

Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

H.248.1

(09/2005)

SERIE H: SISTEMAS AUDIOVISUALES Y
MULTIMEDIOS

Infraestructura de los servicios audiovisuales –
Procedimientos de comunicación

Protocolo de control de las pasarelas: Versión 3

Recomendación UIT-T H.248.1

UIT-T



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

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

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

Recomendación UIT-T H.248.1

Protocolo de control de las pasarelas: Versión 3

Resumen

Para conseguir un sistema con mayor capacidad evolutiva, la presente Recomendación descompone la función pasarela H.323, definida en la Rec. UIT-T H.246, en subcomponentes funcionales y especifica los protocolos utilizados por estos componentes para comunicar. Esto permite que las implementaciones de pasarelas H.323 sean escalables en alto grado e incita a servirse de las capacidades ofrecidas por las redes con conmutación de circuitos (RCC), tales como los conmutadores SS7. Esto también permite que las pasarelas H.323 estén formadas por componentes suministrados por múltiples vendedores, distribuidos a través de múltiples plataformas físicas. La presente Recomendación tiene por objeto añadir capacidades actualmente definidas para sistemas H.323, con el fin de proporcionar nuevas formas de efectuar operaciones ya soportadas en la Recomendación UIT-T H.323.

En esta Recomendación se han hecho algunas mejoras con respecto a la versión 2 de la misma Recomendación:

- la capacidad de definir propiedades de contexto a través de paquetes;
- una propiedad de contexto IEPS;
- una bandera que indica que en la pasarela (MG) hay terminaciones fuera de servicio (OutOfService) que se han de señalar en el momento del registro;
- un nuevo paquete de segmentación de mensajes y procedimientos para el transporte sin segmentación;
- requisitos más precisos para la definición de lotes y nueva plantilla de lote;
- requisitos más precisos para la definición de perfiles y nueva plantilla de perfil;
- nuevas estadísticas en lo que se refiere a los trenes;
- nuevo identificador de petición de señal para distinguir señales similares dentro de una misma lista de señales;
- nuevo parámetro de señal de base para indicar en cuál sentido ha de enviarse la señal;
- dos nuevos tipos de topología;
- nuevo temporizador de retardo entre señales para las señales de una lista de señales;
- nueva forma lingüística ContextIDList (lista de identificadores de contexto) para las respuestas a instrucciones;
- nueva forma lingüística TerminationIDList (lista de identificadores de terminación) para instrucciones y respuestas;
- procedimientos de cambio de servicio más precisos;
- nueva capacidad para que el controlador de pasarela de medios (MGC) regule la velocidad a la que recibe notificaciones;
- nueva capacidad de añadir condiciones de filtrado para las peticiones de auditoría.

Orígenes

La Recomendación UIT-T H.248.1 fue aprobada el 13 de septiembre de 2005 por la Comisión de Estudio 16 (2005-2008) del UIT-T por el procedimiento de la Recomendación UIT-T A.8.

PREFACIO

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

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

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

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

NOTA

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

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

PROPIEDAD INTELECTUAL

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

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

© UIT 2006

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

ÍNDICE

	Página
1 Alcance	1
2 Referencias	1
2.1 Referencias normativas	1
2.2 Referencias informativas	3
3 Definiciones.....	4
4 Abreviaturas, siglas o acrónimos.....	5
5 Convenios	6
6 Modelo de conexión	6
6.1 Contextos.....	8
6.2 Terminaciones	9
6.3 Principios del mecanismo de comodines.....	16
7 Instrucciones.....	18
7.1 Descriptores.....	19
7.2 Interfaz de programación de aplicación para las instrucciones.....	42
8 Transacciones	58
8.1 Parámetros comunes.....	60
8.2 Interfaz de programación de aplicación para las transacciones.....	60
8.3 Mensajes	63
9 Transporte.....	63
9.1 Ordenación de las instrucciones	64
9.2 Protección contra las avalanchas de re arranques.....	64
9.3 Protección contra las avalanchas de Notificaciones.....	65
10 Consideraciones sobre seguridad.....	65
10.1 Protección de las conexiones de protocolo.....	66
10.2 Esquema AH provisional.....	66
10.3 Protección de las conexiones de medios	67
11 Interfaz de control MG-MGC.....	67
11.1 Múltiples MG virtuales.....	67
11.2 Arranque en frío	68
11.3 Negociación de la versión de protocolo	69
11.4 Fallo de una MG.....	69
11.5 Fallo de un MGC	70
11.6 Supervisión de la asociación de control MGC-MG	70
12 Definición de lotes.....	71
12.1 Directrices para la definición de lotes	71
12.2 Directrices para definir los parámetros de eventos y señales	74
12.3 Identificadores	75

	Página
12.4	Registro de lotes 75
13	Definición de perfil..... 75
14	Consideraciones relativas a la IANA..... 76
14.1	Lotes 76
14.2	Códigos de error 76
14.3	Motivos para ServiceChange..... 77
14.4	Perfiles..... 77
Anexo A	– Codificación binaria del protocolo 77
A.1	Codificación de comodines 77
A.2	Especificación de sintaxis en ASN.1 79
A.3	Mapas de dígitos y nombres de trayectos..... 96
Anexo B	– Codificación textual del protocolo..... 97
B.1	Codificación de comodines 97
B.2	Especificación ABNF 97
B.3	Codificación de octetos en hexadecimal 112
B.4	Secuencia de octetos en hexadecimal..... 112
Anexo C	– Rótulos para propiedades de trenes de medios 112
C.1	Atributos generales de los medios 113
C.2	Propiedades de los múltiplex..... 114
C.3	Propiedades generales de los portadores 114
C.4	Propiedades generales de ATM..... 115
C.5	Retransmisión de tramas..... 117
C.6	IP..... 117
C.7	ATM AAL 2 118
C.8	ATM AAL 1 119
C.9	Capacidades portadoras 120
C.10	Propiedades de AAL 5..... 127
C.11	Equivalentes del SDP 127
C.12	H.245 128
Anexo D	– Transporte por IP 129
D.1	Transporte por IP/UDP con empleo de entramado en el nivel de la aplicación (ALF, <i>application level framing</i>)..... 129
D.2	Transporte por TCP 132
Anexo E	– Lotes básicos 134
E.1	Genérico 134
E.2	Lote raíz base (Base Root) 136
E.3	Lote generador de tonos 139
E.4	Lote de detección de tonos (Tone Detection Package) 141
E.5	Lote generador de DTMF básico (Basic DTMF Generator Package)..... 143

	Página
E.6 Lote detección de DTMF (DTMF Detection Package).....	145
E.7 Lote generador de tonos de progresión de la llamada	147
E.8 Lote de detección de tonos de progresión de la llamada	148
E.9 Lote de supervisión de línea analógica.....	149
E.10 Lote de continuidad básico	153
E.11 Lote de red.....	154
E.12 Lote RTP	157
E.13 Lote de circuitos TDM	159
E.14 Lote de segmentación.....	160
E.15 Comportamiento de notificación	163
Anexo F – Procedimientos de cambio de servicio (ServiceChange)	167
F.1 Introducción.....	167
F.2 Definición de asociación de control	169
F.3 Eventos que provocan procedimientos de cambio de servicio	169
F.4 Descripción del elemento de cambio de servicio (ServiceChange)	174
F.5 Utilización de los parámetros de cambio de servicio	176
F.6 ServiceChange comparado con TerminationState.....	179
Apéndice I – Ejemplo de flujos de llamadas.....	180
I.1 Llamada entre dos pasarelas residenciales	180
Apéndice II – Modelo de lote H.248.....	189
Apéndice III – Plantilla para la definición de un perfil H.248.....	192

Recomendación UIT-T H.248.1

Protocolo de control de las pasarelas: Versión 3

1 Alcance

La presente Recomendación define los protocolos que se utilizan entre los elementos de una pasarela multimedia físicamente desglosada. Desde la perspectiva de un sistema no hay diferencias funcionales entre una pasarela desglosada con subcomponentes potencialmente distribuidos entre uno o más dispositivos físicos, y una pasarela monolítica como la descrita en la Rec. UIT-T H.246. La presente Recomendación no define cómo trabajan las pasarelas multipunto, las unidades de control multipunto o las unidades de respuesta vocal interactiva (IVR, *interactive voice response*), sino que crea un marco general adecuado para estas aplicaciones.

Entre las interfaces de redes de paquetes pueden estar las interfaces IP, interfaces ATM y otras. Estas interfaces soportarán una diversidad de sistemas de señalización de redes con conmutación de circuitos (RCC), incluyendo señalización por tonos, red digital de servicios integrados, PU-RDSI, QSIG, y sistema mundial para comunicaciones móviles. También reportarán, en su caso variantes nacionales de estos sistemas de señalización.

Para afirmar que un producto es conforme a la versión 1 de la Rec. UIT-T H.248.1, dicho producto deberá cumplir todas las condiciones obligatorias de la Rec. UIT-T H.248.1 aprobada inicialmente en 06/2000 y publicada nuevamente en 03/2002.

Para afirmar que un producto es conforme a la versión 2 de la Rec. UIT-T H.248.1, dicho producto deberá cumplir todas las condiciones obligatorias de la Rec. UIT-T H.248.1 aprobada inicialmente en 05/2002 y publicada nuevamente en 03/2004.

Para afirmar que un producto es conforme a esta Recomendación, dicho producto deberá cumplir todas las condiciones obligatorias de la Rec. UIT-T H.248.1 (aprobada en 09/2005).

Deberá indicarse cuál es la versión del protocolo de un producto, mencionando ServiceChangeVersion '1' para referirse a la Rec. UIT-T H.248.1 (03/2002), '2' para referirse al corrigendum 1 a la Rec. UIT-T H.248.1 (03/2004) y '3' para referirse a la Rec. UIT-T H.248.1 (aprobada en 09/2005).

2 Referencias

2.1 Referencias normativas

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

- Recomendación UIT-T E.106 (2003), *Plan internacional de preferencias en situaciones de emergencia para actuaciones frente a desastres*.
- Recomendación UIT-T H.225.0 (2003), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.

- Recomendación UIT-T H.235.0 (2005), *Marco de seguridad H.323: Marco de seguridad para sistemas multimedia de la serie H (H.323 y otros basados en H.245)*.
- Recomendación UIT-T H.245 (2005), *Protocolo de control para comunicación multimedia*.
- Recomendación UIT-T H.246 (1998), *Interfuncionamiento de terminales multimedia de la serie H con terminales multimedia de la serie H y terminales vocales/de banda vocal por la RTGC y la RDSI*.
- Recomendación UIT-T H.248.4 (2000), *Protocolo de control de las pasarelas: Transporte por el protocolo de transmisión de control de tren, más corrigendum 1 (2004)*.
- Recomendación UIT-T H.248.5 (2000), *Protocolo de control de las pasarelas: Transporte por redes del modo de transferencia asíncrono*.
- Recomendación UIT-T H.248.8 (2005), *Protocolo de control de las pasarelas: Descripción de los códigos de error y de los motivos de cambio de servicio*.
- Recomendación UIT-T H.248.14 (2002), *Protocolo de control de las pasarelas: Lote de temporizador de inactividad*.
- Recomendación UIT-T H.323 (2003), *Sistemas de comunicación multimedia basados en paquetes*.
- Recomendación UIT-T I.363.1 (1996), *Especificación de la capa de adaptación del modo transferencia asíncrono de la RDSI-BA: Capa de adaptación del modo transferencia asíncrono tipo 1*.
- Recomendación UIT-T I.363.2 (2000), *Especificación de la capa de adaptación del modo de transferencia asíncrono de la RDSI-BA: Capa de adaptación del modo de transferencia asíncrono tipo 2*.
- Recomendación UIT-T I.363.5 (1996), *Especificación de la capa de adaptación del modo transferencia asíncrono de la RDSI-BA: Capa de adaptación del modo transferencia asíncrono tipo 5*.
- Recomendación UIT-T I.366.1 (1998), *Subcapa de convergencia específica del servicio de segmentación y reensamblado para la capa de adaptación del modo transferencia asíncrono tipo 2*.
- Recomendación UIT-T I.366.2 (2000), *Subcapa de convergencia específica de servicio de capa de adaptación del modo de transferencia asíncrono tipo 2 para servicios de banda estrecha, más corrigendum 1 (2002)*.
- Recomendación UIT-T I.371 (2004), *Control de tráfico y control de congestión en RDSI-BA*.
- Recomendación UIT-T Q.763 (1999), *Sistema de señalización N.º 7 – Formatos y códigos de la parte usuario de la RDSI, más enmienda 3 (2004)*.
- Recomendación UIT-T Q.765.5 (2004), *Sistema de señalización N.º 7 – Mecanismo de transporte de aplicación: Control de llamada independiente del portador*.
- Recomendación UIT-T Q.931 (1998), *Especificación de la capa 3 de la interfaz usuario-red de la red digital de servicios integrados para el control de la llamada básica, más enmienda 1 (2002): Extensiones para soportar el equipo de multiplexación digital*.
- Recomendación UIT-T Q.2630.1 (1999), *Protocolo de señalización de la capa de adaptación del modo transferencia asíncrono tipo 2 – Conjunto de capacidades 1*.

- Recomendación UIT-T Q.2931 (1995), *Sistema de señalización digital de abonado N.º 2 – Especificación de la capa 3 de la interfaz usuario-red para el control de llamada/conexión básica*, más enmienda 4 (1999).
- Recomendación UIT-T Q.2941.1 (1997), *Sistema de señalización digital de abonado N.º 2 – Transporte de identificadores genéricos*.
- Recomendación UIT-T Q.2961.1 (1995), *Sistema de señalización digital de abonado N.º 2 – Parámetros de tráfico adicionales: Capacidades de señalización adicionales que soportan parámetros de tráfico para la opción de rotulado y el conjunto de parámetros de velocidad de célula sustentable*.
- Recomendación UIT-T Q.2961.2 (1997), *Sistema de señalización digital de abonado N.º 2 – Parámetros de tráfico adicionales: Soporte de la capacidad de transferencia del modo de transferencia asíncrono en el elemento información de capacidad portadora de banda ancha*, más corrigendum 1 (1999).
- Recomendación UIT-T Q.2965.1 (1999), *Sistema de señalización digital de abonado N.º 2 – Soporte de clases de calidad de servicio*, más enmienda 1 (2000).
- Recomendación UIT-T Q.2965.2 (1999), *Sistema de señalización digital de abonado N.º 2 – Señalización de parámetros de calidad de servicio individuales*.
- Recomendación UIT-T V.76 (1996), *Multiplexor genérico que utiliza procedimientos basados en LAPM de la Recomendación V.42*, más corrigendum 1 (2005).
- Recomendación UIT-T X.213 (2001) | ISO/CEI 8348:2002, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de red*.
- Recomendación UIT-T X.680 (2002) | ISO/CEI 8824-1:2004, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica*, más enmienda 2 (2004); *Armonización con las modificaciones introducidas en la Rec. UIT-T X.660 | ISO/CEI 9834-1 para los identificadores en la notación de valores de indentificador de objeto*.
- Recomendación UIT-T X.690 (2002) | ISO/CEI 8825-1:2002, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica, de las reglas de codificación canónica y de las reglas de codificación distinguida*, más enmienda 1 (2003); *Soporte de EXTENDED-XER*.
- ISO/CEI 10646 (2003), *Information technology – Universal Multiple-Octet Coded character Set (UCS)*.
- ATM Forum (1996), *ATM User-Network Interface (UNI) Signalling Specification – Version 4.0*.
- IETF RFC 1006 (1987), *ISO Transport Service on top of the TCP, Version 3*.
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF*.
- IETF RFC 2327 (1998), *SDP: Session Description Protocol*.
- IETF RFC 2402 (1998), *IP Authentication Header*.
- IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP)*.

2.2 Referencias informativas

- Recomendación UIT-T E.180/Q.35 (1998), *Características técnicas de los tonos para el servicio telefónico*.

- Recomendación UIT-T G.711 (1988), *Modulación por impulsos codificados (MIC) de frecuencias vocales*.
- Recomendación UIT-T H.221 (2004), *Estructura de trama para un canal de 64 a 1920 kbit/s en teleservicios audiovisuales*.
- Recomendación UIT-T H.223 (2001), *Protocolo de multiplexación para comunicación multimedios a baja velocidad binaria*.
- Recomendación UIT-T H.226 (1998), *Protocolo de agregado de canales para funcionamiento multienlace en redes con conmutación de circuitos*.
- Recomendación UIT-T Q.724 (1998), *Procedimientos de señalización de la parte usuario de telefonía*, más enmienda 1 (1993).
- Recomendación UIT-T Q.764 (1999), *Sistema de señalización N.º 7 – Procedimientos de señalización de la parte usuario de la RDSI*, más enmienda 3 (2004).
- Recomendación UIT-T Q.1902.4 (2001), *Protocolo de control de llamada independiente del portador (conjunto de capacidades 2): Procedimientos de llamada básica*, más enmienda 2 (2004).
- IETF RFC 768 (1980), *User Datagram Protocol*.
- IETF RFC 791 (1981), *Internet protocol*.
- IETF RFC 793 (1981), *Transmission control protocol*.
- IETF RFC 1661 (1994), *The Point-to-Point Protocol (PPP)*.
- IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.
- IETF RFC 2460 (1998), *Internet Protocol, Version 6 (IPv6) Specification*.
- IETF RFC 2805 (2000), *Media Gateway Control Protocol Architecture and Requirements*.
- IETF RFC 3261 (2002), *SIP: Session Initiation Protocol*.
- IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.
- IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control*.

3 Definiciones

En esta Recomendación se definen los términos siguientes.

3.1 pasarela de acceso: Tipo de pasarela que proporciona una interfaz usuario-red (UNI), por ejemplo en la RDSI.

3.2 descriptor: Elemento sintáctico del protocolo que agrupa propiedades conexas. Por ejemplo, el controlador puede determinar las propiedades de un flujo de medios en la pasarela incluyendo el descriptor apropiado en una instrucción.

3.3 pasarela de medios (MG, *media gateway*): La pasarela de medios convierte medios proporcionados con un formato dado en un tipo de red, en medios con el formato requerido en otro tipo de red. Por ejemplo, una MG podría terminar canales portadores procedentes de una red con conmutación de circuitos (por ejemplo, DS0) y trenes de medios procedentes de una red de paquetes (por ejemplo, trenes RTP en una red IP). Esta pasarela podría procesar señales de audio, vídeo y multimedia T.120 separadamente o en combinación, y podrá efectuar traslaciones de medios dúplex. La MG también puede reproducir mensajes de audio/vídeo y realizar otras funciones IVR, o efectuar comunicaciones conferencias de medios.

3.4 controlador de pasarela de medios (MGC, *media gateway controller*): Controla las partes del estado de la llamada que atañen al control de la conexión para canales de medios en una MG.

3.5 unidad de control multipunto (MCU, *multipoint control unit*): Entidad que controla el establecimiento y se encarga de la coordinación de una conferencia multiusuario, que incluye habitualmente el procesamiento de señales de audio, vídeo y datos.

3.6 pasarela residencial: Pasarela que interconecta una línea analógica a un red de paquetes. Una pasarela residencial contiene habitualmente una o dos líneas analógicas y está situada en las instalaciones del cliente.

3.7 pasarela de señalización asociada a la facilidad, en redes con conmutación de circuitos: Esta función contiene la interfaz de señalización RCC que termina enlaces de señalización SS7, RDSI y otros, cuando el canal de control de llamada y los canales portadores están coubicados en un mismo tramo físico.

3.8 pasarela de señalización no asociada a la facilidad, en redes con conmutación de circuitos: Esta función contiene la interfaz de señalización RCC que termina enlaces SS7 u otros enlaces de señalización cuando los canales de control de llamadas están separados de los canales portadores.

3.9 tren: Flujo bidireccional de medios o de control recibido/enviado por una pasarela de medios como parte de una llamada o conferencia.

3.10 troncal: Canal de comunicación entre dos sistemas de conmutación, por ejemplo DS0 en una línea T1 o E1.

3.11 pasarela de entronque: Pasarela entre una red con conmutación de circuitos y una red de paquetes, que generalmente termina un gran número de circuitos digitales.

4 Abreviaturas, siglas o acrónimos

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

AAD	Retardo de acuse de recibo medio (<i>average acknowledgement delay</i>)
AAL	Capa de adaptación ATM (<i>ATM adaptation layer</i>)
ADEV	Desviación media (<i>average deviation</i>)
ALF	Entramado de nivel de aplicación (<i>application level framing</i>)
ATM	Modo de transferencia asíncrono (<i>asynchronous transfer mode</i>)
C	Contexto
CAS	Señalización asociada al canal (<i>channel associated signalling</i>)
DNS	Sistema de nombres de dominio (<i>domain name system</i>)
DTMF	Multifrecuencia bitono (<i>dual tone multi-frequency</i>)
FAS	Señalización asociada a la facilidad (<i>facility associated signalling</i>)
GSM	Sistema global para comunicaciones móviles (<i>global system for mobile communications</i>)
GW	Pasarela (<i>gateway</i>)
IANA	Autoridad de asignación de números Internet (<i>Internet assigned numbers authority</i>) (reemplazada por ICANN)
ICANN	Internet corporation for assigned names and numbers

IEPS	Plan internacional de preferencias en situaciones de emergencia (<i>international emergency preference scheme</i>)
IP	Protocolo Internet (<i>Internet protocol</i>)
IS	En servicio (<i>in-service</i>)
IVR	Respuesta vocal interactiva (<i>interactive voice response</i>)
MG	Pasarela de medios (<i>media gateway</i>)
MGC	Controlador de pasarela de medios (<i>media gateway controller</i>)
MWD	Periodo de espera máximo (<i>maximum waiting delay</i>)
NFAS	Señalización no asociada a la facilidad (<i>non-facility associated signalling</i>)
OoS	Fuera de servicio (<i>out-of-service</i>)
PRI	Interfaz de velocidad primaria (<i>primary rate interface</i>)
PU-RDSI	Parte usuario de la red digital de servicios integrados
QoS	Calidad de servicio (<i>quality of service</i>)
RCC	Red con conmutación de circuitos
RTP	Protocolo de transporte en tiempo real (<i>real-time transport protocol</i>)
RTPC	Red telefónica pública conmutada
SC	Cambio de servicio (<i>ServiceChange</i>)
SG	Pasarela de señalización (<i>signalling gateway</i>)
SS7	Sistema de señalización N.º 7 (<i>signalling system N.º 7</i>)
T, Term	Terminación

5 Convenios

En esta Recomendación se utilizan los siguientes convenios en lo relativo al nivel de obligación.

La obligación firme se expresa con el futuro simple del verbo (futuro de mandato) o expresiones con significado de obligación. La conveniencia, es decir una acción aconsejada pero no obligatoria, se expresa con el condicional del verbo modal "deber" o expresiones que indican conveniencia. La opción se expresa mediante el presente de indicativo del verbo "poder" o expresiones de posibilidad.

6 Modelo de conexión

El modelo de conexión para el protocolo describe las entidades lógicas, u objetos, dentro de la pasarela de medios que pueden ser controlados por el controlador de pasarela de medios. Los principales términos utilizados en el modelo de conexión para designar conceptos abstractos son las terminaciones y los contextos.

Una *terminación* actúa como una fuente y/o un sumidero de uno o más trenes. En una conferencia multimedia, una terminación puede ser multimedia y servir de fuente o sumidero para múltiples trenes de medios. Los parámetros de los trenes de medios, así como los parámetros de portador están encapsulados dentro de la terminación.

Un *contexto* es una asociación entre varias terminaciones. Hay un tipo especial de contexto, el contexto *NULL* (*nulo*), que contiene todas las terminaciones que no están presentes en ningún otro contexto y, por ende, no están asociadas con ninguna otra terminación. Por ejemplo, en una pasarela

de acceso fraccionada, todos los canales en reposo se representan por terminaciones en el contexto NULL.

A continuación se presenta una descripción gráfica de estos conceptos. El diagrama de la figura 1 muestra varios ejemplos, y no se pretende que sea exhaustivo. La casilla con asterisco en cada uno de los contextos representa la asociación lógica de terminaciones implicadas por el contexto.

Pasarela de medios

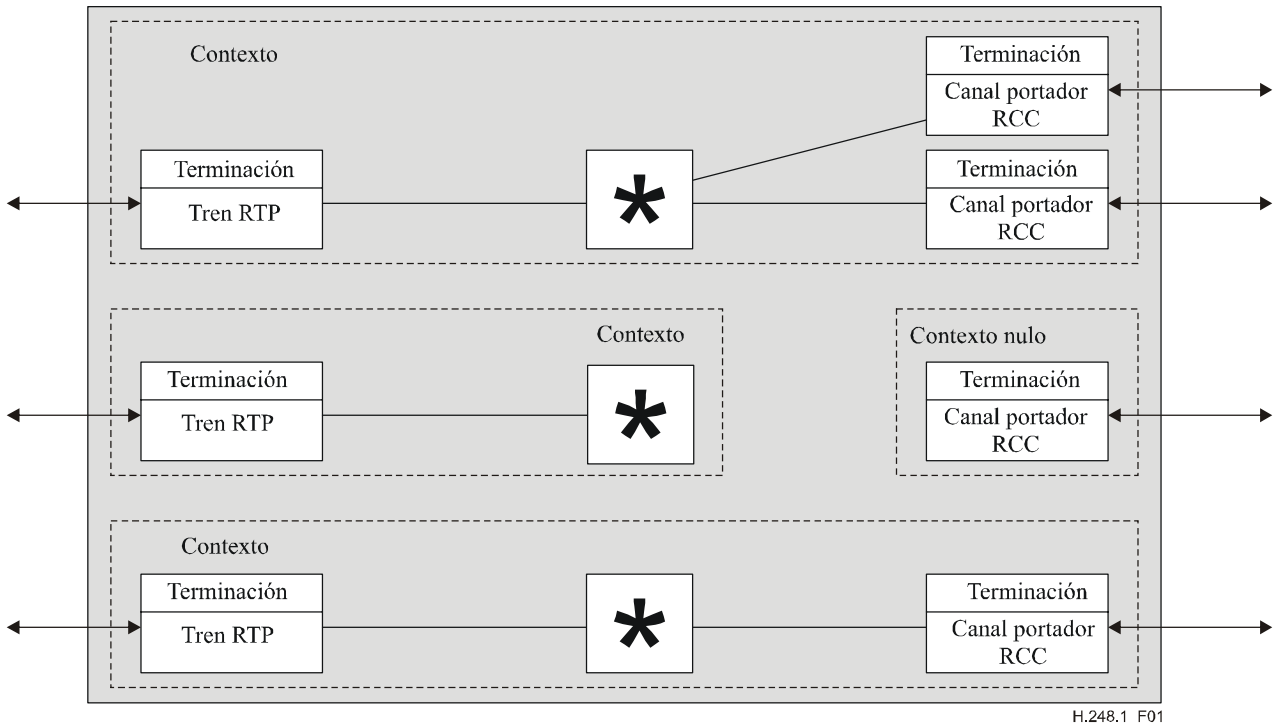


Figura 1/H.248.1 – Ejemplo de modelo de conexión H.248.1

El siguiente ejemplo (figura 2) representa un posible proceso de llamada en espera en una pasarela de acceso fraccionada e ilustra la reubicación de una terminación entre contextos. Las terminaciones T1 y T2 corresponden al contexto C1 en una llamada audio bidireccional. Una segunda llamada audio está en espera de T1 desde la terminación T3. T3 está sola en el contexto C2. T1 acepta la llamada procedente de T3 y coloca a T2 en retención. Como resultado de esta acción T1 pasa al contexto C2 como se muestra en la figura 3.

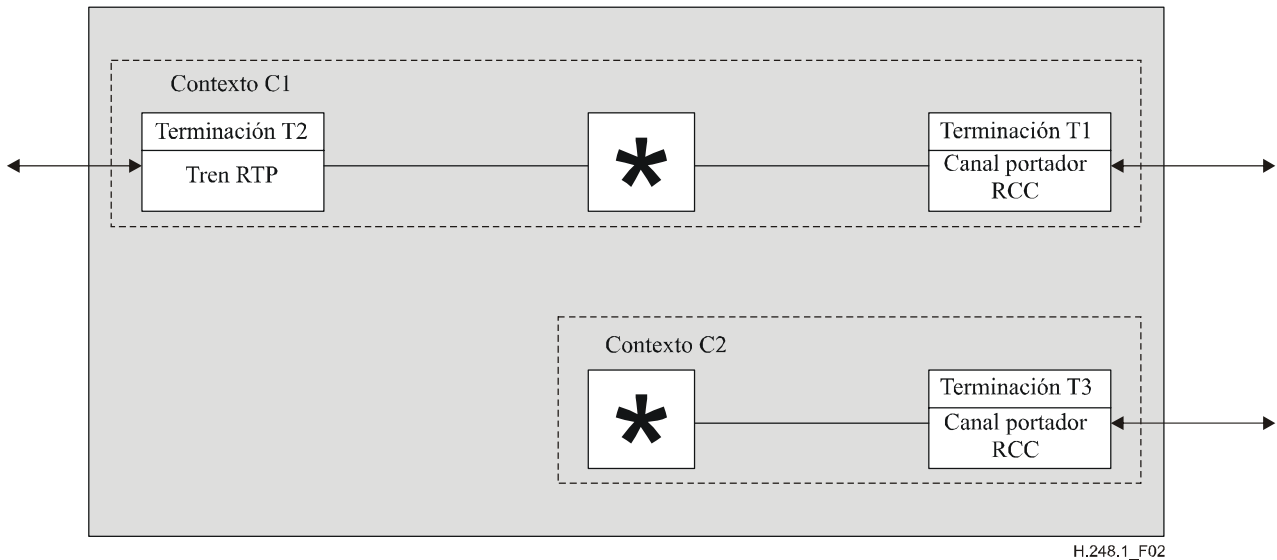


Figura 2/H.248.1 – Ejemplo de situación/aviso de llamada en espera en T1

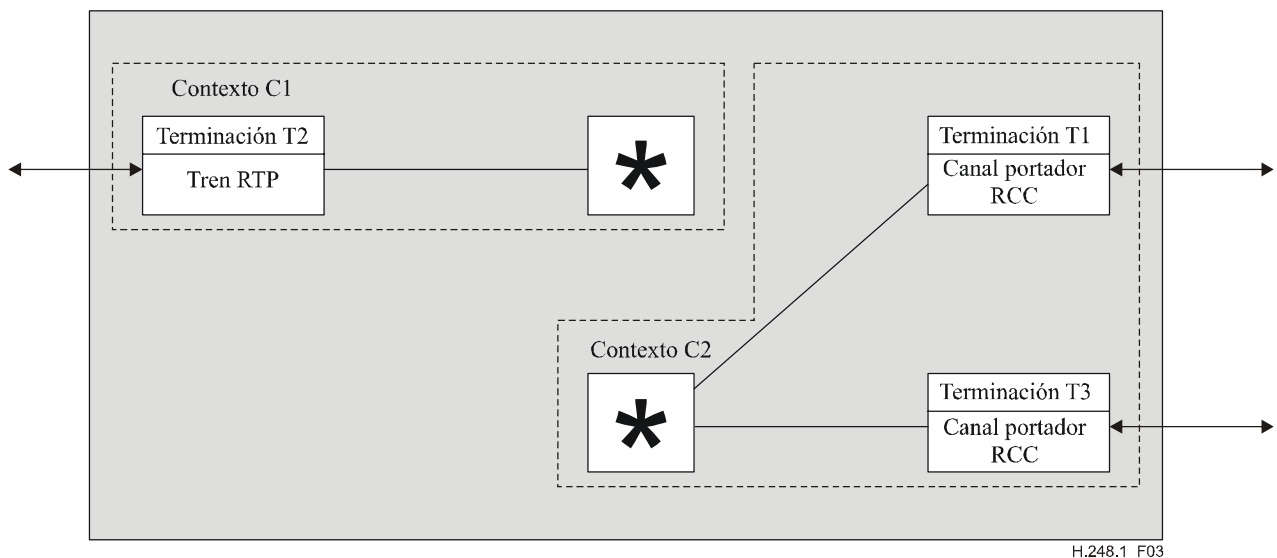


Figura 3/H.248.1 – Ejemplo de situación/contestación de llamada en espera en T1

6.1 Contextos

Un contexto es una asociación entre varias terminaciones. El contexto describe la topología (quién oye/ve a quién) y los parámetros de combinación y/o conmutación de medios si intervienen en la asociación más de dos terminaciones.

Existe un contexto especial denominado contexto *NULL*. Contiene terminaciones que no están presentes en ningún otro contexto y que, por lo tanto, no están asociadas a ninguna otra terminación. Los parámetros de las terminaciones en el contexto *NULL* pueden ser examinados o modificados, y se puede detectar eventos en esas terminaciones.

En general, se utiliza una instrucción Añadir (Add) para añadir terminaciones a contextos. Si el MGC no especifica un contexto existente al cual se añaden las terminaciones, la MG crea un nuevo contexto. Se puede suprimir una terminación de un contexto con una instrucción Substraer

(Subtract) y se puede desplazar una terminación de un contexto a otro con la instrucción Desplazar. Una terminación no puede formar parte de dos contextos al mismo tiempo.

El número máximo de terminaciones en un contexto es una propiedad de la MG. Las pasarelas de medios que ofrecen solamente conectividad punto a punto podrían permitir, como máximo, dos terminaciones por contexto. Las pasarelas de medios que soportan conferencias multipunto podrían permitir tres o más terminaciones por contexto.

6.1.1 Atributos de contextos y descriptores

Los atributos de contextos son:

- ContextID.
- El descriptor de topología (quién oye/ve a quién):
La topología de un contexto describe el flujo de medios entre las terminaciones dentro de un contexto. En cambio, la propiedad modo de una terminación ("SendOnly"/"RecvOnly"/...) describe el flujo de los medios en el egreso/ingreso de la pasarela de medios.
- En un contexto, la prioridad se utiliza para proporcionar a la pasarela de medios información sobre un determinado tratamiento de precedencia para un contexto. El MGC puede también utilizar la prioridad para controlar autónomamente la precedencia del tráfico en la MG de una manera regular en ciertas situaciones (por ejemplo, rearranque), cuando hay que tratar una gran cantidad de contextos simultáneamente. La prioridad 0 es la más baja y la prioridad 15 es la más alta.
- También se puede proporcionar un indicador de una llamada de emergencia para permitir un tratamiento de preferencia en la MG.
- Se proporciona un indicador de llamada IEPS para que se apliquen las características y técnicas de E.106.
- Un descriptor de atributo de contexto (ContextAttribute) que permite definir otros atributos de contexto utilizando el mecanismo de extensión de lotes (véase 7.1.19).

6.1.2 Creación, supresión y modificación de contexto

El protocolo puede utilizarse para crear (implícitamente) contextos y modificar los valores de parámetros de contextos existentes. El protocolo tiene instrucciones para añadir terminaciones a contextos, substraerlas de los contextos, y desplazar terminaciones entre contextos. Los contextos se suprimen implícitamente cuando se retira o se desplaza la última terminación que les queda.

6.2 Terminaciones

Una terminación es una entidad lógica en una MG que actúa como fuente y/o sumidero de trenes de medios y/o control. Una terminación se describe por un número de propiedades que la caracterizan, las cuales se agrupan en un conjunto de descriptores que se incluyen en instrucciones. Las terminaciones tienen identificadores únicos (TerminationID) asignados por la MG en el momento de su creación.

Las terminaciones que representan entidades físicas existen de forma casi permanente. Por ejemplo, una terminación que representa un canal multiplexado por división en el tiempo podría existir durante todo el tiempo en que estuviera en servicio en la pasarela. Las terminaciones que representan flujos efímeros, como los flujos de los protocolos de transporte en tiempo real (RTP, *real time transport protocol*), sólo existen, generalmente, mientras se estén utilizando.

Las terminaciones efímeras se crean por medio de una instrucción Añadir y se suprimen por medio de una instrucción Substraer. En cambio, cuando se añade una terminación física a un contexto, o se substraer de un contexto, se toma del contexto NULL, o se coloca en el contexto NULL, respectivamente.

Se pueden aplicar señales a las terminaciones (véase 7.1.11). Se pueden programar terminaciones para detectar eventos, cuya aparición puede provocar la notificación de mensajes al MGC, o una acción de la MG. En una terminación se pueden acumular estadísticas que se comunican al MGC a petición (por medio de la instrucción AuditValue (auditoría de valores), véase 7.2.5) y cuando la terminación deje de existir o se ponga nuevamente en el contexto NULL mediante la instrucción Substraer.

Las pasarelas multimedia pueden procesar trenes de medios multiplexados. Por ejemplo, la Rec. UIT-T H.221 describe una estructura de trama para varios trenes de medios multiplexados en un número de canales digitales a 64 kbit/s. Este caso se trata en el modelo de conexión de la manera siguiente. Para cada canal portador que transporta una parte de los trenes multiplexados hay una "terminación de portador" física o efímera. Las terminaciones que actúan como fuente/sumidero de canales digitales se conectan a una terminación separada denominada la "terminación multiplexora". La terminación multiplexora es una terminación efímera que representa una sesión orientada a trama. El descriptor de múltiplex para esta terminación describe el múltiplex utilizado (por ejemplo H.221 para una sesión H.320) e indica el orden en que los canales digitales contenidos se ensamblan para formar una trama.

Las terminaciones multiplexoras pueden conectarse en cascada (por ejemplo, múltiplex H.226 de canales digitales que alimentan un múltiplex H.223 que soporta una sesión H.324).

Para describir cada tren de medios se utilizan descriptores de tren en la terminación multiplexora. Estos trenes de medios pueden corresponder a trenes cuya fuente/sumidero depende de terminaciones en el contexto que no son terminaciones de portador que soportan la terminación multiplexora. Cada terminación multiplexora soporta un solo tren de datos. Estos trenes de datos no aparecen explícitamente en la terminación multiplexora y están ocultos del resto del contexto.

En las figuras 4, 5 y 6 se representan aplicaciones típicas de la terminación multiplexora y del descriptor de múltiplex.

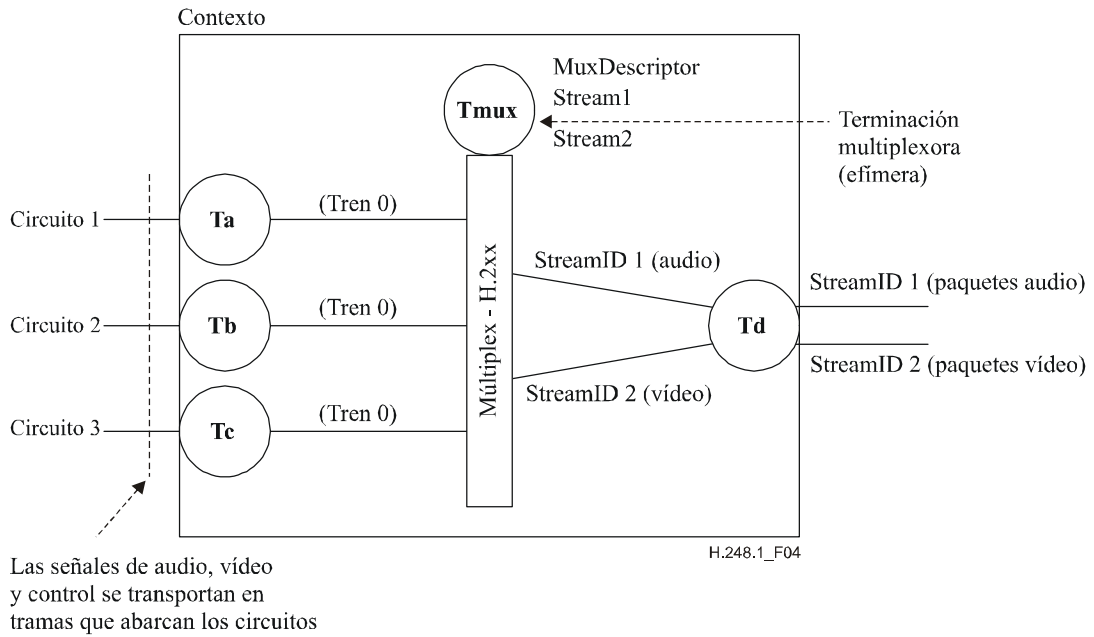


Figura 4/H.248.1 – Escenario de terminación multiplexada – Circuito a paquete

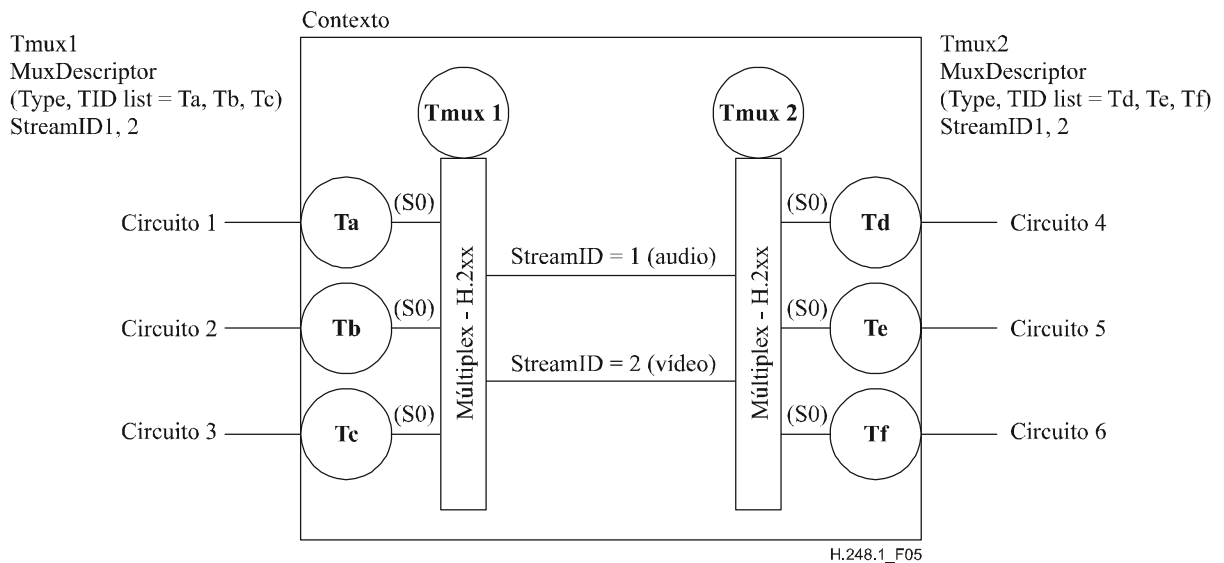


Figura 5/H.248.1 – Escenario de terminación multiplexada – Circuito a circuito

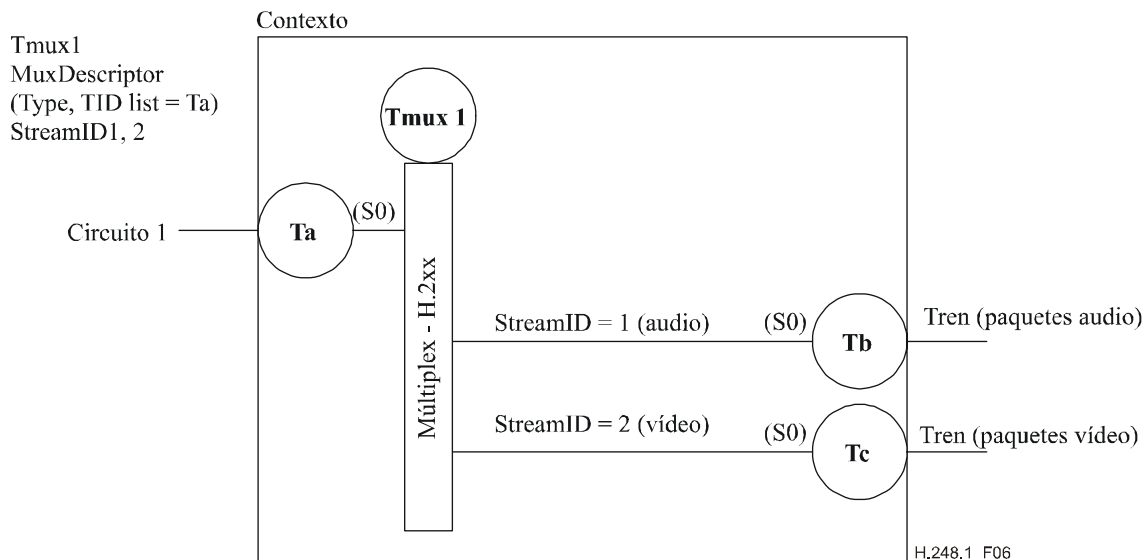


Figura 6/H.248.1 – Escenario de terminación multiplexada – Una a varias terminaciones

A diferencia de las terminaciones multiplexoras descritas en el párrafo anterior, una terminación de portador multiplexado, como puede ser el portador de capa de adaptación ATM de tipo 2 (ATM AAL), no transporta trenes de medios. Sólo se incluyen para hacer los modelos de creación y destrucción del verdadero portador. Cuando debe crearse un nuevo portador multiplexado, se crea una terminación efímera en un contexto establecido para este fin. Cuando la terminación es abstraída, el portador multiplexado se destruye.

6.2.1 Dinámica de las terminaciones

El protocolo puede utilizarse para crear nuevas terminaciones y modificar valores de propiedades de terminaciones existentes. Estas modificaciones incluyen la posibilidad de añadir o suprimir eventos y/o señales. Las propiedades de terminaciones, y los eventos y señales se describen en las subcláusulas que siguen. Un MGC sólo puede liberar/modificar terminaciones, y los recursos que tales terminaciones representan, cuando están en el contexto NULO o se han incluido, por ejemplo mediante la instrucción Añadir.

6.2.2 TerminationID

Se hace referencia a las terminaciones utilizando un TerminationID, que es un esquema arbitrario elegido por la MG.

Los TerminationID de terminaciones físicas se proporcionan en la pasarela de medios. Se puede optar por que los TerminationID consten de una estructura. Por ejemplo, un TerminationID puede consistir en un grupo de circuitos troncales y una troncal dentro del grupo.

Con los TerminationID puede utilizarse un mecanismo que emplea dos tipos de comodines. Estos dos comodines son ALL y CHOOSE. El primero se utiliza para direccionar múltiples terminaciones de una sola vez, mientras que el segundo se utiliza para indicar a una pasarela de medios que tiene que seleccionar una terminación que satisfaga el TerminationID parcialmente especificado. Esto permite, por ejemplo, que un MGC ordene a la MG que elija un circuito perteneciente a un grupo de troncales.

También se pueden especificar los TerminationID en una lista (TerminationIDList) cuya utilización se recomienda en casos en los que no puede haber una estructura jerárquica de TerminationID y cuando no se quieran enviar instrucciones separadamente para cada TerminationID.

6.2.3 Lotes

En los diferentes tipos de pasarelas se pueden implementar terminaciones con características muy diferentes. Las variaciones en las terminaciones se tienen en cuenta en el protocolo permitiendo que las terminaciones tengan propiedades, eventos, señales y estadísticas facultativos implementados por las MG.

Para conseguir la interoperabilidad MG/MGC, esas opciones se agrupan en lotes, y normalmente una terminación realiza un conjunto de esos lotes. En la cláusula 12 puede encontrarse más información sobre la definición de lotes. Un MGC puede hacer una consulta de terminación para determinar los lotes que realiza.

Se utilizan identificadores (ID) para nombrar las propiedades, eventos, señales y estadísticas definidos en lotes, así como los parámetros de los mismos. Los identificadores están delimitados. Para cada lote, los PropertyID, EventID, SignalID, StatisticsID y ParameterID tienen espacios de nombre únicos, pudiéndose utilizar el mismo identificador en cada uno de ellos. Dos PropertyID de lotes diferentes también pueden tener el mismo identificador, etc.

Para soportar un determinado lote, la MG tiene que reconocer todas las propiedades, señales, eventos y estadísticas definidos en ese lote. También tiene que soportar todos los parámetros de señal y de evento lote. Si no se ha implementado la funcionalidad correspondiente a todas esas propiedades, señales, eventos y estadísticas, la MG no devolverá un error de sintaxis o de ID desconocido para cada uno de dichos elementos, sino que en su lugar debería devolver un mensaje de error 501 "No implementado".

La MG puede soportar un subconjunto de los valores de un lote para determinada propiedad o parámetro. Cuando el MGC especifique un valor no soportado, la MG habrá de retornar el error 501, "No implementado".

Al extender los lotes se pueden nombrar las propiedades, eventos, señales y estadísticas definidos en el lote de base utilizando el nombre del lote extendido o el nombre del lote de base. Por ejemplo, si el lote A define el evento e1, y el lote B extiende el lote A, entonces B/e1 es un evento para una terminación que implementa el lote B. Por definición la MG DEBE también implementar el lote de base, pero es facultativo publicar el lote de base como una interfaz permitida. Si en efecto publica a A, entonces A sería informada en el descriptor de lote en AuditValue así como B, y el evento A/e1 estaría disponible en una terminación. Si la MG no publica a A, entonces sólo B/e1 estaría disponible. Si se publica mediante AuditValue, A/e1 y B/e1 son el mismo evento.

Para mejorar la interoperabilidad y la compatibilidad con sistemas anteriores, una MG puede publicar todos los lotes soportados por sus terminaciones, incluidos los lotes de base de los cuales se han derivado lotes extendidos. Una excepción la constituyen los casos en los que los lotes de base están expresamente definidos como "designados para ser extendidos solamente"

6.2.4 Propiedades de las terminaciones y descriptores

Las terminaciones tienen propiedades. Las propiedades tienen PropertyID unívocos. La mayor parte de las propiedades tienen valores por defecto, que están definidos explícitamente en esta especificación de protocolo o en un lote (véase la cláusula 12) o establecidos por el suministro. Si no son suministrados de otra forma, las propiedades en todos los descriptores, a excepción de TerminationState y LocalControl toman el valor por defecto vacío/"ningún valor" cuando una terminación se crea por primera vez o retorna al contexto NULL. Cuando una terminación se crea por primera vez o retorna al contexto NULL, este estado representa una línea, troncal u otra entidad en reposo (*idle*). Los contenidos por defecto de ambas excepciones se describen en 7.1.5 y 7.1.7.

La configuración de una propiedad en la MG prevalece sobre cualquier valor por defecto, tanto si ha sido suministrado en esta especificación de protocolo o en un lote. Por tanto, si es esencial que el MGC ejerza un pleno control sobre los valores de la propiedad de una terminación, deberá suministrar valores explícitos cuando añada la terminación a un contexto. Como otra posibilidad,

para una terminación física el MGC puede determinar cualquier valor de propiedad proporcionado mediante una auditoría de terminación mientras ésta está en el contexto NULL.

Hay propiedades que son comunes a todas las terminaciones, y propiedades que son específicas de trenes de medios. Las propiedades comunes se denominan también propiedades de TerminationState. Cada tren de medios tiene propiedades locales y propiedades de los flujos recibidos y transmitidos.

Las propiedades no incluidas en el protocolo de base se definen en lotes. Estas propiedades se designan por un nombre que consiste en el nombre de lote y un PropertyID. La mayor parte de las propiedades tienen valores por defecto indicados en la descripción del lote. Las propiedades pueden ser de lectura solamente o de lectura/escritura. Se puede comprobar los valores posibles de una propiedad, y también sus valores actuales. En el caso de propiedades de lectura/escritura, el MGC puede fijar sus valores. Una propiedad se puede declarar como "global", que tiene un solo valor compartido por todas las terminaciones que realizan el lote. Las propiedades conexas se agrupan en descriptores por razones de conveniencia.

Cuando se añade una terminación a un contexto, el valor de sus propiedades de lectura/escritura puede fijarse incluyendo los descriptores apropiados como parámetros de la instrucción Añadir. De manera similar, el valor de una propiedad de una terminación en un contexto se puede modificar por la instrucción Modificar. Las propiedades pueden también haber modificado sus valores cuando se desplaza una terminación de un contexto a otro como resultado de una instrucción Desplazar. En algunos casos, se retornan los descriptores como resultado de una instrucción.

Al fijar las propiedades en diferentes terminaciones del mismo contexto, se ordena implícitamente a la MG que realice determinadas funciones. Por ejemplo: si hay un códec G.711 en la Terminación A y un códec G.729 en la B, la MG activará una función de transcodificación tan pronto haya un flujo de medios entre ambas terminaciones es decir, en cada terminación especificará un estado que no sea "Inactivo" para la propiedad de modo.

NOTA – Para evitar la activación innecesaria de recursos de la MG, conviene que el MGC especifique el modo "Inactivo" para una terminación y un tren datos, hasta tanto que se determinen las propiedades que habrá de utilizar dicho tren.

En general, si un descriptor se omite completamente en una de las instrucciones antes mencionadas, las propiedades en ese descriptor mantienen sus anteriores valores para la terminación o terminaciones sobre las que actúa la instrucción. Por otra parte, si algunas propiedades de lectura/escritura se omiten en el descriptor de una instrucción (es decir, si el descriptor se especifica sólo parcialmente), las propiedades serán reinicializadas a sus valores por defecto para la terminación o terminaciones sobre las que actúa la instrucción, a menos que el paquete especifique otro comportamiento. Para más detalles, véase 7.1 que trata de los distintos descriptores.

Este comportamiento también se aplica a las señales y los eventos y sus respectivos parámetros. En la configuración de los descriptores de eventos habría que incluir el RequestID y todos los parámetros de los eventos incluidos. Toda configuración de descriptores de eventos en la MG debería copiarse en el MGC para evitar respuestas de error a instrucciones Notificar de la MG.

En el siguiente cuadro se indican todos los posibles descriptores y su utilización. No todos los descriptores son válidos como parámetros de entrada o de salida de todas las instrucciones.

