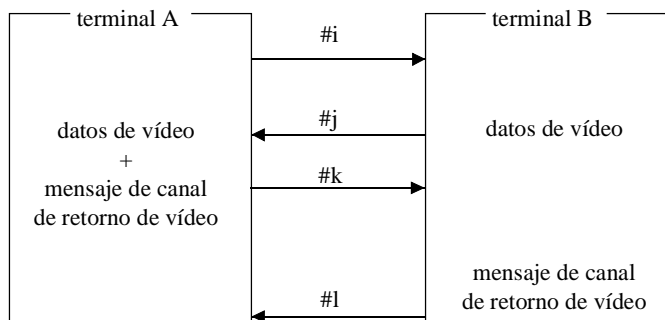


Figura VI.4/H.245 – Modo canal lógico separado en comunicación vídeo unidireccional

Reemplazada por una versión más reciente

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.



T1604390-97

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

H263Capability en mensaje OpenLogicalChannel						
Parámetro de H263Capability	#g	#h	#i	#j	#k	#l
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	ackAndNackMessage	–	ackAndNackMessage	ackAndNackMessage	–	–
separateVideoBackChannel	F	T	F	F	F	T

NP: No presente, T: Verdadero, F: Falso

Reemplazada por una versión más reciente

SERIES DE RECOMENDACIONES DEL UIT-T

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



* 1 6 0 6 4 *

Impreso en Suiza
Ginebra, 1999