Nombre del descriptor	Descripción
Módem	Identifica el tipo de módem, y las propiedades cuando proceda (nota).
Mux	Describe el tipo de múltiplex para terminaciones multimedia (por ejemplo, H.221, H.223, H.225.0) y las terminaciones que forman el múltiplex de entrada.
Medios	Lista de especificaciones de trenes de medios (véase 7.1.4).
TerminationState	Propiedades de una terminación que no son específicas del tren (esas propiedades pueden estar definidas en lotes).
Tren	Lista de descriptores distante (Remote), local y de control local (LocalControl) para un mismo tren.
Local	Contiene propiedades que especifican los flujos de medios que la MG recibe de la entidad distante.
Distante	Contiene propiedades que especifican los flujos de medios que la MG envía a la entidad distante.
LocalControl	Contiene propiedades que son de interés para la MG y el MGC (esas propiedades pueden definirse en lotes).
Eventos	Describe eventos que son detectados por la MG y lo que se debe hacer cuando se detecta un evento.
EventBuffer	Describe eventos que deben ser detectados por la MG cuando está activo el registro temporal de eventos.
Señales	Describe señales (véase 7.1.11) aplicables a las terminaciones.
Auditoría	Identifica la información solicitada en las instrucciones de auditoría.
Lotes	En AuditValue, retorna una lista de lotes realizados por la terminación.
DigitMap	Define patrones con los que deben concordar las secuencias de un conjunto específico de eventos de tal forma que puedan comunicarse como un grupo y no separadamente.
ServiceChange	En ServiceChange, qué cambio de servicio se produjo, por qué motivo, etc.
ObservedEvents	En Notificar o AuditValue, informe de eventos observados.
Estadísticas	En Substraer y Auditoría, informe de estadísticas acumuladas en una terminación o tren.
Topología	Especifica sentidos de flujo entre terminaciones y un contexto.
ContextAttribute	Contiene propiedades (que se pueden definir en lotes) que afectan el contexto como un todo.
Error	Contiene un código de error y, facultativamente, texto explicativo del error; puede aparecer en respuestas a instrucciones y en peticiones de notificación.
NOTA – El descriptor de Modem no fue considerado en la Rec. UIT-T H.248.1, Versión 2 (05/2002).	

6.2.5 Terminación raíz (Root)

Hay casos en que una instrucción tiene que hacer referencia a la pasarela como entidad en sí misma y no a una de sus terminaciones. Para este fin está reservado un TerminationID especial, "Root" (raíz). Se pueden definir lotes en la raíz. Por tanto, la raíz puede tener propiedades, eventos, señales y estadísticas. En consecuencia, el TerminationID raíz puede aparecer en:

- una instrucción Modificar (Modify) – para cambiar una propiedad, enviar una señal o establecer un evento;
- una instrucción Notificar (Notify) – para informar un evento;
- una instrucción AuditValue (Auditoría de valores) – para examinar los valores de propiedades y estadísticas implementadas en Root (raíz);

- una instrucción AuditCapability (Auditoría de capacidades) – para determinar qué propiedades de Root están implementadas;
- una instrucción ServiceChange (Cambio de servicio) – para declarar que toda la pasarela está en servicio o fuera de servicio.

Toda otra utilización del TerminationID raíz es un error. En estos casos deberá retornarse el código de error 410 ("Identificador incorrecto").

6.3 Principios del mecanismo de comodines

En esta cláusula se especifica la utilización de comodines en identificadores de contexto (ContextID) y de terminación (TerminationID) que se aplican a todas las instrucciones. Al procesar estas instrucciones deben tenerse en cuenta dos tipos de comodines:

- 1) Comodín de contexto
- 2) Comodín de terminación

A los efectos de los procedimientos basados en comodines, se considera que una lista (TerminationIDList), con más de un TerminationID es un TerminationID comodín. Al ejecutar una transacción con contextos comodín y, opcionalmente, terminaciones comodín, todas las instrucciones de la transacción se ejecutan ordenadamente para un ejemplar concreto ContextID antes de pasar al siguiente ejemplar ContextID. Cuando haya varias instrucciones en una transacción, es condición esperar a que el TerminationID (específico o de comodín) especificado en la primera instrucción corresponda a un ejemplar específico de ContextID, antes de ejecutar las siguientes instrucciones de la transacción. Si el TerminationID (específico o de comodín) de las siguientes instrucciones de la transacción no corresponda a un ejemplar ContextID específico, se devuelve un código de error 431 ("Ninguna terminación corresponde al comodín") y se detiene el procesamiento de los siguientes ejemplares del ContextID de comodín, a menos que la instrucción que generó el error se haya marcado como facultativa.

A continuación se indica la ejecución de algunas combinaciones de comodines.

6.3.1 ContextID específico con TerminationID comodín

Cuando el ContextID es específico y se utiliza ALL en el TerminationID de la instrucción, es lo mismo que repetir la instrucción para cada uno de los TerminationID que correspondan. La utilización de ALL no afecta a la terminación ROOT. Cada una de estas instrucciones puede generar una respuesta, por lo que el tamaño de toda la respuesta puede ser grande. Así, si el comodín corresponde a más de un TerminationID en el contexto, se intentan todas las correspondencias posibles y se informa de los resultados obtenidos con cada una de ellas. Si ninguna de las terminaciones a las que hace referencia el TerminationID de comodín están en el contexto especificado, se devuelve un código de error 431 ("Ninguna terminación corresponde al comodín"). No se devuelve ningún error en el caso de terminaciones especificadas por el TerminationID de comodín que no estén en dicho contexto.

Por ejemplo: Se supone que una pasarela tiene cuatro terminaciones: t1/1, t1/2, t2/1 y t2/2, que el Contexto 1 tiene t1/1 y t2/1 y que el Contexto 2 tiene t1/2 y t2/2.

La instrucción:

```
Context=1 {Command=t1/* {Descriptor/s}}
```

Provoca en retorno:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

6.3.2 ContextID comodín (ALL) con TerminationID específico

Cuando el ContextID es un comodín (es decir ContextID = ALL) y el TerminationID está totalmente especificado, el efecto es idéntico a una instrucción que indique el contexto no NULL

que contiene la terminación especificada. Por consiguiente, debe buscarse el contexto y ejecutarse sólo un ejemplar de la instrucción. No se señala ningún error en el caso de contextos que no contienen la terminación especificada. Si la terminación no existe en ningún contexto (non-NULL), se devuelve el error 431 ("Ninguna terminación corresponde al comodín"). Si no hay otros contextos diferentes de NULL, se devuelve el código de error 411 ("la transacción se refiere a un ContextID desconocido"). Aunque no se aconseja utilizar esta combinación en lugar de especificar el ContextID, puede resultar útil, por ejemplo, para corregir un estado de discrepancia entre la MG y el MGC.

Por ejemplo: Teniendo en cuenta la anterior configuración de pasarela.

La instrucción:

```
Context=* {Command=t1/1 {Descriptor/s}}
```

Provoca en retorno:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

6.3.3 ContextID comodín (ALL) con TerminationID comodín

Cuando tanto el ContextID como el TerminationID son comodines (es decir, ContextID = ALL), el efecto es idéntico a repetir la instrucción para cada TerminationID que corresponda al comodín de cada contexto no NULL que contiene uno o más de esos TerminationID coincidentes. Por consiguiente, si el comodín coincide con más de un TerminationID en el ejemplar específico del ContextID comodín, se intentan todas las correspondencias posibles y se informa de los resultados de cada una de ellas. No se señala ningún error en el caso de contextos que no contengan una terminación que corresponda con el TerminationID comodín. Tampoco se señala ningún error en el caso de terminaciones especificadas en el TerminationID comodín que no estén en un ejemplar específico del ContextID comodín. Si no hay ninguna correspondencia con el ContextID y el TerminationID comodines, se devuelve un código de error 431 ("Ninguna terminación corresponde al comodín").

Por ejemplo: Teniendo en cuenta la configuración de pasarela anterior.

La instrucción:

```
Context=* {Command=t1/* {Descriptor/s}}
```

Provoca en retorno:

```
Context=1 {Command=t1/1 {Descriptor/s}}
```

```
Context=2 {Command=t1/2 {Descriptor/s}}
```

Cuando un TerminationID comodín y/o una petición de ContextID comodín contienen múltiples instrucciones, si la primera de ellas no corresponde con el primer ejemplar ContextID y TerminationID, la siguiente instrucción de la petición no se ejecutará para dicho ejemplar.

6.3.4 Respuestas con comodín

Cuando no se necesitan respuestas individuales, puede solicitarse una respuesta comodín. En este caso se genera una única respuesta que contiene la UNION de todas las respuestas individuales que se hubiesen generado en otro caso, en la que se han suprimido los valores duplicados. Por ejemplo, en una terminación Ta con propiedades p1 = a, p2 = b y una terminación Tb con propiedades p2 = c, p3 = d, una respuesta UNION estará formada por un TerminationID comodín y la secuencia de propiedades p1 = a, p2 = b, c y p3 = d. La respuesta comodín puede resultar especialmente útil para las instrucciones Audit. Si se utiliza una respuesta UNION comodín junto con un contexto comodín, se envía una única respuesta que contiene la UNION de todas las terminaciones a que hace referencia el TerminationID. La respuesta contendrá Context = ALL, un TerminationID comodín y la secuencia de propiedades.

En caso de error durante la ejecución de una petición comodín que especifica una respuesta comodín, es necesario aplicar un tratamiento especial para proporcionar información útil sobre los errores limitando el tamaño de la respuesta a una dimensión discreta. Cuando se solicita una respuesta comodín, se ejecutarán todos los ejemplares de la instrucción (como se especificó anteriormente), incluso si uno o más de ellos originan errores, pero no se ejecutarán las instrucciones subsiguientes de la transacción (a menos que se especifique que son facultativos). Se devolverán múltiples respuestas de instrucción para la instrucción que ha provocado el error. La primera respuesta de instrucción será una respuesta comodín normal que contenga la UNION de respuestas para las instrucciones que se ejecutaron adecuadamente. Si ninguna de ellas tuvo éxito, la UNION estará vacía. Se devolverán respuestas de instrucción adicionales para cada TransactionID que no ha tenido éxito, con el descriptor de error correspondiente.

Por ejemplo

La instrucción:

```
Context=*{Command=t1/*{Descriptor/s}}
```

Respuesta a un error:

```
Context=*{Command=t1/*{Union response descriptors},  
Command=t1/3{Error=errorcode}}
```

En los anexos A y B se explica la codificación del mecanismo de comodines.

7 Instrucciones

El protocolo proporciona instrucciones para manipular las entidades lógicas del modelo de conexión de protocolo, los contextos y las terminaciones. Las instrucciones permiten controlar el nivel más fino de granularidad soportado por el protocolo, Por ejemplo, existen instrucciones para añadir terminaciones a un contexto, modificar terminaciones, substraer terminaciones de un contexto, y auditoría de propiedades de contextos y terminaciones. Las instrucciones permiten controlar completamente las propiedades de contextos y terminaciones. Esto incluye la especificación de los eventos que una terminación debe informar, las señales/acciones que habrán de aplicarse a una terminación, y la especificación de la topología de un contexto (quién oye/ve a quién).

La mayor parte de las instrucciones son para uso específico por el controlador de pasarela de medios como iniciador de instrucciones para ejercer el control de las pasarelas de medios como respondedores de instrucciones. Las excepciones son la instrucciones Notificar y ServiceChange: Notificar la envía una pasarela de medios a un controlador de pasarela de medios, y ServiceChange puede enviarla cualquiera de las dos entidades. A continuación se presenta una visión general de las instrucciones; en 7.2 se da una explicación más detallada.

- 1) **Añadir:** La instrucción Añadir añade una instrucción a un contexto. La instrucción Añadir en la primera terminación en un contexto se utiliza para crear un contexto.
- 2) **Modificar:** La instrucción Modificar modifica las propiedades, eventos y señales de una terminación.
- 3) **Substraer:** La instrucción Substraer desconecta una terminación de su contexto y retorna estadísticas sobre la participación de la terminación en el contexto. La instrucción Substraer en la última terminación en un contexto suprime el contexto.
- 4) **Desplazar (Move):** La instrucción Desplazar desplaza atómicamente una terminación a otro contexto.
- 5) **AuditValue:** La instrucción AuditValue retorna el estado actual de propiedades, eventos, señales y estadísticas de terminaciones.
- 6) **AuditCapabilities:** La instrucción AuditCapabilities retorna todos los posibles valores de propiedades de terminación, eventos y señales permitidos por la pasarela de medios.

- 7) **Notificar**: La instrucción Notificar permite a la pasarela de medios informar al controlador de pasarela de medios sobre la aparición de eventos en la pasarela de medios.
- 8) **ServiceChange**: La instrucción ServiceChange permite a la pasarela de medios notificar al controlador de pasarela de medios que una terminación o un grupo de terminaciones están a punto de ser puestos fuera de servicio o se han puesto en servicio nuevamente. La MG utiliza también la instrucción ServiceChange para anunciar su disponibilidad a un MGC (registro), y para notificar al MGC que el arranque de la misma es inminente o acaba de efectuarse. El MGC puede anunciar un traspaso a la pasarela enviándole una instrucción ServiceChange. El MGC puede también utilizar la instrucción ServiceChange para ordenar a la MG que ponga en servicio o fuera de servicio una terminación o un grupo de terminaciones.

Estas instrucciones se describen detalladamente en 7.2.1 a 7.2.8.

7.1 Descriptores

Los parámetros de una instrucción se denominan descriptores. Un descriptor consiste en un nombre y una lista de artículos. Algunos artículos pueden tener valores. Muchas instrucciones comparten descriptores comunes. En esta subcláusula se enumeran estos descriptores. Se puede retornar los descriptores como resultado de una instrucción. En cualquiera de esos retornos de contenidos de descriptor, un descriptor vacío está representado por su nombre que no va acompañado de ninguna lista. En la subcláusula que describe la instrucción se describen los parámetros y su utilización específica para un determinado tipo de instrucción.

7.1.1 Especificación de parámetros

Los parámetros de instrucciones están estructurados en varios descriptores. En general, el formato textual de los descriptores es `DescriptorName=<someID>{parm=value, parm=value...}`.

Los parámetros pueden ser: totalmente especificados, subespecificados o sobreespecificados:

- 1) Los parámetros totalmente especificados tienen un valor único, inequívoco, que el iniciador de la instrucción comunica al respondedor de la instrucción para que lo utilice en el parámetro especificado.
- 2) Los parámetros subespecificados, que utilizan el valor CHOOSE, permiten al respondedor de la instrucción elegir cualquier valor que éste pueda soportar.
- 3) Los parámetros sobreespecificados tienen la forma de una lista de valores potenciales. El orden de los valores en la lista corresponde al orden en que el iniciador de la instrucción prefiere que los valores sean seleccionados. El respondedor de la instrucción elige uno de los valores contenidos en la lista ofrecida y retorna ese valor al iniciador de la instrucción.

Si en una instrucción no se especifica (no se incluye) uno de los descriptores requeridos, distinto del descriptor de auditoría, se mantienen los valores precedentes establecidos en dicho descriptor para esa terminación, si existe alguno. Si en una instrucción, diferente de Substraer, no se incluye el descriptor de auditoría, es lo mismo que dejar el descriptor de auditoría vacío. El comportamiento de la MG con respecto a los parámetros no especificados en un descriptor varía con el descriptor en cuestión, tal como se indica en las siguientes subcláusulas. Cuando un parámetro es subespecificado o sobreespecificado, el descriptor que contiene el valor elegido por el respondedor se incluye como resultado de la instrucción.

Cada instrucción especifica el TerminationID sobre el que la instrucción opera. Este TerminationId puede incluir "comodines". Cuando el TerminationID de una instrucción incluye comodines, el efecto será el mismo que si se repitiera la instrucción con cada uno de los TerminationID concordantes.

7.1.2 Descriptor de módem

El descriptor de módem especifica el tipo y los parámetros del módem (en su caso) que es necesario, por ejemplo, en comunicaciones H.324 y de texto. El descriptor incluye los siguientes tipos de módems: V.18, V.22, V.22 *bis*, V.32, V.32 *bis*, V.34, V.90, V.91, RDSI síncrona, y permite ampliaciones. Por defecto, en una terminación no hay descriptor de módem.

En la Rec. UIT-T H.248.1, Versión 2 (05/2002) y las versiones más recientes no se consideró el descriptor de módem. Entonces, el descriptor de módem no se incluirá en el contenido transmitido. Si se recibe un descriptor de módem, las condiciones de la implementación determinarán si se trata o no se tiene en cuenta. Debe indicarse el tipo de módem como un atributo de los trenes de datos en el descriptor local y el descriptor distante.

7.1.3 Descriptor de múltiplex

En las llamadas multimedia se transportan varios trenes de medios en varios portadores (el número es posiblemente diferente). El descriptor de múltiplex asocia los medios y los portadores. El descriptor incluye los tipos de múltiplex:

- H.221;
- H.223;
- H.226;
- V.76;
- $N \times 64K$;
- posibles ampliaciones,

y un conjunto de TerminationID que representan los portadores multiplexados, en orden. Por ejemplo:

$$\text{Mux} = \text{H.221}\{\text{MyT3}/1/2, \text{MyT3}/2/13, \text{MyT3}/3/6, \text{MyT3}/21/22\}$$

El tipo de múltiplex $N \times 64K$ implementa el servicio $N \times 64K$ (conforme a la velocidad de transmisión de información Q.931 o las condiciones de medio de transmisión Q.763, por ejemplo). En lo relativo al contexto, soporta un solo tren de datos de banda ancha. Si se añade implícitamente una terminación de portador a un contexto, como resultado de la creación de una terminación multiplexora $N \times 64K$, el descriptor de tren para la terminación de portador tendrá los mismos valores que el descriptor de tren definido para la terminación multiplexora, sólo que el ancho de banda de la terminación del portador será de 64 kbit/s.

7.1.4 Descriptor de medios

El descriptor de medios especifica los parámetros de todos los trenes de medios. Estos parámetros están estructurados en dos descriptores, un descriptor de estado de la terminación (TerminationState), que especifica las propiedades de una terminación que no dependen del tren, y uno o más descriptores de tren, cada uno de los cuales describe un solo tren de medios.

Un tren se identifica mediante un identificador de tren (StreamID). El StreamID ha de ser un valor entre 1 y 65535 y se utiliza para vincular los trenes en un contexto que corresponden unos con otros. Todos los trenes que salgan de una terminación han de estar sincronizados entre sí. El descriptor de tren comprende hasta cuatro descriptores subsidiarios: los descriptores de control local, local, de estadística y distante. La relación entre estos descriptores es la siguiente:

Descriptor de medios

Descriptor TerminationState
Descriptor de tren (Stream)

Descriptor de control local (LocalControl)
Descriptor local (Local)
Descriptor distante (Remote)
Descriptor de estadística

Por razones de conveniencia, los descriptores de control local, local, distante o de estadísticas pueden incluirse en el descriptor de medios sin un descriptor de tren circundante. En este caso, se supone que el StreamID es uno.

7.1.5 Descriptor del estado de la terminación (TerminationState)

El descriptor del estado de la terminación contiene la propiedad ServiceStates, la propiedad EventBufferControl y propiedades de una terminación (definidas en paquetes) que no son específicas del tren.

La propiedad ServiceStates describe el estado global de la terminación (no específico del tren). Una terminación puede estar en uno de los estados siguientes: "Test", "OutOfService", o "InService". El estado "Test " indica que la terminación se está probando. El estado "OutOfService " indica que la terminación no puede utilizarse para tráfico. El estado "InService" indica que una terminación puede utilizarse para tráfico normal. El estado por defecto es "InService".

Es posible asignar valores simples a las propiedades (entero/cadena/enumeración), o pueden quedar subespecificadas, es decir, se suministra más de un valor y la MG puede elegir:

- Valores alternativos: lista de valores, de los cuales hay que seleccionar uno.
- Gammas: dos valores, uno mínimo y el otro máximo, debiéndose elegir cualquier valor entre el mínimo y el máximo, incluidos los valores límite.
- Mayor que/menor que: el valor tiene que ser mayor/menor que el valor especificado.
- Comodín CHOOSE: la MG elige entre los valores permitidos para la propiedad.

La propiedad EventBufferControl especifica si los eventos son registrados en memoria intermedia después de la detección de un evento en el descriptor de eventos, o son procesados inmediatamente. Para más detalles véase 7.1.9.

7.1.5.1 Propiedades del estado de la terminación

7.1.5.1.1 ServiceStates

Nombre de propiedad: ServiceStates

Descripción: Valor que indica el estado actual de servicio de la terminación.

Tipo: Enumeración

Valores posibles:

InService: La terminación está en servicio y funcionando normalmente.

OutOfService: La terminación está fuera de servicio y no disponible para el tráfico.

Test: La terminación está bajo prueba.

Valor por defecto: InService

Definido en: TerminationState

Características: Lectura/escritura

7.1.5.1.2 Control de memoria intermedia de eventos (EventBufferControl)

Nombre de propiedad: EventBufferControl

Descripción: Especifica si, tras la detección de un evento en el descriptor de eventos, se almacenan los eventos en memoria intermedia o se procesan inmediatamente. Véase 7.1.9.

Tipo: Enumeración

Valores posibles:

LockStep: Se almacenan y procesan los eventos conforme a 7.1.9.
OFF: Se procesan los eventos inmediatamente.

Valor por defecto: OFF

Definido en: TerminationState

Características: Lectura/escritura

7.1.6 Descriptor de tren

Un descriptor de tren especifica los parámetros de un solo tren bidireccional. Estos parámetros están estructurados en tres descriptores: un descriptor que contiene las propiedades de la terminación específicas del tren, un descriptor para los flujos locales y un descriptor para los flujos distantes. El descriptor de tren incluye un StreamID que identifica el tren. Los trenes se crean mediante la especificación de un StreamID nuevo en una de las terminaciones de un contexto. Un tren se suprime fijando los descriptores local y distante vacíos para el tren con ReserveGroup y ReserveValue en LocalControl puesto a "falso" en todas las terminaciones del contexto que soportó anteriormente este tren.

Los identificadores StreamID tienen significación local entre MGC y MG, y son asignados por el MGC. Dentro de un contexto, StreamID permite saber qué flujos de medios están interconectados: los trenes con el mismo StreamID se conectan.

Si se desplaza una terminación de un contexto a otro, el efecto sobre el contexto al que la terminación es desplazada es el mismo que si se añadiera una nueva terminación con los mismos StreamID que la terminación desplazada.

7.1.7 Descriptor de control local (LocalControl)

El descriptor de control local contiene la propiedad modo, las propiedades ReserveGroup y ReserveValue y las propiedades de una terminación (definidas en paquetes) que son específicas del tren, y son a considerar entre la MG y el MGC. Los valores de propiedades se pueden especificar como se indica en 7.1.1.

Los valores permitidos para la propiedad modo son "SendOnly", "RecvOnly", "SendRecv", "Inactive" y "LoopBack". "SendOnly", "RecvOnly" y "LoopBack" han de entenderse con respecto al exterior del contexto, de modo que, por ejemplo, un tren en modo = SendOnly no pasa al contexto los medios recibidos. Cuando el modo de un tren sea "LoopBack" en una terminación, los medios recibidos (descriptor local) en ésta se enviarán en bucle de nuevo hacia el lado de origen (descriptor distante) de la terminación y no se transmitirán medios entre dicha terminación y otras terminaciones en el contexto. Los medios que se devuelven al origen han de ser enviados conforme al descriptor distante. El valor por defecto para la propiedad modo es "Inactive". Las señales y los eventos no son afectados por la propiedad modo. La propiedad modo LocalControl tiene prioridad sobre cualquier otro modo especificado en los descriptores local y distante.

ReserveProperties, ReserveValue y ReserveGroup de una terminación, en valores booleanos, indican lo que se espera que haga la MG cuando recibe un descriptor local y/o distante.

Si el valor de ReserveProperty es "Verdadero" (True), la MG reservará recursos para todas las alternativas especificadas en los descriptores local y/o distante para los cuales tiene actualmente recursos disponibles. Responderá con las alternativas para las cuales reserva los recursos. Si no puede soportar ninguna de las alternativas, enviará al MGC una contestación que contiene descriptores local y/o distante vacíos. Si los medios comienzan a fluir cuando están reservadas más de una alternativa, se puede enviar/recibir paquetes de medios en cualquiera de las alternativas y hay que procesar dichos paquetes aunque una sola alternativa pueda estar activa en un momento dado cualquiera.

Si el valor de ReserveProperty es Falso (False), la MG elegirá una de las alternativas especificadas en el descriptor local (si está presente) y una de las alternativas especificadas en el descriptor distante (si está presente). Si la MG no ha reservado todavía recursos para soportar la alternativa seleccionada, reservará tales recursos. Si, por el contrario, ya reservó recursos para la terminación tratada (debido a un intercambio anterior con ReserveValue y/o ReserveGroup igual a Verdadero), liberará los recursos en exceso que haya reservado anteriormente. Por último, la MG enviará una contestación al MGC conteniendo las alternativas para el descriptor local y/o distante que seleccionó. Si la MG no tiene suficientes recursos para soportar cualquiera de las alternativas especificadas, responderá con el error 510 ("Recursos insuficientes").

El valor por defecto de ReserveValue y ReserveGroup es Falso. En 7.1.8 se suministra más información sobre la utilización de las dos ReserveProperties.

La nueva configuración del descriptor de control local reemplaza completamente a su configuración anterior en la MG. Por ello, para retener información procedente de la anterior, el MGC debe incluir dicha información en la nueva. Si el MGC desea eliminar alguna información del descriptor existente, sencillamente reenvía el descriptor (en una instrucción Modificar) donde se ha retirado la información no deseada.

NOTA – La propiedad modo también se denomina "StreamMode" en las definiciones de codificación de los anexos A y B. En H.248 se puede utilizar cualquiera de las dos expresiones.

7.1.7.1 Propiedades de control local (LocalControl)

7.1.7.1.1 Modo

Nombre de propiedad: StreamMode

Descripción: Su valor indica el estado actual de servicio de la terminación.

Tipo: Enumeración

Valores posibles:

Inactive:	La terminación no pasa medios para el tren.
SendOnly:	La terminación pasa medios para el tren desde el interior hacia el exterior del contexto.
RecvOnly	La terminación pasa medios para el tren desde el exterior hacia el interior del contexto.
SendRecv	La terminación pasa medios para el tren desde el contexto y hacia él.
LoopBack	La terminación reenvía al remitente los medios recibidos para el tren.

Valor por defecto: Inactive

Definido en: LocalControl

Características: Lectura/escritura

7.1.7.1.2 ReserveGroup

Nombre de propiedad: ReserveGroup

Descripción: Especifica si la MG debería reservar los recursos necesarios para soportar un solo grupo de medios o tantos grupos como pueda, conforme a la definición dada en los descriptores local y distante, véase 7.1.8.

NOTA – Por "grupo de medios" se entiende el contenido de la gramática de "PropertyGroup" ASN.1 (véase el anexo A) en la codificación binaria, o la descripción de una sola sesión SDP en la codificación de texto.

Tipo: Booleano

Valores posibles:

True: La MG deberá reservar todos los grupos de medios posibles indicados en los descriptores local y distante.

False: La MG deberá reservar un sólo grupo de medios de cada descriptor, el local y el distante, con arreglo a las políticas descritas en 7.1.8.

Valor por defecto: False

Definido en: LocalControl

Características: Lectura/escritura

7.1.7.1.3 ReserveValue

Nombre de propiedad: ReserveValue

Descripción: Especifica si la MG debería reservar los recursos necesarios para soportar un sólo conjunto de valores de propiedad (por ejemplo, un sólo códec y sus atributos correspondientes) o tantos conjuntos como pueda, conforme a la definición dada en los descriptores local y distante, véase 7.1.8.

Tipo: Booleano

Valores posibles:

True: La MG deberá reservar recursos para utilizar todos los conjuntos de valores de propiedad posibles que se indican en el grupo de medios escogido (si ReserveGroup es False) o en cada grupo de medios (si ReserveGroup es True) en los descriptores local y distante.

False: La MG deberá reservar un solo conjunto de valores de propiedad de entre aquellos indicados en el grupo de medios escogido (si ReserveGroup es False) o en cada grupo de medios (si ReserveGroup es True) en los descriptores local y distante.

Valor por defecto: False

Definido en: LocalControl

Características: Lectura/escritura

7.1.8 Descriptor local y descriptor distante

El MGC utiliza el descriptor local y el descriptor distante para reservar y comprometer recursos de la MG para la codificación y decodificación de medios para el tren, o trenes determinados, y la terminación a la cual se aplican. La MG incluye estos descriptores en su respuesta para indicar que se encuentra realmente preparada para soportarlos. La MG incluirá en su respuesta propiedades

adicionales y sus valores si estas propiedades son obligatorias pero todavía no están presentes en las peticiones realizadas por el MGC (por ejemplo, especificando parámetros de codificación vídeo detallados donde el MGC solamente ha especificado el tipo de cabida útil).

Para que no haya ambigüedad al solicitar a la MG que reserve y comprometa recursos, el MGC debería suministrar toda la información necesaria en caso de subespecificación (es decir, CHOOSE), para que la MG pueda efectuar su selección sin ambigüedad. Así, por ejemplo, si se utiliza CHOOSE sin especificar el tipo de aplicación requerida (por ejemplo, "nombre de medios" cuando se trate de codificación SDP), puede ser necesaria más información (por ejemplo, líneas de atributo, en el caso de la codificación SDP).

Local se refiere a los medios recibidos por la MG y distante se refiere a los enviados por ella.

En el caso de codificación textual del protocolo, los descriptores consisten en descripciones de sesión definidas en el protocolo de descripción de sesión (SDP, *session description protocol*) (RFC 2327). En las descripciones de sesión enviadas desde el MGC a la MG, se permiten las siguientes excepciones a la sintaxis de RFC 2327:

- las líneas "s = ", "t = " y "o = " son facultativas;
- el uso de CHOOSE está permitido en lugar de un valor de parámetro único; y
- el uso de alternativas está permitido en lugar de un valor de parámetro único.

Un descriptor de tren especifica un tren de medios bidireccional en particular, por lo que una descripción de sesión individual no incluirá más de una descripción de medios (línea "m = "). Un descriptor de tren puede contener descripciones de sesión adicionales como alternativas. Cada tren de medios para una terminación aparecerá en descriptores de tren distintos. Cuando se proporcionan múltiples descripciones de sesión en un solo descriptor, las líneas "v = " se requieren como delimitadores; en otro caso, son facultativas en las descripciones de sesión enviadas a la MG. Las implementaciones aceptarán las descripciones de sesión que sean totalmente conformes a RFC 2327, y teniendo en cuenta las restricciones aquí mencionadas. En el caso de protocolo codificado en binario, el descriptor consiste en grupos de propiedades (pares de valores de rótulo), como se especifica en el anexo C. Cada uno de estos grupos contiene los parámetros de una descripción de sesión.

A continuación, se especifica en detalle la semántica de los descriptores local y distante. La especificación consta de dos partes. La primera parte especifica la interpretación del contenido del descriptor. La segunda parte especifica las acciones que debe emprender la MG tras la recepción de los descriptores local y distante. Las acciones que ha de acometer la MG dependen de los valores de las propiedades ReserveValue y ReserveGroup del descriptor de control local.

El descriptor local o el descriptor distante, o ambos, pueden ser:

- no especificado (es decir, ausente),
- vacío,
- subespecificado, utilizando CHOOSE en un valor de propiedad,
- totalmente especificado, o
- sobreespecificado, incluyendo varios grupos de propiedades y posiblemente múltiples valores de propiedad en uno o más de estos grupos.

Los descriptores transferidos del MGC a la MG son interpretados de conformidad con las reglas dadas en 7.1.1 y teniendo en cuenta los siguientes comentarios adicionales:

- a) Si no se especifican los descriptores local o distante, se considera que falta un parámetro obligatorio. La MG utilizará el último parámetro especificado para este descriptor. Si no hubiera ningún valor especificado anteriormente, se ignora ese descriptor en el procesamiento ulterior de la instrucción.

- b) Un descriptor local (distante) vacío en un mensaje procedente del MGC significa una petición de liberar recursos reservados para el flujo de medios recibido (enviado).
- c) Cuando están presentes múltiples grupos de propiedades en un descriptor local o distante, o múltiples valores dentro de un grupo, se aplica el orden de preferencia descendente.
- d) Las propiedades subespecificadas o sobreespecificadas dentro de un grupo de propiedades enviado por el MGC son peticiones para que la MG elija uno o más valores que pueda soportar de cada una de estas propiedades. En caso de una propiedad sobreespecificada, la lista de valores se encuentra en un orden de preferencia descendente.

De acuerdo con las reglas anteriores, la acción siguiente depende de los valores de las propiedades ReserveValue y ReserveGroup en LocalControl.

Si ReserveGroup es "Verdadero", la MG reserva los recursos requeridos para soportar tantas alternativas de grupo de propiedades solicitadas como pueda actualmente soportar. Si ReserveValue es "Verdadero", la MG reserva los recursos requeridos para soportar tantas alternativas de valores de propiedad solicitadas como pueda actualmente soportar.

NOTA – Si un descriptor local o distante contiene múltiples grupos de propiedades, y ReserveGroup es "Verdadero", entonces se pide a la MG que reserve recursos de modo que pueda decodificar o codificar el tren de medios de conformidad con alguna de las alternativas. Por ejemplo, si el descriptor local contiene dos grupos de propiedades, una que especifica señales audio de ley A G.711 paquetizadas y la otra señales audio G.723.1, la MG reserva recursos de modo que pueda decodificar un tren audio codificado, sea en el formato de ley A G.711 o en el formato G.723.1. La MG no ha de reservar recursos para decodificar dos trenes de audio simultáneamente, un tren codificado en ley A G.711 y otro codificado en G.723.1. El mismo principio se aplica para ReserveValue.

Si ReserveGroup es "Verdadero" o ReserveValue es "Verdadero", se aplican las siguientes reglas:

- Si la MG no dispone de recursos suficientes para soportar todas las alternativas solicitadas por el MGC, y el MGC ha solicitado recursos en el descriptor local y el descriptor distante, la MG debe reservar recursos para soportar como mínimo una alternativa en cada uno de los descriptores local o distante.
- Si la MG no dispone de recursos suficientes para soportar como mínimo una de las opciones del descriptor local (distante) recibido del MGC, retornará un descriptor local (distante) vacío como respuesta.
- En su respuesta al MGC que ha incluido los descriptores local y distante, la MG incluirá descriptores local y distante para todos los grupos de propiedades y valores de propiedad para los que reservó recursos. Si la MG no puede soportar como mínimo una de las opciones del descriptor local (distante) recibido del MGC, retornará un descriptor local (distante) vacío.
- Si la propiedad modo del descriptor de control local es "RecvOnly", "SendRecv" o "LoopBack", la MG debe prepararse para recibir medios codificados de acuerdo con cualquiera de las alternativas incluidas en su respuesta al MGC.

Si ReserveGroup es "Falso" y ReserveValue es "Falso", la MG debería aplicar las siguientes reglas para elegir una sola opción tanto para local como para distante:

- La MG elige la primera alternativa en local para la cual es capaz de soportar como mínimo una alternativa en distante.
- Si la MG no es capaz de soportar como mínimo una alternativa local o distante, retorna el código de error 510 ("Recursos insuficientes").
- La MG retorna su alternativa seleccionada en cada uno de los descriptores local y distante.

La nueva configuración del descriptor local o distante reemplaza completamente a su configuración anterior en la MG. Por ello, para retener información de la anterior, el MGC debe incluir esta información en la nueva. Si el MGC desea eliminar alguna información del descriptor existente,

sencillamente reenviará el descriptor (en una instrucción Modificar) con la información no deseada retirada.

7.1.9 Descriptor de eventos

El parámetro descriptor de eventos contiene un RequestID y una lista de eventos que la pasarela de medios debe detectar e informar. El RequestID se utiliza para correlacionar la petición con las notificaciones que ésta pueda ocasionar. Los eventos solicitados incluyen, por ejemplo, tonos de transmisión facsímil, resultados de la prueba de continuidad y transiciones entre los estados colgado y descolgado. El RequestID se omite si el descriptor de eventos está vacío (es decir, si no está especificado ningún evento).

Cada evento en el descriptor contiene el nombre de evento, un StreamID facultativo, una bandera KeepActive facultativa, una bandera NotifyBehaviour facultativa, una bandera ResetEventsDescriptor facultativa y parámetros facultativos. El nombre de evento consiste en un PackageID, nombre de lote (donde el evento está definido) y un EventID. El comodín ALL puede utilizarse para el EventID, lo que indica que todos los eventos procedentes del lote especificado deben ser detectados. El StreamID por defecto es 0, lo que indica que el evento que debe detectarse no está relacionado con un determinado tren de medios. Los eventos pueden tener parámetros. Esto permite que haya alguna variación en el significado de una descripción de evento simple sin que ello entrañe la creación de numerosos eventos individuales. Se definen ulteriores parámetros de eventos en el lote.

Si el descriptor de eventos incluye o implica que se complete el mapa de dígitos, el parámetro EventDM se utiliza para transportar el nombre o el valor del mapa de dígitos asociado. Para más detalles, véase 7.1.14.

Cuando se procesa un evento por referencia al contenido de un descriptor de eventos activo y se comprueba que está presente en ese descriptor ("reconocido"), la acción por defecto de la MG es enviar una instrucción Notificar al MGC. La notificación puede aplazarse si el evento es absorbido en la cadena de marcación actual de un mapa de dígitos activo (véase 7.1.14). El envío de una instrucción Notificar puede verse afectado por la bandera NotifyBehaviour. Además, el reconocimiento del evento puede hacer que se detengan las señales actualmente activas o que deba reemplazarse el descriptor de eventos y/o señales en curso, tal como se describe al final de esta subcláusula. Un descriptor de eventos permanece activo después de que el evento haya sido reconocido, a menos que se reemplace por otro descriptor de eventos.

Si el valor de la propiedad EventBufferControl es igual a "LockStep", después de la detección de tal evento se suspende el tratamiento normal de los eventos. Cualquier evento detectado subsiguientemente en el descriptor EventBuffer se añadirá al final del EventBuffer (una cola FIFO) indicando el momento en que fue detectado. La MG esperará el nuevo EventsDescriptor que ha de cargarse. Se puede cargar un descriptor de eventos nuevo sea como resultado de la recepción de una instrucción con un nuevo descriptor de eventos, sea mediante la activación de un descriptor de eventos insertado.

Si EventBufferControl está desactivado ("Off"), la MG continúa el procesamiento basado en el descriptor de eventos activo.

Cuando se activa un descriptor de eventos insertado la MG continúa el procesamiento del evento basado en el descriptor de eventos recientemente activado.

NOTA 1 – A los fines de tratamiento de EventBuffer, la activación de un descriptor de eventos insertado es equivalente a la recepción de un descriptor de eventos nuevo.

Es posible que se tengan almacenados temporalmente en EventBuffer de la MG uno o más eventos en el momento en que la MG recibe una instrucción con un nuevo descriptor de eventos. El valor de EventBufferControl determina entonces el modo en que la MG tratará estos eventos almacenados temporalmente.

Caso 1

Si EventBufferControl es igual a "LockStep" y la MG recibe un nuevo descriptor de eventos, la MG comprobará la cola de tratamiento cronológico (FIFO) EventBuffer y emprenderá las siguientes acciones:

- 1) Si la EventBuffer está vacía, la MG espera la detección de eventos basados en el nuevo descriptor de eventos.
- 2) Si la EventBuffer no está vacía, la MG procesa la cola de espera FIFO comenzando por el primer evento:
 - a) Si el evento de la cola se encuentra en la lista de eventos del nuevo descriptor de eventos, la MG actúa sobre el evento y o elimina de la EventBuffer. La indicación de tiempo de la instrucción Notificar especificará el instante en que el evento fue realmente detectado. La MG espera entonces un nuevo descriptor de eventos. Mientras espera un nuevo descriptor de eventos, los eventos detectados que aparecen en el EventsBufferDescriptor se colocarán en la EventBuffer. Cuando se recibe un nuevo descriptor de eventos, se repetirá el procesamiento del evento desde el paso 1).
 - b) Si el evento no se encuentra en el nuevo descriptor de eventos, la MG descartará el evento y repetirá desde el paso 1).

Caso 2

Si EventBufferControl es igual a "Off" y la MG recibe un nuevo descriptor de eventos, la MG procesa los nuevos eventos con el nuevo descriptor de eventos.

Si la MG recibe una instrucción que le ordena fijar el valor de EventBufferControl a desactivado, todos los eventos de la EventBuffer serán descartados.

La MG puede comunicar varios eventos en una sola transacción, siempre que no se retrase innecesariamente la comunicación de cada uno de los eventos.

Para los procedimientos de transmisión de la instrucción Notificar, véanse las consideraciones de transporte específicas que se presentan en el anexo apropiado o en la Rec. de la serie H.248.x.

El valor por defecto de EventBufferControl es "Off".

NOTA 2 – Puesto que la propiedad EventBufferControl está en el descriptor TerminationState, la MG puede recibir una instrucción que cambie la propiedad EventBufferControl y no incluya un descriptor de eventos.

Normalmente, el reconocimiento de un evento ocasiona la interrupción de señales activas. Cuando KeepActive está especificado en el evento, la MG no interrumpirá las señales que estén activas en la terminación en que se detecta el evento.

Se puede utilizar la bandera NotifyBehaviour para indicar que la instrucción Notificar:

- es enviada inmediatamente (NotifyImmediate: valor por defecto).
- nunca se envía (NeverNotify),
- es regulada (es decir, enviada o suprimida) según la carga del MGC (RegulatedNotify).

En E.15 se dan más detalles acerca de la utilización de NotifyBehaviour. Si se emplea al mismo tiempo que los mapas de dígitos, hay que esperar a que éste haya sido completado. La Notificación regulada puede incluir un descriptor de eventos o señales regulado, alterno y asociado con ella. Si se regula una notificación (es decir, se suprime) se activará dicho descriptor. Si no, se activa el descriptor incorporado original. Cuando se tenga NotifyImmediate o NeverNotify, al detectar el evento se activará todo descriptor alterno incorporado que ya exista.

De encontrarse la bandera ResetEventDescriptor para un determinado evento, el resultado dependerá de si el descriptor de eventos que contiene el evento es incorporado o no. De serlo, dicha bandera hace que el descriptor de eventos para dicha terminación se ponga de nuevo en el estado

anterior a la activación del descriptor incorporado (es decir, en el último descriptor de eventos determinado explícitamente por una instrucción Modificar o, cuando no se ha emitido ninguna instrucción Modificar desde que se reinició la terminación colocándola en el contexto NULL, en el descriptor de eventos configurado por el MGC). Si se trata de un descriptor de eventos no incorporado, dicha bandera hace que se reinicie el descriptor de eventos activo, para esa terminación, reactivando los eventos de ejecución completa del mapa de dígitos que han pasado el análisis de concordancia y se han desactivado.

La bandera ResetEventsDescriptor estará presente solamente en terminaciones físicas. Se tendrá en cuenta solamente si la terminación está en el contexto NULL. Si está fuera de este contexto, la bandera no tendrá ningún efecto.

Un evento puede incluir un descriptor de señales insertado y/o un descriptor de eventos insertado que, si está presente, sustituye al actual descriptor de señales/eventos cuando se reconoce el evento. Es posible, por ejemplo, especificar que se genere la señal de tono de invitación a marcar cuando se reconozca un evento de descolgado, o que se detenga la señal de tono de invitación a marcar cuando se reconozca un dígito. Un controlador de pasarela de medios no enviará ningún descriptor de eventos para eventos que tengan la indicación KeepActive y que también tengan un descriptor de señales insertado.

Sólo se permite un nivel de inserción. Un descriptor de eventos insertado no contendrá otro descriptor de eventos insertado; un descriptor de eventos insertado puede contener un descriptor de señales insertado.

Un descriptor de eventos recibido por una pasarela de medios sustituye a cualquier descriptor de eventos precedente. La notificación de evento en curso en un proceso será completada, y la ejecución de los eventos detectados después de la instrucción que contiene el nuevo descriptor de eventos se procesarán de acuerdo con el nuevo descriptor de eventos.

Un descriptor de eventos vacío inhabilita todas las operaciones de reconocimiento y señalamiento de eventos. Un descriptor EventBuffer vacío libera la EventBuffer e inhabilita toda la acumulación de eventos en modo "LockStep": los únicos eventos informados serán los que se producen mientras un descriptor de eventos está activo. Si se activa un descriptor de eventos mientras la terminación está funcionando en modo "LockStep", la memoria tampón de eventos se libera inmediatamente.

7.1.10 Descriptor de memoria intermedia de eventos (EventBuffer)

El descriptor de memoria intermedia de eventos contiene una lista de eventos, con sus parámetros si los hay, que la MG debe detectar y almacenar en memoria tampón cuando EventBufferControl es igual a LockStep (véase 7.1.9).

7.1.11 Descriptor de señales

Las señales son medios generados por la MG, tales como tonos y anuncios, y las señales relacionadas con los portadores, tales como señales de colgado/descolgado. Unas señales más complejas pueden incluir una secuencia de esas señales simples entremezcladas con la recepción y el análisis de señales de medios o señales relacionadas con los portadores, y condicionados por estas señales. Son ejemplos la devolución en eco de datos recibidos, como en el paquete prueba de continuidad. Las señales pueden también solicitar la preparación del contenido de medios para futuras señales.

Un descriptor de señales es un parámetro que contiene el conjunto de señales cuya aplicación a una terminación puede pedirse a una pasarela de medios. Un descriptor de señales contiene un número de señales y/o listas de señales secuenciales. Un descriptor de señales puede no contener ninguna señal ni lista de señales secuenciales. El soporte de listas de señales secuenciales es facultativo.

Las señales se definen en paquetes. Las señales se denominarán con un PackageID, nombre de paquete (en el que se define la señal) y un SignalID. El SignalID no contendrá ningún comodín. Las

señales que ocurren en un descriptor de señales tienen un parámetro StreamID facultativo (su valor por defecto es 0, con lo que se indica que la señal no está relacionada con un determinado tren de medios), un tipo de señal facultativo (véase más adelante), una duración facultativa y posiblemente parámetros definidos en el lote que define la señal. Esto permite que haya alguna variación en el significado de una señal, con lo que se evita tener que crear un gran número de señales individuales.

Por último, el parámetro opcional NotifyCompletion permite a un MGC indicar que desea recibir notificación cuando la señal finalice totalmente. Puede ocurrir lo siguiente: ha expirado el periodo de temporización de la señal (o se ha finalizado la señal), la señal ha sido interrumpida por un evento, se ha completado un ciclo/iteración de la señal, la señal ha sido detenida por la sustitución de un descriptor de señales, o la señal ha sido detenida o nunca ha sido activada por algún otro motivo. Si el parámetro NotifyCompletion no está incluido en un descriptor de señales, sólo se genera una notificación si la señal ha sido detenida o nunca ha sido activada por algún otro motivo. Para que tenga lugar la comunicación, el evento de ejecución completa de la señal (véase E.1.2) tiene que estar habilitado en el descriptor de eventos actualmente activo. Es posible asociar el parámetro facultativo "RequestID" a un determinado ejemplar de SignalID cuando se soliciten varias señales cuyo SignalID sea el mismo, con lo cual el MGC puede distinguir entre los ObservedEvents de ejecución completa de la señal para dicho ejemplar. No se incluirá el parámetro SignalID solicitado si no está presente el parámetro NotifyCompletion.

La duración es un valor entero que se expresa en centésimas de segundo.

Hay tres tipos de señales:

- Activado/desactivado (OO, *on/off*): la señal está presente hasta que es desactivada.
- Temporización (TO, *TimeOut*): la señal está presente hasta que es desactivada o expira un determinado periodo de temporización.
- Breve (BR, *brief*): la señal se detendrá por sí misma, a menos que se aplique un nuevo descriptor de señales que cause su parada; no se necesita un valor de temporización.

Si una señal que no es de tipo TO por defecto fuera especificada después como señal de tipo TO en el descriptor de señales, deberá incluirse el parámetro duración.

Si el tipo de señal está especificado en un descriptor de señales, reemplaza al tipo de señal por defecto (véase 12.1.4). Si se especifica la duración para una señal activado/desactivado, no se tendrá en cuenta.

Una lista de señales secuenciales consta de un identificador de lista de señales (SignalListID) y una secuencia de señales que va a ser reproducida secuencialmente. Solamente el elemento final de la secuencia de señales en una lista de señales secuenciales puede ser una señal activado/desactivado. La duración de una lista de señales secuenciales con el tipo temporización es la suma de las duraciones de las señales que contiene más la suma de los retardos entre señales, especificados como parámetros de éstas.

Cuando se especifique un retardo entre señales para una señal que no esté en una lista secuencial de señales o sea el último elemento de una de ellas, no se tendrá en cuenta y se dirá que la señal ha sido completada en la terminación, antes de contar el retardo entre señales. La duración de una señal en una SignalList con retardo entre señales incluye este último.

Los grupos de señales y las listas de señales secuenciales en el mismo descriptor de señales se reproducirán simultáneamente.

Por definición, las señales van desde la terminación hacia el exterior del contexto, a no ser que se especifique otra cosa. Si el sentido de la señal se especifica en el descriptor de señales, reemplaza al sentido por defecto. En el caso de señales que tengan un parámetro de sentido, el parámetro de sentido del protocolo base tiene prelación sobre cualquier otro de dichos parámetros definido en los lotes, cuando ambos se especifiquen. Si el MGC especifica un sentido para una señal y la MG no puede aplicarlo, la MG retornará el código de error 501 ("No implementado"). Cuando una misma

señal se aplica a varias terminaciones dentro de una transacción, la MG debe considerar la utilización del mismo recurso para generar estas señales.

La producción de una señal en una terminación es detenida por la aplicación de un nuevo SignalsDescriptor, o por la detección de un evento en la terminación (véase 7.1.9).

Un nuevo descriptor de señal reemplaza a cualquier descriptor de señal existente. Toda señal aplicada a la terminación y que no esté en el descriptor sustituto será detenida, y se aplicarán las señales nuevas, con las siguientes excepciones. Las señales que estén presentes en el descriptor sustituto y tengan la bandera KeepActive serán continuadas si se están reproduciendo actualmente y todavía no han sido completadas. Si el descriptor de señales sustituto contiene una señal que no se está reproduciendo actualmente y tiene la bandera KeepActive, se ignorará la señal. Si el descriptor sustituto contiene una lista de señales secuenciales con el mismo identificador que el descriptor existente:

- no se tendrán en cuenta el tipo de señal y la secuencia de señales de la lista de señales secuenciales del descriptor sustituto , y
- la reproducción de las señales de la lista de señales secuenciales en el descriptor existente no será interrumpida.

7.1.12 Descriptor de auditoría (Audit)

El descriptor de auditoría especifica la información que se debe revisar. El descriptor de auditoría especifica la lista de descriptores y/o propiedades particulares que el sistema debe retornar. El descriptor de auditoría puede utilizarse en cualquier instrucción para exigir que se transmita en retorno cualquier descriptor que contenga los valores actuales de sus propiedades, eventos, señales y estadísticas incluso si ese descriptor no estaba presente en la instrucción, o no tenía parámetros subespecificados. Son posibles elementos del descriptor de auditoría:

Módem (descartado, véase 7.1.2)	
Mux	
Eventos	
Medios	
Señales	
ObservedEvents	
DigitMap	
Estadísticas	
Lotes	
EventBuffer	
Elementos particulares de Audit:	
Propiedades de medios	Mapa de dígitos
Eventos	Estadísticas
Memoria tampón (o intermedia) de eventos	Lotes
Señales, lista de señales	ContextAttribute

El descriptor de auditoría puede estar vacío, en cuyo caso no se retornan descriptores. Esto es conveniente en el caso de la instrucción Substraer, para inhibir el retorno de estadísticas, especialmente cuando se utilizan comodines.

7.1.13 Descriptor de cambio de servicio (ServiceChange)

El descriptor de cambio de servicio contiene los siguientes parámetros:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeAddress;
- ServiceChangeDelay;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMGCID;
- TimeStamp;
- Extension;
- ServiceChangeInfo;
- ServiceChangeIncompleteFlag.

Véase 7.2.8.

7.1.14 Descriptor de mapa de dígitos (DigitMap)

7.1.14.1 Definición, creación, modificación y supresión de mapa de dígitos

Un mapa de dígitos es un plan de marcación de números que reside en la pasarela de medios y que se utiliza para detectar y comunicar eventos de dígitos recibidos en una terminación. El descriptor mapa de dígitos contiene un nombre de mapa de dígitos y el mapa de dígitos que habrá de asignarse. El mapa de dígitos puede precargarse en la MG por una acción de gestión, y se puede invocar mediante su nombre en un descriptor de eventos, puede ser definido dinámicamente, y se puede invocar después mediante su nombre, o el propio mapa de dígitos puede ser especificado en el descriptor de eventos. Se permite, para un evento de ejecución completa de mapa de dígitos dentro de un descriptor de eventos, utilizar un nombre para referirse a un mapa de dígitos definido por un descriptor de mapa de dígitos dentro de la misma instrucción, con independencia de la petición transmitida de los descriptores respectivos.

Los mapas de dígitos definidos en un descriptor de mapa de dígitos pueden aparecer en cualquiera de las instrucciones normalizadas de manipulación de terminación, del protocolo. Un mapa de dígitos, una vez definido, puede utilizarse en todas las terminaciones especificadas por el TerminationID (posiblemente con comodines) en tal instrucción. Los mapas de dígitos definidos en la terminación Raíz son globales y pueden utilizarse en cada terminación en la MG, siempre que no se haya definido en la terminación dada un mapa de dígitos con el mismo nombre. Cuando un mapa de dígitos se define dinámicamente en un descriptor de mapa de dígitos:

- Se crea un nuevo DigitMap especificando un nombre que no está aún definido. El valor deberá estar presente.
- Se actualiza un valor de DigitMap suministrando un nuevo valor para el nombre que ya está definido. Las terminaciones que actualmente utilizan el mapa de dígitos seguirán empleando la definición antigua; los descriptores de eventos siguientes que especifican el nombre, incluido cualquier descriptor de eventos en la instrucción que contiene el descriptor de mapa de dígitos, utilizarán la definición nueva.
- Se suprime un mapa de dígitos suministrando un valor vacío para un nombre que ya está definido. Las terminaciones que actualmente utilizan el mapa de dígitos seguirán empleando la definición antigua.

7.1.14.2 Temporizadores mapa de dígitos

Los dígitos de un mapa de dígitos pueden protegerse por tres temporizadores: un temporizador de arranque (T, *start timer*), un temporizador corto (S, *short timer*), y un temporizador largo (L, *long timer*).

- 1) El temporizador de arranque (T) se utiliza antes de que haya cualquier dígito disponible para su procesamiento por referencia al mapa de dígitos. Si el temporizador de arranque se reemplaza por un valor cero, (T = 0), quedará inhabilitado. Esto implica que la MG esperará indefinidamente a recibir las cifras.
- 2) Si la pasarela de medios determina que son necesarios uno o más dígitos para que una cadena de dígitos concuerde con cualquiera de los patrones permitidos en el mapa de dígitos, el valor del temporizador entre dígitos debe ser de larga (L) duración (por ejemplo 16 segundos).
- 3) Si la cadena de dígitos concuerda con uno de los patrones de un mapa de dígitos, pero es posible que se reciban más dígitos que ocasionarían la concordancia con un patrón diferente, entonces, en vez de comunicar la concordancia inmediatamente, la MG debe aplicar el temporizador corto (S) y esperar más dígitos.

Los temporizadores son parámetros de un mapa de dígitos que pueden configurarse. Los valores por defecto de estos temporizadores deben ser configurados en la MG pero pueden ser reemplazados por valores especificados dentro del mapa de dígitos.

7.1.14.3 Sintaxis de mapa de dígitos

La sintaxis formal del mapa de dígitos está definida por la regla DigitMap en la descripción de sintaxis formal del protocolo (véanse los anexos A y B). De acuerdo con esta sintaxis, un mapa de dígitos se define como una cadena o como una lista de cadenas. Cada cadena de la lista es una secuencia de eventos alternativa, especificada ya sea como una secuencia de símbolos de mapa de dígitos, o como una expresión regular de símbolos de mapa de dígitos. Estos símbolos del mapa de dígitos, los dígitos de "0" a "9" y las letras de "A" hasta un valor máximo dependiendo del sistema de señalización, pero nunca superior a "K", corresponden a eventos especificados dentro de un lote que ha sido designado en el descriptor de eventos en la terminación a la que se está aplicando el mapa de dígitos. (La correspondencia entre los eventos y los símbolos del mapa de dígitos se define en la documentación sobre lotes asociados con sistemas de señalización asociada al canal como DTMF, MF o R2. Los dígitos de "0" al "9" deben hacerse corresponder con los eventos de los mismos dígitos dentro del sistema de señalización concernido. Las letras deben asignarse de una manera lógica, facilitando el uso de una gama de notación para eventos alternativos.)

La letra "x" se utiliza como comodín para designar cualquier evento correspondiente a símbolos en la gama "0"- "9". La cadena también puede contener gamas explícitas y, de modo más general, conjuntos explícitos de símbolos, que designan eventos alternativos, cualquiera de los cuales satisface la posición del mapa de dígitos. Por último, el símbolo punto ".", representa cero o más repeticiones del selector de eventos (evento, gama de eventos, conjunto de eventos alternativos, o comodín) que le preceda. Como consecuencia de la tercera regla de temporización anterior, para la temporización entre eventos al establecer la concordancia de un símbolo punto de un terminal, se utiliza el temporizador corto por defecto.

Además de estos símbolos de eventos, la cadena puede contener los especificadores de temporización entre eventos "S" y "L" y el modificador de duración "Z". "S" y "L" indican respectivamente que la MG debe utilizar el temporizador corto (S) o el temporizador de larga duración (L) para los eventos subsiguientes, haciendo caso omiso de las reglas de temporización descritas anteriormente. Si hay un especificador de temporización explícito en una secuencia de eventos alternativa, pero no se da ninguno en ninguna otra alternativa candidato, debe utilizarse el valor de temporizador fijado por el especificador de temporización explícito. Si todas las secuencias con controles de temporización explícitos se extraen del conjunto candidato, se restablecerán las

anteriores reglas de temporización por defecto. Los especificadores S y L se ignorarán si se utilizan dentro de una gama de notación. Por último, si especificadores de temporización divergentes están actuando en secuencias alternativas diferentes, se utilizará el temporizador largo.

"Z" designa un evento de duración larga: colocado al principio del símbolo o símbolos que designan el evento o eventos que satisfacen una posición de dígito dada, indica que esa posición es satisfecha solamente si la duración del evento excede el umbral de la duración larga. Se supone que el valor de este umbral está configurado en la MG, pero puede ser modificado por una especificación en DigitMap, de la misma forma que los temporizadores T, L y S. Si el especificador Z no va seguido de un dígito (0-9 o A-K), la MG rechazará el mapa de dígitos al considerarlo un procedimiento no válido. Cuando se utiliza en una gama de notación, el especificador Z se aplica exclusivamente al dígito que le sigue inmediatamente. Cuando se utiliza justo antes de una gama, el modificador Z se aplica a todos los dígitos de la gama (condición de larga duración para concordancia en la gama).

7.1.14.4 Evento de ejecución completa de mapa de dígitos

Un mapa de dígitos está activo mientras el descriptor de eventos que lo invocó permanezca activo y no se haya terminado. Un mapa de dígitos se considera ejecutado completamente cuando:

- ha expirado un temporizador, o
- hay concordancia con una de las secuencias de eventos y no se puede lograr concordancia con ninguna otra secuencia de eventos del mapa de dígitos detectando otro evento (concordancia inequívoca); o
- se ha detectado un evento cuyas características hacen que ya sea imposible una concordancia con una de las secuencias de eventos completa del mapa de dígitos, cualesquiera que sean los eventos adicionales recibidos.

Tras procesarlo, se generará un evento de ejecución completa del mapa de dígitos como se ha definido en el lote que proporciona los eventos que se están estableciendo en el mapa de dígitos. En este punto, el mapa de dígitos es desactivado. Los eventos siguientes en el lote son procesados de esta forma:

- Si EventBufferControl está en "LockStep", los siguientes eventos de dígitos serán tratados de la misma forma que cualquier otro evento.
- Si EventBufferControl está "DESACTIVADO" ("OFF"), el descriptor de eventos activo no ha cambiado, y no hay eventos de dígitos particulares habilitados en ese descriptor, se iniciará el proceso de almacenamiento intermedio de dígitos. Este proceso se realizará durante el tiempo de almacenamiento intermedio especificado en el evento original de ejecución completa del mapa de dígitos, o hasta que el descriptor de eventos activo sea reemplazado.

La forma lógica de la memoria intermedia de dígitos es una cadena de marcación con las cifras como están representadas en el mapa de dígitos, precedidas eventualmente por 'Z'. Se deberá utilizar un valor umbral de duración del tono para identificar los eventos largos igual al que se utilizó con el mapa de dígitos completado más recientemente.

Por defecto vuelve a establecerse un tiempo de almacenamiento intermedio cero (no se hace almacenamiento intermedio), a no ser que se indique explícitamente otra cosa en el evento de ejecución completa del mapa de dígitos. Si el proceso de almacenamiento intermedio se interrumpe porque ha expirado el tiempo correspondiente, se descartará el contenido de esta memoria intermedia de dígitos.

Si un nuevo descriptor de eventos interrumpe el proceso de almacenamiento intermedio, y este descriptor de eventos contiene un nuevo evento de ejecución completa de mapa de dígitos del mismo lote que el anterior, los dígitos retenidos en memoria intermedia serán procesados por referencia al mapa de dígitos como se indica a continuación. Los dígitos en

memoria intermedia que no fueran consumidos por el nuevo mapa de dígitos serán tratados como si hubieran sido observados después de completar ese mapa.

Ahora bien, si el nuevo descriptor de eventos permite la comunicación de eventos de dígitos particulares, todo el conjunto de dígitos almacenados será procesado inmediatamente, se comunicarán los eventos correspondientes y se borrará la memoria intermedia.

Por último, si el nuevo descriptor de eventos no permite un evento de ejecución completa de mapa de dígitos ni la comunicación de eventos de dígitos particulares del paquete correspondiente, se descartará el contenido de la memoria intermedia y se terminará el proceso de almacenamiento.

7.1.14.5 Procedimientos de mapa de dígitos

Hasta la ejecución completa, los sucesivos eventos serán procesados de acuerdo con las reglas siguientes:

- 1) La "cadena de marcación actual", una variable interna, está inicialmente vacía. El conjunto de secuencias de eventos alternativas candidato incluye todas las alternativas especificadas en el mapa de dígitos.
- 2) En cada paso, cuando hay dígitos en la memoria intermedia, se retira el que fue almacenado primero (posiblemente con el indicador de duración larga (Z)) y el procesamiento continúa en el siguiente paso como si se acabara de observar el evento de dígito. Si así no fuera, se fija un temporizador en la situación de espera del evento siguiente, basándose, bien en las reglas de temporización por defecto dadas anteriormente o bien en la temporización explícita especificada en una o más secuencias de eventos alternativas. Si el temporizador expira y un miembro del conjunto de alternativas candidato es totalmente satisfecho, se comunica una compleción de temporización totalmente concordante. Si el temporizador expira y parte o ninguna de las alternativas candidato es satisfecha, se comunica una compleción de temporización con concordancia parcial. En ambos casos, si el evento de compleción de mapa de dígitos permite las comunicaciones detalladas de fin de temporización, la cadena de marcación comunicada terminará en 'L', 'S' o 'T' según el caso.
- 3) Si se detecta un evento antes de que expire el temporizador, se refleja en un símbolo de cadena de dígitos y se añade provisionalmente al final de la cadena de marcación actual. Se señala la duración del evento (larga o no larga) si, y solamente si, dicha duración es pertinente en la posición del símbolo actual (porque al menos una de las secuencias de eventos alternativas candidato incluye el modificador "Z" en esta posición en la secuencia).
- 4) La cadena de marcación actual se compara con las secuencias de eventos alternativas candidato. En caso de concordancia con una secuencia que espera un evento de larga duración en esta posición (es decir, el evento tuvo una duración larga y cumplió la especificación para esta posición), y únicamente en este caso, cualesquiera secuencias de eventos alternativas que no especifiquen un evento de larga duración en esta posición son descartadas, y la cadena de marcación actual es modificada insertando un modificador "Z" al principio del símbolo que representa el evento más reciente. Cualquier secuencia que espere un evento de larga duración en esta posición pero no concuerde con el evento observado es descartada del conjunto candidato. Si quedan secuencias de eventos alternativas que no especifican un evento de larga duración en la posición dada, en el conjunto candidato después de la aplicación de las reglas anteriores, la duración del evento observado se considera irrelevante en la evaluación de las concordancias con ellas.
- 5) Si sólo permanece un candidato que ha tenido una concordancia completa, se genera un evento de ejecución completa que indica que existe una concordancia inequívoca. Si no permanece ningún candidato, se elimina de la cadena de marcación actual el evento más reciente y se genera un evento de compleción indicando una concordancia total si uno de los candidatos del paso anterior fue completamente satisfecho antes que se detectara el

último evento, o una concordancia parcial en los demás casos. El evento eliminado de la cadena de marcación actual será comunicado con un evento separado, será almacenado en memoria intermedia o será descartado, atendiendo a las reglas descritas anteriormente. Se decide con los siguientes criterios:

- a) El evento de ejecución completa del mapa de dígitos puede especificar que se comunique el evento eliminado como un parámetro del evento de ejecución completa. Se hará entonces independientemente del procesamiento ulterior del evento de dígitos.
 - b) El evento de ejecución completa del mapa de dígitos puede especificar que se descarte el dígito adicional. En este caso, será descartado inmediatamente. Sólo los eventos posteriores serán conservados en memoria intermedia o tratados de alguna otra forma.
- 6) Si no se comunica ningún evento de ejecución completa como resultado del paso 5), el procesamiento retorna al paso 2).

7.1.14.6 Activación de mapa de dígitos

Un mapa de dígitos es activado cada vez que se aplica a la terminación un nuevo descriptor de eventos o cuando se activa un descriptor de eventos insertado; ese descriptor de eventos contiene un evento de ejecución completa de mapa de dígitos. Este evento contiene un campo eventDM en el campo acciones solicitadas. Cada nueva activación de un mapa de dígitos se inicia en el paso 1) del procedimiento anterior, con una cadena de marcación actual limpia. Cualquier contenido anterior de la cadena de marcación actual procedente de una activación anterior se pierde.

Un evento de ejecución completa de mapa de dígitos que no contiene el campo eventDM en su campo acciones solicitadas se considera un error. Al recibir tal evento en un descriptor de eventos, una MG contestará con una respuesta de error, con el código de error 457 ("Falta parámetro en señal o evento").

7.1.14.7 Interacción de mapa de dígitos y procesamiento de evento

Mientras el mapa de dígitos esté activado, se permite la detección de todos los eventos definidos en el lote que contiene el evento de ejecución completa del mapa de dígitos especificado. El comportamiento de un evento normal (por ejemplo, la parada de señales a menos que el evento de ejecución completa de dígitos tenga la bandera KeepActive activada) continúa aplicándose a cada uno de tales eventos detectados, salvo que:

- los eventos del lote que contiene el evento de ejecución completa del mapa de dígitos especificado distinto del propio evento de compleción no se notifican y no tienen efectos secundarios a menos que hayan sido habilitados individualmente, y
- un evento que provoca un evento de ejecución completa de concordancia parcial no se reconoce y, por consiguiente, no tiene efectos secundarios hasta que vuelva a ser procesado después del reconocimiento del evento de ejecución completa de mapa de dígitos. Los eventos de dígitos tampoco son reconocidos, y no tienen efectos secundarios antes de que sean procesados.

7.1.14.8 Comodines

Hay que señalar que si un lote contiene un evento de compleción de mapa de dígitos, entonces una especificación de evento compuesta por el nombre del lote con un comodine EventID activará un mapa de dígitos; con ese fin, la especificación del evento incluirá un campo EventDM de acuerdo con 7.1.14.6. Si el lote contiene también los eventos de dígito, esta forma de especificación de evento hará que los eventos individuales sean comunicados al MGC tal como han sido detectados.

7.1.14.9 Ejemplo

A título de ejemplo, considérese el siguiente plan de marcación:

0	Operador local
00	Operador de larga distancia
xxxx	Número de extensión local (arranca con 1-7)
8xxxxxxx	Números locales
#xxxxxxx	Extensión fuera del emplazamiento
*xx	Servicios estrella
91xxxxxxxxxxx	Número de larga distancia
9011 + hasta 15 dígitos	Número internacional

Si se utiliza el lote de detección de DTMF descrito en E.6 para recopilar los dígitos marcados, el plan de marcación mostrado anteriormente da entonces como resultado el siguiente mapa de dígitos:

(0 | 00 | [1-7]xxx | 8xxxxxxx | Fxxxxxxx | Exx | 91xxxxxxxxxxx | 9011x.)

7.1.15 Descriptor de estadísticas

El descriptor de estadísticas proporciona información sobre el estado y la utilización de una terminación durante su existencia dentro de un contexto específico (efímera) o cuando está fuera de un contexto NULL (física). Puede haber información estadística acerca de una terminación o asociada con un descriptor de tren. La opción por defecto es estadísticas de terminaciones. Cuando un tren no pueda soportar una estadística determinada, se retornará el código de error 460 ("No se puede utilizar esta estadística en el tren"). Para cada terminación se mantiene, cuando procede, un conjunto normalizado de estadísticas (por ejemplo, el número de octetos enviados y recibidos). Por defecto, las estadísticas concretas que son comunicadas para una terminación dada corresponden a los lotes tramitados por la terminación. Es posible indicar, utilizando el descriptor, cuáles son las estadísticas que se deben recolectar. La configuración de un descriptor de estadísticas reemplaza a cualquier otro descriptor de estadísticas previamente configurado. En otras palabras, si se desea mantener estadísticas anteriores, hay que incluirlas en el nuevo descriptor de estadísticas y no se han de reiniciar los valores estadísticos. Las estadísticas que han sido suprimidas de un descriptor de estadísticas mantendrán sus valores hasta que se suprima la terminación. No obstante, si una estadística determinada es reactivada por un descriptor de estadísticas ulterior, se restablecerá el valor. Un MGC puede reactivar todas las estadísticas en una terminación o en un tren emitiendo un descriptor de estadísticas que tenga una sola estadística, cuyos PackageID y StatisticID comodines sean "ALL". Para reactivar todas las estadísticas de un determinado lote en una terminación o un tren, se emite un descriptor de estadísticas con una sola estadística, especificando el PackageID y StatisticID comodín ("ALL"). La recepción de un descriptor vacío implica que no se deben recolectar estadísticas en dicha terminación.

Por defecto, salvo si se ha utilizado antes un descriptor de estadísticas vacío para indicar que no se deben recolectar estadísticas, las estadísticas de terminación y tren se notifican cuando deja de existir la terminación o retorna al contexto NULL debido a una instrucción Substraer. Es posible evitar este comportamiento por defecto incluyendo un descriptor de auditoría vacío en dicha instrucción. Cuando se suprime un tren conforme a 7.1.6, la opción por defecto es no informar estadísticas de tren. Se debe efectuar una auditoría del descriptor de estadísticas en el tren, antes de que el tren sea suprimido para recolectar estadísticas. También pueden obtenerse en retorno estadísticas enviando la instrucción AuditValue o cualquier instrucción Añadir/Desplazar/Modificar que utilice el descriptor de auditoría.

Las estadísticas son acumulativas; el hecho de comunicar estadísticas no restablece el estado inicial. El valor de una estadística en una terminación es el resultado de la aplicación de una función superior (por ejemplo, suma o promedio) a los valores, como si se hubiera aplicado a todos los trenes en la terminación. Esta función superior depende del tipo de estadística utilizado. Si no se especifica otra cosa en el lote que define una estadística determinada, el valor por defecto es la suma de los valores. Las estadísticas son reiniciadas cuando una terminación deja de existir o se pone nuevamente en el contexto NULO por efecto de la instrucción Substraer.

7.1.16 Descriptor de lotes

Se utiliza solamente con la instrucción AuditValue y devuelve una lista de lotes tratados por la terminación.

7.1.17 Descriptor de eventos observados (ObservedEvents)

El descriptor de eventos observados se suministra con la instrucción Notificar para informar al MGC del evento o los eventos que fueron detectados. Si se utiliza con la instrucción AuditValue, este descriptor retorna eventos que están almacenados en la memoria tampón de eventos y no han sido notificados. El descriptor de eventos observados contiene el RequestID del descriptor de eventos que activó la notificación, el(los) evento(s) detectados, la hora de detección (es facultativo) y cualesquiera parámetros del evento observado. La hora de detección se comunica con una precisión de centésimas de segundo.

7.1.18 Descriptor de topología

Se utiliza un descriptor de topología para especificar sentidos de flujo entre terminaciones en un contexto. A diferencia de los descriptores presentados en las cláusulas anteriores, el descriptor de topología se aplica a un contexto y no a una terminación. En la topología por defecto de un contexto, la transmisión de cada terminación la reciben todas las demás terminaciones. La implementación del descriptor de topología es facultativa. Si una MG no soporta descriptores de topología y recibe una instrucción que contiene uno, retorna el código de error 444 ("Descriptor no soportado o no conocido") y, facultativamente, incluye una cadena que contiene el nombre del descriptor no soportado ("Topología") en el texto explicativo del error, en el descriptor de error.

El descriptor de topología aparece antes que las instrucciones en una acción. Puede haber una acción que contenga solamente un descriptor de topología, siempre que el contexto al que se aplica la acción ya exista.

Un descriptor de topología consiste en una secuencia de terminaciones asociadas con indicación del tipo de asociación (*T1*, *T2*, *association*[*Streamid*]). *T1* y *T2* especifican las terminaciones dentro del contexto, posiblemente con los comodines ALL y CHOOSE. Si se incluye el campo facultativo StreamID, el tipo de asociación indicado se aplica únicamente al flujo entre *T1* y *T2* identificado por ese StreamID. Cuando no se incluye el campo StreamID, la topología se aplica a todos los flujos en la terminación. En la indicación *asociación* se describen las características de flujos de medios entre estas dos terminaciones, de esta forma:

- (*T1*, *T2*, aislar) significa que las terminaciones que concuerdan con *T2* no reciben medios desde las terminaciones que concuerdan con *T1*, y viceversa.
- (*T1*, *T2*, un solo sentido) significa que las terminaciones que concuerdan con *T2* reciben medios de las terminaciones que concuerdan con *T1*, pero no a la inversa. En este caso, está permitida la utilización del comodín ALL de tal modo que haya terminaciones que concuerden con *T1* y o con *T2*, pero no con ambas.
- (*T1*, *T2*, un solo sentido externo, *onewayexternal*) significa que las terminaciones que concuerdan con *T2* reciben medios enviados externamente por terminaciones que concuerdan con *T1*, pero no lo contrario. En este caso no se permite la utilización del comodín ALL para *T1*.

- (*T1, T2*, un solo sentido ambos, *onewayboth*) significa que las terminaciones que concuerdan con *T2* reciben medios enviados y recibidos externamente por las terminaciones que concuerdan con *T1*, pero no lo contrario. En este caso no se permite la utilización del comodín ALL para *T1* y/o *T2*.
- (*T1, T2*, ambos sentidos; *bothway*) significa que las terminaciones que concuerdan con *T2* reciben medios de las terminaciones que concuerdan con *T1* y viceversa. En este caso se permite utilizar comodines, de modo que haya terminaciones que concuerden con *T1* y *T2*. Sin embargo, si hay una terminación que concuerda con ambas, no se introduce una conexión en bucle.

También pueden utilizarse comodines CHOOSE en *T1* y *T2*, con las siguientes restricciones:

- la acción (véase la cláusula 8) de la que forma parte el descriptor de topología contiene una instrucción Añadir en la que se utiliza un comodín CHOOSE;
- si aparece un comodín CHOOSE en *T1* o *T2*, no se especificará un nombre parcial.

El comodín CHOOSE en un descriptor de topología concuerda con el TerminationID que la MG asigna en la primera instrucción Añadir que utiliza un comodín CHOOSE en la misma acción. Una terminación existente que concuerda con *T1* o *T2* en el contexto al que es añadida una terminación, será conectada a la terminación recién añadida según se especifica en el descriptor de topología. Cuando no se menciona alguna terminación en un descriptor de topología, las topologías correspondientes no serán modificadas. Ahora bien, si se añade otra terminación a un contexto, la forma de asociación entre ésta y las demás terminaciones de ese contexto será por defecto ambos sentidos excepto si se introduce un descriptor de topología con una instrucción diferente (por ejemplo, si se añade *T3* a un contexto que tiene *T1* y *T2* con topología en un solo sentido con *T1* (*T3, T1*, un solo sentido), entonces será conectado en ambos sentidos a *T2*).

Si la topología se aplica a un tren en particular (*T1, T2*, asociación, StreamID), la topología de los otros trenes entre las terminaciones no cambiará.

En un descriptor de topología no aparecerán diferentes asociaciones entre dos terminaciones (*Ti, Tj*) con el campo facultativo StreamID o sin él, para evitar un comportamiento indefinido. Por ejemplo, (*T1, T2* ambos sentidos) y (*T1, T2*, aislar, S1) no deben aparecer en el mismo descriptor. Si la MG recibe un descriptor de topología de este tipo, dará la siguiente respuesta de código de error 421 ("Acción desconocida o combinación de acciones no permitida").

Hay que implementar una conexión unidireccional de tal manera que el cambio en la topología no sea conocido por las otras terminaciones del contexto.

Véanse la figura 7, el cuadro siguiente y la figura 8, como ejemplos de los efectos que resultan de incluir descriptors de topología en acciones. En estos ejemplos se supone que los descriptors de topología se aplican secuencialmente. Las figuras 9 y 10 son ejemplos independientes, en los que se muestran los efectos específicos de la utilización de las topologías *onewayexternal* y *onewayboth*.

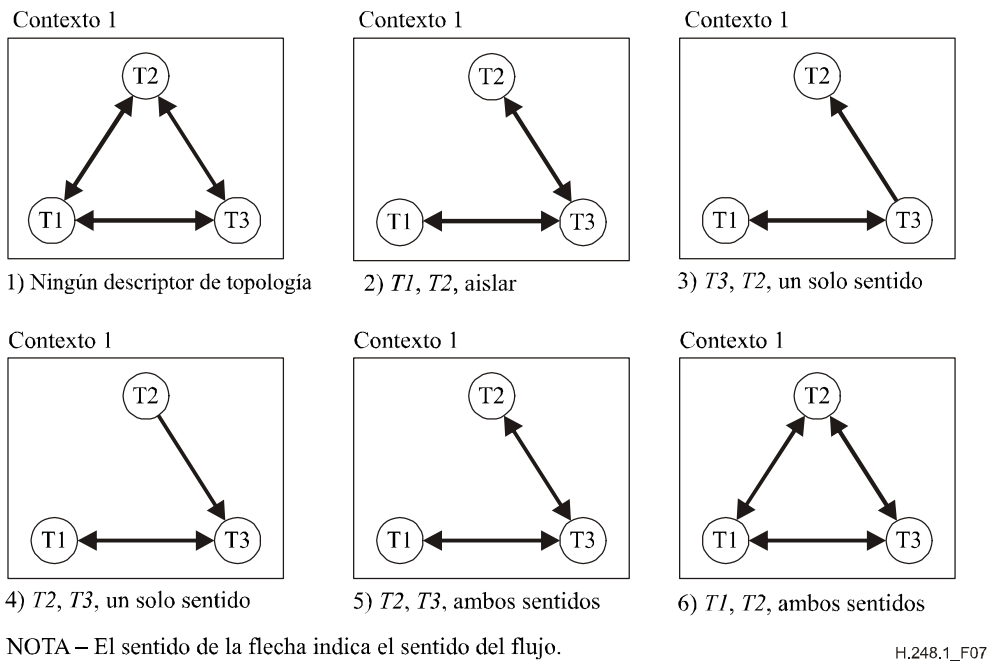


Figura 7/H.248.1 – Ejemplo de topologías

Topología	Descripción
1	Ningún descriptor de topología Cuando no se incluye ningún descriptor de topología, todas las terminaciones tienen una conexión en ambos sentidos con todas las demás terminaciones.
2	<i>T1, T2</i> , aislar Suprime la conexión entre <i>T1</i> y <i>T2</i> . <i>T3</i> tiene una conexión en ambos sentidos con <i>T1</i> y <i>T2</i> . <i>T1</i> y <i>T2</i> tienen conexión en ambos sentidos con <i>T3</i> .
3	<i>T3, T2</i> , un solo sentido Una conexión en un solo sentido de <i>T3</i> a <i>T2</i> (esto es, <i>T2</i> recibe flujo de medios de <i>T3</i>). Una conexión en ambos sentidos entre <i>T1</i> y <i>T3</i> .
4	<i>T2, T3</i> , un solo sentido Una conexión en un solo sentido de <i>T2</i> a <i>T3</i> . Se mantiene la conexión en ambos sentidos entre <i>T1</i> y <i>T3</i>
5	<i>T2, T3</i> , ambos sentidos <i>T2</i> y <i>T3</i> tienen conexión en ambos sentidos. Esta situación es la misma que 2.
6	<i>T1, T2</i> , ambos sentidos (<i>T2, T3</i> , ambos sentidos y <i>T1, T3</i> ambos sentidos pueden ser conexiones implícitas o explícitas). Todas las terminaciones tienen una conexión en ambos sentidos con cada una de las demás.

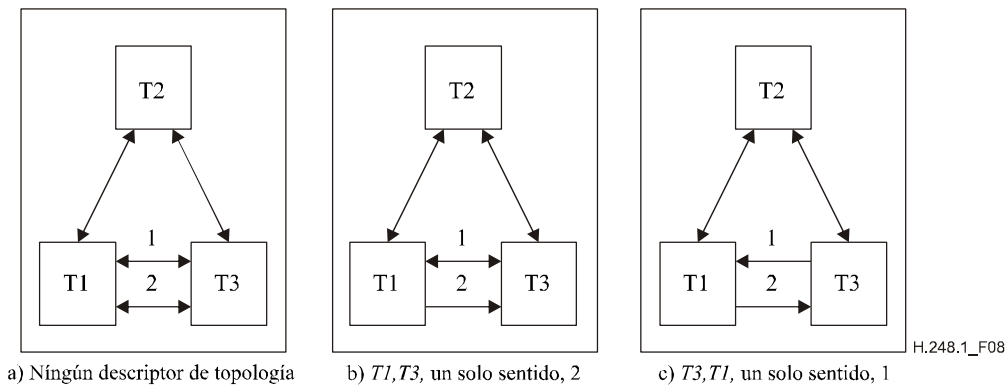


Figura 8/H.248.1 – Ejemplo de topología a nivel de tren

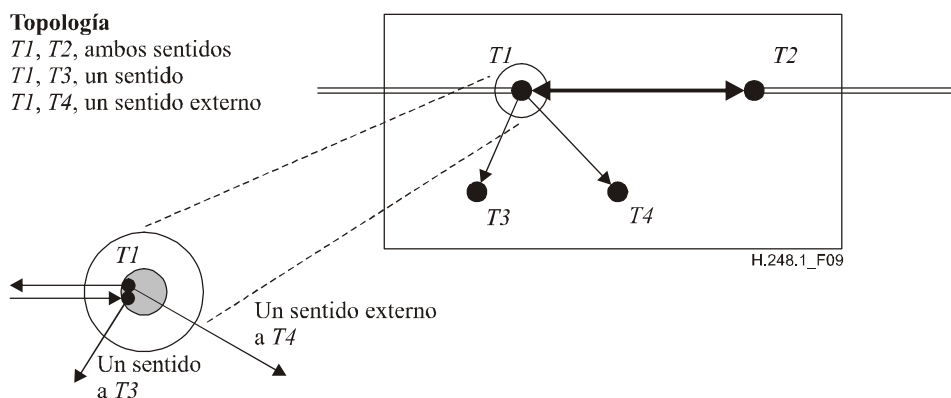


Figura 9/H.248.1 – Comparación de las topologías "un solo sentido externo" y "un solo sentido"

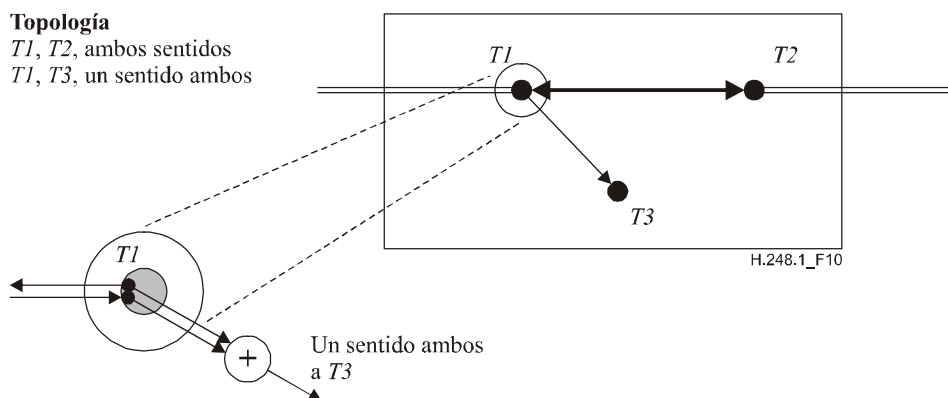


Figura 10/H.248.1 – Funcionamiento de la topología "un solo sentido ambos"

7.1.19 Descriptor de atributo de contexto (ContextAttribute)

Se utiliza para especificar propiedades (definidas en paquetes) que se aplican al contexto como un todo, y se emplea en un contexto y no en una terminación. Las propiedades de terminación no son válidas en el descriptor de atributo de contexto. Este descriptor aparece en una acción antes que las instrucciones. Puede haber una acción que contenga solamente uno de estos descriptors, siempre y cuando el contexto indicado ya exista. Los valores de las propiedades de contexto pueden darse subespecificados conforme a 7.1.1.

Una nueva configuración de este descriptor reemplazará a cualquier otra previamente efectuada para él en la MG. Para conservar información acerca de la configuración anterior el controlador deberá incluirla en la nueva configuración. Para suprimir alguna información del descriptor existente, el MGC simplemente lo reenvía sin incluir dicha información. El hecho de incluir un descriptor de auditoría de contexto o uno de atributo de contexto en una acción que contenga solamente la instrucción AuditValue o AuditCapabilities no implica que se tenga una nueva configuración.

El parámetro ContextIDList se utiliza para proveer una lista compacta de ContextID. Si se quiere solicitar la utilización de dicho parámetro en una respuesta, hay que incluirlo en la petición correspondiente.

7.1.20 Descriptor de error

Si un respondedor se encuentra con un error mientras procesa una petición de transacción, incluirá un descriptor de error en su respuesta. Asimismo, una petición de notificación puede contener un descriptor de error.

Un descriptor de error consiste en un código de error registrado en IANA, facultativamente acompañado por un texto explicativo del error (texto de error). La Rec. UIT-T H.248.8 contiene una lista de códigos de error válidos y de descripciones de errores.

Un descriptor de error se especificará en el "nivel más profundo" que sea semánticamente adecuado para el error que se describe y que sea posible en presencia de eventuales problemas de análisis gramatical en la petición original. Un descriptor de error puede hacer referencia a un bloque de instrucciones sintáctico diferente de aquél en el que aparece. Por ejemplo, el descriptor de error 422 ("Error de sintaxis en acción"), podría aparecer dentro de una instrucción aun cuando hiciera referencia al bloque más amplio, la acción, y no a la instrucción concreta en que aparece.

7.2 Interfaz de programación de aplicación para las instrucciones

A continuación se presenta una interfaz de programación de aplicación (API, *application programming interface*) que describe las instrucciones del protocolo. Se presenta esta API para ilustrar las instrucciones y sus parámetros, sin que se pretenda especificar su implementación (por

ejemplo mediante la utilización de llamadas a funciones de bloqueo). La API describe los parámetros de entrada entre paréntesis después del nombre de instrucción y los valores de retorno antes de la instrucción. Esto tiene una finalidad exclusivamente descriptiva; la sintaxis y la codificación realmente utilizadas para las instrucciones se especifican en subcláusulas ulteriores. El orden de los parámetros de las instrucciones no es fijo. Los descriptores pueden aparecer como parámetros de instrucciones en cualquier orden. Los descriptores serán procesados en el orden en que aparezcan.

Cualquier respuesta a una instrucción puede contener un descriptor de error; la API no muestra esto específicamente.

Todos los parámetros encerrados entre corchetes ([...]) se consideran facultativos.

7.2.1 Añadir (Add)

La instrucción Añadir añade una terminación a un contexto.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationIDList

[, MediaDescriptor]

[, ModemDescriptor] (*)

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

[, StatisticsDescriptor]

)

(*) El descriptor de módem no fue considerado en H.248.1, Versión 2 (05/2002).

La TerminationIDList especifica una o varias terminaciones que habrán de añadirse al contexto. Las terminaciones recrean o se toman del contexto NULL. Si se utiliza un comodín CHOOSE en el TerminationID, el sistema retornará el TerminationID seleccionado. Pueden utilizarse comodines ALL en una instrucción Añadir, pero no es frecuente. Si el comodín concuerda con más de un TerminationID, se intentan todas las concordancias posibles, y se informa el resultado de cada uno de ellos. El orden de los intentos cuando varios TerminationID concuerdan no se especifica.

El descriptor de medios facultativo describe todos los trenes de medios.

El descriptor de múltiplex facultativo especifica un múltiplex, en su caso. Por razones de conveniencia, si un descriptor de múltiplex está presente en una instrucción Añadir e indica terminaciones que no están en ese momento en el contexto, dichas terminaciones se añaden al contexto como si se hubiesen invocado instrucciones de Añadir individuales que indicaran esas terminaciones. Si aparece un error en tal instrucción implícita Añadir, se retornará el error 471 ("Fallo de instrucción añadir implícita para múltiplex") y cesará el procesamiento posterior de la instrucción.

El parámetro descriptor de eventos es facultativo. Si está presente, proporciona la lista de eventos que deben detectarse en la terminación.

El parámetro descriptor de memoria intermedia de eventos es facultativo. Si está presente, proporciona la lista de los eventos que la MG deberá detectar e introducir en memoria tampón cuando EventBufferControl sea igual a LockStep.

El parámetro descriptor de señales es facultativo. Si está presente proporciona la lista de señales que deben aplicarse a la terminación.

El parámetro descriptor de mapa de dígitos es facultativo. Si está presente da una definición de mapa de dígitos que puede utilizarse en un descriptor de eventos.

El parámetro descriptor de auditoría es facultativo. Si está presente, la instrucción retornará los valores de todos los descriptores/propiedades/señales/eventos/estadísticas especificados en el descriptor de auditoría.

El descriptor de estadísticas es facultativo. De haberlo, indica las estadísticas que el MGC desea recolectar para la terminación o tren. Cada vez que se valide la estadística se restablecerá el valor estadístico.

Todos los descriptores que puedan ser modificados pueden ser retornados por la MG si un parámetro fue subespecificado o sobreespecificado. ObservedEvents, Statistics y Packages, y los descriptores EventBuffer solamente son retornados si se solicita en el descriptor de auditoría.

No se utilizará Añadir en una terminación cuyo ServiceState sea "fuera de servicio" ("OutOfservice").

7.2.2 Modificar

La instrucción Modificar modifica las propiedades de una terminación.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Modify(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

```
[, MuxDescriptor]
[, EventsDescriptor]
[, EventBufferDescriptor]
[, SignalsDescriptor]
[, DigitMapDescriptor]
[, AuditDescriptor]
[, StatisticsDescriptor]
)
```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 2 (05/2002).

La TerminationIDList especifica las terminaciones que se han de modificar. El TerminationID puede ser específico si se va a modificar una sola terminación en el contexto. El empleo de comodines en el TerminationID puede ser adecuado para algunas operaciones. Si el comodín concuerda con más de un TerminationID, se intentan todas la concordancias posibles, y se informa del resultado de cada uno de ellos. No se especifica el orden de los intentos cuando múltiples TerminationID concuerdan. La TerminationID CHOOSE es un error, pues la instrucción Modificar sólo puede utilizarse en terminaciones existentes.

Por razones de conveniencia, si un descriptor de múltiplex está presente en una instrucción Modificar:

- Si el nuevo descriptor de múltiplex indica terminaciones que no aparecen entonces en el Contexto, dichas terminaciones se añaden al contexto como si se hubieran invocado instrucciones Añadir en que se indicaran dichas terminaciones.
- Si cualesquiera terminaciones antes indicadas en el descriptor de múltiplex ya no están presentes en el nuevo descriptor de múltiplex, se sustraen del contexto como si se hubieran invocado instrucciones Substraer en que se indicaran esas terminaciones.

Los parámetros restantes de la instrucción Modificar son los mismos que los de Añadir. El sistema puede retomar los mismos valores que en Añadir.

7.2.3 Substraer (Subtract)

La instrucción Substraer desconecta una terminación de su contexto y retorna estadísticas sobre la participación de la terminación en el contexto.

TerminationIDList

```
[,MediaDescriptor]
```

```
[,ModemDescriptor] (*)
```

```
[,MuxDescriptor]
```

```
[,EventsDescriptor]
```

```
[,SignalsDescriptor]
```

```
[,DigitMapDescriptor]
```

```
[,ObservedEventsDescriptor]
```

```
[,EventBufferDescriptor]
```

```
[,StatisticsDescriptor]
```

```
[,PackagesDescriptor]
```

```
Subtract(TerminationIDList
```

```
    [, AuditDescriptor]
```

```
)
```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 2 (05/2002).

La TerminationIDList en los parámetros de entrada representa la terminación o terminaciones que se retiran. El TerminationID puede ser específico o puede ser un valor de comodín que indica que se han de retirar todas las terminaciones (o un conjunto de terminaciones conexas) en el contexto de la instrucción Substraer. Si el comodín concuerda con más de un TerminationID, se intentan todas las concordancias posibles, y se informan los resultados para cada uno de ellos. No se especifica el orden de los intentos cuando múltiples TerminationID concuerdan.

La utilización de CHOOSE en el TerminationID es un error, pues la instrucción Substraer sólo puede utilizarse en terminaciones existentes.

Se puede utilizar ALL para ContextID y para TerminationID en una instrucción Substraer, lo que tendría el efecto de eliminar todos los contextos, eliminar todas las terminaciones efímeras y retornar todas las terminaciones físicas a contexto NULL. No está permitido Substraer una terminación del contexto NULL.

Por razones de conveniencia, si una terminación multiplexora es el objeto de una instrucción Substraer, toda terminación de portador indicada en su descriptor de multiplex se substraer del contexto como si se hubieran invocado instrucciones Substraer que indicaran dichas terminaciones.

Por defecto, salvo si ha sido anulado, el sistema retorna el descriptor de estadísticas, tanto para terminaciones como para trenes, para comunicar información tomada en la terminación o las terminaciones especificadas en la instrucción. La información comunicada vale para la existencia de la terminación (o terminaciones) en el contexto del cual se substraer.

El descriptor de auditoría es facultativo. Si está presente, la instrucción provocará en retorno solamente los ítems especificados en el descriptor de auditoría, que puede estar vacío. Si se omite el descriptor de auditoría, se devuelve por defecto el descriptor de estadísticas. Los valores de retorno posibles son los mismos que los de Añadir.

Si se substraer una terminación configurada de un contexto, se restablecerán los siguientes valores para su descriptor:

- el valor por defecto, si fue especificado para el descriptor y no fue reemplazado por la configuración;
- en caso contrario, el valor configurado.

7.2.4 Desplazar (Move)

La instrucción Desplazar desplaza una terminación de su contexto actual a otro contexto, en una operación atómica. La instrucción Desplazar es la única instrucción que hace referencia a una terminación en un contexto diferente del contexto al que se aplica la instrucción. La instrucción Desplazar no se utilizará para desplazar terminaciones al contexto NULL, o desde dicho contexto.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```

Move( TerminationIDList
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
    [, StatisticsDescriptor]
)

```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 2 (05/2002).

La TerminationIDList especifica la terminación o terminaciones que habrán de desplazarse. Los TerminationID pueden ser comodines, pero no se utilizará CHOOSE . Si el comodín concuerda con más de un TerminationID, se intentan todas las concordancias posibles, y se informan los resultados para cada uno de ellos. No se especifica el orden de los intentos cuando concuerdan varios TerminationID. El contexto al que se desplaza la terminación se indica mediante el ContextID objetivo en la acción. Si se desplaza fuera un contexto la última terminación restante, el contexto se suprime.

La instrucción Desplazar no afecta a las propiedades de la terminación sobre la que actúa, salvo las propiedades explícitamente modificadas por descriptores incluidos en la instrucción Desplazar. El descriptor de auditoría con un descriptor de estadísticas, por ejemplo, retornaría estadísticas en la terminación justamente antes del desplazamiento. Los posibles descriptores que pueden obtenerse en retorno de la instrucción Desplazar son los mismos que para Añadir.

Por razones de conveniencia, si una terminación multiplexora es el objeto de una instrucción Desplazar, toda terminación de portador indicada en su descriptor de múltiplex se desplaza también como si se hubieran invocado instrucciones Desplazar que indicaran dichas terminaciones.

No se utilizará Desplazar en una terminación que se encuentre "fuera de servicio" ("OutOfService").

7.2.5 Auditoría de valores (AuditValue)

La instrucción AuditValue retorna los valores actuales de propiedades, eventos, señales y estadísticas asociados con terminaciones. AuditValue puede pedir el contenido de un descriptor o sólo de una propiedad, un evento, una señal o estadísticas. Puede ser útil para mantener la sincronización de terminaciones entre el MGC y la MG. Puede pedir que los valores retornados sean filtrados basándose en criterios específicos.

```

TerminationIDList
[,MediaDescriptor]
[,ModemDescriptor] (*)
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,DigitMapDescriptor]
[,ObservedEventsDescriptor]
[,EventBufferDescriptor]

```

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
AuditValue(TerminationIDList,  
           AuditDescriptor  
           )
```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 2 (05/2002).

La TerminationIDList especifica una secuencia de TerminationID, que puede ser específica o estar representada por comodines. Si el comodín concuerda con más de un TerminationID, se intentan todas las concordancias posibles, y se informan los resultados para cada uno de ellos. No se especifica el orden de los intentos cuando concuerdan varios TerminationID. Si se solicita una respuesta con comodín, solamente se genera un retorno de instrucción, y su contenido está formado por la unión de los valores de todas las terminaciones concordantes con el comodín. Este convenio puede reducir el volumen de los datos requeridos para efectuar una auditoría en un grupo de terminaciones. La utilización de CHOOSE es un error.

Se puede hacer una auditoría de descriptores o de una propiedad específica, una señal, un evento o estadísticas. Los valores retornados pueden ser filtrados basándose en criterios específicos.

- Se puede solicitar la auditoría de un descriptor especificándolo en el descriptor de auditoría sin ninguna otra información.
- Para hacer la auditoría de una determinada propiedad en el descriptor de medios, se incluyen los StreamID (facultativo), GroupID (facultativo) y PropertyID. El sistema devuelve el valor actual de la propiedad. Se utiliza Group ID en los casos en que se utiliza la bandera local de control Reserve Group. Group ID 1 corresponde al primer grupo (descripción de sesión) reservado, Group ID 2 corresponde al siguiente grupo, y así sucesivamente.
- Para hacer la auditoría de una señal, se proporcionan las identidades SignalIDList y/o SignalID. StreamID y/o RequestID son facultativos. El sistema devuelve los valores de todos los parámetros de la señal únicamente si la señal está activa: indicación KeepActive, tipo de señal, duración, indicación de finalización de la señal y propiedades definidas por el lote.
- Para hacer la auditoría de un evento se suministran las identidades StreamID (facultativo), eventID, requestID (facultativo). El sistema devuelve los valores de todos los parámetros del evento: acciones del evento y parámetros definidos en el lote.
- Para hacer la auditoría de una estadística en nivel de terminación se proporciona la identidad de la estadística. Cuando se trate de una estadística de tren, se incluyen en el descriptor de medios el StreamID (facultativo), el GroupID (facultativo) y el StatisticsID pertinentes. El sistema devuelve el valor actual de esta estadística. No se reinicia la estadística.
- Para hacer la auditoría de un lote, se proporciona la identidad y la versión del lote. El sistema devuelve todas las propiedades, las señales, los eventos y las estadísticas definidas en ese paquete con su valor actual.
- Para filtrar los valores que se obtienen en retorno de auditoría, en la petición de auditoría pueden incluirse criterios de selección, que pueden ser múltiples, en cuyo caso también se pueden incluir operadores lógicos AND y OR para indicar cómo interpretarlos. Si no se incluye ningún operador lógico, se supone que se trata de un AND.

Es posible hacer una auditoría de varios elementos diferentes con una sola petición.

En caso de petición de auditoría de un descriptor, la instrucción AuditValue hace que el sistema devuelva los descriptores apropiados, con los valores actuales para la terminación. Los valores que aparecen en múltiples ejemplares de un descriptor son, por definición, valores alternativos soportados, considerándose que cada parámetro en un descriptor es independiente.

El descriptor de eventos observados produce en retorno una lista de eventos en la memoria intermedia de eventos (EventBuffer). Si se hace una auditoría del descriptor de eventos observados mientras está activo un mapa de dígitos, el descriptor de eventos observados retornado incluye también un evento de finalización de mapa de dígitos que muestra la cadena de marcación actual pero no un método de terminación.

EventBuffer hace que el sistema devuelva el conjunto de eventos y valores de parámetros asociados permitidos actualmente en el descriptor EventBuffer. El descriptor de lotes devuelve una lista de lotes realizados por la terminación. El descriptor de mapa de dígitos devuelve el nombre o el valor del mapa de dígitos actual para la terminación. Si el valor de terminación ID del mapa de dígitos solicitado en una instrucción AuditValue es ALL, el sistema devuelve todos los mapas de dígitos en la pasarela. Estadísticas devuelve los valores actuales de todas las estadísticas que se llevan en la terminación. Una auditoría del descriptor de medios devuelve todas las estadísticas en el tren. Si se especifica un descriptor de auditoría vacío se obtiene como resultado solamente el TerminationID. Esto puede ser conveniente para obtener una lista de los TerminationID cuando se utilizan con comodín. En los anexos A y B figura una sintaxis especial para presentar esa lista en forma condensada, de tal modo que los rótulos de instrucción AuditValue no tienen que repetirse para cada TerminationID.

Los resultados de AuditValue dependen del contexto, (específico, NULL o con comodines). (Obsérvese que ContextID ALL no incluye el Contexto NULL.) El TerminationID puede ser específico, o incluir comodines.

A continuación se presentan ejemplos de lo que se obtiene en retorno cuando el contexto y/o la terminación incluyen comodines y se ha especificado una respuesta que incluye comodines.

Supóngase que la pasarela tiene cuatro terminaciones: t1/1, t1/2, t2/1 y t2/2. Supóngase también que las terminaciones t1/* han implementado lotes aaa y bbb y que las terminaciones t2/* han implementado lotes ccc y ddd. Supóngase que Context 1 tiene t1/1 y t2/1, y que Context 2 tiene t1/2 y t2/2.

La instrucción:

```
Context=1 {AuditValue=t1/1 {Audit {Packages}}}
```

Produce en retorno:

```
Context=1 {AuditValue=t1/1 {Packages {aaa,bbb}}}
```

La instrucción:

```
Context=* {AuditValue=t2/* {Audit {Packages}}}
```

Produce en retorno:

```
Context=1 {AuditValue=t2/1 {Packages {ccc,ddd}}},  
Context=2 {AuditValue=t2/2 {Packages {ccc,ddd}}}
```

La instrucción:

```
Context=* {W-AuditValue=t1/* {Audit {Packages}}}
```

Produce en retorno:

```
Context=* {AuditValue=t1/* {Packages {aaa,bbb}}}
```

NOTA – También puede haber una respuesta con comodín para otras instrucciones, como Substraer.

En el caso de un contexto con comodín raíz, el MGC puede indicar que desea una lista compacta de los ContextID, en lugar de tener cada uno de ellos desglosado en acciones independientes.

Supóngase que hay cuatro contextos en la pasarela, a saber 1, 2, 3 y 4, cada uno de los cuales tiene dos terminaciones (t1-t8). En el siguiente ejemplo se indica cómo respondería la MG a la instrucción de auditoría:

La instrucción:

```
Context=* {AuditValue=Root {Audit {}}}
```

Produce en retorno:

```
Context=1 {AuditValue=Context{*}}, Context=2 {AuditValue=Context{*}},  
Context=3 {AuditValue=Context{*}}, Context=4 {AuditValue=Context{*}}
```

O

```
Context=1 {AuditValue=t1 {}, AuditValue=t2 {}}, Context=2 {AuditValue=t3 {}, AuditValue=  
t4 {}}, Context=3 {AuditValue=t5 {}, AuditValue=t6 {}}, Context=4 {AuditValue=t7 {}, Audit  
Value=t8 {}}
```

La instrucción:

```
Context=* {ContextAttr {ContextList={*}}, AuditValue=Root {Audit {}}}
```

Produce en retorno:

```
Context=* {ContextAttr {ContextList={1,2,3,4}}, AuditValue=Root {}}
```

El siguiente cuadro muestra otras informaciones que pueden obtenerse con la instrucción Auditoría da valor:

ContextID	TerminationID	Información obtenida
Específico	Comodín	Auditoría de terminaciones concordantes en un contexto
Específico	Específico	Auditoría de una terminación individual en un contexto
NULL	Raíz	Auditoría del estado y eventos de una pasarela de medios
NULL	Comodín	Auditoría de todas las terminaciones concordantes en el contexto NULL
NULL	Específico	Auditoría de una terminación individual fuera de todo contexto
TODOS (ALL)	Comodín	Auditoría de todas las terminaciones concordantes que no están en NULL y del contexto al que están asociadas
TODOS (ALL)	Raíz	Lista de todos los ContextID (la lista de ContextID puede devolverse utilizando alguna de las múltiples respuestas de acción, cada una e las cuales contendrá un ContextID tomado de la lista, o utilizando un parámetro ContextIDList. El método de respuesta va determinado por la existencia o no del parámetro ContextIDList en la petición)
TODOS (ALL)	Específico	ContextID (no NULL) en el que la terminación existe en ese momento

7.2.6 AuditCapabilities (Auditoría de capacidades)

La instrucción AuditCapabilities retorna los posibles valores de propiedades, eventos, señales y estadísticas asociados con terminaciones. Puede pedirse AuditCapabilities para el contenido de un descriptor o para una determinada propiedad, evento, señal o estadísticas.

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor](*)
[,MuxDescriptor]
[,EventsDescriptor]
[,SignalsDescriptor]
[,ObservedEventsDescriptor]
[,EventBufferDescriptor]
[,StatisticsDescriptor]
 AuditCapabilities(TerminationIDList,
 AuditDescriptor)

(*) El descriptor de módem no se consideró en H.248.1, Versión 2 (05/2002).

Puede hacerse una auditoría de un descriptor o sólo de determinadas propiedades, señales, eventos y estadísticas.

- Puede hacerse una petición de auditoría de todo un descriptor identificándolo en el descriptor de auditoría sin ninguna otra información.
- Para hacer la auditoría de una determinada propiedad en el descriptor de medios, se indican las identidades StreamID (facultativo) y PropertyID. El sistema devuelve una lista de valores posibles de esta propiedad.
- Para hacer la auditoría de una señal, se suministran las identidades de lista de señales (SignalsIDList) y/o de señales (SignalID). StreamID y/o RequestID son facultativos. El sistema devuelve una lista de valores posibles asociados con cada parámetro de señal (incluyendo las propiedades definidas en el lote). No se devuelve la indicación KeepActive, el tipo de señal, la duración, ni la indicación de finalización de señal.
- Para hacer la auditoría de un evento, se suministran los indicadores pertinentes StreamID (facultativo), EventID, RequestID (facultativo). El sistema devuelve una lista de valores posibles asociados a cada parámetro de evento (acciones de evento y parámetros definidos en el lote).
- Para hacer la auditoría de una estadística en una terminación se suministra la identidad de esa estadística. Si se trata de un tren, se incluyen en el descriptor de medios los identificadores del caso, StreamID (facultativo), GroupID (facultativo) y StatisticsID. El sistema devuelve a los valores posibles de la estadística. La estadística no reinicia.

Cuando se pide la auditoría de un descriptor, la instrucción AuditCapabilities devuelve los descriptores apropiados, con los posibles valores para la terminación. Los descriptores pueden repetirse cuando hay múltiples valores posibles.

Si se solicita una respuesta con comodín, solamente se genera un retorno de instrucción cuyo contenido está formado por la unión de los valores de todas las terminaciones en la lista concordantes con el comodín. Este convenio puede reducir el volumen de los datos requeridos para efectuar una auditoría de un grupo de terminaciones.

Cuando se hace la auditoría de una propiedad, una señal, un evento o estadísticas, AuditCapabilities hace que el sistema retorne las propiedades, las señales, los eventos y las estadísticas apropiados con las capacidades de la terminación.

La interpretación de las capacidades que son solicitadas para los diversos valores de ContextID y TerminationID es la misma que en AuditValue.

En el caso de los valores de propiedades y parámetros de tipo cadena de caracteres o cadena de octetos, la MG devolverá un valor vacío. Para la codificación de texto, las cadenas de caracteres y los caracteres devuelven una forma lingüística quotedString vacía, mientras que las cadenas de

octetos devuelven una respuesta NULL (0x00). Este comportamiento puede ser anulado por la definición del lote.

El descriptor de eventos devuelve la lista de posibles eventos en la terminación junto con la lista de todos los valores posibles de los parámetros de descriptor de eventos. El descriptor de memoria intermedia de eventos (EventBufferDescriptor) devuelve la misma información que el descriptor de eventos. El descriptor de señales devuelve la lista de las posibles señales que podrían aplicarse a la terminación junto con la lista de todos los posibles valores de los parámetros de señales. El descriptor de estadísticas devuelve los valores de las estadísticas que se llevan en la terminación. El descriptor de eventos observados devuelve los nombres de los eventos activos en la terminación. mapa de dígitos y paquetes no legítimos en AuditCapability.

A continuación se presenta otra información que puede obtenerse con la instrucción AuditCapabilities:

ContextID	TerminationID	Información obtenida
Específico	Comodín	Auditoría de terminaciones concordantes en un contexto
Específico	Específico	Auditoría de una sola terminación en un contexto
NULL	Raíz	Auditoría de estados y eventos de MG
NULL	Comodín	Auditoría de todas las terminaciones concordantes en el contexto NULL
NULL	Específico	Auditoría de una sola terminación fuera de todo contexto
ALL	Comodín	Auditoría de todas las terminaciones que no están en NULL concordantes y del contexto a que están asociadas
ALL	Raíz	Lo mismo que para AuditValue
ALL	Específico	Lo mismo que para AuditValue

7.2.7 Notificar

La instrucción Notificar permite a la pasarela de medios notificar al controlador de pasarela de medios los eventos que se producen dentro de ella.

TerminationID

```
Notify(TerminationID,
       ObservedEventsDescriptor,
       [ErrorDescriptor])
```

El parámetro TerminationID especifica la terminación que emite la instrucción Notificar. TerminationID será un nombre completamente calificado.

El descriptor de eventos observados contiene la identidad RequestID y una lista de eventos detectados por la pasarela de medios, en el orden en que fueron detectados. Cada evento en la lista va acompañado por parámetros asociados con el evento y facultativamente una indicación del instante en que se detectó el evento. Los procedimientos para el envío de instrucciones Notificar con RequestID igual a cero quedan en estudio.

Las instrucciones Notificar con el RequestID distinto de cero aparecerán solamente como resultado de la detección de un evento especificado por un descriptor de eventos activo en la terminación considerada.

RequestID produce en retorno el parámetro RequestID del descriptor de eventos que provocó la instrucción Notificar. Se utiliza para correlacionar la notificación con la petición que la provocó. La lista contendrá los eventos solicitados a través de descriptor de eventos que provocó la instrucción o insertados en el descriptor de eventos, a menos que el RequestID sea cero lo que queda en estudio.

El descriptor de errores se puede enviar en la instrucción Notificar como resultado del error 518 ("Memoria tampón de eventos llena").

7.2.8 Cambio de servicio (ServiceChange)

La instrucción ServiceChange permite a la pasarela de medios notificar al controlador de pasarela de medios que una terminación o un grupo de terminaciones están a punto de ser puestos fuera de servicio o acaban justamente de ser puestos en servicio. El controlador de pasarela de medios puede indicar que se deben poner fuera de servicio o se deben restituir al servicio una o más terminaciones. La pasarela de medios puede notificar al MGC que la capacidad de una terminación ha cambiado. Permite también a un MGC traspasar el control de una MG a otro MGC.

TerminationIDList,

[ServiceChangeDescriptor]

```
    ServiceChange(TerminationIDList,  
                  ServiceChangeDescriptor  
                )
```

La TerminationIDList especifica la terminación o terminaciones que habrán de ponerse fuera de servicio o restituirse al servicio. Se permite utilizar comodines para los nombres de las terminaciones, pero no podrá utilizarse el mecanismo CHOOSE. La utilización del TerminationID "Root" (Raíz) indica un ServiceChange que afecta a la totalidad de la pasarela de medios.

NOTA – No es válido utilizar TerminationIDList en la instrucción ServiceChange de la Versión 1 de H.248.1.

7.2.8.1 Contenido del descriptor de cambio de servicio

El descriptor de cambio de servicio contiene los siguientes parámetros según el caso:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeDelay;
- ServiceChangeAddress;
- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMgcID;
- TimeStamp;
- ExtensionParameter;
- ServiceChangeInfo;
- ServiceChangeIncompleteFlag.

7.2.8.1.1 Método de cambio de servicio (ServiceChangeMethod)

El parámetro ServiceChangeMethod especifica el tipo de ServiceChange que se producirá o que se ha producido:

- 1) Paulatino: indica que las terminaciones especificadas serán puestas fuera de servicio después de transcurrido el retardo de cambio de servicio (ServiceChangeDelay) especificado; las conexiones establecidas todavía no son afectadas, pero el controlador de pasarela de medios debe abstenerse de establecer nuevas conexiones y debe tratar de suprimir paulatinamente las conexiones existentes en la terminación o terminaciones afectadas por la instrucción serviceChange. La MG debe establecer el serviceState de la terminación al expirar el ServiceChangeDelay o al retirar la terminación de un contexto activo para ponerla fuera de servicio (lo que ocurra primero).

- 2) Forzado: indica que las terminaciones especificadas fueron puestas fuera de servicio bruscamente y que las eventuales conexiones establecidas que estaban asociadas con ellas pueden haberse perdido. Para terminaciones que no son raíz, el controlador de pasarela de medios es el responsable de liberar el contexto (si existe) al que está asociada la terminación fallida. Como mínimo la terminación deberá retirarse del contexto. La terminación ha de quedar en el estado (serviceState) "fuera de servicio". Para la terminación Raíz, el MGC puede suponer que todas las conexiones se perdieron en la MG y por tanto considerar que se han retirado todas las terminaciones.
- 3) Rearranque: indica que el servicio se restablecerá en las terminaciones especificadas después de transcurrido el ServiceChangeDelay. ServiceState debe pasar a "inService" después de transcurrido el ServiceChangeDelay.
- 4) Desconectado: se aplica siempre con el TerminationID raíz e indica que la MG había perdido la comunicación con el MGC, pero que la comunicación fue restablecida posteriormente al mismo MGC (posiblemente después de intentar con otros MGC en una lista suministrada previamente). Como el estado de la MG puede haber cambiado, el MGC puede tener interés en utilizar la instrucción Auditoría para resincronizar su estado con el de la MG.
- 5) Traspaso: la envía el MGC a la MG para indicar que se retira del servicio y que se debe establecer una asociación con un nuevo MGC. Se envía de la MG al MGC para indicar que la MG intenta establecer una nueva asociación de conformidad con un traspaso recibido del MGC con el cual estuvo anteriormente asociada.
- 6) Cambio-en-caso-de-fallo: se envía de la MG al MGC para indicar que la MG primaria está fuera de servicio y que la MG secundaria la sustituye. Este método serviceChange también se envía de la MG al MGC cuando la MG detecta que el MGC ha fallado
- 7) Otro valor cuyo significado reconocido sea mutuamente por la MG y el MGC.

7.2.8.1.2 Motivo de cambio de servicio y ServiceChangeMgcID (ServiceChangeReason)

El parámetro ServiceChangeReason especifica el motivo por el cual se ha producido o se producirá el ServiceChange. Está constituido por un testigo alfanumérico [registrado por la autoridad de asignación de números Internet (IANA, *Internet assigned numbers authority*)] y, opcionalmente, una cadena explicativa.

7.2.8.1.3 Dirección de cambio de servicio y ServiceChangeMgcID (ServiceChangeAddress)

El parámetro facultativo ServiceChangeAddress especifica la dirección (por ejemplo, número del puerto IP para las redes IP) que habrá de utilizarse para comunicaciones ulteriores. Puede especificarse en el descriptor de parámetros de entrada o el descriptor de resultado retornado. Los parámetros ServiceChangeAddress y ServiceChangeMgcID no deben estar presentes ambos en el descriptor de cambio de servicio o el descriptor de resultado de cambio de servicio (ServiceChangeResultDescriptor). El parámetro ServiceChangeAddress proporciona una dirección que se utilizará dentro del contexto de la asociación que se está negociando actualmente, mientras que el ServiceChangeMgcID proporciona una dirección alternativa donde la MG debe tratar de establecer otra asociación. Se señala que la utilización de ServiceChangeAddress no está recomendada. Los MGC y las MG podrán hacer frente a la situación en que la ServiceChangeAddress es una dirección completa o sólo un número de puerto en el caso de transportes TCP.

7.2.8.1.4 Retardo de cambio de servicio (ServiceChangeDelay)

El parámetro facultativo ServiceChangeDelay se expresa en segundos. Cuando no se incluye el retardo o se indica retardo cero, debe considerarse que el valor del retardo es nulo. En el caso de un ServiceChangeMethod "paulatino" ("Graceful"), un retardo nulo indica que el controlador de la pasarelas de medios debe esperar hasta que las conexiones existentes sean suprimidas de manera

ordinaria y no debe establecer nuevas conexiones. Para el método "paulatino" solamente, un retardo nulo significa que la MG no debe fijar el serviceState "fuera de servicio" ("OutOfService") hasta que la terminación esté en el contexto NULL.

7.2.8.1.5 Perfil de cambio de servicio (ServiceChangeProfile)

El parámetro facultativo ServiceChangeProfile especifica el perfil (si existe) del protocolo soportado. ServiceChangeProfile incluye la versión del perfil soportado. Cuando no esté presente este parámetro, se supone que el valor es "NoProfile".

7.2.8.1.6 Versión de cambio de servicio (ServiceChangeVersion)

El parámetro facultativo ServiceChangeVersion contiene la versión de protocolo y se utiliza si se produce una negociación de la versión de protocolo (véase 11.3).

7.2.8.1.7 Indicación de tiempo (TimeStamp)

El parámetro facultativo TimeStamp especifica la hora registrada por el emisor. No es necesariamente una hora absoluta, por ejemplo en el huso horario local, sino la hora inicial arbitraria con respecto la cual se compararán todas las futuras indicaciones de hora transmitidas por un emisor durante su asociación. Puede ser utilizado por el respondedor para determinar la diferencia que hay entre su noción de hora y la noción de su interlocutor. TimeStamp es enviado con una precisión de centésimas de segundo.

7.2.8.1.8 ExtensionParameter

El parámetro facultativo Extension puede contener cualquier valor cuyo significado sea mutuamente acordado entre la MG y el MGC. El valor "X-SC" se reserva para la utilización de los parámetros ServiceChange de señalización añadidos en la Versión 3 (y versiones ulteriores) de H.248.1. Se utiliza solamente en la instrucción codificada ServiceChange de la Versión 1 de H.248.1, de la MG hacia el MGC, cuando ServiceChangeVersion sea mayor o igual que 3. La estructura del valor se define mediante la siguiente ABNF:

$$X-SC = 1 * (NAME EQUAL paramValue [COMMA])$$

Al añadir nuevos parámetros ServiceChange, conviene atribuir un nombre/valor si el parámetro ha de ser enviado en un ServiceChange inicial. Por ejemplo:

Nombre (Name): Cadena compuesta por, a lo sumo, 64 caracteres alfanuméricos y/o números, sin espacios, que empieza con una letra y que puede incluir el carácter especial de subrayado ("_", *underscore*).

Tipo (Type): Conforme a 12.1.2 "Propiedades"

Valores posibles: Conforme a 12.1.2 "Propiedades"

7.2.8.1.9 Información de cambio de servicio (ServiceChangeInfo)

El parámetro facultativo ServiceChangeInfo puede contener el paquete/propiedad/señal/evento/estadística del motivo para el cambio de servicio.

7.2.8.1.10 ServiceChangeIncompleteFlag

Bandera facultativa que indica que se van a enviar otras instrucciones ServiceChange de la MG al MGC, que señalan el estado de las terminaciones. Se emplea solamente durante el registro o reinicio de la MG (ServiceChange en Raíz con reinicio por ServiceChangeMethod), cuando la MG desee informar al MGC el estado de todas las terminaciones. El MGC que ha recibido esta bandera no generará hacia la MG instrucciones que afecten a las terminaciones, excepto en el caso de una Raíz. Después de haber enviado la ServiceChangeIncompleteFlag en la instrucción inicial de registro/reinicio, la MG tendrá que seguir enviándola en todas las ulteriores instrucciones ServiceChange, hasta que se establezca que ha informado de su estado actual y del estado de sus

terminaciones. Entonces se suprimirá la bandera de la última instrucción ServiceChange que indica dichos estados. Cuando reciba esta ServiceChange sin la ServiceChangeIncompleteFlag, el MGC puede dar de nuevo instrucciones a la MG. Si se envía esta bandera en una instrucción inicial ServiceChange (Versión 1 de H.248.1) utilizando el parámetro de extensión "X-SC" se emplea:

Nombre: SIC

Tipo: Booleano

Valores posibles: ON: Bandera incluida

NOTA – No es necesario el valor OFF, pues las instrucciones subsiguientes serán codificadas conforme a la Versión 3 de H.248.1

7.2.8.1.11 Instrucción y respuesta ServiceChange

Una instrucción ServiceChange con TerminationID "Raíz" y ServiceChangeMethod igual a Restart es una instrucción de registro por la cual una pasarela de medios anuncia su existencia al controlador de pasarela de medios. La pasarela de medios también se puede registrar con "Raíz" (Root) para el TerminationID y ServiceChangeMethod igual a cambio-en-caso-de-fallo (Failover) cuando la MG detecta fallos del MGC. Un mensaje con una instrucción ServiceChange que especifique el TerminationID Raíz y el ServiceChangeMethod de Reinicio (Restart) o Failover no contendrá otras instrucciones, pues éstas tendrían que emplear el nuevo ServiceChangeAddress y la versión negociada del protocolo.

Se espera que estén configurados en la pasarela de medios el nombre de un controlador de pasarela de medios primario y facultativamente de cierto número de controladores de pasarelas de medios alternativos.

El acuse de recibo de la instrucción ServiceChange concluye el proceso de registro, salvo cuando el MGC retorna un ServiceChangeMgcID alternativo descrito en el párrafo siguiente.

La MG puede especificar la ServiceChangeAddress de transporte que habrá de ser utilizada por el MGC para enviar mensajes en el parámetro ServiceChangeAddress del descriptor ServiceChange de entrada. La MG puede especificar una dirección en el parámetro ServiceChangeAddress de la petición ServiceChange, y el MGC puede también hacer lo mismo en la contestación ServiceChange. En cualquier caso, el receptor debe utilizar la dirección suministrada como destino de todas las peticiones de transacción subsiguientes dentro de la asociación. Al mismo tiempo, tal como se indica en la cláusula 9, hay que enviar las respuestas de transacción y las indicaciones pendientes a la dirección desde la cual se han originado las peticiones correspondientes. Se multiplican los mensajes, pero es necesario porque no se pueden poner en un mismo lote las instrucciones y las respuestas.

El parámetro TimeStamp será enviado con una instrucción de registro y su respuesta.

Al registrarse, la MG puede enviar la bandera ServiceChangeIncompleteFlag si ha establecido que la MG está en estado "InService" pero algunas terminaciones pueden encontrarse en estado "OutOfService". Esto con el fin de impedir la toma de terminaciones que no estén en servicio. Las siguientes instrucciones ServiceChange se utilizan para informar las terminaciones "OutOfService".

El controlador de pasarela de medios puede retornar un parámetro ServiceChangeMgcID para indicar el controlador que la pasarela de medios deberá contactar preferentemente para un futuro servicio. En este caso, la pasarela de medios volverá a enviar la instrucción ServiceChange al nuevo controlador de pasarela de medios. El MGC especificado en un ServiceChangeMgcID, si se proporciona, será contactado antes que otros MGC. En un mensaje de traspaso del MGC a la MG, el ServiceChangeMgcID es el nuevo MGC que ocupará la posición del actual MGC.

ServiceChange produce una respuesta vacía, salvo cuando se utiliza el terminationID "Root" (Raíz), que provoca un retorno con los siguientes parámetros:

- ServiceChangeAddress, si el MGC respondedor desea especificar un nuevo destino para los mensajes procedentes de la MG para el resto de la asociación.
- ServiceChangeMgcID, si el MGC respondedor no desea soportar una asociación con la MG.
- ServiceChangeProfile, si el respondedor desea negociar el perfil que se utilizará para la asociación. El sistema retornará el perfil (nombre y versión) en la respuesta únicamente si el MGC no puede soportar los perfiles especificados en ServiceChangeRequest. En la respuesta de retorno se indicará el perfil y la versión soportados o aparecerá la indicación "NoProfile" cuando no se soporta ningún perfil. Cuando la MG recibe una respuesta con indicación de perfil, puede continuar la relación con el MGC actual o contactar MGC secundarios y establecer una relación con ellos. Si el MGC retornó un perfil en una respuesta diferente del que fue suministrado en la petición, la MG:
 - a) continuará con la asociación de control, emitiendo una nueva instrucción ServiceChange con un perfil acordado para confirmar al MGC que está de acuerdo con el perfil escogido, o
 - b) mantendrá activa la asociación de control, para que el MGC pueda utilizar el perfil que envió en la respuesta ServiceChange, o
 - c) iniciará una asociación de control con un MGC diferente, utilizando su perfil original.
- ServiceChangeVersion, si el respondedor desea negociar la versión del protocolo que se utilizará para la asociación.

Las ServiceChangeReasons (motivos de cambio de servicio) se definen en la Rec. UIT-T H.248.8. Esta lista puede ampliarse por un registro en la IANA como se describe en 14.3.

7.2.9 Manipulación y auditoría de los atributos de un contexto

Las instrucciones del protocolo examinadas en las cláusulas precedentes se aplican a terminaciones. En esta cláusula se especifican los procesos de manipulación y auditoría de contextos.

Una acción puede contener instrucciones de manipulación y auditoría de contextos (véase la cláusula 8).

Una petición de acción que se envía a una MG puede incluir una petición de auditoría de los atributos de un contexto. Hay dos tipos de auditoría, a saber:

Auditoría de valor: El MGC puede hacer la auditoría de un determinado contexto para conocer el valor actual de las distintas propiedades de contexto. El MGC puede determinar los valores actuales de todos los contextos existentes (no NULL) especificando ContextID ALL en la petición Audit. Si en la misma acción de la petición Audit se añaden o modifican atributos, el sistema retorna los valores que resultan de ejecutar la acción. Los atributos de contexto se pueden utilizar como criterio de selección para filtrar los valores que retorna la auditoría. Es posible incluir criterios múltiples de selección, en cuyo caso también se pueden incluir operadores lógicos AND y OR para indicar cómo interpretarlos. Si no se incluye ningún operador lógico, se supone que se trata de un AND.

Auditoría de capacidades: El MGC puede efectuar la auditoría de un contexto específico para establecer los valores posibles que pueden tomar las propiedades de contexto. El MGC puede establecer estos valores para toda la MG especificando el ContextID ALL en la petición de auditoría.

En el siguiente cuadro se indica la información que se puede obtener con una auditoría de contexto:

ContextID	TerminationID	AuditValue	AuditCapabilities
Específico	No es aplicable	Valor del atributo en el contexto especificado.	Valores posibles de la propiedad de contexto en el contexto especificado.
NULL	No es aplicable	No autorizado	No autorizado
ALL	No es aplicable	Los valores actuales de todos los contextos existentes (no NULO) si el ContextID es ALL en la petición Audit. Para responder a una petición con el ContextID ALL, se retorna un actionReply por contexto.	Se retorna una respuesta de ContextID ALL con valores posibles de propiedad de contexto en toda la MG.

Una acción puede incluir también una petición de cambio de los atributos de un contexto.

Las propiedades del contexto que pueden incluirse en una contestación de acción se utilizan para retornar información a un MGC. Puede ser información solicitada por una auditoría de los atributos de contexto o una información detallada sobre el efecto de manipulación de un contexto.

En el caso de que una MG reciba una acción que contiene tanto una petición de auditoría de atributos de contexto como una petición de manipular estos atributos, emitirá una respuesta que contendrá los valores de los atributos después del procesamiento de la petición de manipulación.

7.2.10 Sintaxis de instrucción genérica

El protocolo puede codificarse en formato binario o en formato textual. El MGC debe soportar ambos formatos de codificación. Las MG pueden soportar uno o ambos formatos.

La sintaxis de protocolo para el formato binario se define en el anexo A. En el anexo C se especifica la codificación de los descriptores local y distante para utilización con el formato binario.

En el anexo B se presenta una Forma Backus Naur Aumentada (ABNF, *augmented Backus-Naur form*) completa de la codificación textual del protocolo según RFC 2234. El protocolo de descripción de sesión (SDP) se aplica como codificación de los descriptores local y distante en la codificación textual modificada de 7.1.8.

8 Transacciones

Las instrucciones entre el controlador de pasarela de medios y la pasarela de medios se agrupan en transacciones, cada una de las cuales se identifica por un TransactionID. Las transacciones consisten en una o más acciones. Una acción consiste en una serie no vacía de instrucciones, modificaciones de propiedades de contexto, o auditorías de propiedades de contexto que están limitadas a funcionar dentro de un solo contexto. Por eso se suele especificar un ContextID en cada acción. Sin embargo, hay dos casos en los que no se proporciona un ContextID específico con una acción. Uno es la modificación de una terminación fuera de un contexto, y el otro cuando el controlador pide a la pasarela que cree un nuevo contexto. La figura 11 es una representación gráfica de las relaciones entre transacción, acción e instrucción.

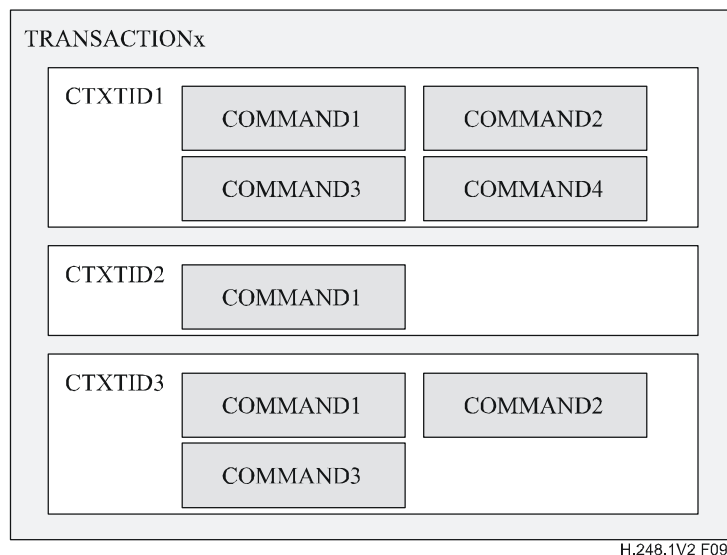


Figura 11/H.248.1 – Transacciones, acciones e instrucciones

Las transacciones son objeto de una petición (TransactionRequests). Las respuestas correspondientes a una TransactionRequest se reciben en una sola contestación, posiblemente precedida por varios mensajes TransactionPending (véase 8.2.3).

Las transacciones garantizan el procesamiento ordenado de las instrucciones, se ejecutan en la misma secuencia. La ordenación de las transacciones no está garantizada; las transacciones pueden ejecutarse en cualquier orden, o simultáneamente; no obstante, las respuestas (TransactionReplies) deben ejecutarse antes de las TransactionRequests, cuando ambas aparezcan en un mensaje.

Al primer fallo de una instrucción en una transacción, se detiene el procesamiento de las instrucciones restantes en esa transacción. Si una instrucción contiene un TerminationID con comodines, se trata de ejecutar la instrucción con cada uno de los TerminationID que concuerdan con el comodín. En TransactionReply se incluye una respuesta para cada TerminationID concordante, incluso si una o más ejemplares generaron un error. Si cualquier TerminationID concordante con un comodín produce un error cuando se ejecuta, no se intentará ninguna de las instrucciones que siguen a la instrucción con el comodín.

Las instrucciones pueden marcarse como "facultativas", lo que puede dejar sin efecto este comportamiento – si una instrucción marcada como facultativa da por resultado un error, las instrucciones subsiguientes en la transacción se ejecutan. Si falla una instrucción, la MG restablecerá tanto como sea posible el estado previo al intento de ejecución de la instrucción antes de continuar el procesamiento de las instrucciones.

Una TransactionReply incluye los resultados para todas las instrucciones en la correspondiente TransactionRequest. TransactionReply incluye los valores de retorno para las instrucciones que se ejecutaron con éxito, y la instrucción y el descriptor de error para cualquier instrucción que haya fracasado. Las respuestas a instrucciones se devuelven en el orden en que aparecen en la correspondiente TransactionRequest. Se utiliza TransactionPending para notificar periódicamente al receptor que una transacción continúa procesándose pero que todavía no ha finalizado.

Las aplicaciones deberían implementar un temporizador en el nivel de aplicación para cada transacción. La expiración del temporizador debería causar una retransmisión de la petición. La recepción de una contestación debe anular el temporizador. La recepción de TransactionPending debe rearrancar el temporizador.

8.1 Parámetros comunes

8.1.1 Identificadores de transacciones

Las transacciones se identifican con un TransactionID que es asignado por el emisor y es único dentro del alcance del emisor. En la respuesta con un descriptor de error para indicar que falta el TransactionID en una petición, TransactionID debe ser cero en la TransactionReply correspondiente.

8.1.2 Identificadores de contexto

Los contextos se identifican mediante un ContextID, que es asignado por la pasarela de medios y es único dentro del alcance de la pasarela de medios. El controlador de pasarela de medios utilizará el ContextID suministrado por la pasarela de medios en todas las transacciones subsiguientes relacionadas con ese contexto. El protocolo hace referencia a un valor característico que puede ser utilizado por el controlador de pasarela de medios cuando se refiere a una terminación que en ese momento no está asociada con un contexto, a saber, el ContextID *NULL*.

El comodín CHOOSE se utiliza para pedir a la pasarela de medios que cree un nuevo contexto.

El MGC puede utilizar el comodín ALL para tratar todos los contextos en la MG. No se incluye el contexto NULL cuando se utiliza el comodín ALL.

El MGC no utilizará los ContextID parcialmente especificados que contengan los comodines CHOOSE o ALL.

8.2 Interfaz de programación de aplicación para las transacciones

A continuación se presenta una interfaz de programación de aplicación (API) que describe las transacciones del protocolo. Esta API se presenta para ilustrar las transacciones y sus parámetros, sin que se pretenda especificar una implementación (por ejemplo, mediante la utilización de llamadas a funciones de bloqueo). Describirá los parámetros de entrada y los valores de retorno que, según se espera, serán utilizados por las diversas transacciones del protocolo desde un nivel muy alto. La sintaxis y las codificaciones de las transacciones se especifican en ulteriores cláusulas.

8.2.1 TransactionRequest

La petición TransactionRequest la invoca el emisor y se realiza una transacción para cada invocación de petición. Una petición contiene una o más acciones, cada una de las cuales especifica el contexto deseado.

```
TransactionRequest(TransactionID {  
    ContextID {Command ... Command},  
    ...  
    ContextID {Command ... Command})
```

El parámetro TransactionID especificará un valor para ulterior correlación con la respuesta TransactionReply o TransactionPending del receptor.

El parámetro ContextID especificará un valor para establecer la relación con todas las instrucciones que seguirán, sea hasta que se especifique un nuevo parámetro ContextID o hasta que finalice la TransactionRequest, de los dos eventos el que suceda primero.

El parámetro Command representa una de las instrucciones mencionadas en 7.2 ("Interfaz de programación de aplicación para las instrucciones").

8.2.2 TransactionReply

La respuesta TransactionReply la invoca el receptor y se realiza una transacción por cada invocación de contestación. Una contestación contiene una o más acciones, cada una de las cuales especifica el contexto deseado y una o más contestaciones por contexto. Una contestación puede fragmentarse en varios mensajes. TransactionReply la invoca el respondedor cuando ha procesado TransactionRequest.

El procesamiento de TransactionRequest termina:

- cuando se han procesado todas las acciones de la petición; o
- cuando el procesamiento de la petición tropieza con un error, a menos que se trate de un error en una instrucción facultativa.

Si un mensaje contiene más TransactionRequest que las que puede procesar el receptor, éste devolverá el descriptor de error 413 ("El número de transacciones en el mensaje rebasa el máximo permitido").

Una instrucción ha sido procesada cuando todos sus descriptores han sido procesados.

Se considera que un descriptor de señales ha sido procesado cuando se ha determinado que es sintácticamente válido, las señales solicitadas están soportadas y se han introducido en cola para ser aplicadas.

Se considera que un descriptor de eventos o uno de memoria intermedia de eventos han sido procesados cuando se ha determinado que es sintácticamente válido, los eventos solicitados pueden ser observados, toda señal insertada puede ser generada, todo evento insertado puede ser detectado, y la MG ha sido pasada a un estado en el que los eventos serán detectados.

```
TransactionReply(TransactionID {  
    ContextID {Response ... Response},  
    ...  
    ContextID {Response ... Response}})
```

El parámetro TransactionID debe ser el mismo que el de la correspondiente TransactionRequest.

El parámetro ContextID especificará un valor para establecer la relación con todas las respuestas para la acción. El ContextID puede ser específico, ALL o NULL.

Cada uno de los parámetros Response representa un valor de retorno como se ha indicado en 7.2 o un descriptor de error si la ejecución de la instrucción encontró un error. Las instrucciones no son procesadas más allá del punto en que aparece un error y, por lo tanto, no se emiten respuestas para ellas.

Una excepción es el caso en que una instrucción se ha marcado como facultativa en la TransactionRequest. Si la instrucción facultativa genera un error, la transacción continúa no obstante ejecutándose, por lo que la contestación, en este caso, tendría respuestas después de un error.

En la cláusula 7.1.20 ("Descriptor de error") se especifica la generación de descriptores de error. En el texto a continuación se examinan varios casos concretos.

Si el receptor encuentra un error al procesar un ContextID, la respuesta a la acción solicitada consistirá en el identificador de contexto y un solo descriptor de error, 422 ("Error de sintaxis en acción"). Si el receptor no puede analizar sintácticamente el ContextID, devolverá una TransactionReply que conste del TransactionID y de un solo descriptor de error, 422 ("Error de sintaxis en acción").

Si el receptor encuentra un error que impide determinar una acción legítima, retornará una TransactionReply que consistirá en el TransactionID y un solo descriptor de error, 422 ("Error de sintaxis en acción"). Si el final de una acción no puede ser determinado de forma fiable pero una o más instrucciones pueden ser analizadas gramaticalmente, el receptor las procesará y enviará entonces 422 ("Error de sintaxis en acción") como la última acción para la transacción. Si el receptor encuentra un error que impide determinar una acción legítima, retornará una TransactionReply con un TransactionID NULL y un solo descriptor de error 403 ("Error de sintaxis en TransactionRequest").

Si el final de una transacción no puede determinarse de manera fiable y es posible analizar gramaticalmente una o más acciones el receptor las procesará y retornará entonces 403 ("Error de sintaxis en TransactionRequest") como la última contestación para la transacción. Si no hay acciones que puedan ser analizadas gramaticalmente, el receptor las retornará entonces 403 ("Error de sintaxis en TransactionRequest") como la única contestación para la transacción.

Si el terminationID no puede determinarse de manera fiable, enviará 442 ("Error de sintaxis en instrucción") como la contestación de acción.

Si el final de una instrucción no puede determinarse de manera fiable, retornará 442 ("Error de sintaxis en instrucción"), como contestación a la última acción que puede analizar gramaticalmente.

8.2.3 TransactionPending

La instrucción TransactionPending la invoca el receptor para indicar que la transacción continúa procesándose, pero que todavía no ha finalizado. Se utiliza para impedir que el emisor suponga que se ha perdido TransactionRequest cuando se requiera cierto tiempo para la finalización de la transacción.

TransactionPending(TransactionID { })

El parámetro TransactionID debe ser el mismo que el de la correspondiente TransactionRequest. El MGC puede especificar una propiedad de la terminación Raíz (normalMGExecutionTime) para indicar un plazo para la respuesta de la MG a cualquier transacción (sin incluir el retardo de red). El MGC también puede especificar otra propiedad (normalMGCEExecutionTime) para indicar un plazo para la respuesta del MGC a la MG a cualquier transacción (sin incluir el retardo de red). MGProvisionalResponseTimerValue indica el plazo previsto para una respuesta pendiente de la MG al MGC si no se puede completar una transacción (fijado inicialmente como normalMGExecutionTime más el retardo de red, pero que puede ser disminuido). MGCProvisionalResponseTimerValue tiene el mismo significado, pero para la MG. Los emisores pueden recibir más de una TransactionPending para una instrucción. Si se recibe una petición duplicada cuando se encuentra en curso la transacción, el respondedor puede enviar un duplicado de pendiente inmediatamente, o continuar esperando a su temporizador para activar otra transacción pendiente.

El MGC puede determinar una propiedad de la terminación Raíz (MGOrioginatedPendingLimit) para indicar el número de TransactionPendings que se pueden recibir de la MG. Si se llega a sobrepasar el valor expresado por esta propiedad, la MG interrumpirá el tratamiento de transacciones y enviará un TransactionReply, para evitar que el MGC considere que hay un error en la transacción.

El MGC puede determinar otra propiedad de la terminación Raíz (MGCOrioginatedPendingLimit) para indicar el número de TransactionPendings que se pueden recibir del MGC. Si se sobrepasa el valor expresado por esta propiedad, el MGC interrumpirá el tratamiento de transacciones y enviará un TransactionReply, para impedir que la MG considere que hay un error en la transacción.

Se podría sobrepasar el valor de la propiedad xxxOrioginatedPendingLimit (MGOrioginatedPendingLimit o MGCOrioginatedPendingLimit) si el procesamiento de la instrucción es largo o debido a un error (por ejemplo la instrucción provoca un bucle). En ambos casos, la respuesta del receptor de TransactionRequest original será un TransactionReply que contiene un descriptor de error, como

parámetro de respuesta, para indicar la instrucción demasiado larga o la instrucción que provocó el error. Las siguientes instrucciones de la transacción no serán procesadas. Se utilizará el código error 506 ("Se ha rebasado el número máximo de transacciones pendientes").

NOTA – Cuando se sobrepasa el valor de xxxOriginatedPendingLimit (MGOOriginatedPendingLimit o MGCOOriginatedPendingLimit) debido a un error, y el receptor de la TransactionRequest original sigue transmitiendo TransactionPending, es necesario un mecanismo de protección para que el receptor de la TransactionRequest original active las acciones apropiadas de recuperación. El emisor de la TransactionRequest original puede registrar el número de Pendientes recibidos y poner en marcha las acciones correctivas.

8.3 Mensajes

Se pueden concatenar varias transacciones para formar un mensaje. Los mensajes tienen un encabezamiento, que incluye la identidad del emisor. El identificador de mensaje (MID, *message identifier*) será uno de los nombres configurados (por ejemplo dirección del dominio/nombre del dominio/nombre del dispositivo) de la entidad que transmite el mensaje. El nombre del dominio es un valor por defecto sugerido. Una entidad H.248.1 (MG MGC) utilizará de forma coherente el mismo MID en todos los mensajes que origina por todo el tiempo que dure la asociación de control con la entidad par (MGC MG).

Cada mensaje contiene un número que identifica la versión del protocolo de ese mensaje. Las versiones constan de uno o dos dígitos, comenzando con la versión 1, la Versión 3 del protocolo es la actual.

Las transacciones en un mensaje se tratan independientemente, sin ningún orden particular ni acuse de recibo de mensaje por la aplicación o el protocolo. Un mensaje es esencialmente un mecanismo de transporte. Por ejemplo, a un mensaje X que contiene las peticiones de transacción A, B y C se puede responder con un mensaje Y que contiene las respuestas a A y C, y un mensaje Z que contiene la respuesta a B. Asimismo, a un mensaje L que contiene la petición D y a un mensaje M que contiene la petición E se puede responder con un mensaje N que contiene respuestas a D y E.

9 Transporte

El mecanismo de transporte para el protocolo debe permitir el transporte fiable de transacciones entre un MGC y una MG. El transporte seguirá siendo independiente de las distintas instrucciones que se estén enviando y será aplicable a todos los estados de las aplicaciones. Hay varios transportes definidos para el protocolo en anexos a esta Recomendación y a otras Recomendaciones de la subserie H.248 (por ejemplo, H.248.4 y H.248.5). Podrán definirse transportes adicionales como Recomendaciones adicionales de la subserie H.248. Para el transporte del protocolo a través de IP, los MGC implementarán los protocolos TCP y UDP/ALF; una MG implementará el protocolo TCP o el protocolo UDP/ALF o ambos.

La MG se configura con un nombre o dirección (tal como el nombre de servidor de nombres de dominio (DNS, *domain name server*) o una dirección IP) de un MGC primario y de MGC secundarios (posiblemente ninguno) (véase 7.2.8), que es la dirección que la MG utiliza para enviar mensajes al MGC. Si se utilizan TCP o UDP como transporte de protocolo, y no se conoce el puerto al que se enviará la petición ServiceChange inicial, la petición debe enviarse al número de puerto por defecto para el protocolo. Este número de puerto es 2944 para operación con codificación textual o 2945 para operación con codificación en binario, sea para UDP o sea para TCP. El MGC recibe el mensaje que contiene la petición ServiceChange de la MG y a partir del mismo puede determinar la dirección de la MG. Como se describe en 7.2.8, tanto la MG como el MGC pueden proporcionar una dirección en el parámetro ServiceChangeAddress a la cual deben dirigirse las peticiones de transacción subsiguientes, pero las respuestas (incluida la respuesta a la petición ServiceChange inicial) deben enviarse siempre hacia atrás a la dirección que ha sido fuente de la

petición correspondiente. Por ejemplo, en las redes IP, ésta es la dirección fuente en el encabezamiento IP y el número de puerto fuente en el encabezamiento TCP/UDP/SCTP.

9.1 Ordenación de las instrucciones

Esta Recomendación no prescribe que el protocolo de transporte subyacente garantice la secuenciación de las transacciones enviadas a una entidad. Esta propiedad es una garantía de ejecución oportuna de las acciones, pero tiene algunos inconvenientes. Por ejemplo:

- Las instrucciones de notificación pueden ser demoradas y llegar al MGC después de que se haya transmitido una nueva instrucción que cambie el descriptor de eventos.
- Si se transmite una nueva instrucción antes de que se haya acusado recibo de una instrucción precedente, no hay garantía de que la instrucción precedente será ejecutada antes que la nueva.

Las siguientes reglas para los controladores permiten garantizar el funcionamiento correcto de la pasarela de medios. Estas reglas se refieren a instrucciones que se encuentran en diferentes transacciones. Las instrucciones que se encuentran en la misma transacción son ejecutadas por orden (véase la cláusula 8).

- 1) Cuando una pasarela de medios trata varias terminaciones, es posible que instrucciones pertenecientes a terminaciones diferentes se transmitan en paralelo, por ejemplo siguiendo un modelo en el que cada terminación (o grupo de terminaciones) sea controlada por su propio proceso o por su propio hilo.
- 2) Normalmente, en una terminación debe haber como máximo una instrucción pendiente (Añadir o Modificar o Desplazar), a menos que las instrucciones pendientes se encuentren en la misma transacción. En cambio, una instrucción Substraer puede emitirse en cualquier momento. Por consiguiente, una pasarela de medios puede a veces recibir una instrucción Modificar aplicable a terminaciones que hayan sido anteriormente substraídas. Tales instrucciones deberán ser ignoradas y se debe retornar un código de error.
- 3) En el caso de transportes que no garantizan la entrega de mensajes en secuencia (como en el UDP), en una determinada terminación no habrá normalmente más de una instrucción Notificar pendiente en un momento dado cualquiera.
- 4) En algunos casos, una instrucción Substraer acompañada implícita o explícitamente de comodines que se aplica a un grupo de terminaciones puede situarse antes de una instrucción Añadir pendiente. El controlador de pasarela de medios debe suprimir individualmente todas las terminaciones cuya compleción estaba pendiente en el momento de la instrucción Substraer global. Además, no podrán enviarse nuevas instrucciones Añadir para terminaciones denominadas por medio de comodines (o implícitas en un descriptor de múltiplex), mientras no se haya acusado recibo de la instrucción Substraer acompañada de comodines.
- 5) Las instrucciones AuditValue y AuditCapability no están sometidas a secuenciación.
- 6) ServiceChange será siempre la primera instrucción enviada por una MG, como se define por el procedimiento de re arranque. Toda otra instrucción o respuesta se entregará después de esta instrucción ServiceChange.

Estas reglas no afectan al respondedor de instrucciones, que siempre responderá a las instrucciones.

9.2 Protección contra las avalanchas de re arranques

En el caso en que un gran número de pasarelas de medios fueran activadas simultáneamente, y si todas ellas debieran iniciar una transacción emitiendo una instrucción ServiceChange, el controlador de pasarela de medios seguramente se vería inundado, situación que conduciría a pérdidas de mensajes y a congestión de la red durante el periodo crítico de restablecimiento del servicio. Para evitar esas avalanchas se propone el siguiente comportamiento:

- 1) La pasarela de medios activada debe poner en marcha un temporizador de re arranque que se fija a un valor aleatorio distribuido uniformemente entre cero y un periodo de espera máximo (MWD, *maximum waiting delay*). Se deben tomar precauciones para evitar que se produzcan sincronizaciones entre los números aleatorios generados por múltiples pasarelas de medios que utilizasen el mismo algoritmo.
- 2) La pasarela de medios debe entonces esperar hasta que termine este temporizador o hasta que se detecte una actividad del usuario local, por ejemplo una transición a descolgado en una pasarela de medios residencial.
- 3) Cuando ha expirado el temporizador, o cuando se detecta una actividad del usuario local, la pasarela de medios debe iniciar el procedimiento de re arranque.

El procedimiento de re arranque exige, simplemente, que la MG garantice que el primer mensaje que el controlador de pasarela de medios perciba, procedente de la misma, sea un mensaje ServiceChange que informe al controlador de pasarela de medios sobre el re arranque.

NOTA – El valor de MWD es un parámetro de configuración que depende del tipo de la pasarela de medios. Se puede adoptar el siguiente razonamiento para determinar el valor de este retardo en las pasarelas residenciales.

Habitualmente los controladores de pasarelas de medios se dimensionan para que traten la carga de tráfico en la hora punta, durante la cual, en promedio, el 10% de las líneas están ocupadas por llamadas con una duración promedio típica es de tres minutos. El procesamiento de una llamada suele consistir en cinco a seis transacciones del controlador de pasarela de medios, entre cada pasarela de medios y el controlador de pasarela de medios. De un cálculo simple se deduce que el controlador de pasarela de medios trata de cinco a seis transacciones para cada terminación cada 30 minutos en promedio, o, dicho de otra manera, que trata aproximadamente una transacción por cada terminación cada cinco o seis minutos en promedio. Esto hace pensar que un valor razonable de MWD para una pasarela residencial sería de 10 a 12 minutos. Si no hay una configuración explícita, las pasarelas residenciales deben adoptar un valor de 600 segundos para MWD.

Un razonamiento similar lleva a la conclusión de que el valor de MWD debe ser mucho menor en el caso de pasarelas de entronque o de pasarelas comerciales, porque estas pasarelas tratan un gran número de terminaciones, y también porque la tasa de utilización de estas terminaciones es mucho mayor que el 10% durante la hora punta (habitualmente 60%). Se espera que estas terminaciones contribuyan, durante la hora punta, aproximadamente con una transacción por minuto a la carga del controlador de pasarela de medios. Un algoritmo razonable consiste en hacer que el valor de MWD por cada terminación "troncal" sea seis veces más corto que el MWD por cada pasarela residencial, y que también sea inversamente proporcional al número de terminaciones que se re arrancan. Por ejemplo, el MWD debe fijarse a 2,5 segundos para una pasarela que trate una línea T1, o a 60 milisegundos para una pasarela que trate una línea T3.

9.3 Protección contra las avalanchas de Notificaciones

Cuando se acumule una cantidad importante de notificaciones en una MG, bien sea debido a dificultades de transmisión o bien porque la pasarela ha reconocido un cierto número de eventos en un corto lapso de tiempo, la MG debería enviar las notificaciones de una manera restringida hasta tanto que se solucione el atasco.

10 Consideraciones sobre seguridad

Esta cláusula cubre la seguridad cuando se utiliza el protocolo en un entorno IP.

10.1 Protección de las conexiones de protocolo

Es evidente que se necesita un mecanismo de seguridad para impedir que entidades no autorizadas utilicen el protocolo definido en esta Recomendación para el establecimiento de comunicaciones no autorizadas o para interferir en llamadas autorizadas. El mecanismo de seguridad para el protocolo cuando éste se transporta por redes IP es el mecanismo de seguridad del protocolo Internet IPSec (RFC 2401 a RFC 2411).

El encabezamiento AH (RFC 2402) proporciona autenticación del origen de datos, integridad sin conexión y protección facultativa antirreproducción de mensajes cursados entre la MG y el MGC. El encabezamiento ESP (RFC 2406) proporciona confidencialidad de mensajes, si se desea. Por ejemplo, se debe solicitar el servicio de criptación ESP si las descripciones de sesión se utilizan para transportar claves de sesión, tal como se definen en el protocolo SDP.

En las implementaciones del protocolo definido en esta Recomendación con el encabezamiento ESP deben respetarse las condiciones de la sección 5 de RFC 2406, que define un conjunto mínimo de algoritmos para la verificación de la integridad y criptación. De manera similar, en las implementaciones con el encabezamiento AH deben cumplirse las condiciones de la sección 5 de RFC 2402, que define un conjunto mínimo de algoritmos para la verificación de la integridad mediante el empleo de claves manuales.

Las implementaciones deberían utilizar IKE (RFC 2409) para permitir opciones con claves más fuertes. Las implementaciones que emplean IKE DEBEN soportar la autenticación con firmas RSA y la criptación por clave pública RSA.

10.2 Esquema AH provisional

La implementación de la seguridad de protocolo Internet (IPSec, *Internet protocol security*) requiere el encabezamiento AH o ESP inmediatamente después del encabezamiento IP, lo que no puede hacerse fácilmente en el nivel de aplicación. Por consiguiente, hay un problema de implantación del protocolo definido en esta Recomendación cuando la implementación de la red subyacente no soporta IPSec.

Como una solución provisional, se define un encabezamiento AH facultativo dentro del encabezamiento de protocolo H.248.1. Los campos del encabezamiento son exactamente los mismos campos SPI, SEQUENCE NUMBER y DATA definidos en RFC 2402. Las semánticas de los campos de encabezamiento son las mismas que las del "modo transporte" de RFC 2402, salvo el cálculo del valor de la verificación de integridad (ICV, *integrity check value*). En IPSec, el ICV se calcula para la totalidad del paquete IP, incluido el encabezamiento IP. Con esto se evita la piratería de las direcciones IP. Para mantener la misma funcionalidad, el cálculo de ICV debe realizarse para todas las transacciones (concatenadas) en el mensaje precedidas de un encabezamiento IP sintetizado constituido por una dirección IP fuente de 32 bits, una dirección de destino de 32 bits y un puerto de destino UDP de 16 bits codificado como 20 dígitos hexadecimales. Cuando se emplea el mecanismo AH provisional siendo TCP la capa de transporte, el puerto UDP antes mencionado pasa a ser el puerto TCP, y todas las demás operaciones son las mismas.

Las implementaciones del protocolo H.248.1 implementarán el mecanismo de seguridad IPSec cuando el sistema operativo subyacente y la red de transporte lo soporten. Las implementaciones del protocolo que utilizan IPv4 implementarán el esquema AH provisional. Sin embargo, este esquema provisional no se utilizará cuando la capa de red subyacente soporta IPsec. Se supone que las implementaciones IPv6 soportan IPsec y no utilizarán el esquema AH provisional.

Todas las implementaciones del mecanismo provisional AH cumplirán con la sección 5 de RFC 2402 que define un conjunto mínimo de algoritmos para la verificación de la integridad utilizando claves manuales.

Un esquema AH provisional no proporciona protección contra la escucha/observación ilícita de conexiones, por lo que no impide que terceras partes supervisen las conexiones establecidas por una determinada terminación. Tampoco da protección contra ataques por reproducción. Estos procedimientos no protegen necesariamente contra los ataques que consisten en la denegación del servicio por los MG o MGC que se comportan incorrectamente. No obstante, proporcionarán una identificación de las entidades de este tipo que se comportan incorrectamente, las cuales deberán ser ulteriormente privadas de sus respectivas autorizaciones mediante procedimientos de mantenimiento.

10.3 Protección de las conexiones de medios

El protocolo permite al MGC proporcionar "claves de sesión" a las MG, que pueden utilizarse para criptar los mensajes audio y proteger contra la escucha/observación ilícita de las conexiones.

Un problema peculiar de las redes de paquetes es la "admisión no controlada". Este ataque consiste en dirigir paquetes de medios a la dirección IP y al puerto UDP utilizados por una conexión. Si no se ha implementado una protección contra este ataque, hay que descomprimir los paquetes y reproducir las señales en el "lado de la línea".

Una protección básica contra este ataque es aceptar solamente paquetes procedentes de fuentes conocidas, comprobando, por ejemplo, que la dirección fuente IP y el puerto fuente UDP concuerdan con los valores anunciados en el descriptor distante. Esto tiene dos inconvenientes: demora el establecimiento de la conexión, y puede ser burlado por piratería de la fuente:

- Para la protección basada en la dirección, el MGC obtendrá la descripción de sesión distante de la MG de egreso y la pasará a la MG de ingreso. Esto requiere por lo menos un recorrido de ida y retorno a través de la red, y plantea el siguiente dilema: o bien se permite que la llamada prosiga sin esperar hasta que se complete el recorrido de ida y retorno, con el riesgo, por ejemplo, del "recorte" de un anuncio distante, o se espera a que se complete el recorrido de ida y retorno a expensas de que los procedimientos de establecimiento de la comunicación sean más dilatados.
- La piratería de la fuente sólo es eficaz si el pirata puede obtener pares válidos de direcciones y puertos de fuente y de destino, por ejemplo escuchando una fracción del tráfico. Para combatir la piratería de la fuente se podría tratar de controlar todos los puntos de acceso a la red. Sin embargo, esto es muy difícil de conseguir en la práctica.

Una alternativa a la verificación de la dirección de fuente es criptar y autenticar los paquetes utilizando una clave secreta que se transporta durante el procedimiento de establecimiento de la comunicación. Esto no demora el establecimiento de la comunicación, y proporciona una protección adecuada contra la piratería de direcciones.

11 Interfaz de control MG-MGC

La asociación de control entre MG y MGC se inicia por un arranque en frío de la MG, y se anuncia por un mensaje ServiceChange, pero puede ser modificada por eventos subsiguientes, tales como fallos o eventos del servicio manual.

NOTA – Si bien el protocolo no tiene un mecanismo explícito para el soporte de múltiples MGC que controlan una MG física, se ha diseñado para el soporte de múltiples MG lógicas (pertenecientes a una sola MG física) que pueden ser asociadas con diferentes MGC.

11.1 Múltiples MG virtuales

Una pasarela de medios física puede dividirse en una o más MG virtuales (VMG). Una MG virtual consiste en un conjunto de terminaciones físicas estáticamente divididas y/o conjuntos de terminaciones efímeras. Una terminación física es controlada por un MGC. El modelo no requiere que se atribuyan estáticamente otros recursos, sino sólo terminaciones. El mecanismo para atribuir

terminaciones a MG virtuales es un método de gestión que está fuera del ámbito de esta Recomendación. Cada una de las MG virtuales aparece ante el MGC como un cliente MG completo.

Una MG física puede tener una sola interfaz de red, que deberá ser compartida por MG virtuales. En tal caso, la terminación en el lado lote/célula es compartida. Debe señalarse, sin embargo, que, en condiciones de utilización, esas interfaces requieren que se cree un ejemplar efímero de la terminación por cada flujo, por lo que la compartición de la terminación es inmediata. Este mecanismo no da lugar a complicaciones, por ejemplo que la MG tenga siempre que saber cuál de los MGC que la controlan debe ser notificado si se produce un evento en la interfaz.

En operación normal, el MGC ordenará a la MG virtual que cree flujos de red (si se trata del lado de origen) o que espere peticiones de flujo (si se trata del lado de terminación), y no habrá ninguna confusión. Sin embargo, si se produce un evento inesperado, la MG virtual tiene que saber lo que debe hacer con respecto a los recursos físicos que controla.

Si la recuperación después del evento requiere la manipulación del estado de una interfaz física, esto sólo puede hacerlo un MGC. Estas cuestiones se resuelven permitiendo que cualquiera de los MGC cree el descriptor de eventos para que se le notifiquen tales eventos, pero sólo un MGC puede tener acceso de lectura/escritura a las propiedades de la interfaz física; todos los demás MGC tienen acceso de sólo lectura. El mecanismo de gestión se utiliza para designar el MGC que tiene capacidad de lectura/escritura, el cual se designará por MGC maestro.

Cada MG virtual tiene su propia terminación raíz. En la mayor parte de los casos, los valores de las propiedades de la terminación raíz pueden ser fijados independientemente por cada MGC. Donde sólo pueda haber un valor, el parámetro es de lectura solamente para todos los MGC, excepto el MGC maestro.

ServiceChange sólo puede aplicarse a una terminación o conjunto de terminaciones divididas para la MG virtual, o creadas (en el caso de terminaciones efímeras) por esa MG virtual.

11.2 Arranque en frío

Mediante un mecanismo de gestión que está fuera del ámbito de esta Recomendación, inicialmente se configura en una MG un MGC primario y (facultativamente) una lista ordenada de MGC secundarios. Tras un arranque en frío de la MG, ésta enviará una instrucción ServiceChange con un método "rearranque", en la terminación raíz, a su MGC primario. Si el MGC acepta la MG, le envía una respuesta de transacción que no incluye un parámetro ServiceChangeMgcID. Si el MGC no acepta el registro de la MG, envía una respuesta de transacción que proporciona la dirección de un MGC alternativo que habrá de ser contactado incluyendo un parámetro ServiceChangeMgcID.

Si la MG recibe una respuesta de transacción que incluye un parámetro ServiceChangeMgcID, envía un ServiceChange al MGC especificado en el ServiceChangeMgcID. La MG continúa este proceso hasta que obtiene un MGC de control que acepte su registro, o hasta que fracase en su intento de recibir una contestación. Una vez fracasada la tentativa de obtener una contestación, del MGC primario o de un sucesor designado, la MG intenta con los MGC secundarios configurados inicialmente, en su orden. Si la MG no puede establecer una relación de control con ningún MGC, deberá esperar un periodo de tiempo aleatorio como se describe en 9.2, e iniciar entonces nuevamente el contacto con su MGC primario y, si es necesario, con su MGC secundario.

Puede ocurrir que se pierda la contestación a una instrucción ServiceChange con Rearranque, y que la MG reciba una instrucción antes de la recepción de la respuesta ServiceChange. La MG emitirá un error 505 ("Solicitud de transacción recibida antes de haber recibido una respuesta de cambio de servicio").

11.3 Negociación de la versión de protocolo

Una instrucción ServiceChange procedente de una MG asignada a un MGC contendrá el número de versión del protocolo soportado por la MG en el parámetro ServiceChangeVersion. Cualquiera que sea la versión que aparece en el parámetro ServiceChangeVersion, el mensaje que contiene la instrucción será codificado como mensaje de versión 1. Después de la recepción de este mensaje, y si el MGC soporta solamente una versión inferior, el MGC enviará entonces una ServiceChangeReply con la versión inferior y más tarde todos los mensajes entre la MG y el MGC se conformarán a la versión inferior del protocolo. Si la MG no puede cumplimentar lo anterior y ha establecido una conexión de transporte con el MGC, debe cerrar dicha conexión. En cualquier caso, debe rechazar todas las peticiones subsiguientes procedentes del MGC con código de error 406 ("Versión no soportada").

Si el MGC sólo soporta una versión o versiones superiores a la versión de la MG, rechazará la asociación mediante el código de error 406 ("Versión no soportada").

Si el MGC soporta la versión indicada por la MG, se conformará a dicha versión en todos sus mensajes siguientes. En este caso, el MGC puede devolver una versión en la ServiceChangeReply (facultativo).

La negociación de la versión del protocolo también puede ocurrir en los ServiceChanges "traspaso" (Handoff) y "cambio-en-caso-de-fallo" (Failover).

Cuando se amplía el protocolo con nuevas versiones, deben cumplirse las siguientes reglas:

- 1) No deben cambiarse los elementos de protocolo existentes, es decir, procedimientos, parámetros, descriptor, propiedad, valores, a menos que haya que corregir un error de protocolo o sea necesario cambiar la operación del servicio que está soportando el protocolo.
- 2) No debe modificarse la semántica de una instrucción, parámetro, descriptor, propiedad o valor.
- 3) No deben modificarse las reglas establecidas para la formatación y la codificación de los mensajes y parámetros.
- 4) Cuando se encuentran elementos de información obsoletos pueden marcarse como no utilizados. No obstante, el identificador para este elemento de información se marcará como reservado. De este modo no podrá utilizarse en versiones futuras.

11.4 Fallo de una MG

Si una MG falla, pero es capaz de enviar un mensaje al MGC, le envía un ServiceChange con un método apropiado (Gradual o forzado) y especifica el TerminationID raíz. Cuando retorna al servicio, envía un ServiceChange con un ServiceChangeMethod "rearranque".

El hecho de que el MGC pueda enviar mensajes duplicados a ambas MG es conveniente para funcionar con un par de MG con la posibilidad de cambio-en-caso-de-fallo redundante de una de ellas. Solamente la MG en funcionamiento aceptará o rechazará transacciones. Tras un cambio-en-caso-de-fallo, la MG primaria envía una instrucción ServiceChange con un método "cambio en caso de fallo" y un motivo "fallo inminente de MG". El MGC utiliza entonces la MG secundaria como la MG activa. Una vez eliminada la condición de error, la MG en funcionamiento puede enviar un "ServiceChange" con un ServiceChangeMethod "rearranque".

NOTA – Las MG que sufren un cambio-en-caso-de-fallo redundante requieren un transporte fiable, porque el protocolo no proporciona un medio fiable para que una MG que ejecuta un ALF acuse recibo de mensajes enviados desde el MGC.

11.5 Fallo de un MGC

Si la MG detecta un fallo del MGC que ejerce el control sobre ella, tratará de conectarse con el MGC que sigue orden en la lista que le fue previamente suministrada. La MG comienza los intentos por el MGC que aparece al principio de la lista (el MGC primario), a menos que el MGC primario haya sido el que ha fallado, en cuyo caso comienza por el primer MGC secundario. La MG envía un mensaje ServiceChange con un método "cambio-en-caso-de-fallo" y el motivo "Fallo inminente del MGC". Si la MG no puede establecer una relación de control con ningún MGC, esperará durante un tiempo determinando de forma aleatoria como se describe en 9.2, transcurrido el cual comenzará de nuevo a contactar a su MGC primario y, (si es necesario), a sus MGC secundarios. Cuando la MG contacta al MGC que estaba ejerciendo el control anteriormente, le envía el mensaje ServiceChange con método "Desconectado".

En caso de fallo parcial o por razones de mantenimiento manual, podría necesitarse que el MGC indique a la MG controlada que utilice un MGC diferente. Para hacer esto, envía un ServiceChange a la MG con método "Traspaso", y su sustituto designado en ServiceChangeMgcID. Si "Traspaso" está soportado, la MG enviará un mensaje ServiceChange con método "Traspaso" y un motivo "MGC ordenó cambio" al MGC designado. Si no obtiene una contestación del MGC designado, se comportará como si su MGC hubiera fallado, y comenzará a contactar MGC secundarios como se especifica en el párrafo anterior. Si la MG no puede establecer una relación de control con ningún MGC, deberá esperar un tiempo determinado de forma aleatoria como se describe en 9.2, y entonces iniciar otra vez el contacto con su MGC primario, y si es necesario, con sus MGC secundarios.

No se formula ninguna recomendación acerca del modo en que los MGC implicados en el traspaso (HandOff) mantienen la información de estado; se considera que esta materia cae fuera del alcance de la presente Recomendación. El MGC y la MG pueden realizar el siguiente proceso en caso de traspaso. El proceso de traspaso iniciado por el MGC debe ser transparente para las operaciones en la pasarela de medios. Las transacciones pueden ejecutarse en cualquier orden, y pudieran estar en curso cuando se ejecuta el ServiceChange. En consecuencia, continúan las instrucciones en curso y las contestaciones a todas las instrucciones procedentes del MGC original deben enviarse a la dirección de transporte desde la que se enviaron. Si la relación de servicio con el MGC emisor ha terminado, las contestaciones deben descartarse. La MG puede recibir contestaciones de transacción pendientes, del nuevo MGC. No se enviará ningún mensaje nuevo al nuevo MGC hasta que la asociación de control haya sido establecida. Las peticiones de transacción repetidas serán dirigidas al nuevo MGC. La MG mantendrá el estado en todas las terminaciones y contextos.

El MGC puede estar implementado de forma que un MGC que falla sea reemplazado por un MGC activo cuando la identidad del nuevo MGC es la misma que la identidad del que ha fallado. En tal caso, ServiceChangeMgcID se especificaría con el valor anterior y la MG se comportará como si el valor hubiera cambiado y enviará un mensaje ServiceChange en la forma descrita.

Los pares de MGC que pueden hacer cambio-en-caso-de-fallo redundante, pueden utilizar el mecanismo antes mencionado para notificar a las MG controladas el cambio-en-caso-de-fallo.

11.6 Supervisión de la asociación de control MGC-MG

Para garantizar una alta disponibilidad de la red es fundamental la supervisión de la asociación de control MGC-MG, algo que se puede lograr verificando continuamente la interconexión (estado de enlace) entre el MGC y la MG. Puesto que en muchos protocolos de transporte ya se cuenta con esta funcionalidad, no es necesario especificar un mecanismo de protocolo obligatorio.

Si el protocolo de transporte no proporciona la supervisión de estado de enlace, se puede obtener esta funcionalidad en el protocolo H.248 utilizando los mensajes existentes. Si no hay estos mensajes, el MGC puede utilizar una instrucción AuditValue en Raíz con un descriptor de auditoría

vacío para detectar la pérdida de comunicación con la MG. La misma MG puede detectar la pérdida de comunicación mediante el lote de temporizador de inactividad (Rec. UIT-T H.248.14).

12 Definición de lotes

El mecanismo primario para la ampliación (o extensión) se basa en lotes. Los lotes definen propiedades adicionales que pueden estar presentes en terminaciones y contextos, así como eventos, señales y estadísticas que pueden producirse en terminaciones.

Los lotes definidos por Grupo IETF se especificarán en otros documentos (RFC).

Los lotes definidos por el UIT-T pueden especificarse en las Recomendaciones pertinentes (por ejemplo, como Recomendaciones de la subserie H.248).

- 1) Se debe especificar el documento público o el documento de un foro de normalización en el que se describe el lote, como se ha indicado antes.
- 2) El documento especificará la versión del lote que describe.
- 3) El documento deberá estar disponible en un servidor web público y debe tener un URL estable. El sitio debe proporcionar un mecanismo para ofrecer comentarios y retornar respuestas apropiadas.

12.1 Directrices para la definición de lotes

Los lotes definen propiedades, eventos, señales y estadísticas.

Los lotes también pueden definir nuevos códigos de error de acuerdo con las directrices dadas en 14.2. Lo determina la tramitación de documentos: la documentación de lotes se somete a la IANA para soportar el registro de códigos de error. Si se modifica un lote, no es necesario proporcionar a la IANA una nueva referencia de documento de soporte del código de error, a menos que se modifique la descripción del propio código de error.

Los nombres de estas definiciones se formarán con el PackageID (que identifica unívocamente el lote), y el ID del artículo (que identifica unívocamente al artículo en ese lote). En la codificación textual se separarán los dos identificadores mediante un carácter de barra de fracción derecha ("/"). Por ejemplo: tonegen/playtone es la codificación textual para hacer referencia a la señal de tono de reproducción en el lote de generación de tonos.

Un lote contendrá las siguientes secciones y palabras clave (en negritas). En el apéndice II se incluye una plantilla para la definición de lotes.

12.1.1 Lote

Descripción global del lote; se especifica:

Nombre del lote: Descriptivo solamente

PackageID: Es un identificador

Descripción: Es una descripción del lote.

Versión:

Una nueva versión de un lote sólo puede añadir propiedades, eventos, señales o estadísticas adicionales, así como nuevos valores posibles para un parámetro existente descrito en el lote original. No se permitirán supresiones ni modificaciones. Una versión es un número entero de 1 a 99.

Previsto solamente para ser extendido (facultativo): Sí

Esto indica que el lote ha sido designado expresamente para ser extendido por otros, y no para que se haga referencia al mismo directamente. Por ejemplo, el lote puede que

no tenga ninguna función propia, o que no tenga sentido por sí solo. La MG no publicará este PackageID cuando informe lotes.

Extiende: Identificador y versión de lote existente

Un lote puede extender (ampliar) un lote existente. La versión del lote original debe especificarse. Cuando un lote extiende otro lote sólo añadirá propiedades, eventos, señales o estadísticas adicionales, y nuevos valores posibles de un parámetro existente descrito en el lote original. Un lote extendido no redefinirá ni sobrecargará un identificador definido en el lote original, ni lotes que pueda haber extendido (múltiples niveles de extensión). En consecuencia, si el lote B versión 1 extiende el lote A versión 1, la versión 2 de B no podrá ampliar la versión 2 de A si la versión 2 de A define un nombre que ya está en la versión 1 de B. Si el lote no extiende otro lote, especificará "ninguno".

12.1.2 Propiedades

Propiedades definidas por el lote; se especifica:

Nombre de propiedad: Descriptivo solamente

PropertyID: Es un identificador

Descripción: Es una descripción de la función de la propiedad

Tipo: Uno de los siguientes:

Booleano

Cadena: Cadena UTF-8

Cadena de octetos: Un número de octetos. Para la codificación, véanse anexo A y B.3

Entero: Entero con signo 4 octetos

Doble: Entero con signo 8 octetos

Carácter: Codificación UTF-8 Unicode de una sola letra. Podría ser de más de un octeto

Enumeración: Un valor de una lista de valores únicos posibles

Sublista: Una lista de varios valores de una lista. También se especificará el tipo de sublista. El tipo se elegirá entre los tipos especificados en esta cláusula (salvo sublista). Por ejemplo, tipo: sublista de enumeración. La codificación de sublistas se especifica en el anexo A y B.2.

Valores posibles:

Un lote especificará sea un determinado conjunto de valores, sea una descripción de la manera de determinar los valores. Un lote especificará también un valor por defecto o el comportamiento por defecto cuando el valor se omite en su descriptor. Por ejemplo, un lote puede especificar que los procedimientos relacionados con la propiedad se suspendan cuando se omita su valor.

Valor por defecto:

Un valor por defecto es una de las características que se pueden configurar (no los procedimientos).

Definido en:

El descriptor H.248.1 en el que está definida la propiedad. LocalControl es para propiedades dependientes del tren. TerminationState es para propiedades independientes del tren. ContextAttribute se utiliza cuando se trata de propiedades que afecten el contexto como un todo, es decir, propiedades de grupo. Se prevé que éstos sean los casos más comunes, pero es posible que las propiedades se definan en otros

descriptores. Las propiedades de contexto se deben definir en el descriptor ContextAttribute.

Características: Sólo lectura o Lectura/escritura y (facultativamente), global:

Indica si una propiedad es de lectura solamente o lectura/escritura, y si es global. Cuando se ha omitido global, la propiedad no es global. Si una propiedad se declara como global, el valor de la propiedad es compartido por todas las terminaciones que realizan el lote. Si una propiedad de contexto se declara como global, la propiedad es compartida por todos los contextos que realizan el lote.

12.1.3 Eventos

Eventos definidos por el lote; se especifica:

Nombre de evento: Solamente descriptivo

EventID: Es un identificador

Descripción: Es una descripción de la función del evento

Parámetros del descriptor de eventos (EventsDescriptor):

Parámetros utilizados por el MGC para configurar el evento, y encontrados en el descriptor de eventos. Véase 12.2. Si no hay parámetros para este descriptor, se especificará "ninguno".

Parámetros del descriptor de eventos observados (ObservedEventsDescriptor):

Parámetros retornados al MGC en peticiones Notificación y en contestaciones a peticiones de instrucción procedentes del MGC que efectúa una auditoría del descriptor de eventos observados (ObservedEventsDescriptor), y encontrados en el ObservedEventsDescriptor. Véase 12.2. Si no hay parámetros para este descriptor, se especificará "ninguno".

12.1.4 Señales

Señales definidas por el lote; se especifica:

Nombre de señal: Solamente descriptivo

SignalID: Es un identificador. SignalID se utiliza en un descriptor de señales (SignalsDescriptor).

Descripción: Es una descripción de la función de la señal

SignalType: Uno de los siguientes:

OO (OnOff)

TO (TimeOut)

BR (Brief)

NOTA – SignalType puede definirse de tal manera que dependa del valor de uno o más parámetros. El lote especificará un tipo de señal por defecto. Si el tipo por defecto es TO, el lote especificará una duración por defecto que puede suministrarse en el proceso de provisión. Una duración por defecto no tiene sentido en señales de tipo BR.

Duración: En centésimas de segundo

Parámetros adicionales: Véase 12.2.

12.1.5 Estadísticas

Estadísticas definidas por el lote; se especifica:

Nombre de la estadística: Solamente descriptivo

StatisticID: Es un identificador

StatisticID se utiliza en un descriptor de estadísticas

Descripción: Una descripción de la estadística.

Tipo: Uno de los siguientes:

Booleano

Cadena: Cadena UTF-8

Cadena de octetos: Varios octetos. Véase la codificación en el anexo A y B.3

Entero: Entero de 4 bytes con signo

Doble: Entero de 8 bytes con signo

Carácter: Codificación UTF-8 Unicode para un sólo carácter. Podría tener más de un octeto.

Enumeración: Uno de una lista de valores únicos posibles.

Sublista: Una lista de varios valores tomados de otra lista. Se especificará también el tipo de sublista. El tipo se escogerá entre los tipos especificados en esta subcláusula (salvo sublista). Por ejemplo, Tipo: sublista de enumeración. En el anexo A y B.2 se especifica la codificación de sublistas.

Valores posibles:

Un lote debe indicar la unidad de medida, por ejemplo milisegundos o lotes, bien sea aquí o junto con el tipo del párrafo anterior, así como toda restricción del intervalo.

Nivel: Especifica si se pueden mantener las estadísticas en la terminación, el tren o ambos.

12.1.6 Códigos de error

Si el lote no define ningún código de error, se puede omitir esta sección. De lo contrario, describe códigos de error definidos por el lote, especificando:

de código de error: El número del código de error.

Nombre: Nombre del error

Definición: Descripción del código de error.

Texto en el descriptor de error:

Descripción del texto que se ha de devolver en el descriptor de error.

Observación: Cualquier otro comentario acerca de la utilización del código de error.

12.1.7 Procedimientos

Orientaciones adicionales sobre la utilización del lote.

12.2 Directrices para definir los parámetros de eventos y señales

Nombre de parámetro: Solamente descriptivo

ParameterID: Es un identificador. El ParameterID textual de parámetros para eventos y señales no comenzará por "EPA" ni "SPA", respectivamente. El ParameterID textual tampoco será "ST", "Stream", "SY", "SignalType", "DR", "Duration", "NC", "NotifyCompletion", "KA", "KeepActive", "EB", "Embed", "DM", "DigitMap", "SPADI", "SPADirection", "SPARQ" o "SPARquestID".

Descripción: Es una descripción de la función del parámetro.

Tipo: Uno de los siguientes:

Booleano

Cadena: Cadena de octetos UTF-8

Cadena de octetos: Un número de octetos, Para la codificación, véanse el anexo A y B.3

Entero: Número entero con signo 4 octetos

Doble: Número entero con signo 8 octetos

Carácter: Codificación UTF-8 Unicode de una sola letra. Podría ser de más de un octeto.

Enumeración: Uno de una lista de posibles valores únicos

Sublista: Una lista de varios valores tomados de una lista (no soportados para estadísticas). El tipo de sublista se especificará también. El tipo se escogerá entre los tipos especificados en esta cláusula (salvo sublista). Por ejemplo, tipo: sublista de enumeración. La codificación de sublistas se especifica en el anexo A y B.2.

Facultativo: Sí/No

Indica si el parámetro se puede omitir en la señal o el evento.

Posibles valores:

Un lote especificará sea un determinado conjunto de valores, sea una descripción de la manera de determinar los valores. Un lote especificará también un valor por defecto o el comportamiento por defecto cuando el valor se omite en su descriptor. Por ejemplo, un lote puede especificar que los procedimientos relacionados con la propiedad se suspendan cuando se omita su valor.

Valor por defecto:

Un valor por defecto es una de las características que se pueden configurar (no los procedimientos).

12.3 Identificadores

Los identificadores en la codificación textual serán cadenas de no más de 64 caracteres, sin espacios, comenzarán por un carácter alfabético y estarán constituidas por caracteres alfanuméricos y/o dígitos, y posiblemente incluyan el carácter de subrayado ("_").

Los identificadores en la codificación en binario tienen una longitud de 2 octetos.

Para cada identificador se especificará el valor textual y el valor binario, incluidos los identificadores utilizados como valores en tipos enumerados.

12.4 Registro de lotes

Un lote puede registrarse en la IANA por razones de interoperabilidad. Para las consideraciones relativas a la IANA véase la cláusula 14.

13 Definición de perfil

Pueden definirse perfiles para determinar de una forma más precisa cómo se utiliza el protocolo H.248.1 y cuáles son las funciones soportadas por una MG. El perfil sólo especifica las capacidades de la interfaz MGC/MG H.248. En el perfil se indica cuáles son las opciones de H.248.1 que se utilizan. Por ejemplo: transporte y paquetes utilizados para una aplicación.

Cada perfil se identifica con un nombre (registrado en la IANA) y una versión. El nombre es una cadena, que no depende de mayúsculas o minúsculas, que puede tener hasta 64 caracteres, y la versión es un número entre 1 y 99.

El perfil propiamente dicho es un documento que indica las opciones para una determinada aplicación. No hay ningún formato especificado para este documento. La única condición es que se incluya una sección donde se indique el nombre y la versión, así como el resumen del perfil.

Los dos siguientes puntos son las únicas secciones de un perfil obligatorias:

- Identificación del perfil: Nombre y versión del perfil que se envía en una instrucción cambio de servicio.
- Resumen: Descripción del perfil.

En el apéndice III se incluye una plantilla que debería utilizarse como base para la definición de perfiles.

14 Consideraciones relativas a la IANA

14.1 Lotes

Para el registro de un lote en la IANA se deberán satisfacer las siguientes consideraciones:

- 1) Para cada lote se registra un nombre de cadena único, un número de serie único y un número de versión. El nombre de cadena se utiliza con la codificación textual. El número de serie se utilizará con la codificación en binario. Los números de serie 0x8000 a 0xFFFF están reservados para uso privado. El número de serie 0 está reservado.
- 2) Se registrará un nombre de contacto, así como las direcciones postal y de correo electrónico de ese contacto. La información relativa al contacto será actualizada por la organización definidora cuando sea necesario.
- 3) Una referencia a un documento, público, que describe el lote. El documento especificará la versión del lote allí descrito.
Si el documento es público, debe estar ubicado en un servidor web público y debe tener un URL estable. El sitio deberá proporcionar un mecanismo para ofrecer comentarios y devolverá respuestas apropiadas.
- 4) Los lotes registrados por otros organismos que no sean organismos de normalización reconocidos tendrán un nombre de lote con una longitud mínima de 8 caracteres.
- 5) Para todos los demás nombres de lotes, el tratamiento se hará por orden de registro, si se han satisfecho todas las demás condiciones.

14.2 Códigos de error

Para el registro de un código de error en la IANA deberán satisfacerse las siguientes consideraciones:

- 1) Para cada error se registra un número de error y una cadena formada por una línea (80 caracteres como máximo).
- 2) Se incluirá, en un documento disponible públicamente, una descripción completa de las condiciones en las que se detectó el error. La descripción será lo suficientemente clara para diferenciar el error de todos los demás códigos de error existentes.
- 3) El documento deberá estar disponible en un servidor web público y debe tener un URL estable.
- 4) Los números de error registrados por organismos de normalización reconocidos tendrán 3 ó 4 caracteres.
- 5) Los números de error registrados por todas las demás organizaciones o individuos tendrán 4 caracteres.
- 6) Un número de error no será redefinido ni modificado, salvo por la organización o individuo que lo definió inicialmente, o por sus sucesores o representantes.

14.3 Motivos para ServiceChange

Para el registro, en la IANA, de motivos del cambio de servicio deberán satisfacerse las siguientes consideraciones:

- 1) Para cada motivo se registra un código de motivo único constituido por una sola frase con una longitud máxima de 80 caracteres.
- 2) Una descripción completa de las condiciones en las que se utiliza el motivo se incluirá en un documento disponible públicamente. La descripción será lo suficientemente clara para distinguir el motivo de todos los demás motivos existentes.
- 3) El documento deberá estar disponible en un servidor web público y tener un URL estable.

14.4 Perfiles

Para el registro de un perfil en la IANA deberán satisfacerse las siguientes consideraciones:

- 1) Para cada perfil se registra un nombre único y el número de versión (puede omitirse la versión si el nombre del perfil incluye un carácter comodín).
- 2) Se registrará un nombre de contacto, así como las direcciones postal y de correo electrónico de ese contacto. La información relativa al contacto será actualizada por la organización definidora cuando sea necesario.
- 3) Los perfiles registrados por otros organismos que no sean organismos de normalización reconocidos serán identificados con un nombre de una longitud de 6 caracteres.
- 4) Se aceptarán nombres de perfiles con un carácter comodín "*" al final si se especifican completamente los primeros 6 caracteres. Se supone que la organización destinataria del nombre de perfil completará el espacio del carácter comodín. La IANA no producirá otros nombres de perfiles con el formato "nombre*".

Para todos los demás nombres de perfiles, el tratamiento se hará por orden de registro, si se han satisfecho todas las demás condiciones.

Anexo A

Codificación binaria del protocolo

Este anexo especifica la sintaxis de los mensajes con la notación definida en la Rec. UIT-T X.680, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica*. Los mensajes deben codificarse para la transmisión aplicando las reglas de codificación básica especificadas en la Rec. UIT-T X.690, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básica (BER), de las reglas de codificación canónica (CER) y de las reglas de codificación distinguida (DER)*.

A.1 Codificación de comodines

La utilización de los comodines ALL y CHOOSE en el protocolo, permite que un MGC especifique parcialmente identificadores de terminaciones, y que la MG escoja entre los valores que son conformes a la especificación parcial. Los identificadores de terminaciones pueden codificar una jerarquía de nombres que está configurada. Por ejemplo, un TerminationID puede consistir en un grupo de troncales, una troncal perteneciente a ese grupo, y un circuito. La utilización de comodines será posible en todos los niveles. En los párrafos siguientes se explica el procedimiento.

En la descripción ASN.1, los TerminationID están constituidos por cadenas de octetos con una longitud máxima de 8 octetos. Esto significa que los identificadores de terminación constan de un máximo de 64 bits. Un identificador de terminación completamente especificado puede ir precedido de una secuencia de campos de comodines. Un campo de comodín tiene una longitud de un octeto. El bit 7 (bit más significativo) de este octeto especifica el tipo de comodín que se invoca: si el valor del bit es 1 se utiliza el comodín ALL; si el valor del bit es 0 se utiliza el comodín CHOOSE. El bit 6 del campo de comodín especifica si el comodín pertenece a un nivel del esquema de denominación jerárquico (valor 0 del bit) o al nivel de la jerarquía especificado en el campo de comodín más todos los niveles inferiores (valor 1 del bit). Los bits 0 a 5 del campo de comodín especifican la posición de bit de comienzo de comodín en el identificador de terminación.

Se explica este esquema con algunos ejemplos en los que el bit más significativo de una cadena de bits aparece a la izquierda.

Supóngase que los identificadores de terminación tienen una longitud de tres octetos y que cada octeto representa un nivel en un esquema de denominación jerárquico. Un ID de terminación válido es:

00000001 00011110 01010101.

El direccionamiento de TODOS (ALL) los nombres con el prefijo 00000001 00011110 se efectúa como sigue:

campo de uso de comodín: 10000111

ID de terminación: 00000001 00011110 xxxxxxxx.

El valor de los bits señalados por "x" es indiferente y será ignorado por el receptor.

Una indicación al receptor de que debe escoger un nombre que tenga 00011110 como segundo octeto se efectúa como sigue:

campos de uso de comodín: 00010111 seguido de 00000111

ID de terminación: xxxxxxxx 00011110 xxxxxxxx.

El primer campo corresponde a un comodín CHOOSE para el nivel de la jerarquía de denominación que comienza en el bit 23, que es el nivel más alto en el esquema de denominación del ejemplo. El segundo campo corresponde a un comodín CHOOSE para el nivel de la jerarquía de denominación que comienza en el bit 7, que es el nivel más bajo en el esquema de denominación del ejemplo.

Por último, un nombre al que se ha aplicado el comodín CHOOSE con el nivel más alto del nombre igual a 00000001 se especifica como sigue:

campo de uso de comodín: 01001111

ID de terminación: 00000001 xxxxxxxx xxxxxxxx.

El valor 1 de bit en la posición de bit 6 del primer octeto del campo de uso comodín indica que el uso de comodín comprende el nivel especificado en la jerarquía de denominación y todos los niveles inferiores.

También se pueden utilizar comodines en ID de contexto. Sin embargo, en el caso de ID de contexto no se permite la especificación de nombres parciales. Se utilizará el ID de contexto 0x0 para indicar el contexto NULL, el ID de contexto 0xFFFFFFFFE para indicar un comodín CHOOSE, y el ID de contexto 0xFFFFFFFF para indicar un comodín ALL.

Se utilizará el ID de terminación 0xFFFFFFFFFFFFFFFF para indicar la terminación Raíz.

A.2 Especificación de sintaxis en ASN.1

Esta cláusula contiene la especificación de la sintaxis del protocolo H.248.1 en notación de sintaxis abstracta uno (ASN.1).

NOTA 1 – En caso de que se utilice un mecanismo de transporte que emplee entramado en el nivel de aplicación, la definición de **Transaction** que se da más adelante cambia. Para la definición aplicable a ese caso, véase el anexo o la Recomendación de la serie H.248.x que define el mecanismo de transporte.

NOTA 2 – Esta especificación de sintaxis no impone todas las restricciones para valores e introducción de elementos. Algunas restricciones adicionales se establecen en comentarios y otras restricciones aparecen en el texto de esta Recomendación. Estas restricciones adicionales forman parte del protocolo aunque no vengan impuestas por esta Recomendación.

NOTA 3 – El módulo ASN.1 en este anexo utiliza tipos cadena de octetos para codificar valores y estadísticas de parámetro de propiedad, parámetro de señal y parámetro de evento. Los tipos que se utilizan efectivamente para estos valores pueden ser diferentes y se especifican en el anexo C o en la definición de paquete pertinente.

Un valor se codifica primero según las reglas BER según su tipo, utilizando el siguiente cuadro. El resultado de esta codificación BER se codifica como una cadena de octetos ASN.1, aplicando "doble desbordamiento de línea" al valor. El formato especificado en el anexo C o en los paquetes se relaciona con la codificación BER de acuerdo con el siguiente cuadro:

Tipo especificado en el paquete	Tipo BER ASN.1
String (cadena)	IA5String o UTF8String (nota 4)
Integer (entero) (4 octetos)	INTEGER
Double (doble) (entero con signo, 8 octetos)	INTEGER (nota 3)
Character (carácter) (UTF-8) (nota 1)	IA5String
Enumeration (enumeración)	ENUMERATED
Boolean (booleano)	BOOLEAN
Unsigned Integer (entero sin signo) (nota 2)	INTEGER (nota 3)
Octet (String) [cadena (de octetos)]	OCTET STRING
NOTA 1 – Puede tener más de un octeto. NOTA 2 – En el anexo C se hace referencia al entero sin signo. NOTA 3 – La codificación BER de INTEGER no entraña el uso de 4 octetos. NOTA 4 – String (cadena) debe codificarse como IA5String si sólo contiene caracteres ASCII, y como UTF8String si contiene caracteres que no sean ASCII.	

Véase 8.7/X.690, para la definición de la codificación de un valor de tipo cadena de octetos.

```

MEDIA-GATEWAY-CONTROL {itu-t(0) recommendation(0) h(8) h248(248) modules(0)
media-gateway-control(0) version3(3)}
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
MegacoMessage      ::= SEQUENCE
{
    authHeader      AuthenticationHeader OPTIONAL,
    mess            Message
}

AuthenticationHeader ::= SEQUENCE
{
    secParmIndex    SecurityParmIndex,
    seqNum          SequenceNum,
    ad              AuthData
}

```

```

}

SecurityParmIndex ::= OCTET STRING(SIZE(4))
SequenceNum       ::= OCTET STRING(SIZE(4))
AuthData          ::= OCTET STRING(SIZE(12..32))
Message           ::= SEQUENCE
{
    version                INTEGER(0..99),
    -- The version of the protocol defined here is equal to 3.
    mId                   Mid, -- Name/address of message originator
    messageBody           CHOICE
    {
        messageError      ErrorDescriptor,
        transactions      SEQUENCE OF Transaction
    },
    ...
}

Mid                ::= CHOICE
{
    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName          DomainName,
    deviceName          PathName,
    mtpAddress          OCTET STRING(SIZE(2..4)),
    -- Addressing structure of mtpAddress:
    --   25 - 15         0
    --   | PC           | NI |
    --   24 - 14 bits   2 bits
    -- NOTE - 14 bits are defined for international use.
    -- Two national options exist where the point code is 16 or 24 bits.
    -- To octet align the mtpAddress, the MSBs shall be encoded as 0s.
    ...
}

DomainName         ::= SEQUENCE
{
    name                IA5String,
    -- The name starts with an alphanumeric digit followed by a sequence
    -- of alphanumeric digits, hyphens and dots. No two dots shall occur
    -- consecutively.
    portNumber          INTEGER(0..65535) OPTIONAL
}

IP4Address         ::= SEQUENCE
{
    address             OCTET STRING(SIZE(4)),
    portNumber          INTEGER(0..65535) OPTIONAL
}

IP6Address         ::= SEQUENCE
{
    address             OCTET STRING(SIZE(16)),
    portNumber          INTEGER(0..65535) OPTIONAL
}

PathName           ::= IA5String(SIZE(1..64))
-- See A.3

Transaction ::= CHOICE
{
    transactionRequest  TransactionRequest,
    transactionPending  TransactionPending,
    transactionReply    TransactionReply,
}

```



```

    transactionResponseAck      TransactionResponseAck,
    -- use of response acks is dependent on underlying transport
    ...,
    segmentReply                SegmentReply
}

TransactionId ::= INTEGER(0..4294967295) -- 32-bit unsigned integer

TransactionRequest ::= SEQUENCE
{
    transactionId      TransactionId,
    actions            SEQUENCE OF ActionRequest,
    ...
}

TransactionPending ::= SEQUENCE
{
    transactionId      TransactionId,
    ...
}

TransactionReply ::= SEQUENCE
{
    transactionId      TransactionId,
    immAckRequired     NULL OPTIONAL,
    transactionResult  CHOICE
    {
        transactionError      ErrorDescriptor,
        actionReplies         SEQUENCE OF ActionReply
    },
    ...,
    segmentNumber       SegmentNumber OPTIONAL,
    segmentationComplete NULL OPTIONAL
}

SegmentReply ::= SEQUENCE
{
    transactionId      TransactionId,
    segmentNumber      SegmentNumber,
    segmentationComplete NULL OPTIONAL,
    ...
}

SegmentNumber ::= INTEGER(0..65535)

TransactionResponseAck ::= SEQUENCE OF TransactionAck
TransactionAck ::= SEQUENCE
{
    firstAck           TransactionId,
    lastAck            TransactionId OPTIONAL
}

ErrorDescriptor ::= SEQUENCE
{
    errorCode          ErrorCode,
    errorText          ErrorText OPTIONAL
}

ErrorCode ::= INTEGER(0..65535)
-- See clause 14 for IANA considerations with respect to error codes

ErrorText ::= IA5String

ContextID ::= INTEGER(0..4294967295)

```

```

-- Context NULL Value: 0
-- Context CHOOSE Value: 4294967294 (0xFFFFFFFFE)
-- Context ALL Value: 4294967295 (0xFFFFFFFF)

```

```

ActionRequest ::= SEQUENCE
{
    contextId          ContextID,
    contextRequest     ContextRequest OPTIONAL,
    contextAttrAuditReq ContextAttrAuditRequest OPTIONAL,
    commandRequests    SEQUENCE OF CommandRequest
}

```

```

ActionReply ::= SEQUENCE
{
    contextId          ContextID,
    errorDescriptor    ErrorDescriptor OPTIONAL,
    contextReply       ContextRequest OPTIONAL,
    commandReply       SEQUENCE OF CommandReply
}

```

```

ContextRequest ::= SEQUENCE
{
    priority            INTEGER(0..15) OPTIONAL,
    emergency           BOOLEAN OPTIONAL,
    topologyReq        SEQUENCE OF TopologyRequest OPTIONAL,
    ...,
    iepscallind        BOOLEAN OPTIONAL,
    contextProp        SEQUENCE OF PropertyParm OPTIONAL,
    contextList        SEQUENCE OF ContextID OPTIONAL
}

```

```

-- When returning a contextList, the contextId in the ActionReply construct will
-- return the contextId from the associated ActionRequest.

```

```

ContextAttrAuditRequest ::= SEQUENCE
{
    topology           NULL OPTIONAL,
    emergency           NULL OPTIONAL,
    priority           NULL OPTIONAL,
    ...,
    iepscallind        NULL OPTIONAL,
    contextPropAud     SEQUENCE OF IndAudPropertyParm OPTIONAL,
    selectpriority     INTEGER(0..15) OPTIONAL,
    -- to select given priority
    selectemergency    BOOLEAN OPTIONAL,
    -- to select if emergency set/not set (T/F)
    selectiepscallind  BOOLEAN OPTIONAL,
    -- to select if IEPS set/not set (T/F)
    selectLogic        SelectLogic OPTIONAL -- default is AND
}

```

```

SelectLogic ::= CHOICE
{
    andAUDITSelect     NULL, -- all filter conditions satisfied
    orAUDITSelect      NULL, -- at least one filter condition satisfied
    ...
}

```

```

CommandRequest ::= SEQUENCE
{
    command            Command,
    optional           NULL OPTIONAL,
    wildcardReturn     NULL OPTIONAL,
    ...
}

```

```

Command ::= CHOICE
{
    addReq           AmmRequest,
    moveReq          AmmRequest,
    modReq           AmmRequest,
    -- Add, Move, Modify requests have the same parameters
    subtractReq      SubtractRequest,
    auditCapRequest  AuditRequest,
    auditValueRequest AuditRequest,
    notifyReq        NotifyRequest,
    serviceChangeReq ServiceChangeRequest,
    ...
}

CommandReply ::= CHOICE
{
    addReply          AmmsReply,
    moveReply         AmmsReply,
    modReply          AmmsReply,
    subtractReply     AmmsReply,
    -- Add, Move, Modify, Subtract replies have the same parameters
    auditCapReply     AuditReply,
    auditValueReply   AuditReply,
    notifyReply       NotifyReply,
    serviceChangeReply ServiceChangeReply,
    ...
}

TopologyRequest ::= SEQUENCE
{
    terminationFrom      TerminationID,
    terminationTo        TerminationID,
    topologyDirection    ENUMERATED
    {
        bothway(0),
        isolate(1),
        oneway(2)
    },
    ...,
    streamID              StreamID OPTIONAL,
    topologyDirectionExtension ENUMERATED
    {
        onewayexternal(0),
        onewayboth(1),
        ...
    }
}

AmmRequest ::= SEQUENCE
{
    terminationID        TerminationIDList,
    descriptors          SEQUENCE OF AmmDescriptor,
    -- At most one descriptor of each type (see AmmDescriptor)
    -- allowed in the sequence.
    ...
}

AmmDescriptor ::= CHOICE
{
    mediaDescriptor      MediaDescriptor,
    modemDescriptor      ModemDescriptor,
    muxDescriptor        MuxDescriptor,
    eventsDescriptor     EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
}

```

```

    signalsDescriptor      SignalsDescriptor,
    digitMapDescriptor     DigitMapDescriptor,
    auditDescriptor       AuditDescriptor,
    ...,
    statisticsDescriptor  StatisticsDescriptor
}

AmmsReply                ::= SEQUENCE
{
    terminationID         TerminationIDList,
    terminationAudit      TerminationAudit OPTIONAL,
    ...
}

SubtractRequest         ::= SEQUENCE
{
    terminationID         TerminationIDList,
    auditDescriptor       AuditDescriptor OPTIONAL,
    ...
}

AuditRequest            ::= SEQUENCE
{
    terminationID         TerminationID,
    auditDescriptor       AuditDescriptor,
    ...,
    terminationIDList     TerminationIDList OPTIONAL
}
-- terminationID shall contain the first termination in the
-- list when using the terminationIDList construct in AuditRequest

AuditReply              ::= CHOICE
{
    contextAuditResult    TerminationIDList,
    error                 ErrorDescriptor,
    auditResult           AuditResult,
    ...,
    auditResultTermList   TermListAuditResult
}

AuditResult             ::= SEQUENCE
{
    terminationID         TerminationID,
    terminationAuditResult TerminationAudit
}

TermListAuditResult     ::= SEQUENCE
{
    terminationIDList     TerminationIDList,
    terminationAuditResult TerminationAudit,
    ...
}

TerminationAudit        ::= SEQUENCE OF AuditReturnParameter

AuditReturnParameter    ::= CHOICE
{
    errorDescriptor       ErrorDescriptor,
    mediaDescriptor       MediaDescriptor,
    modemDescriptor       ModemDescriptor,
    muxDescriptor         MuxDescriptor,
    eventsDescriptor      EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor     SignalsDescriptor,
    digitMapDescriptor     DigitMapDescriptor,

```

```

    observedEventsDescriptor      ObservedEventsDescriptor,
    statisticsDescriptor          StatisticsDescriptor,
    packagesDescriptor            PackagesDescriptor,
    emptyDescriptors              AuditDescriptor,
    ...
}

AuditDescriptor ::= SEQUENCE
{
    auditToken                BIT STRING
    {
        muxToken(0),
        modemToken(1),
        mediaToken(2),
        eventsToken(3),
        signalsToken(4),
        digitMapToken(5),
        statsToken(6),
        observedEventsToken(7),
        packagesToken(8),
        eventBufferToken(9)
    } OPTIONAL,
    ...,
    auditPropertyToken        SEQUENCE OF IndAuditParameter OPTIONAL
}

IndAuditParameter ::= CHOICE
{
    indaudmediaDescriptor      IndAudMediaDescriptor,
    indaudeventsDescriptor     IndAudEventsDescriptor,
    indaudeventBufferDescriptor IndAudEventBufferDescriptor,
    indaudsignalsDescriptor    IndAudSignalsDescriptor,
    indauidigitMapDescriptor   IndAudDigitMapDescriptor,
    indaudstatisticsDescriptor  IndAudStatisticsDescriptor,
    indaudpackagesDescriptor   IndAudPackagesDescriptor,
    ...
}

IndAudMediaDescriptor ::= SEQUENCE
{
    termStateDescr            IndAudTerminationStateDescriptor OPTIONAL,
    streams                    CHOICE
    {
        oneStream              IndAudStreamParms,
        multiStream            SEQUENCE OF IndAudStreamDescriptor
    } OPTIONAL,
    ...
}

IndAudStreamDescriptor ::= SEQUENCE
{
    streamID                  StreamID,
    streamParms               IndAudStreamParms
}

IndAudStreamParms ::= SEQUENCE
{
    localControlDescriptor    IndAudLocalControlDescriptor OPTIONAL,
    localDescriptor           IndAudLocalRemoteDescriptor OPTIONAL,
    remoteDescriptor          IndAudLocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor      IndAudStatisticsDescriptor OPTIONAL
}

```

```

IndAudLocalControlDescriptor ::= SEQUENCE
{
    streamMode          NULL OPTIONAL,
    reserveValue        NULL OPTIONAL,
    reserveGroup        NULL OPTIONAL,
    propertyParms       SEQUENCE OF IndAudPropertyParm OPTIONAL,
    ...,
    streamModeSel       StreamMode OPTIONAL
}
-- must not have both streamMode and streamModeSel
-- if both are present only streamModeSel shall be honoured

IndAudPropertyParm ::= SEQUENCE
{
    name                PkgdName,
    ...,
    propertyParms       PropertyParm OPTIONAL
}

-- to select based on property values
-- AND/OR selection logic is specified at context level

IndAudLocalRemoteDescriptor ::= SEQUENCE
{
    propGroupID         INTEGER(0..65535) OPTIONAL,
    propGrps            IndAudPropertyGroup,
    ...
}

IndAudPropertyGroup ::= SEQUENCE OF IndAudPropertyParm

IndAudTerminationStateDescriptor ::= SEQUENCE
{
    propertyParms       SEQUENCE OF IndAudPropertyParm,
    eventBufferControl NULL OPTIONAL,
    serviceState        NULL OPTIONAL,
    ...,
    serviceStateSel     ServiceState OPTIONAL
}
-- must not have both serviceState and serviceStateSel
-- if both are present only serviceStateSel shall be honoured

IndAudEventsDescriptor ::= SEQUENCE
{
    requestID           RequestID OPTIONAL,
    pkgdName            PkgdName,
    streamID            StreamID OPTIONAL,
    ...
}

IndAudEventBufferDescriptor ::= SEQUENCE
{
    eventName           PkgdName,
    streamID            StreamID OPTIONAL,
    ...
}

IndAudSignalsDescriptor ::= CHOICE
{
    signal              IndAudSignal,
    seqSigList          IndAudSeqSigList,
    ...
}

```

```

IndAudSeqSigList      ::= SEQUENCE
{
    id                INTEGER(0..65535),
    signalList        IndAudSignal OPTIONAL
}

IndAudSignal          ::= SEQUENCE
{
    signalName        PkgdName,
    streamID          StreamID OPTIONAL,
    ...,
    signalRequestID   RequestID OPTIONAL
}

IndAudDigitMapDescriptor ::= SEQUENCE
{
    digitMapName      DigitMapName OPTIONAL
}

IndAudStatisticsDescriptor ::= SEQUENCE
{
    statName          PkgdName
}

IndAudPackagesDescriptor ::= SEQUENCE
{
    packageName       Name,
    packageVersion    INTEGER(0..99),
    ...
}

NotifyRequest         ::= SEQUENCE
{
    terminationID     TerminationIDList,
    observedEventsDescriptor ObservedEventsDescriptor,
    errorDescriptor   ErrorDescriptor OPTIONAL,
    ...
}

NotifyReply           ::= SEQUENCE
{
    terminationID     TerminationIDList,
    errorDescriptor   ErrorDescriptor OPTIONAL,
    ...
}

ObservedEventsDescriptor ::= SEQUENCE
{
    requestId         RequestID,
    observedEventLst  SEQUENCE OF ObservedEvent
}

ObservedEvent         ::= SEQUENCE
{
    eventName         EventName,
    streamID          StreamID OPTIONAL,
    eventParList      SEQUENCE OF EventParameter,
    timeNotation      TimeNotation OPTIONAL,
    ...
}

EventName             ::= PkgdName

EventParameter        ::= SEQUENCE
{
    eventParameterName Name,

```

```

    value                Value,
    -- For use of extraInfos see the comment related to PropertyParm
    extraInfo            CHOICE
    {
        relation         Relation,
        range             BOOLEAN,
        sublist          BOOLEAN
    } OPTIONAL,
    ...
}

ServiceChangeRequest ::= SEQUENCE
{
    terminationID        TerminationIDList,
    serviceChangeParms  ServiceChangeParm,
    ...
}

ServiceChangeReply ::= SEQUENCE
{
    terminationID        TerminationIDList,
    serviceChangeResult  ServiceChangeResult,
    ...
}

-- For ServiceChangeResult, no parameters are mandatory. Hence the
-- distinction between ServiceChangeParm and ServiceChangeResParm.
ServiceChangeResult ::= CHOICE
{
    errorDescriptor      ErrorDescriptor,
    serviceChangeResParms ServiceChangeResParm
}

WildcardField ::= OCTET STRING(SIZE(1))

TerminationID ::= SEQUENCE
{
    wildcard             SEQUENCE OF WildcardField,
    id                  OCTET STRING(SIZE(1..8)),
    ...
}

-- See A.1 for explanation of wildcarding mechanism.
-- TerminationID 0xFFFFFFFFFFFFFFFF indicates the Root Termination.

TerminationIDList ::= SEQUENCE OF TerminationID

MediaDescriptor ::= SEQUENCE
{
    termStateDescr      TerminationStateDescriptor OPTIONAL,
    streams             CHOICE
    {
        oneStream        StreamParms,
        multiStream      SEQUENCE OF StreamDescriptor
    } OPTIONAL,
    ...
}

StreamDescriptor ::= SEQUENCE
{
    streamID            StreamID,
    streamParms         StreamParms
}

```



```

StreamParms          ::= SEQUENCE
{
    localControlDescriptor    LocalControlDescriptor OPTIONAL,
    localDescriptor           LocalRemoteDescriptor OPTIONAL,
    remoteDescriptor          LocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor      StatisticsDescriptor OPTIONAL
}

LocalControlDescriptor ::= SEQUENCE
{
    streamMode                StreamMode OPTIONAL,
    reserveValue              BOOLEAN OPTIONAL,
    reserveGroup              BOOLEAN OPTIONAL,
    propertyParms             SEQUENCE OF PropertyParm,
    ...
}

StreamMode           ::= ENUMERATED
{
    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
    ...
}

-- In PropertyParm, value is a SEQUENCE OF octet string. When sent
-- by an MGC the interpretation is as follows:
-- empty sequence means CHOOSE
-- one element sequence specifies value
-- If the sublist field is not selected, a longer sequence means
-- "choose one of the values" (i.e., value1 OR value2 OR ...)
-- If the sublist field is selected,
-- a sequence with more than one element encodes the value of a
-- list-valued property (i.e., value1 AND value2 AND ...).
-- The relation field may only be selected if the value sequence
-- has length 1. It indicates that the MG has to choose a value
-- for the property. E.g., x > 3 (using the greaterThan
-- value for relation) instructs the MG to choose any value larger
-- than 3 for property x.
-- The range field may only be selected if the value sequence
-- has length 2. It indicates that the MG has to choose a value
-- in the range between the first octet in the value sequence and
-- the trailing octet in the value sequence, including the
-- boundary values.
-- When sent by the MG, only responses to an AuditCapability request
-- may contain multiple values, a range, or a relation field.

PropertyParm        ::= SEQUENCE
{
    name                    PkgdName,
    value                   SEQUENCE OF OCTET STRING,
    extraInfo              CHOICE
    {
        relation            Relation,
        range               BOOLEAN,
        sublist             BOOLEAN
    } OPTIONAL,
    ...
}

```

```

Name                ::= OCTET STRING(SIZE(2))

PkgdName            ::= OCTET STRING(SIZE(4))
-- represents Package Name (2 octets) plus property, event,
-- signal names or StatisticsID. (2 octets)
-- To wildcard a package use 0xFFFF for first two octets, CHOOSE
-- is not allowed. To reference native property tag specified in
-- Annex C, use 0x0000 as first two octets.
-- To wildcard a PropertyID, EventID, SignalID, or StatisticsID, use
-- 0xFFFF for last two octets, CHOOSE is not allowed.
-- Wildcarding of Package Name is permitted only if PropertyID,
-- EventID, SignalID, or StatisticsID are also wildcarded.

Relation            ::= ENUMERATED
{
    greaterThan(0),
    smallerThan(1),
    unequalTo(2),
    ...}

LocalRemoteDescriptor ::= SEQUENCE
{
    propGrps          SEQUENCE OF PropertyGroup,
    ...
}

PropertyGroup       ::= SEQUENCE OF PropertyParm

TerminationStateDescriptor ::= SEQUENCE
{
    propertyParms     SEQUENCE OF PropertyParm,
    eventBufferControl EventBufferControl OPTIONAL,
    serviceState      ServiceState OPTIONAL,
    ...
}

EventBufferControl ::= ENUMERATED
{
    off(0),
    lockStep(1),
    ...
}

ServiceState        ::= ENUMERATED
{
    test(0),
    outOfSvc(1),
    inSvc(2),
    ...
}

MuxDescriptor       ::= SEQUENCE
{
    muxType           MuxType,
    termList          SEQUENCE OF TerminationID,
    nonStandardData   NonStandardData OPTIONAL,
    ...
}

MuxType             ::= ENUMERATED
{
    h221(0),
    h223(1),
    h226(2),
    v76(3),

```

```

    ...,
    nx64k(4)
}

StreamID ::= INTEGER(0..65535) -- 16-bit unsigned integer

EventsDescriptor ::= SEQUENCE
{
    requestID RequestID OPTIONAL,
    -- RequestID must be present if eventList
    -- is non empty
    eventList SEQUENCE OF RequestedEvent,
    ...
}

RequestedEvent ::= SEQUENCE
{
    pkgdName PkgdName,
    streamID StreamID OPTIONAL,
    eventAction RequestedActions OPTIONAL,
    evParList SEQUENCE OF EventParameter,
    ...
}

RegulatedEmbeddedDescriptor ::= SEQUENCE
{
    secondEvent SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...
}

NotifyBehaviour ::= CHOICE
{
    notifyImmediate NULL,
    notifyRegulated RegulatedEmbeddedDescriptor,
    neverNotify NULL,
    ...
}

RequestedActions ::= SEQUENCE
{
    keepActive BOOLEAN OPTIONAL,
    eventDM EventDM OPTIONAL,
    secondEvent SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...,
    notifyBehaviour NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventDM ::= CHOICE
{
    digitMapName DigitMapName,
    digitMapValue DigitMapValue
}

SecondEventsDescriptor ::= SEQUENCE
{
    requestID RequestID OPTIONAL,
    eventList SEQUENCE OF SecondRequestedEvent,
    ...
}

```

```

SecondRequestedEvent ::= SEQUENCE
{
    pkgdName          PkgdName,
    streamID          StreamID OPTIONAL,
    eventAction       SecondRequestedActions OPTIONAL,
    evParList         SEQUENCE OF EventParameter,
    ...
}
SecondRequestedActions ::= SEQUENCE
{
    keepActive        BOOLEAN OPTIONAL,
    eventDM           EventDM OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...,
    notifyBehaviour   NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventBufferDescriptor ::= SEQUENCE OF EventSpec

EventSpec ::= SEQUENCE
{
    eventName         EventName,
    streamID          StreamID OPTIONAL,
    eventParList      SEQUENCE OF EventParameter,
    ...
}

SignalsDescriptor ::= SEQUENCE OF SignalRequest

SignalRequest ::= CHOICE
{
    signal            Signal,
    seqSigList        SeqSigList,
    ...
}

SeqSigList ::= SEQUENCE
{
    id                INTEGER(0..65535),
    signalList        SEQUENCE OF Signal
}

Signal ::= SEQUENCE
{
    signalName        SignalName,
    streamID          StreamID OPTIONAL,
    sigType           SignalType OPTIONAL,
    duration          INTEGER(0..65535) OPTIONAL,
    notifyCompletion   NotifyCompletion OPTIONAL,
    keepActive        BOOLEAN OPTIONAL,
    sigParList        SEQUENCE OF SigParameter,
    ...,
    direction         SignalDirection OPTIONAL,
    requestID         RequestID OPTIONAL,
    intersigDelay     INTEGER(0..65535) OPTIONAL
}

SignalType ::= ENUMERATED
{
    brief(0),
    onOff(1),
    timeOut(2),
    ...
}

```

```

}

SignalDirection      ::= ENUMERATED
{
    internal(0),
    external(1),
    both(2),
    ...
}

SignalName           ::= PkgdName

NotifyCompletion     ::= BIT STRING
{
    onTimeOut(0), onInterruptByEvent(1),
    onInterruptByNewSignalDescr(2), otherReason(3), onIteration(4)
}

SigParameter        ::= SEQUENCE
{
    sigParameterName  Name,
    value              Value,
    -- For use of extraInfo see the comment related to PropertyParm
    extraInfo         CHOICE
    {
        relation      Relation,
        range          BOOLEAN,
        sublist       BOOLEAN
    } OPTIONAL,
    ...
}

-- For an AuditCapReply with all events, the RequestID shall be ALL.
-- ALL is represented by 0xffffffff.
RequestID           ::= INTEGER(0..4294967295) -- 32-bit unsigned integer

ModemDescriptor     ::= SEQUENCE
{
    mt1                SEQUENCE OF ModemType,
    mpl                SEQUENCE OF PropertyParm,
    nonStandardData    NonStandardData OPTIONAL
}

ModemType           ::= ENUMERATED
{
    v18(0),
    v22(1),
    v22bis(2),
    v32(3),
    v32bis(4),
    v34(5),
    v90(6),
    v91(7),
    synchISDN(8),
    ...
}

DigitMapDescriptor ::= SEQUENCE
{
    digitMapName       DigitMapName OPTIONAL,
    digitMapValue      DigitMapValue OPTIONAL
}

DigitMapName        ::= Name

```

```

DigitMapValue ::= SEQUENCE
{
    startTimer          INTEGER(0..99) OPTIONAL,
    shortTimer          INTEGER(0..99) OPTIONAL,
    longTimer           INTEGER(0..99) OPTIONAL,
    digitMapBody        IA5String,
    -- Units are seconds for start, short and long timers, and hundreds
    -- of milliseconds for duration timer. Thus start, short, and long
    -- range from 1 to 99 seconds and duration from 100 ms to 9.9 s
    -- See A.3 for explanation of DigitMap syntax
    ...,
    durationTimer       INTEGER (0..99) OPTIONAL
}

ServiceChangeParm ::= SEQUENCE
{
    serviceChangeMethod ServiceChangeMethod,
    serviceChangeAddress ServiceChangeAddress OPTIONAL,
    serviceChangeVersion INTEGER(0..99) OPTIONAL,
    serviceChangeProfile ServiceChangeProfile OPTIONAL,
    serviceChangeReason Value,
    -- A serviceChangeReason consists of a numeric reason code
    -- and an optional text description.
    -- The serviceChangeReason shall be a string consisting of
    -- a decimal reason code, optionally followed by a single
    -- space character and a textual description string.
    -- This string is first BER-encoded as an IA5String.
    -- The result of this BER-encoding is then encoded as
    -- an ASN.1 OCTET STRING type, "double wrapping" the
    -- value as was done for package elements.
    serviceChangeDelay  INTEGER(0..4294967295) OPTIONAL,
    -- 32-bit unsigned integer
    serviceChangeMgcId  Mid OPTIONAL,
    timeStamp           TimeNotation OPTIONAL,
    nonStandardData     NonStandardData OPTIONAL,
    ...,
    serviceChangeInfo   AuditDescriptor OPTIONAL,
    serviceChangeIncompleteFlag NULL OPTIONAL
}

ServiceChangeAddress ::= CHOICE
{
    portNumber          INTEGER(0..65535), -- TCP/UDP port number
    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName          DomainName,
    deviceName          PathName,
    mtpAddress          OCTET STRING(SIZE(2..4)),
    ...
}

ServiceChangeResParm ::= SEQUENCE
{
    serviceChangeMgcId  Mid OPTIONAL,
    serviceChangeAddress ServiceChangeAddress OPTIONAL,
    serviceChangeVersion INTEGER(0..99) OPTIONAL,
    serviceChangeProfile ServiceChangeProfile OPTIONAL,
    timeStamp           TimeNotation OPTIONAL,
    ...
}

ServiceChangeMethod ::= ENUMERATED
{

```

```

    failover(0),
    forced(1),
    graceful(2),
    restart(3),
    disconnected(4),
    handOff(5),
    ...
}

ServiceChangeProfile ::= SEQUENCE
{
    profileName          IA5String(SIZE (1..67))
    -- 64 characters for name, 1 for "/", 2 for version to match ABNF
}

PackagesDescriptor ::= SEQUENCE OF PackagesItem

PackagesItem ::= SEQUENCE
{
    packageName          Name,
    packageVersion      INTEGER(0..99),
    ...
}

StatisticsDescriptor ::= SEQUENCE OF StatisticsParameter

StatisticsParameter ::= SEQUENCE
{
    statName            PkgdName,
    statValue           Value OPTIONAL
}
-- If statistic consists of a sub-lists there will be more than
-- one octetstring in statValue.

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

NonStandardIdentifier ::= CHOICE
{
    object                OBJECT IDENTIFIER,
    h221NonStandard       H221NonStandard,
    experimental          IA5String(SIZE(8)),
    -- first two characters should be "X-" or "X+"
    ...
}

H221NonStandard ::= SEQUENCE
{
    t35CountryCode1      INTEGER(0..255),
    t35CountryCode2      INTEGER(0..255),      -- country, as per T.35
    t35Extension         INTEGER(0..255),      -- assigned nationally
    manufacturerCode     INTEGER(0..65535),    -- assigned nationally
    ...
}

TimeNotation ::= SEQUENCE
{
    date                  IA5String(SIZE(8)), -- yyyyymmdd format
    time                  IA5String(SIZE(8)) -- hhmmssss format
    -- per ISO 8601:2004
}

```

Value ::= SEQUENCE OF OCTET STRING

END

A.3 Mapas de dígitos y nombres de trayectos

Desde el punto de vista sintáctico, los mapas de dígitos son cadenas sujetas a restricciones sintácticas. La sintaxis de mapas de dígitos válidos se especifica en ABNF (RFC 2234). La sintaxis de esta cláusula se incluye sólo para ilustrar el proceso. En caso de divergencias entre las dos, prevalece la definición de digitMap que figura en el anexo B.

```
digitMap          = (digitString / LWSP "(" LWSP digitStringList LWSP
                    ")" LWSP)

digitStringList  = digitString *(LWSP "|" LWSP digitString)

digitString      = 1*(digitStringElement)

digitStringElement = digitPosition [DOT]

digitPosition    = digitMapLetter / digitMapRange

digitMapRange    = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter      = *((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter   = DIGIT                ;Basic event symbols
                  / %x41-4B / %x61-6B ; a-k, A-K
                  / "L" / "S" / "T"    ;Inter-event timers
                  ; (long, short, start)
                  / "Z"                ;Long duration modifier

DOT              = %x2E                ; "."

SP               = %x20                ; space

HTAB            = %x09                ; horizontal tab

CR              = %x0D                ; Carriage return

LF              = %x0A                ; linefeed

LWSP            = *(WSP / COMMENT / EOL)

EOL             = (CR [LF] / LF)

WSP             = SP / HTAB           ; white space

SafeChar        = DIGIT / ALPHA / "+" / "-" / "&" /
                  "!" / "~" / "/" / "!" / "?" / "@" /
                  "^" / "`" / "~" / "*" / "$" / "\" /
                  "(" / ")" / "%" / "|" / "."

RestChar        = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
                  "<" / ">" / "="

DIGIT           = %x30-39             ; 0-9

ALPHA          = %x41-5A / %x61-7A   ; A-Z / a-z
```

Un nombre de trayecto también es una cadena sometida a restricciones sintácticas. La gramática ABNF que la define se ha tomado del anexo B.


```

; Total length of pathNAME must not exceed 64 chars.
pathname      = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$" )
               ["@" pathDomainName ]
; ABNF allows two or more consecutive "." although it is meaningless
; in a path domain name.
pathDomainName = (ALPHA / DIGIT / "*" ) *63(ALPHA / DIGIT / "-" / "*" / ".")
NAME           = ALPHA *63(ALPHA / DIGIT / "_")

```

Anexo B

Codificación textual del protocolo

B.1 Codificación de comodines

En la codificación textual del protocolo se determinan de forma arbitraria los TerminationID, pero se podrá aprovechar más el comodín "*" eligiendo el nombre cuidadosamente. El comodín establecerá una concordancia con todos los TerminationID que tienen los mismos caracteres precedentes y siguientes (si existen). Por ejemplo, si existieran los TerminationID R13/3/1, R13/3/2 y R13/3/3, el TerminationID R13/3/* concordaría con todos ellos. En ciertas circunstancias hay que hacer referencia a todas las terminaciones. Entonces basta con utilizar el TerminationID "*", que significa ALL. El TerminationID "\$" (CHOOSE) puede utilizarse para señalar a la MG que tiene que crear una terminación efímera o seleccionar una terminación física en reposo.

B.2 Especificación ABNF

La sintaxis de protocolo se presenta en ABNF de acuerdo con RFC 2234.

NOTA 1 – Esta especificación de sintaxis no abarca todas las restricciones para valores o introducción de elementos. Algunas restricciones adicionales se establecen en comentarios y otras restricciones aparecen en el texto de esta Recomendación. Estas restricciones adicionales forman parte del protocolo aunque no vengan impuestas por esta Recomendación.

NOTA 2 – La sintaxis depende del contexto. Por ejemplo, "Add" puede ser el AddToken o un NAME, según el contexto en que aparezca.

Todo lo escrito en ABNF y en codificación de texto es insensible al hecho de que utilicen mayúsculas o minúsculas. Esto incluye los TerminationID, los identificadores de mapa de dígitos (DigitMapID), etc. SPD es sensible a la utilización de mayúsculas o minúsculas según RFC 2327.

```

; NOTE - The ABNF in this section uses the VALUE construct (or lists of
; VALUE constructs) to encode various package element values (properties,
; signal parameters, etc.). The types of these values vary and are
; specified in the relevant package definition. Several such types are
; described in 12.2.
;
; The ABNF specification for VALUE allows a quotedString form or a
; collection of SafeChars. The encoding of package element values into
; ABNF VALUES is specified below. If a type's encoding allows characters
; other than SafeChars, the quotedString form must be used for all values
; of that type, even for specific values that consist only of SafeChars.
;
; String: A string must use the quotedString form of VALUE and can
; contain anything allowable in the quotedString form.
;
; Integer, Double, and Unsigned Integer: Decimal values can be encoded
; using characters 0-9. Hexadecimal values must be prefixed with '0x'
; and can use characters 0-9,a-f,A-F. An octal format is not supported.

```

```

; Negative integers start with '-' and must be Decimal. The SafeChar
; form of VALUE must be used.
;
; Character: A UTF-8 encoding of a single letter surrounded by double
; quotes.
;
; Enumeration: An enumeration must use the SafeChar form of VALUE
; and can contain anything allowable in the SafeChar form.
;
; Boolean: Boolean values are encoded as "on" and "off" and are
; case insensitive. The SafeChar form of VALUE must be used.
;
; Future types: Any defined types must fit within
; the ABNF specification of VALUE. Specifically, if a type's encoding
; allows characters other than SafeChars, the quotedString form must
; be used for all values of that type, even for specific values that
; consist only of SafeChars.
;
; Note that there is no way to use the double quote character within
; a value.
;
; Note that SDP disallows whitespace at the beginning of a line, Megaco
; ABNF allows whitespace before the beginning of the SDP in the
; Local/Remote descriptor. Parsers should accept whitespace between the
; LBRKT following the Local/Remote token and the beginning of the SDP.

megacoMessage          = LWSP [authenticationHeader SEP] message

authenticationHeader   = AuthToken EQUAL SecurityParmIndex COLON
                        SequenceNum COLON AuthData

SecurityParmIndex      = "0x" 8 (HEXDIG)
SequenceNum            = "0x" 8 (HEXDIG)
AuthData               = "0x" 24*64 (HEXDIG)

Message                = MegacopToken SLASH Version SEP mId SEP messageBody
; The version of the protocol defined here is equal to 3.

messageBody            = (errorDescriptor / transactionList)

transactionList        = 1*(transactionRequest / transactionReply /
                        transactionPending / transactionResponseAck /
                        segmentReply)
;Use of response acks is dependent on underlying transport

transactionPending     = PendingToken EQUAL TransactionID LBRKT RBRKT

transactionResponseAck = ResponseAckToken LBRKT transactionAck
                        *(COMMA transactionAck) RBRKT

transactionAck         = TransactionID / (TransactionID "-" TransactionID)

transactionRequest     = TransToken EQUAL TransactionID LBRKT
                        actionRequest *(COMMA actionRequest) RBRKT

actionRequest          = CtxToken EQUAL ContextID LBRKT ((contextRequest
                        [COMMA commandRequestList]) /
                        commandRequestList) RBRKT

contextRequest         = ((contextProperties [COMMA contextAudit])
                        / contextAudit)

contextProperties      = contextProperty *(COMMA contextProperty)

; at-most-once

```

```

; EmergencyOff to be used in MG to MGC direction only in H.248.1 V1 and V2
; EmergencyToken or EmergencyOffToken, but not both
contextProperty          = (topologyDescriptor / priority / EmergencyToken /
                           EmergencyOffToken / ieepsValue /
                           contextAttrDescriptor)

contextAudit             = ContextAuditToken LBRKT (contextAuditProperties
                           *(COMMA contextAuditProperties)) /
                           indAudcontextAttrDescriptor      RBRKT

; at-most-once except contextAuditSelector
contextAuditProperties   = (TopologyToken / EmergencyToken /
                           PriorityToken / IEPSToken/ pkgdName /
                           contextAuditSelector)

; at-most-once
contextAuditSelector     = priority / emergencyValue / ieepsValue /
                           contextAttrDescriptor / auditSelectLogic

auditSelectLogic        = [ AndAUDITselectToken / OrAUDITselectToken ]
; If empty, AND of selection conditions is assumed.

indAudcontextAttrDescriptor
                        = ContextAttrToken LBRKT contextAuditProperties
                           *(COMMA contextAuditProperties) RBRKT

; "O-" indicates an optional command
; "W-" indicates a wildcarded response to a command
commandRequestList      = ["O-"] ["W-"] commandRequest *(COMMA ["O-"]
                           ["W-"]commandRequest)

commandRequest           = (ammRequest / subtractRequest / auditRequest /
                           notifyRequest / serviceChangeRequest)

transactionReply         = ReplyToken EQUAL TransactionID [SLASH segmentNumber
                           [SLASH SegmentationCompleteToken]] LBRKT
                           [ImmAckRequiredToken COMMA]
                           (errorDescriptor / actionReplyList) RBRKT

segmentReply             = MessageSegmentToken EQUAL TransactionID SLASH
                           segmentNumber [SLASH SegmentationCompleteToken]

segmentNumber            = UINT16

actionReplyList          = actionReply *(COMMA actionReply)

actionReply              = CtxToken EQUAL ContextID [LBRKT (errorDescriptor /
                           commandReply /(commandReply COMMA
                           errorDescriptor)) RBRKT]

commandReply             = ((contextProperties [COMMA commandReplyList]) /
                           commandReplyList)

commandReplyList         = commandReplies *(COMMA commandReplies)

commandReplies           = (serviceChangeReply / auditReply / ammsReply /
                           notifyReply)

;Add, Move and Modify have the same request parameters
ammRequest               = (AddToken / MoveToken / ModifyToken) EQUAL
                           termIDList [LBRKT ammParameter *(COMMA
                           ammParameter) RBRKT]

```

```

;at-most-once
ammParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /
eventsDescriptor / signalsDescriptor /
digitMapDescriptor / eventBufferDescriptor /
auditDescriptor / statisticsDescriptor)

ammsReply = (AddToken / MoveToken / ModifyToken / SubtractToken)
EQUAL termIDList [LBRKT terminationAudit RBRKT]

subtractRequest = SubtractToken EQUAL termIDList [LBRKT
auditDescriptor RBRKT]

auditRequest = (AuditValueToken / AuditCapToken) EQUAL
termIDList LBRKT auditDescriptor RBRKT

auditReply = (AuditValueToken / AuditCapToken)
(contextTerminationAudit / auditOther)

auditOther = EQUAL termIDList [LBRKT terminationAudit RBRKT]

terminationAudit = auditReturnParameter *(COMMA auditReturnParameter)

contextTerminationAudit = EQUAL CtxToken (terminationIDList / LBRKT
errorDescriptor RBRKT)

auditReturnParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /
eventsDescriptor / signalsDescriptor /
digitMapDescriptor / observedEventsDescriptor /
eventBufferDescriptor / statisticsDescriptor /
packagesDescriptor / errorDescriptor /
auditReturnItem)

auditReturnItem = (MuxToken / ModemToken / MediaToken /
DigitMapToken / StatsToken / ObservedEventsToken /
PackagesToken)

auditDescriptor = AuditToken LBRKT [auditItem *(COMMA auditItem)]
RBRKT

notifyRequest = NotifyToken EQUAL termIDList LBRKT
(observedEventsDescriptor [COMMA errorDescriptor])
RBRKT

notifyReply = NotifyToken EQUAL termIDList [LBRKT errorDescriptor
RBRKT]

serviceChangeRequest = ServiceChangeToken EQUAL termIDList LBRKT
serviceChangeDescriptor RBRKT

serviceChangeReply = ServiceChangeToken EQUAL termIDList [LBRKT
(errorDescriptor / serviceChangeReplyDescriptor)
RBRKT]

errorDescriptor = ErrorToken EQUAL ErrorCode LBRKT [quotedString]
RBRKT

ErrorCode = 1*4(DIGIT) ; could be extended

TransactionID = UINT32

mId = ((domainAddress / domainName) [":" portNumber]) /
mtpAddress / deviceName

```

; ABNF allows two or more consecutive "." although it is meaningless

```

; in a domain name.
domainName          = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" / ".")
                    ">"

deviceName          = pathNAME

;The values 0x0, 0xFFFFFFFFE and 0xFFFFFFFFF are reserved.
;'-' is used for NULL context. '*' is ALL. '$' is CHOOSE.
ContextID           = (UINT32 / "*" / "-" / "$")

domainAddress       = "[" (IPv4address / IPv6address) "]"

;RFC 2373 contains the definition of IPv6 addresses.
IPv6address         = hexpart [":" IPv4address]

IPv4address         = V4hex DOT V4hex DOT V4hex DOT V4hex

V4hex               = 1*3(DIGIT) ; "0".."255"

; this production, while occurring in RFC 2373, is not referenced
; IPv6prefix        = hexpart SLASH 1*2DIGIT

hexpart             = hexseq ":@" [hexseq] / ":@" [hexseq] / hexseq

hexseq              = hex4 *(":" hex4)

hex4                = 1*4HEXDIG

portNumber          = UINT16

; Addressing structure of mtpAddress:
; 25 - 15           0
;   | PC           | NI |
;   24 - 14 bits   2 bits
; NOTE - 14 bits are defined for international use.
; Two national options exist where the point code is 16 or 24 bits.
; To octet align the mtpAddress the MSBs shall be encoded as 0s.
; An octet shall be represented by 2 hex digits.
mtpAddress          = MTPToken LBRKT 4*8 (HEXDIG) RBRKT

termIDList          = (TerminationID / LBRKT TerminationID 1*(COMMA
                    TerminationID) RBRKT)

terminationIDList   = LBRKT TerminationID *(COMMA TerminationID) RBRKT

; Total length of pathNAME must not exceed 64 chars.
pathNAME            = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$")
                    ["@" pathDomainName]

; ABNF allows two or more consecutive "." although it is meaningless
; in a path domain name.
pathDomainName      = (ALPHA / DIGIT / "*") *63(ALPHA / DIGIT / "-" /
                    "*" / ".")

; '*' is ALL. '$' is CHOOSE.
TerminationID       = "ROOT" / pathNAME / "$" / "*"

mediaDescriptor     = MediaToken LBRKT mediaParm *(COMMA mediaParm) RBRKT

; at-most one terminationStateDescriptor
; and either streamParm(s) or streamDescriptor(s) but not both
mediaParm           = (streamParm / streamDescriptor /
                    terminationStateDescriptor)

```

```

; at-most-once per item
streamParm          = (localDescriptor / remoteDescriptor /
                      localControlDescriptor / statisticsDescriptor)

streamDescriptor    = StreamToken EQUAL StreamID LBRKT streamParm *(COMMA
                      streamParm) RBRKT

localControlDescriptor = LocalControlToken LBRKT localParm *(COMMA localParm)
                      RBRKT

; at-most-once per item except for propertyParm
localParm           = (streamMode / propertyParm / reservedValueMode /
                      reservedGroupMode)

reservedValueMode   = ReservedValueToken EQUAL ("ON" / "OFF")

reservedGroupMode   = ReservedGroupToken EQUAL ("ON" / "OFF")

streamMode          = ModeToken EQUAL streamModes

streamModes         = (SendonlyToken / RecvonlyToken / SendrecvToken /
                      InactiveToken / LoopbackToken)

propertyParm        = pkgdName parmValue

; the (Safe)Char '$' means CHOOSE
; the (Safe)Char '*' means ALL
parmValue           = (EQUAL alternativeValue / INEQUAL VALUE)

alternativeValue     = (VALUE
                      / LBRKT VALUE *(COMMA VALUE) RBRKT
                      ; sublist (i.e., A AND B AND ...)
                      / LBRKT VALUE *(COMMA VALUE) RBRKT
                      ; alternatives (i.e., A OR B OR ...)
                      / LBRKT VALUE COLON VALUE RBRKT)
                      ; range

INEQUAL             = LWSP ">" / "<" / "#") LWSP ; '#' means "not equal"

LSBRKT              = LWSP "[" LWSP

RSBRKT              = LWSP "]" LWSP

; NOTE - The octet zero is not among the permitted characters in octet
; string. As the current definition is limited to SDP, and a zero octet
; would not be a legal character in SDP, this is not a concern.
localDescriptor     = LocalToken LBRKT octetString RBRKT

remoteDescriptor    = RemoteToken LBRKT octetString RBRKT

eventBufferDescriptor = EventBufferToken [LBRKT eventSpec*(COMMA eventSpec)
                      RBRKT]

eventSpec           = pkgdName [LBRKT eventSpecParameter *(COMMA
                      eventSpecParameter) RBRKT]

eventSpecParameter  = (eventStream / eventOther)

eventBufferControl   = BufferToken EQUAL eventBufferControlValue
eventBufferControlValue = ("OFF" / LockStepToken)

terminationStateDescriptor = TerminationStateToken LBRKT terminationStateParm
                      *(COMMA terminationStateParm) RBRKT

```

```

; at-most-once per item except for propertyParm
terminationStateParm    = (propertyParm / serviceStates / eventBufferControl)

serviceStates           = ServiceStatesToken EQUAL serviceStatesValue
serviceStatesValue      = (TestToken / OutOfSvcToken / InSvcToken)

muxDescriptor           = MuxToken EQUAL MuxType terminationIDList

MuxType                 = (H221Token / H223Token / H226Token / V76Token /
extensionParameter / Nx64kToken)

StreamID                = UINT16

pkgdName                = (PackageName SLASH ItemID);specific item
                        / (PackageName SLASH "*") ;all items in package
                        / ("*" SLASH "*")      ; all items supported by the MG

PackageName            = NAME

ItemID                  = NAME

eventsDescriptor        = EventsToken [EQUAL RequestID LBRKT requestedEvent
*(COMMA requestedEvent) RBRKT]

requestedEvent          = pkgdName [LBRKT eventParameter *(COMMA
eventParameter) RBRKT]

notifyRegulated         = NotifyRegulatedToken [LBRKT (embedWithSig/
embedNoSig) RBRKT]

notifyBehaviour         = NotifyImmediateToken / notifyRegulated /
NeverNotifyToken

; at-most-once each of KeepActiveToken, notifyBehaviour, eventDM,
; ResetEventsDescriptor and eventStream
; at most one of either embedWithSig or embedNoSig but not both
; KeepActiveToken and embedWithSig must not both be present
eventParameter         = (embedWithSig / embedNoSig / KeepActiveToken /
eventDM / eventStream / eventOther /
notifyBehaviour / ResetEventsDescriptorToken)

embedWithSig            = EmbedToken LBRKT signalsDescriptor [COMMA
embedFirst] RBRKT

embedNoSig              = EmbedToken LBRKT embedFirst RBRKT

; at-most-once of each
embedFirst              = EventsToken [EQUAL RequestID LBRKT
secondRequestedEvent *(COMMA secondRequestedEvent)
RBRKT]

secondRequestedEvent    = pkgdName [LBRKT secondEventParameter *(COMMA
secondEventParameter) RBRKT]

; at-most-once each of embedSig, KeepActiveToken, notifyBehaviour, eventDM
; ResetEventsDescriptor or eventStream
; KeepActiveToken and embedSig must not both be present
secondEventParameter   = (embedSig / KeepActiveToken / eventDM /
eventStream / eventOther / notifyBehaviour /
ResetEventsDescriptorToken)

embedSig                = EmbedToken LBRKT signalsDescriptor RBRKT

```

```

eventStream           = StreamToken EQUAL StreamID
eventOther            = eventParameterName parmValue
eventParameterName   = NAME
eventDM               = DigitMapToken EQUAL((digitMapName) / (LBRKT
digitMapValue RBRKT))
signalsDescriptor     = SignalsToken [LBRKT signalParm *(COMMA signalParm)
RBRKT]
signalParm            = signalList / signalRequest
signalRequest         = signalName [LBRKT sigParameter *(COMMA sigParameter)
RBRKT]
signalList            = SignalListToken EQUAL signalListId LBRKT
signalListParm *(COMMA signalListParm) RBRKT
signalListId         = UINT16
;exactly once signalType, at most once duration and every signal
;parameter
signalListParm       = signalRequest
signalName            = pkgdName
;at-most-once sigStream, at-most-once sigSignalType,
;at-most-once sigDuration, at-most-once sigDirection,
;at-most-once sigRequestID, at-most-once sigIntsigDelay
;every signalParameterName at most once
sigParameter         = sigStream / sigSignalType / sigDuration / sigOther /
notifyCompletion / KeepActiveToken /
sigDirection / sigRequestID / sigIntsigDelay
sigStream             = StreamToken EQUAL StreamID
sigOther              = sigParameterName parmValue
sigParameterName     = NAME
sigSignalType        = SignalTypeToken EQUAL signalType
signalType            = (OnOffToken / TimeOutToken / BriefToken)
sigDuration           = DurationToken EQUAL UINT16
sigDirection         = DirectionToken EQUAL direction
direction            = ExternalToken / InternalToken / BothToken
sigRequestID         = RequestIDToken EQUAL RequestID
sigIntsigDelay       = IntsigDelayToken EQUAL UINT16
notifyCompletion      = NotifyCompletionToken EQUAL (LBRKT
notificationReason *(COMMA notificationReason)
RBRKT)
notificationReason   = TimeOutToken / InterruptByEventToken /
InterruptByNewSignalsDescrToken / OtherReasonToken /
IterationToken

```



```

observedEventsDescriptor
    = ObservedEventsToken EQUAL RequestID LBRKT
      observedEvent *(COMMA observedEvent) RBRKT

; time per event, because it might be buffered
observedEvent
    = [TimeStamp LWSP COLON] LWSP pkgdName [LBRKT
      observedEventParameter *(COMMA
      observedEventParameter) RBRKT]

; at-most-once eventStream, every eventParameterName at most once
observedEventParameter = eventStream / eventOther

; For an AuditCapReply with all events, the RequestID should be ALL.
RequestID
    = UINT32 / "*"

modemDescriptor
    = ModemToken ((EQUAL modemType) / (LSBRKT modemType
      *(COMMA modemType) RSBKRT)) [LBRKT propertyParm
      *(COMMA propertyParm) RBRKT]

; at-most-once except for extensionParameter
modemType
    = V32bisToken / V22bisToken / V18Token / V22Token /
      V32Token / V34Token / V90Token / V91Token /
      SynchISDNToken / extensionParameter

digitMapDescriptor
    = DigitMapToken EQUAL ((LBRKT digitMapValue RBRKT) /
      (digitMapName [LBRKT digitMapValue RBRKT]))

digitMapName
    = NAME

digitMapValue
    = ["T" COLON Timer COMMA] ["S" COLON Timer COMMA] ["L"
      COLON Timer COMMA] ["Z" COLON Timer COMMA] digitMap

Timer
    = 1*2DIGIT
; Units are seconds for T, S, and L timers, and hundreds of
; milliseconds for Z timer. Thus T, S, and L range from 1 to 99
; seconds and Z from 100 ms to 9.9 s

digitMap
    = (digitString / LWSP "(" LWSP digitStringList LWSP
      ")" LWSP)

digitStringList
    = digitString *(LWSP "|" LWSP digitString)

digitString
    = 1*(digitStringElement)

digitStringElement
    = digitPosition [DOT]

digitPosition
    = digitMapLetter / digitMapRange

digitMapRange
    = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter
    = *((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter
    = DIGIT ;Basic event symbols
      / %x41-4B / %x61-6B ; a-k, A-K
      / "L" / "S" / "T" ;Inter-event timers
      ; (long, short, start)
      / "Z" ;Long duration modifier

; at-most-once, and DigitMapToken and PackagesToken are not allowed
; in AuditCapabilities command
auditItem
    = auditReturnItem / SignalsToken / EventBufferToken /
      EventsToken / indAudterminationAudit

indAudterminationAudit = indAudauditReturnParameter *(COMMA

```

```

indAudauditReturnParameter)

indAudauditReturnParameter
    = indAudmediaDescriptor / indAudeventsDescriptor /
      indAudsignalsDescriptor /
      indAuddigitMapDescriptor /
      indAudeventBufferDescriptor /
      indAudstatisticsDescriptor /
      indAudpackagesDescriptor

indAudmediaDescriptor    = MediaToken LBRKT indAudmediaParm *(COMMA
                          indAudmediaParm) RBRKT

; either streamParm or streamDescriptor but not both
indAudmediaParm         = indAudstreamParm / indAudstreamDescriptor /
                          indAudterminationStateDescriptor

; at-most-once
indAudstreamParm       = (indAudlocalControlDescriptor /
                          indAudstatisticsDescriptor /
                          indAudremoteDescriptor / indAudlocalDescriptor )

indAudremoteDescriptor = RemoteToken LBRKT octetString RBRKT
indAudlocalDescriptor  = LocalToken LBRKT octetString RBRKT

indAudstreamDescriptor = StreamToken EQUAL StreamID LBRKT indAudstreamParm
                          RBRKT

indAudlocalControlDescriptor
    = LocalControlToken LBRKT indAudlocalParm
      *(COMMA indAudlocalParm) RBRKT

; at-most-once per item
indAudlocalParm        = ModeToken [(EQUAL/INEQUAL) streamModes]/ pkgdName /
                          propertyParm / ReservedValueToken /
                          ReservedGroupToken

; propertyParm and streamModes are used only to specify audit selection
; criteria. AND/OR selection logic is specified at context level.

indAudterminationStateDescriptor
    = TerminationStateToken LBRKT
      indAudterminationStateParm RBRKT

; at-most-once per item
indAudterminationStateParm
    = pkgdName / propertyParm / ServiceStatesToken
      [(EQUAL/INEQUAL) serviceStatesValue ] / BufferToken

; When values are included a Select operation is implied.
; AND/OR logic is specified at context level.

indAudeventBufferDescriptor
    = EventBufferToken LBRKT indAudeventSpec RBRKT

indAudeventSpec        = pkgdName [LBRKT indAudeventSpecParameter RBRKT]

indAudeventSpecParameter= eventStream / eventParameterName

indAudeventsDescriptor = EventsToken [EQUAL RequestID] LBRKT
                          indAudrequestedEvent RBRKT

indAudrequestedEvent   = pkgdName

indAudsignalsDescriptor = SignalsToken LBRKT [indAudsignalParm] RBRKT

```

```

indAudsignalParm          = indAudsignalList / indAudsignalRequest
indAudsignalRequest      = signalName [LBRKT indAudsignalRequestParm
                                *(COMMA indAudsignalRequestParm) RBRKT]
indAudsignalRequestParm  = sigStream / sigRequestID
indAudsignalList        = SignalListToken EQUAL signalListId [LBRKT
                                indAudsignalListParm RBRKT]
indAudsignalListParm     = indAudsignalRequest
indAuddigitMapDescriptor= DigitMapToken EQUAL (digitMapName)
indAudstatisticsDescriptor
                        = StatsToken LBRKT pkgdName RBRKT
indAudpackagesDescriptor= PackagesToken LBRKT packagesItem RBRKT
serviceChangeDescriptor  = ServicesToken LBRKT serviceChangeParm *(COMMA
                                serviceChangeParm) RBRKT

; each parameter at-most-once, except auditItem
; at most one of either serviceChangeAddress or serviceChangeMgcId but
; not both
; serviceChangeMethod and serviceChangeReason are REQUIRED
serviceChangeParm        = serviceChangeMethod / serviceChangeReason /
                                serviceChangeDelay / serviceChangeAddress /
                                serviceChangeProfile / extension / TimeStamp /
                                serviceChangeMgcId / serviceChangeVersion /
                                ServiceChangeIncompleteToken / auditItem

serviceChangeReplyDescriptor
                        = ServicesToken LBRKT servChgReplyParm *(COMMA
                                servChgReplyParm) RBRKT

; at-most-once. Version is REQUIRED on first ServiceChange response
; at most one of either serviceChangeAddress or serviceChangeMgcId but
; not both
servChgReplyParm        = serviceChangeAddress / serviceChangeMgcId /
                                serviceChangeProfile / serviceChangeVersion /
                                TimeStamp

serviceChangeMethod      = MethodToken EQUAL (FailoverToken / ForcedToken /
                                GracefulToken / RestartToken / DisconnectedToken /
                                HandOffToken / extensionParameter)

; A serviceChangeReason consists of a numeric reason code
; and an optional text description.
; A serviceChangeReason must be encoded using the quotedString
; form of VALUE.
; The quotedString shall contain a decimal reason code,
; optionally followed by a single space character and a
; textual description string.
serviceChangeReason      = ReasonToken EQUAL VALUE

serviceChangeDelay       = DelayToken EQUAL UINT32

serviceChangeAddress     = ServiceChangeAddressToken EQUAL (mId / portNumber)

serviceChangeMgcId       = MgcIdToken EQUAL mId

serviceChangeProfile     = ProfileToken EQUAL NAME SLASH Version

```

```

serviceChangeVersion    = VersionToken EQUAL Version
extension               = extensionParameter parmValue
packagesDescriptor     = PackagesToken LBRKT packagesItem *(COMMA
                        packagesItem) RBRKT
Version                = 1*2(DIGIT)
packagesItem           = NAME "-" UINT16
TimeStamp              = Date "T" Time ; per ISO 8601:2004
; Date = yyyyymmdd
Date                   = 8(DIGIT)
; Time = hhmmsssss
Time                   = 8(DIGIT)
statisticsDescriptor   = StatsToken LBRKT statisticsParameter *(COMMA
                        statisticsParameter) RBRKT
;at-most-once per item
statisticsParameter    = pkgdName [EQUAL VALUE /
                        (LSBRKT VALUE *(COMMA VALUE) RSBKRT)]
topologyDescriptor     = TopologyToken LBRKT topologyTriple *(COMMA
                        topologyTriple) RBRKT
topologyTriple         = terminationA COMMA terminationB COMMA
                        topologyDirection [COMMA eventStream]
terminationA           = TerminationID
terminationB           = TerminationID
topologyDirection      = BothwayToken / IsolateToken / OnewayToken /
                        OnewayExternalToken / OnewayBothToken
priority               = PriorityToken EQUAL UINT16
iepsValue              = IEPSToken EQUAL ("ON" / "OFF")
emergencyValue         = EmergencyValueToken EQUAL (EmergencyToken /
                        EmergencyOffToken)
contextAttrDescriptor  = ContextAttrToken LBRKT (contextIdList /
                        propertyParm *(COMMA propertyParm)) RBRKT
; When using the contextIdList construct, the ContextID in the
; actionReply construct shall be the same as the ContextID in the
; associated actionRequest
contextIdList          = ContextListToken EQUAL LBRKT ContextID
                        *(COMMA ContextID) RBRKT
extensionParameter     = "X" ("- / "+) 1*6(ALPHA / DIGIT)
; octetString is used to describe SDP defined in RFC 2327.
; Caution should be taken if CRLF in RFC 2327 is used.
; To be safe, use EOL in this ABNF.
; Whenever "}" appears in SDP, it is escaped by "\", e.g., "\}"
octetString            = *(nonEscapeChar)
nonEscapeChar          = ("\}" / %x01-7C / %x7E-FF)

```

```

; NOTE - The double-quote character is not allowed in quotedString.
quotedString      = DQUOTE *(SafeChar / EOL / %x80-FF / RestChar / WSP)
                   DQUOTE

UINT16           = 1*5 (DIGIT) ; %x0-FFFF

UINT32           = 1*10 (DIGIT) ; %x0-FFFFFFFF

NAME             = ALPHA *63 (ALPHA / DIGIT / "_")

VALUE           = quotedString / 1*(SafeChar / %x80-FF)

SafeChar         = DIGIT / ALPHA / "+" / "-" / "&" /
                 "!" / " " / "/" / "'" / "?" / "@" /
                 "^" / "`" / "~" / "*" / "$" / "\" /
                 "(" / ")" / "%" / "|" / "."

EQUAL           = LWSP %x3D LWSP      ; "="

COLON           = %x3A                ; ":"

LBRKT           = LWSP %x7B LWSP      ; "{"

RBRKT           = LWSP %x7D LWSP      ; "}"

COMMA           = LWSP %x2C LWSP      ; ","

DOT             = %x2E                ; "."

SLASH           = %x2F                ; "/"

ALPHA           = %x41-5A / %x61-7A   ; A-Z / a-z

DIGIT           = %x30-39            ; 0-9

DQUOTE         = %x22                ; " (Double Quote)

HEXDIG         = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"

SP             = %x20                ; space

HTAB           = %x09                ; horizontal tab

CR             = %x0D                ; Carriage return

LF             = %x0A                ; linefeed

LWSP           = *(WSP / COMMENT / EOL)

EOL            = (CR [LF] / LF)

WSP            = SP / HTAB           ; white space

SEP            = (WSP / EOL / COMMENT) LWSP

COMMENT        = ";" *(SafeChar/ RestChar / WSP / %x22) EOL

RestChar       = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
                 "<" / ">" / "="

```

```

; New Tokens added to sigParameter must take the format of SPA*
; * may be of any form i.e., SPAM
; New Tokens added to eventParameter must take the form of EPA*

```

; * may be of any form i.e., EPAD

AddToken	= ("Add"	/ "A")
AndAUDITSelectToken	= ("ANDLgc")	
AuditToken	= ("Audit"	/ "AT")
AuditCapToken	= ("AuditCapability"	/ "AC")
AuditValueToken	= ("AuditValue"	/ "AV")
AuthToken	= ("Authentication"	/ "AU")
BothToken	= ("Both"	/ "B")
BothwayToken	= ("Bothway"	/ "BW")
BriefToken	= ("Brief"	/ "BR")
BufferToken	= ("Buffer"	/ "BF")
CtxToken	= ("Context"	/ "C")
ContextAuditToken	= ("ContextAudit"	/ "CA")
ContextAttrToken	= ("ContextAttr"	/ "CT")
ContextListToken	= ("ContextList"	/ "CLT")
DigitMapToken	= ("DigitMap"	/ "DM")
DirectionToken	= ("SPADirection"	/ "SPADI")
DisconnectedToken	= ("Disconnected"	/ "DC")
DelayToken	= ("Delay"	/ "DL")
DurationToken	= ("Duration"	/ "DR")
EmbedToken	= ("Embed"	/ "EM")
EmergencyToken	= ("Emergency"	/ "EG")
EmergencyOffToken	= ("EmergencyOff"	/ "EGO")
EmergencyValueToken	= ("EmergencyValue"	/ "EGV")
ErrorToken	= ("Error"	/ "ER")
EventBufferToken	= ("EventBuffer"	/ "EB")
EventsToken	= ("Events"	/ "E")
ExternalToken	= ("External"	/ "EX")
FailoverToken	= ("Failover"	/ "FL")
ForcedToken	= ("Forced"	/ "FO")
GracefulToken	= ("Graceful"	/ "GR")
H221Token	= ("H221")	
H223Token	= ("H223")	
H226Token	= ("H226")	
HandOffToken	= ("HandOff"	/ "HO")
IEPSToken	= ("IEPSCall"	/ "IEPS")
ImmAckRequiredToken	= ("ImmAckRequired"	/ "IA")
InactiveToken	= ("Inactive"	/ "IN")
InternalToken	= ("Internal"	/ "IT")
IntsigDelayToken	= ("Intersignal"	/ "SPAIS")
IsolateToken	= ("Isolate"	/ "IS")
InSvcToken	= ("InService"	/ "IV")
InterruptByEventToken	= ("IntByEvent"	/ "IBE")
InterruptByNewSignalsDescrToken		
	= ("IntBySigDescr"	/ "IBS")
IterationToken	= ("Iteration"	/ "IR")
KeepActiveToken	= ("KeepActive"	/ "KA")
LocalToken	= ("Local"	/ "L")
LocalControlToken	= ("LocalControl"	/ "O")
LockStepToken	= ("LockStep"	/ "SP")
LoopbackToken	= ("Loopback"	/ "LB")
MediaToken	= ("Media"	/ "M")
MegacopToken	= ("MEGACO"	/ "!")
MessageSegmentToken	= ("Segment"	/ "SM")
MethodToken	= ("Method"	/ "MT")
MgcIdToken	= ("MgcIdToTry"	/ "MG")
ModeToken	= ("Mode"	/ "MO")
ModifyToken	= ("Modify"	/ "MF")
ModemToken	= ("Modem"	/ "MD")
MoveToken	= ("Move"	/ "MV")
MTPToken	= ("MTP")	
MuxToken	= ("Mux"	/ "MX")
NeverNotifyToken	= ("NeverNotify"	/ "NBNN")

```

NotifyToken = ("Notify" / "N")
NotifyCompletionToken = ("NotifyCompletion" / "NC")
NotifyImmediateToken = ("ImmediateNotify" / "NBIN")
NotifyRegulatedToken = ("RegulatedNotify" / "NBRN")
Nx64kToken = ("Nx64Kservice" / "N64")
ObservedEventsToken = ("ObservedEvents" / "OE")
OnewayToken = ("Oneway" / "OW")
OnewayBothToken = ("OnewayBoth" / "OWB")
OnewayExternalToken = ("OnewayExternal" / "OWE")
OnOffToken = ("OnOff" / "OO")
OrAUDITselectToken = ("ORLgc")
OtherReasonToken = ("OtherReason" / "OR")
OutOfSvcToken = ("OutOfService" / "OS")
PackagesToken = ("Packages" / "PG")
PendingToken = ("Pending" / "PN")
PriorityToken = ("Priority" / "PR")
ProfileToken = ("Profile" / "PF")
ReasonToken = ("Reason" / "RE")
RecvonlyToken = ("ReceiveOnly" / "RC")
ReplyToken = ("Reply" / "P")
ResetEventsDescriptorToken = ("ResetEventsDescriptor" / "RSE")
RestartToken = ("Restart" / "RS")
RemoteToken = ("Remote" / "R")
RequestIDToken = ("SPAResultID" / "SPARQ")
ReservedGroupToken = ("ReservedGroup" / "RG")
ReservedValueToken = ("ReservedValue" / "RV")
SegmentationCompleteToken = ("END" / "&")
SendonlyToken = ("SendOnly" / "SO")
SendrecvToken = ("SendReceive" / "SR")
ServicesToken = ("Services" / "SV")
ServiceStatesToken = ("ServiceStates" / "SI")
ServiceChangeIncompleteToken = ("ServiceChangeInc" / "SIC")
ServiceChangeToken = ("ServiceChange" / "SC")
ServiceChangeAddressToken = ("ServiceChangeAddress" / "AD")
SignalListToken = ("SignalList" / "SL")
SignalsToken = ("Signals" / "SG")
SignalTypeToken = ("SignalType" / "SY")
StatsToken = ("Statistics" / "SA")
StreamToken = ("Stream" / "ST")
SubtractToken = ("Subtract" / "S")
SynchISDNToken = ("SynchISDN" / "SN")
TerminationStateToken = ("TerminationState" / "TS")
TestToken = ("Test" / "TE")
TimeOutToken = ("TimeOut" / "TO")
TopologyToken = ("Topology" / "TP")
TransToken = ("Transaction" / "T")
ResponseAckToken = ("TransactionResponseAck" / "K")
V18Token = ("V18")
V22Token = ("V22")
V22bisToken = ("V22b")
V32Token = ("V32")
V32bisToken = ("V32b")
V34Token = ("V34")
V76Token = ("V76")
V90Token = ("V90")
V91Token = ("V91")
VersionToken = ("Version" / "V")

```

B.3 Codificación de octetos en hexadecimal

La codificación de octetos en hexadecimal es un medio de representar una cadena de octetos como una cadena de dígitos hexadecimales, representando cada octeto por dos dígitos. Esta codificación de octetos debe utilizarse cuando se codifiquen cadenas de octetos en la versión textual del protocolo.

Para cada octeto, la secuencia de 8 bits se codifica como dos dígitos hexadecimales. El bit en la posición 0 (brevemente bit 0) es el primero que se transmite, y el bit en la posición 7 es el último.

La codificación de los bits 7-4 constituye el primer dígito hexadecimal, siendo el bit 7 el bit más significativo (MSB) y el bit 4 el menos significativo (LSB). Los bits 3-0 codificados constituyen el segundo dígito hexadecimal, siendo el bit 3 el MSB y el bit 0 el LSB.

Ejemplos:

Esquema de los bits en los octetos	Codificación hexadecimal
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

B.4 Secuencia de octetos en hexadecimal

Una secuencia de octetos en hexadecimal es un número par de dígitos hexadecimales terminados por un carácter <CR>.

Anexo C

Rótulos para propiedades de trenes de medios

Los parámetros para descriptores Local (local), Remote (distante) y LocalControl (de control local) se especifican como pares de valores de rótulo si se utiliza la codificación binaria para el protocolo. En este anexo se indican los nombres de las propiedades (Identificador de propiedad, PropertyID), los rótulos (Rótulo de propiedad), el tipo de la propiedad (Tipo) y los valores (Valor). Los valores presentados en el campo Valor cuando éste contiene referencias se considerarán "información". La referencia contiene los valores normativos. Si un campo de valor no contiene una referencia, los valores en ese campo pueden considerarse "normativos".

Si bien la referencia a las propiedades del anexo C se hace conforme a la estructura PackageID/PropertyID, este anexo no es en sí mismo un lote. Se considera que un rótulo del anexo C tiene PackageID 0x0000 para la codificación binaria y "anxc" para la textual. En el caso de codificación textual H.248.1, el anexo C se empleará únicamente si la propiedad requerida no está definida por un lote o no está representada por un SDP. No se permite anidar las propiedades de este anexo dentro de otras.

Los rótulos se indican como números hexadecimales en este anexo. Cuando se determina el valor de una propiedad, un controlador de pasarela de medios (MGC) puede subespecificar el valor de acuerdo con uno de los mecanismos especificados en 7.1.1.

El soporte de las propiedades descritas en este anexo o en cualquiera de sus subcláusulas es facultativo. Por ejemplo, es posible implementar sólo tres propiedades de C.3 y sólo 5 propiedades de C.8.

El valor de "enumeración" es el valor indicado entre paréntesis, por ejemplo, Send (0), Receive (1). Las propiedades en el anexo C con los tipos "N bits" o "M octetos" deben tratarse como cadenas de octetos cuando se codifica el protocolo. Las propiedades con "entero de N bits" se tratarán como enteros. Las "cadenas" se tratarán como una IA5string cuando se codifique el protocolo.

Si el valor de un tipo ocupa menos de un octeto, se almacenará en los bits de orden inferior de una cadena con un solo octeto.

C.1 Atributos generales de los medios

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
Medios	1001	Enumeración	Audio(0), Video(1), Data(2)
Modo de transmisión	1002	Enumeración	Send(0), Receive(1), Send&Receive(2)
Número de canales	1003	Entero sin signo	0-255
Velocidad de muestreo	1004	Entero sin signo	0-2 ³²
Velocidad binaria	1005	Entero	(0..4294967295) NOTA – Unidades de 100 bit/s.
ACodec	1006	Cadena de octetos	Tipo de códec de audio: Ref.: Rec. UIT-T Q.765.5 Los códecs ajenos al UIT-T están definidos por la correspondiente organización de normalización y tienen un identificador organizacional definido.
Samplepp	1007	Entero sin signo	Número máximo de muestras o tramas por paquete: 0..65535
Silencesupp	1008	Booleano	Supresión de silencio: Verdadero/Falso
Encrypttype	1009	Cadena de octetos	Ref.: Rec. UIT-T H.245
Encryptkey	100A	Tamaño de la cadena de octetos (0..65535)	Clave de criptación Ref.: Rec. UIT-T H.235.0
Echocanc	100B		No se utiliza. Véase E.13 para un ejemplo de posibles propiedades de control de eco.
Gain	100C		No se utiliza. Véase E.13 para una propiedad de ganancia disponible.
Jitterbuff	100D	Entero sin signo	Tamaño de la memoria tampón para fluctuación en ms: 0..65535
PropDelay	100E	Entero sin signo	Tiempo de propagación: 0..65535 Máximo tiempo de propagación, en milisegundos, para la conexión portadora entre dos pasarelas de medios. Dependerá de la tecnología del portador.

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
RTPpayload	100F	Entero	Tipo de cabida útil en perfil de protocolo en tiempo real (RTP) para conferencias de audio y vídeo con control mínimo. Ref.: RFC 1890
Ptime	1010	Entero	Tiempo de paquetización Establece la longitud de tiempo, en milisegundos, representada por los medios en un paquete. Ref.: RFC 2327 del IETF

C.2 Propiedades de los múltiplex

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
H222	2001	Cadena de octetos	H222LogicalChannelParameters Ref.: Rec. UIT-T H.245
H223	2002	Cadena de octetos	H223LogicalChannelParameters Ref.: Rec. UIT-T H.245
V76	2003	Cadena de octetos	V76LogicalChannelParameters Ref.: Rec. UIT-T H.245
H2250	2004	Cadena de octetos	H2250LogicalChannelParameters Ref.: Rec. UIT-T H.245

C.3 Propiedades generales de los portadores

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
Mediatx	3001	Enumeración	Tipo de transporte de medios TDM Circuit(0), ATM(1), FR(2), Ipv4(3), Ipv6(4), ...
BIR	3002	4 octetos	El valor depende de la tecnología de transporte
NSAP	3003	1-20 octetos	Véase NSAP. Ref.: Anexo A/X.213

C.4 Propiedades generales de ATM

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
AESA	4001	20 octetos	Dirección de sistema de extremo ATM
VPVC	4002	4 octetos: VPCI en los primeros dos octetos menos significativos, VCI en los dos octetos siguientes	VPCI/VCI Ref.: Rec. UIT-T Q.2931
SC	4003	Enumeración	Categoría de servicio: CBR(0), nrt-VBR1(1), nrt-VBR2(2), nrt-VBR3(3), rt-VBR1(4), rt-VBR2(5), rt-VBR3(6), UBR1(7), UBR2(8), ABR(9). Ref.: ATM Forum UNI 4.0
BCOB	4004	Entero de 5 bits	Clase portador de banda ancha Ref.: Rec. UIT-T Q.2961.2
BBTC	4005	Entero de 7 bits	Capacidad de transferencia de banda ancha Ref.: Rec. UIT-T Q.2961.1
ATC	4006	Enumeración	Capacidad de tráfico ATM I.371 DBR(0), SBR1(1), SBR2(2), SBR3(3), ABT/IT(4), ABT/DT(5), ABR(6) Ref.: Rec. UIT-T I.371
STC	4007	2 bits	Sensibilidad al recorte: Bits <u>2</u> <u>1</u> 0 0 no es sensible al recorte 0 1 sensible al recorte Ref.: Rec. UIT-T Q.2931
UPCC	4008	2 bits	Configuración de la conexión en el plano de usuario: Bits <u>2</u> <u>1</u> 0 0 punto a punto 0 1 punto a multipunto Ref.: Rec. UIT-T Q.2931
PCR0	4009	Entero de 24 bits	Velocidad de células de cresta (para CLP = 0) Ref.: Rec. UIT-T Q.2931
SCR0	400A	Entero de 24 bits	Velocidad de células sostenible (para CLP = 0) Ref.: Rec. UIT-T Q.2961.1
MBS0	400B	Entero de 24 bits	Tamaño máximo de ráfaga (para CLP = 0) Ref.: Rec. UIT-T Q.2961.1
PCR1	400C	Entero de 24 bits	Velocidad de células de cresta (para CLP = 0 + 1) Ref.: Rec. UIT-T Q.2931
SCR1	400D	Entero de 24 bits	Velocidad de células sostenible (para CLP = 0 + 1) Ref.: Rec. UIT-T Q.2961.1

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor	
MBS1	400E	Entero de 24 bits	Tamaño máximo de ráfaga (para CLP = 0 + 1) Ref.: Rec. UIT-T Q.2961.1	
BEI	400F	Booleano	Indicador de mejor esfuerzo El valor 1 indica que se debe incluir BEI en la señalización ATM; el valor 0 indica que no se debe incluir BEI en la señalización ATM. Ref.: ATM Forum UNI 4.0	
TI	4010	Booleano	Indicador de rotulado El valor 0 indica que no se permite el rotulado; el valor 1 indica que se ha solicitado el rotulado. Ref.: Rec. UIT-T Q.2961.1	
FD	4011	Booleano	Descarte de trama El valor 0 indica que no se permite el descarte de trama; el valor 1 indica que se permite el descarte de trama. Ref.: ATM Forum UNI 4.0	
A2PCDV	4012	Entero de 24 bits	CDV 2 puntos aceptable Ref.: Rec. UIT-T Q.2965.2	
C2PCDV	4013	Entero de 24 bits	CDV 2 puntos acumulativa Ref.: Rec. UIT-T Q.2965.2	
APPCDV	4014	Entero de 24 bits	CDV P-P aceptable Ref.: ATM Forum UNI 4.0	
CPPCDV	4015	Entero de 24 bits	CDV P-P acumulativa Ref.: ATM Forum UNI 4.0	
ACLR	4016	Entero de 8 bits	Relación de pérdida de células aceptable Ref.: Rec. UIT-T Q.2965.2, ATM Forum UNI 4.0	
MEETD	4017	Entero de 16 bits	Retardo máximo de tránsito de extremo a extremo Ref.: Rec. UIT-T Q.2965.2, ATM Forum UNI 4.0	
CEETD	4018	Entero de 16 bits	Retardo acumulativo de tránsito de extremo a extremo Ref.: Rec. UIT-T Q.2965.2, ATM Forum UNI 4.0	
QoSClass	4019	Entero 0-5	Clase de calidad de servicio (QoS)	
			Clase de QoS	Significado
			0	QoS por defecto asociada con la capacidad ATC definida en la Rec. UIT-T Q.2961.2
			1	Restringida
			2	Tolerante
			3	Binivel
			4	No limitada
			5	Restringida binivel
			Ref.: Rec. UIT-T Q.2965.1	

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
AALtype	401A	1 octeto	Tipo AAL Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 AAL para voz 0 0 0 0 0 0 0 1 AAL tipo 1 0 0 0 0 0 0 1 0 AAL tipo 2 0 0 0 0 0 0 1 1 AAL tipo 3/4 0 0 0 0 0 1 0 1 AAL tipo 5 0 0 0 1 0 0 0 0 AAL definida por el usuario Ref.: Rec. UIT-T Q.2931

C.5 Retransmisión de tramas

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
DLCI	5001	Entero sin signo	Identificador de conexión de enlace de datos
CID	5002	Entero sin signo	Identificador de subcanal
SID/Noiselevel	5003	Entero sin signo	Descriptor de inserción de silencio
Tipo cabida útil primaria	5004	Entero sin signo	Tipo cabida útil primaria Abarca FAX y códecs

C.6 IP

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
IPv4	6001	Dirección Ipv4 de 32 bits	Dirección Ipv4 Ref.: RFC del IETF 791
IPv6	6002	128 bits	Dirección IPv6 Ref.: RFC del IETF 2460
Port	6003	Entero sin signo	0..65535
Porttype	6004	Enumerado	TCP(0), UDP(1), SCTP(2)
RtcpbwRS	6005	Entero	El modificador de ancho de banda RS del RTCP indica el ancho de banda atribuido a los remitentes activos de datos (definidos por la especificación del RTP) Ref.: RFC del IETF 3556
RtcpbwRR	6006	Entero	El modificador de ancho de banda RR del RTCP indica el ancho de banda atribuido a otros participantes en la sesión RTP (es decir, los receptores) Ref.: RFC del IETF 3556

C.7 ATM AAL 2

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
AESA	7001	20 octetos	Dirección de punto extremo de servicio AAL 2, definida en la Recomendación a que se hace referencia. ESEA NSEA Ref.: Rec. UIT-T Q.2630.1
BIR	Véase C.3	4 octetos	Referencia generada por el usuario servido, definida en la Recomendación a que se hace referencia. SUGR Ref.: Rec. UIT-T Q.2630.1
ALC	7002	12 octetos	Características del enlace AAL 2, definidas en la Recomendación a que se hace referencia. Velocidad binaria máxima/promedio de la CPS-SDU Tamaño máximo/promedio de la CPS-SDU Ref.: Rec. UIT-T Q.2630.1
SSCS	7003	I.366.2: Audio (8 octetos); Multivelocidad (3 octetos), o I.366.1: con SAR (14 octetos); sin SAR (7 octetos)	Información de subcapa de convergencia específica del servicio, definida en: – Rec. UIT-T Q.2630.1, y utilizada en: – Rec. UIT-T I.366.2: audio/multivelocidad; – Rec. UIT-T I.366.1: SAR asegurada/no asegurada. Ref.: Recs. UIT-T Q.2630.1, I.366.1 e I.366.2
SUT	7004	1..254 octetos	Parámetro de transporte del usuario servido, definido en la Recomendación a que se hace referencia. Ref.: Rec. UIT-T Q.2630.1
TCI	7005	Booleano	Indicador de conexión de prueba, definido en la Recomendación a que se hace referencia. Ref.: Rec. UIT-T Q.2630.1
Timer_CU	7006	Entero de 32 bits	Temporizador CU Milisegundos durante los cuales se retiene una célula parcialmente llena, antes de enviarla.
MaxCPSSDU	7007	Entero de 8 bits	Máxima unidad de datos del servicio de subcapa de la parte común Ref.: Rec. UIT-T Q.2630.1
CID	7008	8 bits	Identificador de subcanal: 0-255 Ref.: Rec. UIT-T I.363.2

C.8 ATM AAL 1

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
BIR	Véase el cuadro en C.3	4-29 octetos	Transporte de identificador genérico (GIT, <i>generic identifier transport</i>) Ref.: Rec. UIT-T Q.2941.1
AAL1ST	8001	1 octeto	Subtipo AAL 1 Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 nulo 0 0 0 0 0 0 0 1 transporte de señal en banda vocal a 64 kbit/s 0 0 0 0 0 0 1 0 transporte de circuito 0 0 0 0 0 1 0 0 transporte de señal audio de alta calidad 0 0 0 0 0 1 0 1 transporte de señal vídeo Ref.: Rec. UIT-T Q.2931
CBRR	8002	1 octeto	Velocidad binaria constante (CBR) Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 1 64 kbit/s 0 0 0 0 0 1 0 0 1544 kbit/s 0 0 0 0 0 1 0 1 6312 kbit/s 0 0 0 0 0 1 1 0 32 064 kbit/s 0 0 0 0 0 1 1 1 44 736 kbit/s 0 0 0 0 1 0 0 0 97 728 kbit/s 0 0 0 1 0 0 0 0 2048 kbit/s 0 0 0 1 0 0 0 1 8448 kbit/s 0 0 0 1 0 0 1 0 34 368 kbit/s 0 0 0 1 0 0 1 1 139 264 kbit/s 0 1 0 0 0 0 0 0 n × 64 kbit/s 0 1 0 0 0 0 0 1 n × 8 kbit/s Ref.: Rec. UIT-T Q.2931
MULT	Véase el cuadro en C.9		Multiplicador, o n × 64k/8k/300 Ref.: Rec. UIT-T Q.2931
SCRI	8003	1 octeto	Método de recuperación de la frecuencia del reloj fuente Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 nulo 0 0 0 0 0 0 0 1 SRTS 0 0 0 0 0 0 1 0 ACM Ref.: Rec. UIT-T Q.2931
ECM	8004	1 octeto	Método de corrección de errores Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 nulo 0 0 0 0 0 0 0 1 FEC – Loss 0 0 0 0 0 0 1 0 FEC – Delay Ref.: Rec. UIT-T Q.2931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
SDTB	8005	Entero de 16 bits	Tamaño de bloque para la transferencia de datos estructurados Tamaño de bloque del servicio SDT CBR. Ref.: Rec. UIT-T I.363.1
PFCI	8006	Entero de 8 bits	Identificador de células parcialmente llenas: 1-47 Ref.: Rec. UIT-T I.363.1

C.9 Capacidades portadoras

Los asientos del cuadro que remiten a la Rec. UIT-T Q.931 se refieren a la codificación del elemento de información capacidad portadora Q.931, y no al elemento de información de capa baja.

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
TMR	9001	1 octeto	Medio de transmisión requerido (Rec. UIT-T Q.763) Bits <u>8 7 6 5 4 3 2 1</u> 0 0 0 0 0 0 0 0 habla 0 0 0 0 0 0 0 1 reserva 0 0 0 0 0 0 1 0 64 kbit/s sin restricciones 0 0 0 0 0 0 1 1 audio 3,1 kHz 0 0 0 0 0 1 0 0 reservado para habla (servicio 2)/64 kbit/s sin restricciones (servicio 1) alternados 0 0 0 0 0 1 0 1 reservado para 64 kbit/s sin restricciones (servicio 1)/habla (servicio 2) alternados 0 0 0 0 0 1 1 0 64 kbit/s preferido 0 0 0 0 0 1 1 1 2 × 64 kbit/s sin restricciones 0 0 0 0 1 0 0 0 384 kbit/s sin restricciones 0 0 0 0 1 0 0 1 1536 kbit/s sin restricciones 0 0 0 0 1 0 1 0 1920 kbit/s sin restricciones 0 0 0 0 1 0 1 1 a reserva 0 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 3 × 64 kbit/s sin restricciones 0 0 0 1 0 0 0 1 4 × 64 kbit/s sin restricciones 0 0 0 1 0 0 1 0 5 × 64 kbit/s sin restricciones 0 0 0 1 0 0 1 1 reserva 0 0 0 1 0 1 0 0 7 × 64 kbit/s sin restricciones 0 0 0 1 0 1 0 1 8 × 64 kbit/s sin restricciones 0 0 0 1 0 1 1 0 9 × 64 kbit/s sin restricciones 0 0 0 1 0 1 1 1 10 × 64 kbit/s sin restricciones 0 0 0 1 1 0 0 0 11 × 64 kbit/s sin restricciones 0 0 0 1 1 0 0 1 12 × 64 kbit/s sin restricciones 0 0 0 1 1 0 1 0 13 × 64 kbit/s sin restricciones 0 0 0 1 1 0 1 1 14 × 64 kbit/s sin restricciones 0 0 0 1 1 1 0 0 15 × 64 kbit/s sin restricciones 0 0 0 1 1 1 0 1 16 × 64 kbit/s sin restricciones 0 0 0 1 1 1 1 0 17 × 64 kbit/s sin restricciones 0 0 0 1 1 1 1 1 18 × 64 kbit/s sin restricciones 0 0 1 0 0 0 0 0 19 × 64 kbit/s sin restricciones 0 0 1 0 0 0 0 1 20 × 64 kbit/s sin restricciones 0 0 1 0 0 0 1 0 21 × 64 kbit/s sin restricciones

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
			0 0 1 0 0 0 1 1 22 × 64 kbit/s sin restricciones 0 0 1 0 0 1 0 0 23 × 64 kbit/s sin restricciones 0 0 1 0 0 1 0 1 reserva 0 0 1 0 0 1 1 0 25 × 64 kbit/s sin restricciones 0 0 1 0 0 1 1 1 26 × 64 kbit/s sin restricciones 0 0 1 0 1 0 0 0 27 × 64 kbit/s sin restricciones 0 0 1 0 1 0 0 1 28 × 64 kbit/s sin restricciones 0 0 1 0 1 0 1 0 29 × 64 kbit/s sin restricciones 0 0 1 0 1 0 1 1 a reserva 1 1 1 1 1 1 1 1 Ref.: Rec. UIT-T Q.763
TMRSR	9002	1 octeto	Subvelocidad del medio de transmisión requerido 0 no especificada 1 8 kbit/s 2 16 kbit/s 3 32 kbit/s
Contcheck	9003	Booleano	Prueba de continuidad 0 no se requiere en este circuito 1 se requiere en este circuito Ref.: Rec. UIT-T Q.763
ITC	9004	5 bits	Capacidad de transferencia de información Bits <u>5 4 3 2 1</u> 0 0 0 0 habla 0 1 0 0 información digital sin restricciones 0 1 0 1 información digital con restricciones 1 0 0 0 audio 3,1 kHz 1 0 0 1 información digital sin restricciones con tonos/anuncios 1 1 0 0 vídeo Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.763
TransMode	9005	2 bits	Modo transferencia Bits <u>2 1</u> 0 0 Modo circuito 1 0 Modo paquete Ref.: Rec. UIT-T Q.931
TransRate	9006	5 bits	Velocidad de transferencia Bits <u>5 4 3 2 1</u> 0 0 0 0 Se utilizará este código para llamadas en modo paquete 1 0 0 0 64 kbit/s 1 0 0 1 2 × 64 kbit/s 1 0 0 1 384 kbit/s 1 0 1 0 1536 kbit/s 1 0 1 1 1920 kbit/s 1 1 0 0 Multivelocidad (velocidad de base: 64 kbit/s)

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
			Ref.: Rec. UIT-T Q.931
MULT	9007	7 bits	Multiplicador de velocidad: Cualquier valor de 2 a n (número máximo de canales B) Ref.: Rec. UIT-T Q.931
layer1prot	9008	5 bits	Protocolo de capa 1 de información de usuario Bits <u>5 4 3 2 1</u> 0 0 0 0 1 Adaptación de velocidad V.110 y X.30 0 0 0 1 0 Ley μ de la Rec. UIT-T G.711 0 0 0 1 1 Ley A de la Rec. UIT-T G.711 0 0 1 0 0 MICDA a 32 kbit/s de la Rec. UIT-T G.726 y Rec. UIT-T I.460 0 0 1 0 1 Recs. UIT-T H.221 y H.242 0 0 1 1 0 Recs. UIT-T H.223 y H.245 0 0 1 1 1 Adaptación de velocidad no normalizada por el UIT-T 0 1 0 0 0 Adaptación de velocidad V.120. 0 1 0 0 1 Adaptación de velocidad X.31, inserción de banderas HDLC Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.931
syncasync	9009	Booleano	Síncrono/asíncrono 0 Datos síncronos 1 Datos asíncronos Ref.: Rec. UIT-T Q.931
negotiation	900A	Booleano	Negociación 0 Es posible la negociación dentro de banda 1 No es posible la negociación dentro de banda Ref.: Rec. UIT-T Q.931
Userrate	900B	5 bits	Velocidad de usuario Bits <u>5 4 3 2 1</u> 0 0 0 0 0 La velocidad se indica por bits E especificados en la Rec. UIT-T I.460 o puede negociarse dentro de banda 0 0 0 0 1 0,6 kbit/s Recs. UIT-T V.6 y X.1 0 0 0 1 0 1,2 kbit/s Rec. UIT-T V.6 0 0 0 1 1 2,4 kbit/s Recs. UIT-T V.6 y X.1 0 0 1 0 0 3,6 kbit/s Rec. UIT-T V.6 0 0 1 0 1 4,8 kbit/s Recs. UIT-T V.6 y X.1 0 0 1 1 0 7,2 kbit/s Rec. UIT-T V.6 0 0 1 1 1 8 kbit/s Rec. UIT-T I.460 0 1 0 0 0 9,6 kbit/s Recs. UIT-T V.6 y X.1 0 1 0 0 1 14,4 kbit/s Rec. UIT-T V.6 0 1 0 1 0 16 kbit/s Rec. UIT-T I.460 0 1 0 1 1 19,2 kbit/s Rec. UIT-T V.6 0 1 1 0 0 32 kbit/s Rec. UIT-T I.460 0 1 1 0 1 38,4 kbit/s Rec. UIT-T V.110 0 1 1 1 0 48 kbit/s Recs. UIT-T V.6 y X.1 0 1 1 1 1 56 kbit/s Rec. UIT-T V.6

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
			1 0 0 1 0 57,6 kbit/s Rec. UIT-T V.14 ampliada 1 0 0 1 1 28,8 kbit/s Rec. UIT-T V.110 1 0 1 0 0 24 kbit/s Rec. UIT-T V.110 1 0 1 0 1 0,1345 kbit/s Rec. UIT-T X.1 1 0 1 1 0 0,100 kbit/s Rec. UIT-T X.1 1 0 1 1 1 0,075/1,2 kbit/s Recs. UIT-T V.6 y X.1 1 1 0 0 0 1,2/0,075 kbit/s Recs. UIT-T V.6 y X.1 1 1 0 0 1 0,050 kbit/s Recs. UIT-T V.6 y X.1 1 1 0 1 0 0,075 kbit/s Recs. UIT-T V.6 y X.1 1 1 0 1 1 0,110 kbit/s Recs. UIT-T V.6 y X.1 1 1 1 0 0 0,150 kbit/s Recs. UIT-T V.6 y X.1 1 1 1 0 1 0,200 kbit/s Recs. UIT-T V.6 y X.1 1 1 1 1 0 0,300 kbit/s Recs. UIT-T V.6 y X.1 1 1 1 1 1 12 kbit/s Rec. UIT-T V.6 Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.931
INTRATE	900C	2 bits	Velocidad intermedia Bits <u>2 1</u> 0 0 No se utiliza 0 1 8 kbit/s 1 0 16 kbit/s 1 1 32 kbit/s Ref.: Rec. UIT-T Q.931
nictx	900D	Booleano	Reloj independiente de la red (NIC, <i>network independent clock</i>) en transmisión 0 No se requiere el envío de datos con reloj independiente de la red 1 Se requiere el envío de datos con reloj independiente de la red Ref.: Rec. UIT-T Q.931
nicrx	900E	Booleano	Reloj independiente de la red (NIC) en recepción 0 No puede aceptar datos con reloj independiente de la red (es decir, el emisor no soporta este procedimiento facultativo) 1 Puede aceptar datos con reloj independiente de la red (es decir, el emisor soporta este procedimiento facultativo) Ref.: Rec. UIT-T Q.931
flowconttx	900F	Booleano	Control de flujo en transmisión (Tx) 0 No se requiere el envío de datos con mecanismo de control de flujo 1 Se requiere el envío de datos con mecanismo de control de flujo Ref.: Rec. UIT-T Q.931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
flowcontrx	9010	Booleano	Control de flujo en recepción (Rx) 0 No puede aceptar datos con mecanismo de control de flujo (es decir, el emisor no soporta este mecanismo facultativo) 1 Puede aceptar datos con mecanismo de control de flujo (es decir, el emisor soporta este mecanismo facultativo) Ref.: Rec. UIT-T Q.931
rateadapthdr	9011	Booleano	Con/sin encabezamiento de adaptación de velocidad 0 Encabezamiento de adaptación de velocidad no incluido 1 Encabezamiento de adaptación de velocidad incluido Ref.: Rec. UIT-T Q.931
multiframe	9012	Booleano	Soporte del establecimiento de múltiples tramas en el enlace de datos 0 No soporta el establecimiento de múltiples tramas. Sólo se permiten tramas UI 1 Soporta el establecimiento de múltiples tramas. Ref.: Rec. UIT-T Q.931
OPMODE	9013	Booleano	Modo de funcionamiento 0 Modo de funcionamiento transparente a los bits 1 Modo de funcionamiento sensible al protocolo Ref.: Rec. UIT-T Q.931
llidnegot	9014	Booleano	Negociación del identificador de enlace lógico 0 Valor por defecto, LLI = 256 solamente 1 Negociación de la totalidad del protocolo Ref.: Rec. UIT-T Q.931
assign	9015	Booleano	Asignante/asignado 0 El originador del mensaje es "asignado por defecto" 1 El originador del mensaje es "sólo asignante" Ref.: Rec. UIT-T Q.931
inbandneg	9016	Booleano	Negociación dentro de banda/fuera de banda 0 La negociación se efectúa con mensajes INFORMACIÓN DE USUARIO mediante una conexión de señalización temporal 1 La negociación se efectúa dentro de banda mediante el enlace lógico cero Ref.: Rec. UIT-T Q.931
stopbits	9017	2 bits	Número de bits de parada Bits <u>2 1</u> 0 0 No se utilizan 0 1 1 bit 1 0 1,5 bits 1 1 2 bits Ref.: Rec. UIT-T Q.931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
databits	9018	2 bits	Número de bits de datos, sin contar el bit de paridad, si está presente Bits <u>2 1</u> 0 0 No se utilizan 0 1 5 bits 1 0 7 bits 1 1 8 bits Ref.: Rec. UIT-T Q.931
parity	9019	3 bits	Información de paridad Bits <u>3 2 1</u> 0 0 0 Impar 0 1 0 Par 0 1 1 Ninguna 1 0 0 Obligado a 0 1 0 1 Obligado a 1 Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.931
duplexmode	901A	Booleano	Modo dúplex 0 Semidúplex 1 Dúplex Ref.: Rec. UIT-T Q.931
modem	901B	6 bits	Tipo de módem Bits <u>6 5 4 3 2 1</u> 0 0 0 0 0 a Uso nacional 0 0 0 1 0 1 0 1 0 0 0 1 Rec. UIT-T V.21 0 1 0 0 1 0 Rec. UIT-T V.22 0 1 0 0 1 1 Rec. UIT-T V.22 <i>bis</i> 0 1 0 1 0 0 Rec. UIT-T V.23 0 1 0 1 0 1 Rec. UIT-T V.26 0 1 1 0 0 1 Rec. UIT-T V.26 <i>bis</i> 0 1 0 1 1 1 Rec. UIT-T V.26 <i>ter</i> 0 1 1 0 0 0 Rec. UIT-T V.27 0 1 1 0 0 1 Rec. UIT-T V.27 <i>bis</i> 0 1 1 0 1 0 Rec. UIT-T V.27 <i>ter</i> 0 1 1 0 1 1 Rec. UIT-T V.29 0 1 1 1 0 1 Rec. UIT-T V.32 0 1 1 1 1 0 Rec. UIT-T V.34 1 0 0 0 0 a Uso nacional 1 0 1 1 1 1 1 1 0 0 0 0 a Especificado por el usuario 1 1 1 1 1 1 Ref.: Rec. UIT-T Q.931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
layer2prot	901C	5 bits	<p>Protocolo de capa 2 de información de usuario</p> <p>Bits</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Rec. UIT-T Q.921/I.441</p> <p>0 0 1 1 0 Rec. UIT-T X.25, capa enlace</p> <p>0 1 1 0 0 Control de enlace lógico LAN (ISO/CEI 8802-2)</p> <p>Todos los demás valores están reservados.</p> <p>Ref.: Rec. UIT-T Q.931</p>
layer3prot	901D	5 bits	<p>Protocolo de capa 3 de información de usuario</p> <p>Bits</p> <p><u>5 4 3 2 1</u></p> <p>0 0 0 1 0 Rec. UIT-T Q.931</p> <p>0 0 1 1 0 Rec. UIT-T X.25, capa paquetes</p> <p>0 1 0 1 1 ISO/CEI TR 9577 (Identificación de protocolo en la capa de red)</p> <p>Todos los demás valores están reservados.</p> <p>Ref.: Rec. UIT-T Q.931</p>
addlayer3prot	901E	Octeto	<p>Protocolo de capa 3 de información de usuario adicional</p> <p>Bits Bits</p> <p><u>4 3 2 1</u> <u>4 3 2 1</u></p> <p>1 1 0 0 1 1 0 0 Protocolo Internet (RFC 791) (ISO/CEI TR 9577)</p> <p>1 1 0 0 1 1 1 1 Protocolo punto a punto (RFC 1661)</p> <p>Ref.: Rec. UIT-T Q.931</p>
DialledN	901F	30 octetos	Número marcado
DiallingN	9020	30 octetos	Número desde el que se marca
ECHOCI	9021		No utilizado. Véase E.13 para un ejemplo de posibles propiedades de compensadores de eco
NCI	9022	1 octeto	<p>Indicadores de la naturaleza de la conexión</p> <p>Bits</p> <p><u>2 1</u> <i>Indicador de satélite</i></p> <p>0 0 ningún circuito de satélite en la conexión</p> <p>0 1 un circuito de satélite en la conexión</p> <p>1 0 dos circuitos de satélite en la conexión</p> <p>1 1 reserva</p> <p>Bits</p> <p><u>4 3</u> <i>Indicador de prueba de continuidad</i></p> <p>0 0 no se requiere prueba de continuidad</p> <p>0 1 se requiere prueba de continuidad en este circuito</p> <p>1 0 prueba de continuidad efectuada en un circuito anterior</p> <p>1 1 reserva</p> <p>Bit</p> <p><u>5</u> <i>Indicador de dispositivo de control de eco</i></p> <p>0 dispositivo de control de eco de salida no incluido</p> <p>1 dispositivo de control de eco de salida incluido</p> <p>Bits</p> <p><u>8 7 6</u> Reserva</p> <p>Ref.: Rec. UIT-T Q.763</p>

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
USI	9023	Cadena de octetos	Información de servicio de usuario Ref.: 3.57/Q.763

C.10 Propiedades de AAL 5

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
FMSDU	A001	Entero de 32 bits	Tamaño máximo de la unidad de datos de servicio (SDU) de la subcapa de convergencia de la parte común (CPCS) en el sentido de ida: Tamaño máximo de la SDU de la CPCS enviado en el sentido del usuario llamante al usuario llamado. Ref.: Rec. UIT-T Q.2931
BMSDU	A002	Entero de 32 bits	Tamaño máximo de la unidad de datos de servicio (SDU) de la subcapa de convergencia de la parte común (CPCS) en el sentido de retorno: Tamaño máximo de la SDU de la CPCS enviado en el sentido del usuario llamado al usuario llamante. Ref.: Rec. UIT-T Q.2931
SSCS	Véase el cuadro en C.7	Véase el cuadro en C.7	Véase el cuadro en C.7 Valores adicionales: VPI/VCI

C.11 Equivalentes del SDP

Los equivalentes del SDP están sujetos a las excepciones SDP indicadas en 7.1.8 para la codificación textual del protocolo. Se permite, por ejemplo, utilizar el comodín CHOOSE en el sentido del MGC a la MG sin importar cuál sea la codificación del protocolo (binaria o textual).

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
SDP_V	B001	Cadena	Versión del protocolo Ref.: RFC 2327
SDP_O	B002	Cadena	ID de propietario/creador y de sesión Ref.: RFC 2327
SDP_S	B003	Cadena	Nombre de sesión Ref.: RFC 2327
SDP_I	B004	Cadena	Identificador de sesión Ref.: RFC 2327
SDP_U	B005	Cadena	URI de descriptor Ref.: RFC 2327
SDC_E	B006	Cadena	Dirección de correo electrónico Ref.: RFC 2327
SDP_P	B007	Cadena	Número de teléfono Ref.: RFC 2327

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
SDP_C	B008	Cadena	Información de conexión Ref.: RFC 2327
SDP_B	B009	Cadena	Información de anchura de banda Ref.: RFC 2327
SDP_Z	B00A	Cadena	Ajuste de zona de tiempo Ref.: RFC 2327
SDP_K	B00B	Cadena	Clave de criptación Ref.: RFC 2327
SDP_A	B00C	Cadena	Cero o más atributos de sesión Ref.: RFC 2327
SDP_T	B00D	Cadena	Tiempo de sesión activa Ref.: RFC 2327
SDP_R	B00E	Cadena	Cero o más repeticiones Ref.: RFC 2327
SDP_M	B00F	Cadena	Tipo de medio, puerto, transporte y formato Ref.: RFC 2327

C.12 H.245

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
OLC	C001	Cadena de octetos	Valor de la estructura de H.245 OpenLogicalChannel Ref.: Rec. UIT-T H.245
OLCack	C002	Cadena de octetos	Valor de la estructura de H.245 OpenLogicalChannelAck Ref.: Rec. UIT-T H.245
OLCcnf	C003	Cadena de octetos	Valor de la estructura de H.245 OpenLogicalChannelConfirm Ref.: Rec. UIT-T H.245
OLCrej	C004	Cadena de octetos	Valor de la estructura de H.245 OpenLogicalChannelReject Ref.: Rec. UIT-T H.245
CLC	C005	Cadena de octetos	Valor de la estructura de H.245 CloseLogicalChannel Ref.: Rec. UIT-T H.245
CLCack	C006	Cadena de octetos	Valor de la estructura de H.245 CloseLogicalChannelAck Ref.: Rec. UIT-T H.245
LNC	C007	Entero	Valor del número de canal local de H.245, 0-65535. Ref.: Rec. UIT-T H.245

Anexo D

Transporte por IP

D.1 Transporte por IP/UDP con empleo de entramado en el nivel de la aplicación (ALF, *application level framing*)

Los mensajes de protocolo definidos en esta Recomendación pueden transmitirse por UDP. Cuando el par no ha proporcionado un puerto (véase 7.2.8), las instrucciones deben enviarse al número de puerto por defecto: 2944 para la operación de codificación textual, o 2945 para la operación de codificación binaria. Las respuestas deben enviarse a la dirección y puerto desde el cual fueron enviadas las instrucciones correspondientes.

El entramado ALF es un conjunto de técnicas que permite a una aplicación, no a una pila, influir sobre la manera en que se envían mensajes al otro lado. Una técnica ALF típica consiste en permitir que una aplicación cambie el orden en que se envían los mensajes cuando hay una cola, después de haberlos puesto en cola. No hay una especificación formal de ALF. Los procedimientos descritos en D.1 contienen un conjunto mínimo sugerido de comportamientos ALF.

Los implementadores que utilizan IP/UDP con ALF deben tener presente las restricciones de la MTU sobre el tamaño de mensaje máximo.

D.1.1 Configuración de la función "una vez como máximo"

Puede haber pérdidas en los mensajes que se transmiten por el protocolo UDP. Si no se recibe una respuesta en tiempo oportuno, se repiten las instrucciones. La mayoría de las instrucciones no son idempotentes. El estado de la MG sería imprevisible si, por ejemplo, se ejecutaran varias veces unas instrucciones Añadir. Por tal razón, los procedimientos de transmisión proporcionarán una funcionalidad "una vez como máximo".

Las entidades de protocolo por conservan normalmente en memoria una lista de las respuestas que envían para transacciones recientes y una lista de las transacciones que están en curso en cada momento. El identificador de transacción (TransactionID) de cada mensaje entrante se compara con los identificadores de transacción de las respuestas recientes enviadas al mismo MID. Si se encuentra una concordancia, la entidad no ejecuta la transacción sino que, simplemente, repite la respuesta. Si no se encuentra una concordancia, el mensaje se comparará con la lista de transacciones en curso en cada momento. Si se encuentra una concordancia en esa lista, lo que indica que una transacción ha sido duplicada, la entidad no ejecuta la transacción (véase D.1.4 para los procedimientos de envío de TransactionPending).

El procedimiento utiliza un temporizador con un largo periodo de temporización, que en lo sucesivo se designará LONG-TIMER. Este temporizador deberá fijarse a un valor mayor que la duración máxima de una transacción, para lo cual se habrá de tener en cuenta el número máximo de repeticiones, el valor máximo del temporizador de repetición y el retardo máximo de propagación de un paquete en la red. Un valor sugerido es 30 segundos.

Las copias de las respuestas pueden suprimirse después de LONG-TIMER segundos desde la emisión de la respuesta o cuando la entidad recibe la confirmación de que se ha recibido la respuesta, mediante el "parámetro acuse de recibo de respuesta". En el caso de las transacciones de las que se acusa recibo por medio de este parámetro, la entidad conservará una copia del identificador de transacción durante LONG-TIMER segundos a partir del instante en que se emitió la respuesta, para poder detectar y hacer caso omiso de las copias duplicadas de la petición de la transacción que pudieran haber sido producidas por la red.

D.1.2 Identificadores de transacciones y toma de contacto tridireccional

D.1.2.1 Identificadores de transacciones (TransactionID)

Los identificadores de transacción son números enteros de 32 bits. Un controlador de pasarela de medios puede optar por utilizar un determinado espacio de número para cada MG sobre la que ejerce control, o utilizar el mismo espacio de número para todas las MG que pertenezcan a uno u otro grupo. Los MGC pueden repartir, entre varios procedimientos independientes, la carga del ejercicio del control sobre una MG grande. Estos procesos compartirán el mismo espacio de número de transacción. Hay varias formas de implementar esta compartición, por ejemplo mediante la asignación centralizada de identificadores de transacción, o la previa asignación de gamas no superpuestas de identificadores a diferentes procesos. Las implementaciones garantizarán que se asignarán identificadores de transacción unívocos a todas las transacciones que tengan su origen en un MGC lógico (MID idéntico). Para detectar las transacciones duplicadas, las MG no tienen más que examinar el identificador de transacción y el MID.

D.1.2.2 Toma de contacto tridireccional

El parámetro acuse de recibo de respuesta de transacción puede encontrarse en cualquier mensaje. Este parámetro transporta un conjunto de "gamas de identificadores de transacción confirmadas". Las entidades pueden optar por suprimir las copias de las respuestas de las transacciones cuyo identificador esté comprendido en las "gamas de identificadores de transacción confirmadas" recibidas en los mensajes de respuesta de transacción. Dichas entidades deben descartar ulteriores instrucciones sin más tratamiento cuando el identificador de transacción esté comprendido en esas gamas.

Los valores de las "gamas de identificadores de transacción confirmadas" no se utilizarán si han transcurrido más de LONG-TIMER segundos desde que la MG envió su última respuesta al MGC en cuestión, o cuando una MG reanuda el servicio. En esta situación, las transacciones deberán aceptarse y procesarse sin ninguna comprobación del identificador de transacción.

Los mensajes que transportan el parámetro "acuse de recibo de respuesta de transacción" (TransactionResponseAcknowledgement) pueden transmitirse en cualquier orden. La entidad retendrá las "gamas de identificadores de transacción confirmadas" recibidas durante LONG-TIMER segundos.

En la codificación binaria, si sólo aparece `firstAck` en un acuse de recibo de respuesta (véase A.2), se acusa recibo de una sola transacción. Si aparecen `firstAck` y `lastAck`, se acusa recibo de la gama de transacciones desde `firstAck` hasta `lastAck`. En la codificación textual, se utiliza un guión horizontal para indicar que se acusa recibo de una gama de transacciones (véase B.2).

D.1.3 Cálculo de los temporizadores de retransmisión

Incumbe a la entidad solicitante proporcionar periodos de temporización adecuados para todas las transacciones pendientes, y volver a intentar las transacciones cuando han expirado esos periodos de temporización. Además, cuando no se acusa recibo de transacciones repetidas, incumbe a la entidad solicitante procurarse servicios redundantes y/o liberar conexiones existentes o en curso de establecimiento.

En la presente especificación se ha evitado, intencionalmente, especificar valores para los temporizadores de retransmisión. Estos valores dependen típicamente de la red. Normalmente, para el cálculo de los temporizadores de retransmisión se debe medir el tiempo que transcurre entre el envío de una instrucción y el retorno de una respuesta. En la implementación se hará lo necesario para que el algoritmo que calcula la temporización de la retransmisión reduzca con variaciones exponenciales el tiempo de retransmisión para cada retransmisión o repetición después de la primera.

NOTA – Una posibilidad es la utilización del algoritmo implementado en TCP-IP, que emplea dos variables:

- El retardo promedio de acuse de recibo (AAD, *average acknowledgement delay*) calculado como un promedio alisado exponencialmente de los retardos observados.
- La desviación promedio (ADEV, *average deviation*) calculada como un promedio alisado exponencialmente del valor absoluto de la diferencia entre el retardo observado y el promedio actual. El temporizador de retransmisión, en TCP, se fija a la suma del retardo promedio más N veces la desviación promedio. Se debe, sin embargo, limitar el valor máximo del temporizador para el protocolo definido en esta Recomendación, para garantizar que las pasarela no recibirán ningún paquete repetido después de LONG-TIMER segundos. Un valor máximo sugerido es 4 segundos.

Después de una retransmisión, la entidad debería :

- multiplicar por 2 el valor estimado del retardo promedio, AAD.
- calcular un valor aleatorio uniformemente distribuido entre 0,5 AAD y AAD.
- fijar el temporizador de retransmisión a la suma de ese valor aleatorio y N veces la desviación promedio.

Este procedimiento tiene un doble efecto. Por el hecho de incluir un componente que aumenta exponencialmente, reduce automáticamente la velocidad de transmisión del tren de mensajes en caso de congestión. Por el hecho de incluir un componente aleatorio, elimina la sincronización que podría haber entre las notificaciones ocasionadas por un mismo evento externo.

D.1.4 Respuestas provisionales

La ejecución de algunas transacciones necesita mucho tiempo. Los largos tiempos de ejecución pueden influir en el procedimiento de retransmisión basado en temporizadores: puede resultar un número irregular de retransmisiones, o los valores de temporizador pueden ser demasiado largos para ser eficaces. Si una entidad puede determinar que una transacción requerirá un largo tiempo de ejecución, puede enviar una respuesta provisional "transacción pendiente". Debería enviar esta respuesta si recibe la repetición de una transacción que todavía se está ejecutando.

Las entidades que reciben una respuesta provisional de transacción pendiente conmutarán a un temporizador de repetición diferente para peticiones de repetición. Al recibir una respuesta final después de respuestas provisionales, se enviará una confirmación inmediata y a partir de ahí se utilizarán temporizadores de repetición normales. Una entidad que envía una respuesta provisional, incluirá el campo `immAckRequired` en la respuesta final que sigue, indicando que se espera una confirmación inmediata. No se tendrá en cuenta un aviso de transacción pendiente recibido después de una contestación.

D.1.5 Repetición de peticiones, respuestas y acuses de recibo

El protocolo se organiza como un conjunto de transacciones, cada una de las cuales se compone de una petición y una respuesta (acuse de recibo). Los mensajes de protocolo, que se transportan por UDP, pueden sufrir pérdidas. Si no se recibe una respuesta oportunamente, las transacciones se repiten. Las entidades conservan normalmente en memoria una lista de las respuestas que han enviado para transacciones recientes, es decir una lista de todas las respuestas que han enviado en los últimos LONG-TIMER segundos, y una lista de las transacciones que se están ejecutando en ese momento.

Se utiliza el mecanismo de repetición para la protección contra tres tipos de errores posibles:

- Errores de transacción cuando, por ejemplo, se pierde un paquete debido a ruido en la línea o congestión en una cola.
- Fallo de un componente, por ejemplo, cuando no está disponible una interfaz con una entidad.
- Fallo de una entidad, por ejemplo, cuando no está disponible toda una entidad.

Las entidades deben poder obtener, a partir de los datos históricos recogidos, una estimación de la tasa de pérdida de paquetes como consecuencia de errores de transmisión. En un sistema debidamente configurado, esta tasa de pérdida debe mantenerse lo más baja posible, por lo general inferior al 1%. Si un controlador de pasarela de medios o una pasarela de medios tiene que repetir un mensaje cierto número de veces, es lógico pensar que se está en presencia de algo más que un error de transmisión. Por ejemplo, dada una tasa de pérdida del 1%, la probabilidad de que fracasen cinco intentos consecutivos de transmisión es de 1×10^{-11} , evento que debe ocurrir con una frecuencia menor que una vez cada 10 días en una pasarela de medios que procesa 1000 transacciones por segundo. (En realidad, el número de repeticiones que se considere excesivo dependerá de la tasa de pérdida de paquetes resultante.) Debe señalarse que el "umbral de sospecha", que se designará por "Max1", normalmente es inferior al "umbral de desconexión", que se debe fijar a un valor mayor.

Un algoritmo clásico de retransmisión simplemente contaría el número de repeticiones sucesivas, y llegaría a la conclusión de que la asociación se ha interrumpido cuando el paquete se hubiera retransmitido un número de veces considerado excesivo (típicamente entre 7 y 11 veces). A fin de tener en cuenta la posibilidad de un "cambio-en-caso-de-fallo" en curso o no detectado, se modifica el algoritmo clásico de modo que, si la pasarela de medios recibe un mensaje ServiceChange válido que anuncia un cambio-en-caso-de-fallo, comenzará a transmitir las instrucciones pendientes a ese nuevo MGC. Las respuestas a instrucciones se siguen transmitiendo a la dirección fuente de las respectivas instrucciones.

Con el fin de adaptar automáticamente a la carga de la red, en esta Recomendación se especifican temporizadores que aumentan exponencialmente. Si el temporizador inicial se fija a 200 milisegundos, la pérdida de una quinta retransmisión se detectará después de aproximadamente 6 segundos. Éste probablemente sea un tiempo de espera aceptable para detectar un cambio-en-caso-de-fallo. Las repeticiones deben continuar después de transcurrido ese periodo, no sólo para resolver quizás un problema de la conectividad transitoria, sino también para dar un poco más de tiempo para la ejecución de un cambio-en-caso-de-fallo (probablemente sea aceptable un periodo total de 30 segundos).

Sin embargo, es importante limitar el periodo máximo de retransmisiones. Antes de cualquier retransmisión se debería verificar que el tiempo transcurrido desde el envío del datagrama inicial no es superior a T-MAX. Si ha transcurrido un periodo de tiempo mayor que T-MAX, la MG supone que el MGC ha fallado, y comienza su proceso de recuperación como se describe en 11.5. Si la MG intenta de nuevo conectarse al MGC actual, utilizará un ServiceChange con ServiceChangeMethod puesto en desconectado de tal manera que el MGC tendrá conocimiento de que la MG ha perdido una o más transacciones. El valor T-MAX está relacionado con el valor LONG-TIMER: el valor LONG-TIMER se obtiene sumando a T-MAX el tiempo máximo de propagación en la red.

D.2 Transporte por TCP

Los mensajes de protocolo definidos en esta Recomendación pueden transmitirse por TCP. Si el otro lado no especifica ningún puerto (véase 7.2.8), las instrucciones deben enviarse al puerto por defecto. El protocolo definido tiene mensajes como la unidad de transferencia, mientras que TCP es un protocolo orientado a trenes. TPKT, de acuerdo con RFC 1006 se utilizará para delinear mensajes dentro del tren TCP.

En un protocolo orientado a las transacciones pueden perderse peticiones o respuestas de transacciones. Por tanto, se recomienda que las entidades que usen el transporte TCP implementen temporizadores a nivel de aplicación para cada petición y cada respuesta, similares a los especificados para el entramado a nivel de aplicación por UDP.

D.2.1 Función "una vez como máximo"

Los mensajes transportados por TCP no sufren pérdidas, pero en las implementaciones reales se puede observar la pérdida de peticiones de transacción o de contestaciones a éstas. En ausencia de una respuesta en tiempo debido, se repiten las instrucciones. La mayor parte de las instrucciones no son idempotentes. El estado de la MG sería imprevisible, si por ejemplo, se ejecutaran varias veces instrucciones Añadir.

Para evitar tales pérdidas, se recomienda que las entidades sigan los procedimientos de D.1.1.

D.2.2 Identificadores de transacción y toma de contacto tridireccional

Por las mismas razones, es posible que las contestaciones a las transacciones se pierdan incluso cuando se utiliza un protocolo de entrega fiable como TCP. Se recomienda que las entidades sigan los procedimientos de D.1.2.2.

D.2.3 Cálculo de los temporizadores de retransmisión

Un sistema de entrega fiable reduce al mínimo los casos de pérdida de una petición de transacción o de una contestación a la transacción. Por consiguiente, sólo se requieren mecanismos de temporizador simples. Normalmente no es necesario utilizar algoritmos de reducción exponencial pero podrían emplearse en un MGC, por ejemplo, donde ya era necesario el código correspondiente, porque los MGC tienen que implementar ALF/UDP así como TCP.

D.2.4 Respuestas provisionales

Al igual que en el caso del UDP, la ejecución de algunas transacciones necesita mucho tiempo. Si una entidad puede determinar que una transacción requerirá un largo tiempo de ejecución, puede enviar una respuesta provisional, "transacción pendiente". Debería enviar estas respuestas si recibe una repetición de una transacción que todavía se está ejecutando.

Las entidades que reciben una respuesta provisional de transacción pendiente conmutarán a un temporizador de repetición más largo para esa transacción.

Las entidades retendrán transacciones y contestaciones hasta que sean confirmadas. Debe seguirse el procedimiento básico de D.1.4, aunque debe bastar con simples valores de temporizador. No es necesario enviar una confirmación inmediata tras la recepción de una respuesta final.

D.2.5 Ordenación de instrucciones

El protocolo TCP proporciona una entrega ordenada de las transacciones. No se requieren procedimientos especiales. Debe señalarse que ALF/UDP permite a la entidad emisora modificar su comportamiento en caso de congestión y, en particular, podría registrar transacciones cuando se encuentra una situación de congestión. TCP no podría obtener los mismos resultados.

Anexo E

Lotes básicos

Este anexo contiene definiciones de algunos paquetes que se utilizan con la presente Recomendación.

E.1 Genérico

Nombre de lote: Genérico

PackageID: g (0x0001)

Descripción: Lote genérico para los elementos frecuentes

Versión: 2

Extiende: Ninguno

E.1.1 Propiedades

Ninguna.

E.1.2 Eventos

E.1.2.1 Causa

Nombre de evento: Causa

EventID: Causa (0x0001)

Descripción: Evento de error genérico

Parámetros del descriptor de eventos (EventsDescriptor): Ninguno

Parámetros del descriptor de eventos observados (ObservedEventsDescriptor):

Causa general

Nombre de parámetro: Causa General

ParameterID: Generalcause (0x0001)

Descripción: Este parámetro agrupa los fallos en seis grupos, sobre los cuales el MGC puede actuar.

Tipo: Enumeración

Facultativo: No

Valores posibles:	"NR" (0x0001)	Liberación normal	(Normal Release)
	"UR" (0x0002)	Recursos no disponibles	(Unavailable Resources)
	"FT" (0x0003)	Fallo, Temporal	(Failure, Temporary)
	"FP" (0x0004)	Fallo, Permanente	(Failure, Permanent)
	"IW" (0x0005)	Error de interfuncionamiento	(Interworking Error)
	"UN" (0x0006)	No soportado	(Unsupported)

Valor por defecto: Ninguno

Causa del fallo

Nombre de parámetro: Causa del fallo

ParameterID: Failurecause (0x0002)

Descripción: La causa de fallo es el valor generado por el equipo liberado, es decir, una conexión de red liberada. Este valor se define en el protocolo de control de portador apropiado.

Tipo: CADENA DE OCTETOS

Facultativo: Sí

Valores posibles: CADENA DE OCTETOS

Valor por defecto: Ninguno

E.1.2.2 Ejecución completa de señal

Nombre de evento: Ejecución completa de señal

EventID: sc (0x0002)

Descripción: Indica la terminación de señales cuando el parámetro NotifyCompletion habilita la generación de informes de evento de ejecución completa. Para una descripción más detallada del procedimiento, véanse 7.1.1, 7.1.11 y 7.2.7.

Parámetros del descriptor de eventos: Ninguno

Parámetros del descriptor de eventos observados:

Identidad de la señal

Nombre de parámetro: Identidad de la señal

ParameterID: SigID (0x0001)

Descripción: Este parámetro identifica la señal que ha terminado. En el caso de una señal que aparece en una lista de señales (SignalList), también se devolverá el parámetro identidad de lista de señales (SignalListID) y se indicará la lista pertinente.

Tipo: Binario: (cadena de) octeto, Texto: cadena

Facultativo: No

Valores posibles: Una señal que ha terminado. Una señal se identificará utilizando la sintaxis pkgdName sin comodines.

Valor por defecto: Ninguno

Método de terminación

Nombre de parámetro: Método de terminación

ParameterID: Meth (0x0002)

Descripción: Indica el motivo de terminación de la señal.

Tipo: Enumeración

Facultativo: No

Valores posibles:

"TO" (0x0001) Expiración de temporización o finalización normal de la señal

"EV" (0x0002) Interrumpida por evento

"SD" (0x0003) Detenida por nuevo descriptor de señales

"NC" (0x0004) No completada, otra causa

"PI" (0x0005) De la primera de la penúltima iteración. Para la última iteración, utilícese TO

Valor por defecto: Ninguno

ID de lista de señales

Nombre de parámetro: ID de lista de señales

ParameterID: SLID (0x0003)

Descripción: Indica a qué lista de señales pertenece la señal en cuestión. El SignalListID sólo se produce en retorno cuando la señal está contenida en una lista de señales.

Tipo: Entero

Facultativo: Sí (sólo se soporta si se utilizan listas de señales)

Valores posibles: 1-65535

Valor por defecto: Ninguno

ID de petición

Nombre de parámetro: Request ID

ParameterID: RID (0x0004)

Descripción: Indica a cuál petición NotifyCompletion se asocia el SignalID

Tipo: Entero

Facultativo: Sí (sólo se soporta si se utilizan listas de señales)

Valores posibles: 1-4294967295

Valor por defecto: Ninguno

E.1.3 Señales

Ninguna.

E.1.4 Estadísticas

Ninguna.

E.2 Lote raíz base (Base Root)

Nombre de lote: raíz base

PackageID: root (0x0002)

Descripción: Este lote define propiedades generales de la pasarela

Versión: 2

Extiende: Ningun

E.2.1 Propiedades

E.2.1.1 Número máximo de contextos

Nombre de propiedad: Número máximo de contextos

Nombre de propiedad: Número máximo de contextos

PropertyID: maxNumberOfContexts (0x0001)

Descripción: El valor de esta propiedad da el número máximo de contextos que pueden existir en un momento dado. El contexto NULL no se incluye en este número.

Tipo: Doble

Valores posibles: 1 y más

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura solamente

E.2.1.2 Número máximo de terminaciones por contexto

Nombre de propiedad: Número máximo de terminaciones por contexto

PropertyID: maxTerminationsPerContext (0x0002)

Descripción: Número máximo de terminaciones permitido en un contexto; véase 6.1.

Tipo: Entero

Valores posibles: Cualquier entero positivo

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura solamente

E.2.1.3 Tiempo normal para ejecución en la MG

Nombre de propiedad: Tiempo normal para ejecución en la MG

PropertyID: normalMGExecutionTime (0x0003)

Descripción: Fijado por el MGC para indicar el plazo de espera en el MGC de una respuesta a cualquier transacción de la MG (sin tener en cuenta el retardo de red).

Tipo: Entero

Valores posibles: Cualquier entero positivo; en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.1.4 Tiempo normal para ejecución en el MGC

Nombre de propiedad: Tiempo normal para ejecución en el MGC

PropertyID: normalMGCExecutionTime (0x0004)

Descripción: Fijado por el MGC para indicar el plazo de espera en la MG de una respuesta a cualquier transacción del MGC (sin tener en cuenta el retardo de red).

Tipo: Entero

Valores posibles: Cualquier entero positivo; en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.1.5 Temporización para respuesta provisional de la MG

Nombre de propiedad: Temporización para respuesta provisional de la MG

PropertyID: MGProvisionalResponseTimerValue (0x0005)

Descripción: Indica el plazo para recibir en el MGC una respuesta pendiente de la MG si no se puede completar una transacción. Inicialmente equivale a normalMGExecutionTime, más el retardo de red, pero puede reducirse.

Tipo: Entero

Valores posibles: Cualquier entero positivo; en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.1.6 Temporización para respuesta provisional del MGC

Nombre de propiedad: Temporización para respuesta provisional del MGC

PropertyID: MGCProvisionalResponseTimerValue (0x0006)

Descripción: Indica el plazo para recibir en la MG una respuesta pendiente del MGC si no se puede completar una transacción. Inicialmente se fija a normalMGCExecutionTime, más el retardo de red, pero puede reducirse.

Tipo: Entero

Valores posibles: Cualquier entero positivo; en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.1.7 Número de transacciones pendientes originadas en el MGC

Nombre de propiedad: Número de transacciones pendientes originadas en el MGC

PropertyID: MGCOriantedPendingLimit (0x0007)

Descripción: Indica el número de TransactionPendings que se pueden recibir del MGC. Si se sobrepasa este valor, el MGC debe emitir un TransactionReply con error 506 ("Se ha rebasado el número máximo de transacciones pendientes"). Sin este mensaje, la MG podría concluir que hay un error de transacción.

Tipo: Entero

Valores posibles: Cualquier entero positivo

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.1.8 Número de transacciones pendientes originadas en la MG

Nombre de propiedad: Número de transacciones pendientes originadas en la MG

PropertyID: MGOriginatedPendingLimit (0x0008)

Descripción: Indica el número de TransactionPendings que se pueden recibir de la MG. Si se sobrepasa este límite la MG debe producir un TransactionReply con error 506 ("Se ha rebasado el número máximo de transacciones pendientes"). Sin este mensaje, el MGC podría deducir que hay un error de transacción.

Tipo: Entero

Valores posibles: Cualquier entero positivo

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.2.2 Eventos

Ninguno.

E.2.3 Señales

Ninguna.

E.2.4 Estadísticas

Ninguna.

E.2.5 Procedimientos

Ninguno.

E.3 Lote generador de tonos

Nombre de lote: Lote generador de tonos

PackageID: tonegen (0x0003)

Descripción: Este lote define señales para generar tonos de audio. No especifica valores de parámetros y se puede extender. Generalmente, los tonos se definen como una señal individual con un parámetro, ind, que significa retardo "interdígito", y un identificador (ID) destinado a los tonos para la reproducción. El ID de tono debe ser el mismo cada vez que se genera el mismo tono. Cabe esperar que en las MG se van a configurar las características de los tonos apropiados para el país en que están ubicadas.

Versión: 2

Previsto para ser extendido solamente: Sí

Extiende: Ninguno

E.3.1 Propiedades

Ninguna.

E.3.2 Eventos

Ninguno.

E.3.3 Señales

E.3.3.1 Tono para la reproducción (Play Tone)

Nombre de señal: Tono para la reproducción

SignalID: pt (0x0001)

Descripción: Reproduce un tono de audio por un canal de audio

Tipo de señal: Breve

Duración: Configurada

Parámetros adicionales:

Lista de ID de tonos

Nombre de parámetro: Lista de ID de tonos

ParameterID: tl (0x0001)

Descripción: Lista de tonos que habrán de ser reproducidos en secuencia.

Tipo: Sublista de enumeración

Facultativo: No

Valores posibles: La sublista contendrá uno o más ID de tonos

Valor por defecto: Ninguno

Duración entre señales

Nombre de parámetro: Duración entre señales

ParameterID: ind (0x0002)

Descripción: Periodo de temporización entre dos tonos consecutivos, en milisegundos

Tipo: Entero

Facultativo: Sí

Valores posibles: Cualquier entero mayor o igual que 0

Valor por defecto: Configurado

Sentido del tono

Nombre de parámetro: Sentido del tono

ParameterID: btd (0x0003)

Descripción: Sentido de emisión del tono desde la terminación.

Tipo: Enumeración

Facultativo: Sí

Valores posibles: "EXT" (0x0001) Exterior
"INT" (0x0002) Interior
"BOTH" (0x0003) Ambos

Valor por defecto: Exterior ("External")

E.3.4 Estadísticas

Ninguna.

E.3.5 Procedimientos

No se especifican ID de tonos en este lote. Los lotes que amplían este lote pueden añadir valores posibles para ID de tono y también añadir señales de tono individuales.

E.4 Lote de detección de tonos (Tone Detection Package)

Nombre de lote: Lote de detección de tonos

PackageID: tonedet (0x0004)

Descripción: Este lote define eventos para la detección de tonos de audio. Los tonos se seleccionan por sus nombres (ID de tono). Cabe esperar que en las MG se van a configurar las características de los tonos apropiados para el país en que están ubicadas.

Este lote no especifica valores de parámetros. Se pretende que sea extensible.

Versión: 1

Previsto para ser extendido solamente: Sí

Extiende: Ninguno

E.4.1 Propiedades

Ninguna.

E.4.2 Eventos

E.4.2.1 Comienzo de tono detectado (Start tone detected)

Nombre de evento: Comienzo de tono detectado

EventID: std (0x0001)

Descripción: Detecta el comienzo de un tono. Las características de la detección positiva de tonos dependen de la implementación.

Parámetros del descriptor de eventos:

Lista de ID de tonos

Nombre de parámetro: Lista de ID de tonos

ParameterID: tl (0x0001)

Tipo: Sublista de enumeración

Facultativo: No

Valores posibles: El único ID de tono definido en este lote es "comodín", que es "*" en codificación textual y 0x0000 en codificación binaria. Las extensiones a este lote añadirían otros ID de tonos posibles. Si tl es "comodín", se detecta cualquier tono.

Valor por defecto: Ninguno

Parámetros del descriptor de eventos observados:

ID de tono

Nombre de parámetro: ID de tono

ParameterID: tid (0x0003)

Descripción: El ID del tono detectado

Tipo: Enumeración

Facultativo: No

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían otros valores posibles de ID de tono.

Valor por defecto: Ninguno

E.4.2.2 Fin de tono detectado

Nombre de evento: Fin de tono detectado

EventID: etd (0x0002)

Descripción: Detecta el fin de un tono.

Parámetros del descriptor de eventos:

Lista de ID de tonos

Nombre de parámetro: Lista de ID de tonos

ParameterID: tl (0x0001)

Descripción: Una lista de ID de tonos que se han de detectar

Tipo: Sublista de enumeración

Facultativo: No

Valores posibles: No se especifican valores posibles en este lote. Las extensiones a este lote añadirían otros ID de tonos posibles.

Valor por defecto: Ninguno

Parámetros del descriptor de eventos observados:

ID de tono

Nombre de parámetro: ID de tono

ParameterID: tid (0x0003)

Descripción: El ID del tono detectado

Tipo: Enumeración

Facultativo: No

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían otros ID de tonos posibles.

Valor por defecto: Ninguno

Duración

Nombre de parámetro: Duración

ParameterId: dur (0x0002)

Descripción: Este parámetro contiene la duración del tono desde la primera detección hasta que se para.

Tipo: Entero

Facultativo: Sí

Valores posibles: Cualquier entero positivo, en milisegundos.

Valor por defecto: Ninguno

E.4.2.3 Tono largo detectado

Nombre de evento: Tono largo detectado

EventID: ltd, (0x0003)

Descripción: Detecta que un tono ha estado reproduciéndose durante un determinado tiempo.

Parámetros de descriptor de eventos:

Lista de ID de tonos

Nombre de parámetro: Lista de ID de tonos

ParameterID: tl (0x0001)

Descripción: Una lista de ID de tono que han de ser detectados

Tipo: Sublista de enumeración

Facultativo: No

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían otros ID de tonos posibles.

Valor por defecto: Ninguno

Duración

Nombre de parámetro: Duración

ParameterID: dur (0x0002)

Descripción: Duración con respecto a la cual se hace la comparación

Tipo: Entero,

Facultativo: Sí

Valores posibles: Cualquier entero positivo válido, expresado en milisegundos

Valor por defecto: Configurado

Parámetros ObservedEventsDescriptor:

ID de tono

Nombre de parámetro: ID de tono

ParameterID: tid (0x0003)

Descripción: El ID de tono del tono detectado

Tipo: Enumeración

Facultativo: No

Valores posibles: No se especifican valores posibles en este lote. Las extensiones a este lote añadirían otros ID de tonos posibles.

Valor por defecto: Ninguno

E.4.3 Señales

Ninguna.

E.4.4 Estadísticas

Ninguna.

E.4.5 Procedimientos

Ninguno.

E.5 Lote generador de DTMF básico (Basic DTMF Generator Package)

Nombre de lote: Lote generador de DTMF básico

PackageID: dg (0x0005)

Descripción: Este lote define los tonos DTMF básicos como señales y extiende los valores permitidos del parámetro tl de tono reproducido en tonogen.

Versión: 2

Extiende: tonegen versión 2

E.5.1 Propiedades

Ninguna.

E.5.2 Eventos

Ninguno.

E.5.3 Señales

E.5.3.1 Carácter DTMF 0

Nombre de señal: Carácter DTMF 0

SignalID: d0 (0x0010)

Descripción: Generar tono DTMF 0. Las características físicas del tono DTMF 0 se definen en la pasarela.

Tipo de señal: Breve

Duración: Configurada

Parámetros adicionales:

Sentido del tono

Nombre de parámetro: Sentido del tono

ParameterID: btd (0x0001)

Descripción: Sentido de emisión del tono desde la terminación.

Tipo: Enumeración

Optional: Sí

Valores posibles: "EXT" (0x0001) Exterior
"INT" (0x0002) Interior
"BOTH" (0x0003) Ambos

Valor por defecto: Exterior ("External")

Valores adicionales: d0 (0x0010) se define como un ID de tono para reproducción

Los demás caracteres DTMF se especifican exactamente de la misma manera. Se incluye un cuadro con todos los nombres de señales e identificadores de señales (SignalID). Obsérvese que cada carácter DTMF se define como señal y como ID de tono, extendiendo de este modo el lote generación de tonos básico. Obsérvese también que los SignalID DTMF son diferentes de los nombres utilizados en un mapa de dígitos.

Nombre de señal	ID de señal/ID de tono
Carácter DTMF 0	d0 (0x0010)
Carácter DTMF 1	d1 (0x0011)
Carácter DTMF 2	d2 (0x0012)
Carácter DTMF 3	d3 (0x0013)
Carácter DTMF 4	d4 (0x0014)
Carácter DTMF 5	d5 (0x0015)
Carácter DTMF 6	d6 (0x0016)
Carácter DTMF 7	d7 (0x0017)
Carácter DTMF 8	d8 (0x0018)
Carácter DTMF 9	d9 (0x0019)
Carácter DTMF *	ds (0x0020)
Carácter DTMF #	do (0x0021)
Carácter DTMF A	da (0x001a)
Carácter DTMF B	db (0x001b)
Carácter DTMF C	dc (0x001c)
Carácter DTMF D	dd (0x001d)

E.5.4 Estadísticas

Ninguna.

E.5.5 Procedimientos

Ninguno.

E.6 Lote detección de DTMF (DTMF Detection Package)

Nombre de lote: Lote detección de DTMF

PackageID: dd (0x0006)

Descripción: Este lote define la detección de tonos DTMF básicos. Extiende los valores posibles de ID de tono en los eventos comienzo de tono detectado fin de tono detectado y tono largo detectado.

Los valores de ID de tono adicionales son, todos ellos, ID de tonos descritos en el lote dg (lote generador DTMF básico).

El cuadro siguiente establece la correspondencia entre los eventos DTMF y los símbolos del mapa de dígitos como se describe en 7.1.14.

DTMF	Símbolo del evento
d0	"0"
d1	"1"
d2	"2"
d3	"3"
d4	"4"
d5	"5"
d6	"6"
d7	"7"
d8	"8"
d9	"9"
da	"A" o "a"
db	"B" o "b"
dc	"C" o "c"
dd	"D" o "d"
ds	"E" o "e"
do	"F" o "f"

Versión: 1

Extiende: tonedet versión 1

E.6.1 Propiedades

Ninguna.

E.6.2 Eventos

E.6.2.1 Dígitos DTMF

Nombre de evento: Dígitos DTMF

EventID: Los EventID se definen con los mismos nombres que los SignalID definidos en el cuadro de E.5.3.

Descripción: Se genera cuando la MG detecta un dígito

Parámetros de descriptor de eventos: Ninguno

Parámetros de descriptor de eventos observados: Ninguno

E.6.2.2 Evento de ejecución completa de mapa de dígitos

Nombre de evento: Evento de ejecución completa de mapa de dígitos

EventID: ce (0x0004)

Descripción: Generado cuando se completa un mapa de dígitos como se describe en 7.1.14.

Parámetros de descriptor de eventos: Ninguno

Parámetros de descriptor de eventos observados:

DigitString

Nombre de parámetro: DigitString

ParameterID: ds (0x0001)

Descripción: La porción de la cadena de marcación actual descrita en 7.1.14 que concuerda con parte o con la totalidad de una secuencia de eventos alternativa especificada en el mapa de dígitos.

Tipo: Cadena de símbolos de mapa de dígitos (posiblemente vacía) retornada como una quotedString

Facultativo: No

Valores posibles: Una secuencia de los caracteres "0" a "9", "A" a "F", y el modificador "Z" de larga duración.

Valor por defecto: Ninguno

Método de terminación

Nombre de parámetro: Método de terminación

ParameterID: Meth (0x0003)

Descripción: Indica el motivo de generación del evento. Véanse los procedimientos de 7.1.14.

Tipo: Enumeración

Facultativo: No

Valores posibles:

"UM" (0x0001)	Concordancia inequívoca
"PM" (0x0002)	Concordancia parcial, finalización por expiración del temporizador o evento no concordante
"FM" (0x0003)	Concordancia total, finalización por expiración del temporizador o evento no concordante

Valor por defecto: Ninguno

E.6.3 Señales

Ninguna.

E.6.4 Estadísticas

Ninguna.

E.6.5 Procedimientos

El procesamiento de mapa de dígitos comienza solamente si se activa un descriptor de eventos que contiene un evento de ejecución completa de mapa de dígitos como se define en E.6.2 y ese evento de ejecución completa de mapa de dígitos contiene un campo EventDM en las acciones solicitadas como se define en 7.1.9. En el descriptor de eventos también puede haber otros parámetros como KeepActive o descriptores de señales o eventos insertados que no influyen en la activación del procesamiento de mapa de dígitos.

E.7 Lote generador de tonos de progresión de la llamada

Nombre de lote: Lote generador de tonos de progresión de la llamada

PackageID: cg 0x0007

Descripción: Este lote define los tonos de progresión de la llamada básicos como señales y extiende los valores permitidos del parámetro tl de tono reproducido en tonegen.

Versión: 2

Extiende: tonegen versión 2

E.7.1 Propiedades

Ninguna.

E.7.2 Eventos

Ninguno.

E.7.3 Señales

E.7.3.1 Tono de invitación a marcar

Nombre de señal: Tono de invitación a marcar

SignalID: dt (0x0030)

Descripción: Generar el tono de invitación a marcar. Las características físicas del tono de invitación a marcar están disponibles en la pasarela.

Tipo de señal: Temporización

Duración: Configurada

Parámetros adicionales: Ninguno

Valores adicionales: dt (0x0030) se define como un ID de tono para reproducción

Los demás tonos de este lote se especifican exactamente de la misma manera. Se incluye un cuadro con todos los nombres de señales e identificadores de señales (SignalID). Obsérvese que cada tono se define como señal y como ID de tono, extendiéndose de este modo el lote de generación de tonos básico.

Nombre de señal	ID de señal/ID de tono
Tono de marcar	dt (0x0030)
Tono de llamada	rt (0x0031)
Tono de ocupado	bt (0x0032)
Tono de congestión	ct (0x0033)
Tono especial de información	sit (0x0034)
Tono de advertencia (grabación)	wt (0x0035)
Tono de reconocimiento de teléfono de previo pago	prt (0x0036)
Tono de llamada en espera	cw (0x0037)
Tono de llamante en espera	cr (0x0038)

E.7.4 Estadísticas

Ninguna.

E.7.5 Procedimientos

NOTA – El conjunto requerido de ID de tonos corresponde a los definidos en la Rec. UIT-T E.180/Q.35. Véase dicha Recomendación para la definición del significado de estos tonos.

E.8 Lote de detección de tonos de progresión de la llamada

Nombre de lote: Lote de detección de tonos de progresión de la llamada

PackageID: cd (0x0008)

Descripción: Este lote define los tonos básicos de detección de progresión de la llamada. Este lote extiende los valores posibles de ID de tono en los eventos Comienzo de tono detectado, Fin de tono detectado y Tono largo detectado.

Valores adicionales

Se definen valores de ID de tonos para Comienzo de tono detectado, Fin de tono detectado y Tono largo detectado con los mismos valores que en el lote cg (lote de generación de tonos de progresión de la llamada).

El conjunto requerido de ID de tonos corresponde a la Rec. UIT-T E.180/Q.35. Véase dicha Recomendación para la definición del significado de estos tonos.

Versión: 1

Extiende: tonedet versión 1

E.8.1 Propiedades

Ninguna.

E.8.2 Eventos

Los eventos se definen como en el lote generador de tonos de progresión de la llamada (cg) para los tonos listados en el cuadro de E.7.3.

E.8.3 Señales

Ninguna.

E.8.4 Estadísticas

Ninguna.

E.8.5 Procedimientos

Ninguno.

E.9 Lote de supervisión de línea analógica

Nombre de lote: Lote de supervisión de línea analógica

PackageID: al 0x0009

Descripción: Este lote define eventos y señales para una línea analógica.

Versión: 1

Extiende: Ninguno

E.9.1 Propiedades

Ninguna.

E.9.2 Eventos

E.9.2.1 Colgado (On-hook)

Nombre de evento: Colgado

EventID: on (0x0004)

Descripción: Detecta la acción de colgar el microteléfono. Si se activa un descriptor de eventos que solicita la supervisión de un evento de colgado y la línea está ya en situación de colgado, el comportamiento de la MG está dictado por las características del parámetro "estricto".

Parámetros del descriptor de eventos:

Transición estricta

Nombre de parámetro: Transición estricta

ParameterID: estricto (0x0001)

Descripción: Indica cómo se detecta el evento de colgado

Tipo: Enumeración

Facultativo: Sí

Valores posibles: "exacto" (0x0000), significa que únicamente se reconocerá una transición efectiva del conmutador a la situación de colgado

"estado" (0x0001), se reconocerá el evento si la transición del conmutador es detectada o si el conmutador ya ha pasado a la situación de colgado;

"error por fallo" ("*failWrong*") (0x0002), si el conmutador ya está colgado, la instrucción falla y se informa un error

Valor por defecto: Exacto

Parámetros del descriptor de eventos observados:

Estado inicial

Nombre de parámetro: Estado inicial

ParameterID: init (0x0002)

Descripción: Motivo por el cual se informa de la transición a colgado. Sólo se devuelve cuando el parámetro de transición ha sido definido con el valor "estado"

Tipo: Booleano

Valores posibles:

"Verdadero" significa que la línea ya estaba en situación de colgado cuando el descriptor de eventos que contiene dicho evento fue activado.

"Falso" significa que se detectó efectivamente una transición a la situación de colgado.

Valor por defecto: Ninguno

E.9.2.2 Descolgado (Off-hook)

EventID: of (0x0005)

Descripción: Detecta la acción de descolgar el microteléfono. Si se activa un descriptor de eventos que solicita la supervisión de un evento de descolgado, y la línea está ya en situación de descolgado, el comportamiento de la MG está dictado por las características del parámetro "estricto".

Parámetros del descriptor de eventos:

Transición estricta

Nombre de parámetro: Transición estricta

ParameterID: strict (0x0001)

Descripción: Indica cómo se detecta el evento de descolgado

Tipo: Enumeración

Valores posibles:

"exacto" (0x0000), significa que únicamente se reconocerá una transición efectiva del conmutador a la situación de descolgado

"estado" (0x0001), significa que se reconocerá el evento si la transición del conmutador es detectada o si el conmutador ya ha pasado a la situación de descolgado

"error por fallo" ("*failWrong*") (0x0002), significa que si el conmutador ya está descolgado, la instrucción falla y se informa un error

Valor por defecto: Exacto

Parámetros de descriptor de eventos observados:**Estado inicial**

Nombre de parámetro: Estado inicial

ParameterID: init (0x0002)

Descripción: Motivo por el cual se informa de la transición a descolgado. Sólo se devuelve cuando el parámetro de transición ha sido definido con el valor "estado"

Tipo: Booleano

Facultativo: Sí

Valores posibles:

"Verdadero" significa que la línea ya estaba en situación de descolgado cuando el descriptor de eventos que contiene dicho evento fue activado.

"Falso" significa que el evento representa una transición efectiva a la situación de descolgado.

Valor por defecto: Ninguno

E.9.2.3 Golpe de conmutador (Flashhook)

Nombre de evento: Golpe de conmutador

EventID: fl (0x0006)

Descripción: Detecta un golpe de conmutación del microteléfono: cuando se cuelga y luego se descuelga en un lapso de tiempo comprendido entre un máximo y un mínimo.

Parámetros del descriptor de eventos:**Duración mínima**

Nombre de parámetro: Duración mínima

ParameterID: mindur (0x0004)

Descripción: Mínimo lapso de tiempo que debe transcurrir entre un colgado y un descolgado para que se pueda detectar una señal de intermitencia (flash)

Tipo: Entero

Facultativo: Sí

Valores posibles: Cualquier entero positivo, en milisegundos

Valor por defecto: Configurado

Duración máxima

Nombre de parámetro: Duración máxima

ParameterID: maxdur (0x0005)

Descripción: Máximo lapso de tiempo que debe transcurrir entre un colgado y un descolgado para que se pueda detectar una señal como de intermitencia (flash).

Tipo: Entero

Facultativo: Sí

Valores posibles: Cualquier entero positivo, en milisegundos

Valor por defecto: Configurado

Parámetros del descriptor de eventos observados: Ninguno

E.9.3 Señales

E.9.3.1 Timbre

Nombre de señal: Timbre

SignalID: ri (0x0002)

Descripción: Aplica corriente de timbre a la línea

Tipo de señal: Temporización

Duración: Configurada

Parámetros adicionales:

Cadencia

Nombre de parámetro: Cadencia

ParameterID: cad (0x0006)

Descripción: Representa duraciones de segmentos alternos activado y desactivado, que constituyen un ciclo completo de señales de timbre que comienzan con un segmento activado.. Las MG de función restringida pueden no tener en cuenta valores de cadencia que ellas sean incapaces de generar.

Tipo: Sublista de enteros

Facultativo: Sí

Valores posibles: Cualquier entero positivo, en milisegundos

Valor por defecto: Configurado

Frecuencia

Nombre de parámetro: Frecuencia

ParameterID: freq (0x0007)

Descripción: Es la frecuencia del timbre. Las MG de función restringida pueden no tener en cuenta valores de cadencia que ellas sean incapaces de generar.

Tipo: Entero

Facultativo: Sí

Valores posibles: Cualquier entero positivo en Hz

Valor por defecto: Configurado

E.9.4 Estadísticas

Ninguna.

E.9.5 Códigos de error

Código de error #: 540

Nombre: Estado inicial no esperado del conmutador

Definición: Se genera cuando el MGC ha tratado de solicitar un evento de transición de estado del conmutador con la opción "error por fallo" para el parámetro Estricto, y el estado del gancho ya es el que implica dicha transición.

Texto de error en el descriptor de error: –

Observaciones: –

E.9.6 Procedimientos

Si el MGC establece un descriptor de eventos que contiene un evento de transición de estado del conmutador (colgado o descolgado) con la opción "error por fallo" para el parámetro "estricto" (0x0001), y el estado del conmutador ya es el que implica la transición, no se ejecuta la instrucción que contiene ese descriptor de eventos. La MG incluirá el código de error 540 ("Estado inicial no esperado del conmutador") en su respuesta.

E.10 Lote de continuidad básico

Nombre de lote: Lote de continuidad básico

PackageID: ct (0x000a)

Descripción: Este lote define eventos y señales para pruebas de continuidad. La prueba de continuidad incluye la provisión de una conexión en bucle o de una función de tranceptor.

Versión: 1

Extiende: Ninguno

E.10.1 Propiedades

Ninguna.

E.10.2 Eventos

E.10.2.1 Ejecución completa de prueba de continuidad

Nombre de evento: Ejecución completa

EventID: cmp (0x0005)

Descripción: Este evento detecta la finalización de una prueba de continuidad.

Parámetros del descriptor de eventos: Ninguno

Parámetros del descriptor de eventos observados:

Resultado de prueba de continuidad

Nombre de parámetro: Resultado

ParameterID: res (0x0008)

Descripción: Indica el resultado de la prueba de continuidad

Tipo: Enumeración

Facultativo: No

Valores posibles: Satisfactorio ("SUCCESS") (0x0001), insatisfactorio ("FAILURE") (0x0000).

Valor por defecto: Ninguno

E.10.3 Señales

E.10.3.1 Prueba de continuidad

Nombre de señal: Prueba de continuidad

SignalID: ct (0x0003)

Descripción: Inicia el envío del tono de prueba de continuidad en la terminación a la cual se aplica.

Tipo de señal: Temporización

Valor por defecto: Configurado

Parámetros adicionales: Ninguno

E.10.3.2 Responder a la prueba de continuidad

Nombre de señal: Responder

SignalID: rsp (0x0004)

Descripción: La señal se utiliza para responder a una prueba de continuidad. Véase E.10.5 para una explicación más detallada.

Tipo de señal: activado/desactivado

Valor por defecto: Configurado

Parámetros adicionales: Ninguno

E.10.4 Estadísticas

Ninguna.

E.10.5 Procedimientos

Para iniciar una prueba de continuidad, el MGC envía una instrucción a la MG que contiene:

- un descriptor de señales con la señal ct, y
- un descriptor de eventos que contiene el evento cmp.

La MG que recibe la instrucción que contiene la señal ct y el evento cmp inicia el tono de prueba de continuidad para la terminación especificada. Si se detecta el tono de retorno y se satisfacen otras condiciones antes de la expiración de la señal, se genera el evento cmp, dando el valor "satisfactorio" al parámetro resultado. Si así no fuera, se genera el evento cmp, dando el valor "satisfactorio" al parámetro resultado.

Cuando un MGC desea que la MG conteste a una prueba de continuidad, envía a la MG una instrucción que contiene un descriptor de señales con la señal rsp. Tras la recepción de una instrucción con la señal rsp, la MG aplica un bucle o bien (en circuitos de dos hilos) espera la recepción del tono de prueba de continuidad. En el primer caso, cualquier información entrante se transmitirá como información saliente. En el segundo caso de circuitos de dos hilos, cada vez que se reciba el tono de prueba adecuado, deberá enviarse el tono de respuesta adecuado. El MGC determina cuándo se elimina la señal rsp.

Para hacer una prueba de continuidad en una terminación, no puede haber en ella ningún dispositivo de eco o códecs activos.

Verificar la seguridad del trayecto vocal como parte de la prueba de continuidad es algo que se define mediante un acuerdo bilateral entre los operadores de red.

NOTA – En las cláusulas 7 y 8/Q.724, 2.1.8/Q.764 y Rec. UIT-T Q.1902.4 se presentan ejemplos detallados de tonos y procedimientos de prueba.

E.11 Lote de red

Nombre de lote: Lote de red

PackageID: nt (0x000b)

Descripción: Este lote define propiedades de terminaciones de red independientes del tipo de red. Esto incluye, aunque no solamente, TDM, IP y ATM.

Versión: 1

Extiende: Ninguno

E.11.1 Propiedades

E.11.1.1 Tamaño máximo de la memoria intermedia para la fluctuación

Nombre de propiedad: Tamaño máximo de la memoria intermedia

PropertyID: jit (0x0007)

Descripción: Esta propiedad establece un tamaño máximo de la memoria intermedia para la fluctuación.

Tipo: Entero

Valores posibles: Cualquier entero positivo, en milisegundos.

Valor por defecto: Configurado

Definida en: El descriptor de control local

Características: Lectura/escritura

E.11.2 Eventos

E.11.2.1 Fallo de red

EventID: netfail (0x0005)

Descripción: La terminación genera este evento cuando detecta un fallo por un motivo relacionado con la red interna o con una red externa.

Parámetros del descriptor de eventos: Ninguno

Parámetros del descriptor de eventos observados:

Causa

Nombre de parámetro: Causa

ParameterID: cs (0x0001)

Descripción: Puede incluirse este parámetro con el evento de fallo para proporcionar información de diagnóstico sobre el motivo de fallo.

Tipo: Cadena

Facultativo: Sí

Valores posibles: Cualquier cadena de texto

Valor por defecto: Ninguno

E.11.2.2 Aviso de calidad

Nombre de evento: Aviso de calidad

EventID: qualert (0x0006)

Descripción: Este evento permite a la MG indicar una pérdida de calidad de la conexión de red. La MG puede hacerlo midiendo la pérdida de lotes, la fluctuación entre llegadas, y el tiempo de propagación, lo que indicará como un porcentaje de pérdida de calidad.

Parámetros del descriptor de eventos:

Umbral

Nombre de parámetro: Umbral

ParameterId: th (0x0001)

Descripción: Umbral para el porcentaje de pérdida de calidad medida, calculado por el método que se ha configurado, que podría tomar en consideración la pérdida de lotes, la fluctuación, y el retardo, por ejemplo. Se activa este evento cuando el valor calculado rebasa el umbral.

Tipo: Entero

Facultativo: Sí

Valores posibles: 0 a 99

Valor por defecto: Configurado

Parámetros del descriptor de eventos observados:

Umbral

Nombre de parámetro: Umbral

ParameterId: th (0x0001)

Descripción: Umbral para el porcentaje de pérdida de calidad medida, calculado por el método que se ha configurado, que podría tomar en consideración la pérdida de lotes, la fluctuación, y el retardo, por ejemplo.

Tipo: Entero

Facultativo: Sí

Valores posibles: 0 a 99

Valor por defecto: Ninguno

E.11.3 Señales

Ninguna.

E.11.4 Estadísticas

E.11.4.1 Duración

Nombre de estadística: Duración

StatisticsID: dur (0x0001)

Descripción: Proporciona el lapso de tiempo durante el cual la terminación ha existido o ha estado fuera del contexto NULL.

Tipo: Doble

Valores posibles: Cualquier entero positivo, en milisegundos

Nivel: Uno u otro

E.11.4.2 Octetos enviados

Nombre de estadística: Octetos enviados

StatisticID: os (0x0002)

Descripción: Proporciona la cantidad de octetos enviados desde la terminación o tren, durante el tiempo que la terminación ha existido o ha estado fuera del contexto NULL. Los octetos representan el flujo de medios saliente, excluyendo toda tara de transporte. En el nivel de la terminación, es igual a la suma de los flujos salientes (de egreso) en todos los trenes.

Para medios de transmisión analógica, se pone a cero la cuenta de octetos.

Tipo: Doble

Valores posibles: Cualquier entero de 64 bits igual a cero o mayor

Nivel: Uno u otro

E.11.4.3 Octetos recibidos

Nombre de estadística: Octetos recibidos

StatisticID: or (0x0003)

Tipo: Doble

Descripción: Proporciona la cantidad de octetos recibidos en la terminación o tren, durante el tiempo que la terminación ha existido o ha estado fuera del contexto NULL. Los octetos representan el flujo de medios entrante, excluyendo toda tara de transporte. En el nivel de la terminación, es igual a la suma de los flujos entrantes en todos los trenes.

Para medios de transmisión analógica, se pone a cero la cuenta de octetos.

Valores posibles: Cualquier entero de 64 bits, igual a cero o mayor

Nivel: Uno u otro

E.11.5 Procedimientos

E.11.5.1 Procedimientos para la aplicación de los eventos Aviso de calidad y Fallo de red

La aplicación de los dos eventos definidos por el paquete depende de:

- el tipo específico de portador (por ejemplo, TDM, línea analógica, RTP, IP, AAL 2, AAL 1, etc.),
- la tecnología de capa de transmisión subyacente (por ejemplo, analógica, PDH, SDH, SONET, xDSL, 802.3, 802.11, etc.),
- si el evento está asociado con recursos internos o externos a la MG.

E.11.5.1.1 Comunes a ambos eventos

Para la aplicación de cualquiera de los dos eventos es necesario que el MGC y la MG se pongan de acuerdo. Si estas dos entidades no tienen una imagen coherente de lo que significan dichos eventos, el MGC no podrá actuar eficazmente sobre ellos.

Está fuera del alcance del protocolo H.248.1 definir las recomendaciones explícitas a este respecto. La utilización particular de estos eventos debería definirse en especificaciones de perfil H.248.

E.11.5.1.2 Propios del evento "Fallo de red"

Es posible emplear el parámetro Causa para distinguir entre diversos modos de fallo. El tipo de terminación también es útil para interpretar el evento.

E.11.5.1.3 Propios del evento "Aviso de calidad"

Conviene que la lógica de detección se base en una o varias métricas de calidad de funcionamiento (o desempeño) relacionadas con la QoS, a partir de lo cual es posible calcular un valor comparable al parámetro Umbral.

Se pueden especificar en un perfil H.248 los detalles de los procesos subyacentes de medición y cálculo (las variables de observación relacionadas con la QoS, los mecanismos de filtrado, la duración de los periodos de actualización, etc.)

E.11.5.1.4 Relación con los procedimientos de cambio de servicio (ServiceChange)

Puede ocurrir que un evento Aviso de calidad o Fallo de red active un procedimiento de cambio de servicio en la terminación o terminaciones del caso. La MG o el MGC puede iniciar, dependiendo de las circunstancias particulares, la transición hacia el estado fuera de servicio (OutOfService).

E.12 Lote RTP

Nombre de lote: Lote RTP

PackageID: rtp (0x000c)

Descripción: Este lote se utiliza para soportar la transferencia de datos multimedia basada en lotes por medio de protocolo de transporte en tiempo real (RTP) (RFC 3550).

Versión: 1

Extiende: nt versión 1

E.12.1 Propiedades

Ninguna.

E.12.2 Eventos

E.12.2.1 Transición de cabida útil

Nombre de evento: Transición de cabida útil

EventID: pltrans (0x0001)

Descripción: Este evento detecta y notifica las transiciones de un formato de cabida útil RTP a otro formato.

Parámetros del descriptor de eventos: Ninguno

Parámetros del descriptor de eventos observados:

Tipo de cábida útil RTP

Nombre de parámetro: Cabida útil RTP (rtppayload)

ParameterID: rtppltype (0x01)

Descripción: Formato de cabida útil resultante de una transición

Tipo: Sublista deenumeración

Facultativo: No

Valores posibles: El método de codificación se especificará utilizando uno o más nombres de codificación válidos, sea del perfil AV RTP o los registrados en la IANA.

Valor por defecto: Ninguno

E.12.3 Señales

Ninguna.

E.12.4 Estadísticas

E.12.4.1 Lotes enviados

Nombre de estadística: Lotes enviados

StatisticID: ps (0x0004)

Descripción: Suministra el número de lotes enviados desde la terminación o tren, desde que la terminación existe o desde que está fuera del contexto NULL.

Tipo: Doble

Valores posibles: Cualquier entero de 64 bits, igual a cero o mayor

Nivel: Uno u otro

E.12.4.2 Lotes recibidos

Nombre de estadística: Lotes recibidos

StatisticID: pr (0x0005)

Descripción: Suministra el número de lotes recibidos en la terminación o tren, desde que la terminación existe o desde que está fuera del contexto NULL.

Tipo: Doble

Valores posibles: Cualquier entero de 64 bits, igual a cero o mayor

Nivel: Uno u otro

E.12.4.3 Pérdida de lotes

Nombre de estadística: Pérdida de lotes

StatisticID: pl (0x0006)

Descripción: Tasa actual de pérdida de lotes en un tren RTP, como se define en RFC 3550. La pérdida de lotes se expresa como un porcentaje: número de lotes perdidos en el intervalo entre dos informes de recepción, dividido por el número de lotes que se esperan en ese intervalo.

Tipo: Doble

Valores posibles: Número entero de 32 bits y una fracción de 32 bits

Nivel: Uno u otro

E.12.4.4 Fluctuación (Jitter)

Nombre de estadística: Fluctuación

StatisticID: jit (0x0007)

Descripción: Pide el valor actual de la fluctuación entre llegadas en un tren RTP como se define en RFC 3550. La fluctuación mide la variación del tiempo entre llegadas para lotes de datos RTP.

Tipo: Doble

Valores posibles: Cualquier entero de 64 bits, igual a cero o mayor

Nivel: Uno u otro

E.12.4.5 Retardo

Nombre de estadística: Retardo

StatisticID: delay (0x0008)

Descripción: Pide el valor actual del tiempo de propagación de lote en unidades de indicación de tiempo. Equivale a la latencia promedio.

Tipo: Doble

Valores posibles: Cualquier entero de 64 bits, igual a cero o mayor

Nivel: Uno u otro

E.12.5 Procedimientos

Si el RTCP está asociado a un tren RTP, no se verá afectado por la propiedad modo H.248.1 del descriptor de control local.

Si además, la MG recibe un descriptor distante vacío para el tren en cuestión, la MG parará el tren RTCP y el tren RTP correspondiente.

E.13 Lote de circuitos TDM

Nombre de lote: Lote de circuitos TDM

PackageID: tdmc (0x000d)

Descripción: Este lote puede utilizarlo cualquier terminación que soporta control de ganancia y de eco. Inicialmente estaba previsto para uso en circuitos TDM, pero su campo de aplicación puede ser más amplio.

Las nuevas versiones o extensiones de este lote deben tener en cuenta utilidades diferentes de TDM.

Versión: 1

Extiende: Lote de red, versión 1

E.13.1 Propiedades

E.13.1.1 Compensación de eco

Nombre de propiedad: Compensación de eco

PropertyID: ec (0x0008)

Tipo: Booleano

Valores posibles:

"True" (cuando se pide compensación de eco) y

"False" (cuando se desactiva)

Valor por defecto: Configurado

Definido en: Descriptor de control local

Características: Lectura/escritura

E.13.1.2 Control de ganancia

Nombre de propiedad: Control de ganancia

PropertyID: gain (0x000a)

Descripción: El control de ganancia, o la adaptación del nivel de señal y la reducción del nivel de ruido, se utiliza para adaptar el nivel de la señal de salida. Sin embargo es necesario desactivar esta función, por ejemplo en llamadas en que intervienen módems. Cuando su valor es "automático", la terminación funciona como un control de nivel automático (ALC, *automatic level control*), con un objetivo de nivel configurado en la MG y sentido saliente.

Tipo: Entero

Valores posibles: El control de ganancia especifica la ganancia en decibelios (positiva o negativa); el máximo entero positivo 214748647 (0xffffffff), está reservado para "automático"

Valor por defecto: Configurado

Definido en: Descriptor de control local

Características: Lectura/escritura

E.13.2 Eventos

Ninguno.

E.13.3 Señales

Ninguna.

E.13.4 Estadísticas

Ninguna.

E.13.5 Procedimientos

Ninguno.

E.14 Lote de segmentación

Nombre de lote: Lote de segmentación

PackageID: seg (0x00A3)

Descripción: Define propiedades que se utilizan cuando se aplica la segmentación basada en H.248 a transportes sin segmentación.

Versión: 1

Extiende: Raíz, versión 2

E.14.1 Propiedades

E.14.1.1 Valor de temporizador de segmentación de la MG

Nombre de propiedad: Valor de temporizador de segmentación de la MG

PropertyID: MGSegmentationTimerValue (0x0009)

Descripción: Indica el plazo para recepción en el MGC de segmentos de un mensaje pendiente de la MG, desde el momento de recepción del testigo SegmentationCompleteToken. Si bien inicialmente tiene el valor MGProvisionalResponseTimerValue, puede modificarse.

Tipo: Entero

Valores posibles: Cualquier entero positivo, en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.14.1.2 Valor de temporizador de segmentación del MGC

Nombre de propiedad: Valor de temporizador de segmentación del MGC

PropertyID: MGCSegmentationTimerValue (0x000A)

Descripción: Indica el plazo para recepción en la MG de segmentos de un mensaje pendiente del MGC, desde el momento de recepción del testigo SegmentationCompleteToken. Si bien inicialmente tiene el valor MGCProvisionalResponseTimerValue, puede modificarse.

Tipo: Entero

Valores posibles: Cualquier entero positivo, en milisegundos

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.14.1.3 Tamaño máximo de PDU de la MG

Nombre de propiedad: Tamaño máximo de PDU de la MG

PropertyID: MGMaxPDUSize (0x000B)

Descripción: Indica el tamaño máximo de la unidad de datos de protocolo entrante de la MG para el protocolo de transporte de la asociación de control. No es recomendable que el MGC produzca mensajes que superen este tamaño.

Tipo: Entero

Valores posibles: Cualquier entero positivo, en bytes

Valor por defecto: Configurado

Definido en: TerminationState

Características: Sólo lectura

E.14.1.4 Tamaño máximo de PDU del MGC

Nombre de propiedad: Tamaño máximo de PDU del MGC

PropertyID: MGCMaxPDUSize (0x000C)

Descripción: Indica el tamaño máximo de la unidad de datos de protocolo entrante del MGC para el protocolo de transporte de la asociación de control. No es recomendable que la MG produzca mensajes que superen este tamaño.

Tipo: Entero

Valores posibles: Cualquier entero positivo, en bytes

Valor por defecto: Configurado

Definido en: TerminationState

Características: Lectura/escritura

E.14.2 Eventos

Ninguno.

E.14.3 Señales

Ninguna.

E.14.4 Estadísticas

Ninguna.

E.14.5 Códigos de error

Código de error #: 459

Nombre: Segmentos no recibidos

Definición: Indica que el receptor de una TransactionReply segmentada ha dejado de esperar a que se le entreguen todos los segmentos.

Texto de error en el descriptor de error: Se incluyen los números de los segmentos faltantes.

Observaciones: –

E.14.6 Procedimientos

Puesto que en algunos transportes no se segmentan automáticamente los mensajes que rebasan el tamaño máximo de unidad de transmisión, es posible que algunos mensajes resulten truncados. Si el mensaje en cuestión contiene varias TransactionReplies, el remitente debería enviar cada una en un mensaje aparte. Si quedan mensajes que rebasan la longitud MTU, el remitente puede aplicar los siguientes procedimientos de segmentación para las TransactionReplies. No se segmentarán las TransactionRequests.

Al segmentar una respuesta de transacción, el remitente deberá garantizar que todos los mensajes contienen instrucciones y/o acciones completas. Cada segmento se indica utilizando información adicional de segmento en el nivel de transacción. Cada segmento utilizará el mismo transactionID, y no repetirá en segmentos posteriores instrucciones o acciones ya completadas. Se numerarán los segmentos en serie empezando por 1 hasta el último, señalado por el testigo SegmentationCompleteToken. Los mensajes segmentados siempre serán combinaciones de instrucciones (*constructs*) válidas desde el punto de vista sintáctico. El receptor, por su parte, responde a cada segmento utilizando la respuesta de segmentación. Tratándose los segmentos de mensajes completos en sí mismos, no es necesario que la entidad receptora espere la llegada de otros de éstos antes de procesar un segmento dado.

Al recibir el último segmento, el receptor establecerá si le han llegado todos los segmentos del mensaje, en cuyo caso responderá mediante un acuse TransactionResponseAcknowledgement.

De lo contrario, iniciará un temporizador y esperará la llegada de los segmentos faltantes. Si bien el receptor ha de poder configurar la duración de dicho temporizador, conviene que ésta sea igual al valor ProvisionalResponseTimerValue (véase E.2.1) de la otra entidad. Si pasado este tiempo no han llegado los segmentos que faltan, el receptor devolverá un código de error 459 ("Segmentos no recibidos").

Ejemplo 1:

```
Sender:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}
Receiver:    !/3 [12.34.56.79]:2944 SM=1/1
Sender:      !/3 [12.34.56.78]:2944 P=1/2{C=1{AV=term3{...}}, C=2{AV=term4{...}}}
Receiver:    !/3 [12.34.56.79]:2944 SM=1/2
Sender:      !/3 [12.34.56.78]:2944 P=1/3/#{C=3{AV=term5{...}}}
Receiver:    !/3 [12.34.56.79]:2944 SM=1/3/#
Receiver:    !/3 [12.34.56.79]:2944 K=1
```

Ejemplo 2:

```
Sender:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}
Receiver:    !/3 [12.34.56.79]:2944 SM=1/1
Sender:      !/3 [12.34.56.78]:2944 P=1/4/#{C=3{AV=term5{...}}}
Receiver:    !/3 [12.34.56.79]:2944 SM=1/4/#
/* Segmentation Timer Expires */
Receiver:    !/3 [12.34.56.79]:2944 ER=459{"2,3"}
```

E.15 Comportamiento de notificación

Nombre de lote: Lote comportamiento de notificación

PackageID: nb (0x009a)

Descripción: Lote que permite a la MG, a petición del MGC, regular el envío de instrucciones Notificar. Este lote interactúa con la bandera NotifyBehaviour descrita en 7.1.9. Esta versión del lote describe el comportamiento de regulación en términos de porcentaje. Quedan en estudio otros tipos de regulación).

Versión: 1

Extiende: Ninguno

E.15.1 Propiedades

E.15.1.1 Regulación de notificación

Nombre de propiedad: Regulación de notificación

PropertyId: notreg (0x0001)

Descripción: Indica el porcentaje de notificaciones en la MG que deberían suprimirse. Un valor de 0% quiere decir que no se suprime ninguna, mientras que 100% implica que habría que suprimirlas todas. Esta propiedad se definirá solamente en la terminación Raíz. Cuando se envíe "Notificación regulada" en determinado evento, la MG utilizará esta propiedad Regulación de Notificación para establecer el porcentaje correcto de notificaciones que se han de regular (es decir, las que hay que suprimir).

Tipo: Entero

Valores posibles: 0-100

Valor por defecto: Ninguno

Definido en: TerminationState

E.15.2 Eventos

Ninguno.

E.15.3 Señales

Ninguna.

E.15.4 Estadísticas

Ninguna.

E.15.5 Procedimientos

E.15.5.1 Comportamiento de notificación (NotifyBehaviour)

El mecanismo Comportamiento de notificación permite al MGC decidir cómo debería recibir las instrucciones de petición de notificación de la MG.

E.15.5.1.1 No notificar nunca (NeverNotify)

Puede ocurrir que el MGC no desee recibir ninguna instrucción de petición de notificación. Por ejemplo: la detección de este evento puede ser fundamental activar una señal o evento incorporados, pero no es necesario informar al MGC. La atribución al parámetro NotifyBehaviour del valor "NeverNotify" permite garantizar que no se notificará al MGC la detección del evento, aunque sí se ejecutan las señales o eventos incorporados. NeverNotify no guarda ninguna relación con la propiedad de regulación de notificación (nb/notreg) y los eventos que tengan este valor no se incluyen en los cálculos de regulación.

E.15.5.1.2 Notificar inmediatamente (NotifyImmediate)

Es posible que el MGC quiera recibir notificación de eventos importantes tales como una "Llamada de emergencia", aun si se están regulando otros eventos. Entonces se asigna el valor "NotifyImmediate" al parámetro NotifyBehaviour para el evento en cuestión. Éste es el comportamiento por defecto cuando se detectan eventos. NotifyImmediate no guarda ninguna relación con la propiedad de regulación de notificación (nb/notreg) y los eventos que tengan este valor no se incluyen en los cálculos de regulación.

E.15.5.1.3 RegulatedNotify

E.15.5.1.3.1 Notificación regulada en porcentaje

Cuando el valor del parámetro NotifyBehaviour es "NotifyRegulated" en un determinado evento, la decisión de enviar al MGC una notificación para cualquier evento dado depende de la propiedad de regulación de notificación (nb/notreg) en la terminación Root. El MGC establece el valor de esta propiedad. Por ejemplo, se establece una tasa de regulación que corresponde al nivel de congestión del MGC. La MG determinará el porcentaje correcto de instrucciones Notificar que se han de suprimir, entre los eventos marcados como "NotifyRegulated" en toda la MG.

La MG puede utilizar cualquier algoritmo que permita realizar el porcentaje de regulación de notificaciones. Es recomendable seleccionar uno que minimice el riesgo de sincronización entre las diferentes MG. Se sugiere la utilización del siguiente algoritmo:

Para un porcentaje de regulación dado (por ejemplo, 10%) y para los eventos que tengan la bandera RegulatedNotify, la MG:

- 1) Suprime varias notificaciones (por ejemplo, Notificación 1).
- 2) Acepta varias notificaciones ulteriores, hasta que se alcance el porcentaje de NotificationRegulation (por ejemplo, aceptar 9 Notificaciones).

- 3) Repite los pasos 1 y 2 hasta que se cambie el porcentaje de regulación.
- 4) Debería empezar, para evitar problemas de sincronización, en una posición aleatoria dentro de la secuencia que se acaba de sugerir.

Si para un evento determinado el valor del parámetro NotifyBehaviour es "RegulatedNotify" y la propiedad de regulación de notificación es cero, el resultado será idéntico al del caso en que dicho parámetro equivalga a "ImmediateNotify", que es el comportamiento por defecto de la notificación de evento.

Conviene utilizar con precaución el mecanismo NotifyBehaviour para evitar los estados incorrectos. Así, por ejemplo, si la MG suprime la notificación de un evento "descolgado" ("off-hook"), el MGC no podrá proporcionar la respuesta correcta (por ejemplo, tono de invitación a marcar, tono de congestión). Esto se puede solucionar utilizando el mecanismo de señales o eventos incorporados a fin de devolver la respuesta adecuada sin que en ello participe el MGC. Es posible que el MGC utilice un descriptor de señales o eventos incorporados que se active siempre que se haya detectado y aceptado un evento con "NotifyRegulated" (es decir, notificación no suprimida) y un descriptor de señales o eventos incorporados independiente de aquél, que se active cuando se detecte y regule un evento (es decir, notificación suprimida).

NOTA – Aun en este caso, la MG sigue sin conocer la composición de la llamada, ya que simplemente está realizando una tarea preprogramada y no una función de control de llamada.

E.15.5.2 Ejemplo

Considérese el caso en el que un MGC y una MG terminen líneas analógicas. El comportamiento del controlador y de la pasarela puede cambiar si el MGC está sobrecargado o no, y si la línea analógica (ALN, *analogue line*) está originando una llamada de emergencia o no. En el cuadro E.1 se presentan los diversos comportamientos posibles:

Cuadro E.1/H.248.1 – Comportamiento posible de MGC/MG en caso de sobrecarga

Comportamiento de la MG		1) Tipo de llamada/contexto	
Tarea: "Procesamiento de señalización de llamada por línea analógica (ALS, <i>analogue line call signalling</i>)" = tráfico de señalización de control de llamada para terminales POTS (ALN, interfaces de línea analógica)		Llamada de emergencia	Llamada que no es de emergencia
2) Estado de carga del MGC NOTA – Se supone que hay dos estados.	Estado sin sobrecarga (p. ej. carga baja, media y alta)	I) MG: Procesamiento ALS normal con <i>la más alta</i> prioridad	II) MG: Procesamiento ALS normal con <i>baja</i> prioridad
	Estado de sobrecarga	III) MG: Procesamiento ALS normal ==> Se reenvía la señalización de control de llamada al MGC de "modo habitual H.248"	IV) MG: Procesamiento ALS local programado ==> El MGC <i>no</i> participa ==> Protocolo de llamada conforme rechazado mediante el "tono de congestión"

E.15.5.2.1 Comportamiento en los casos I y II

Al no estar el MGC en estado de sobrecarga, y poder entonces recibir todos los mensajes (notificaciones) de la MG, ésta no requiere ningún NotifyBehaviour especial. Sin embargo, tan pronto el MGC haya establecido el tipo de llamada, puede imponer una prioridad a la MG utilizando el atributo de contexto que venga al caso.

E.15.5.2.2 Comportamiento en los casos III y IV

Cuando el MGC haya establecido que se encuentra en un estado de sobrecarga, puede enviar la siguiente instrucción a todas las terminaciones en el contexto NULL o a un grupo seleccionado de ellas.

Example message:

```
Transaction=1234{
  Context = - {
    Modify = aln/* {           ; select terminations as desired
      Events = 1234 {
        al/of {
          RegulatedNotify {
            Embed {
              Signals {cg/dt} ,
; If the off-hook is regulated apply dial tone.
              Events = 1235 {
                xdd/xce {
                  DigitMap = PriorityDialPlan1,
                  bc = 20,
                  mp = enhanced,
                  ImmediateNotify
                },
; If emergency generate notify to MGC
                xdd/xce {
                  DigitMap = NonPriorityDialPlan1,
                  bc = 20,
                  mp = enhanced,
                  NeverNotify,
                  Embed{
                    Signals {cg/ct}
                  }
                },
; If non-priority send congestion tone and never notify
                al/on {
                  NeverNotify,
                  ResetEventsDescriptor
                }
; If on-hook is detected then reset the EventsDescriptor
; and never notify
              }
            }
          }
        }
      }
    }
  }
}
```

NOTA – La identidad de contexto puede ser específica o NULL, y el terminationID puede ser comodín o específico, dependiendo de cómo y cuándo se haya impuesto este comportamiento a la MG. La utilización de la respuesta comodín limitará la cantidad de respuestas si el comportamiento en caso de sobrecarga se activa mientras el MGC entra en dicho estado.

E.15.5.2.3 Propiedad de regulación de notificación diferente de cero (estado de sobrecarga)

En caso de regulación, al detectar un evento descolgado se observaría el siguiente comportamiento de la MG:

- Suprimir la notificación de descolgado al MGC.
- Emitir el tono de invitación a marcar.
- Recibir los dígitos marcados por el abonado.

- Si el usuario ha marcado un número de emergencia, generar siempre la notificación de correspondencia con el "plan de marcación de prioridad".
- Si el usuario ha marcado un número no prioritario, suprimir siempre la notificación de correspondencia con el "plan de marcación no prioritaria" y emitir el tono de congestión.
- Habilitar la detección de "colgado" (utilizando NeverNotify). Al recibirse un evento de colgado, el descriptor de eventos en la terminación volvería al estado en que se activó la incorporación (o el descriptor de eventos configurado por la MG).

E.15.5.2.4 Propiedad de regulación de notificación igual a cero (Estado sin sobrecarga)

En este caso, la MG:

- Enviaría una notificación de descolgado al MGC.

Si el descriptor de eventos con comportamiento de sobrecarga se establece mediante instrucciones explícitas Modificar del MGC (en lugar de utilizar la configuración por la MG), mientras el MGC sigue en la condición de sobrecarga, dicho descriptor se recargará en cada terminación, como parte de la secuencia de liberación de toda llamada o intento de llamada notificados al MGC.

Cuando haya terminado la condición de sobrecarga del MGC, éste podrá restablecer para las terminaciones del contexto NULL el comportamiento sin sobrecarga por defecto. El MGC podría también dar a la propiedad de regulación de notificación el valor cero.

Anexo F

Procedimientos de cambio de servicio (ServiceChange)

F.1 Introducción

En este anexo se describe la utilización de los procedimientos de cambio de servicio (ServiceChange) provocados por ciertos eventos que ocurren en una MG o en un MGC. Se pretende aclarar lo descrito en 7.2.8 y 11. De haber alguna discrepancia entre los procedimientos de este anexo y los de aquellas cláusulas, estos últimos tienen prioridad.

Los eventos que pueden provocar el envío de una instrucción de cambio de servicio, como se describe en este anexo, son:

- 1) Registro de la MG – Ésta se registra con un MGC y se establece una asociación de control.
- 2) Nuevo registro de la MG – Ésta se registra con un MGC después de que un MGC se lo ha solicitado.
- 3) Cancelación de servicio de la MG – La MG informa al MGC que se han de poner fuera de servicio algunas terminaciones o toda la MG.
- 4) Restauración de servicio de la MG – La MG se ha recuperado tras un fallo y lo informa al MGC.
- 5) Reemplazo por una MG redundante – Una MG secundaria reemplaza a la primaria en caso de fallo o mantenimiento de ésta.
- 6) Pérdida de comunicación de la MG – La MG informa al MGC que había perdido comunicación con él, pero se ha restablecido.
- 7) Cambio de capacidad de la MG – La MG informa al MGC que han cambiado las capacidades de algunas terminaciones o de toda la MG.

- 8) Nuevo registro de la MG iniciado por el MGC – El MGC informa a la MG que debe registrarse nuevamente con el mismo MGC o con otro.
- 9) Restauración de servicio iniciada por el MGC – El MGC se ha recuperado tras un fallo y ordena a la MG poner las terminaciones del caso en un estado inicial.
- 10) Cancelación de servicio iniciada por el MGC – El MGC informa a la MG que se están poniendo fuera de servicio algunas terminaciones o toda la MG.
- 11) Derivación en caso de fallo hacia un MGC redundante – El MGC primario está fallando e instruye a la MG para que se registre con un determinado MGC secundario.

En la figura F.1 se muestran estos eventos de activación de cambio de servicio, la entidad en la que está presente el evento y los pares de eventos correlacionados. La figura F.2 es un modelo de estados que vale para la MG y sus terminaciones.

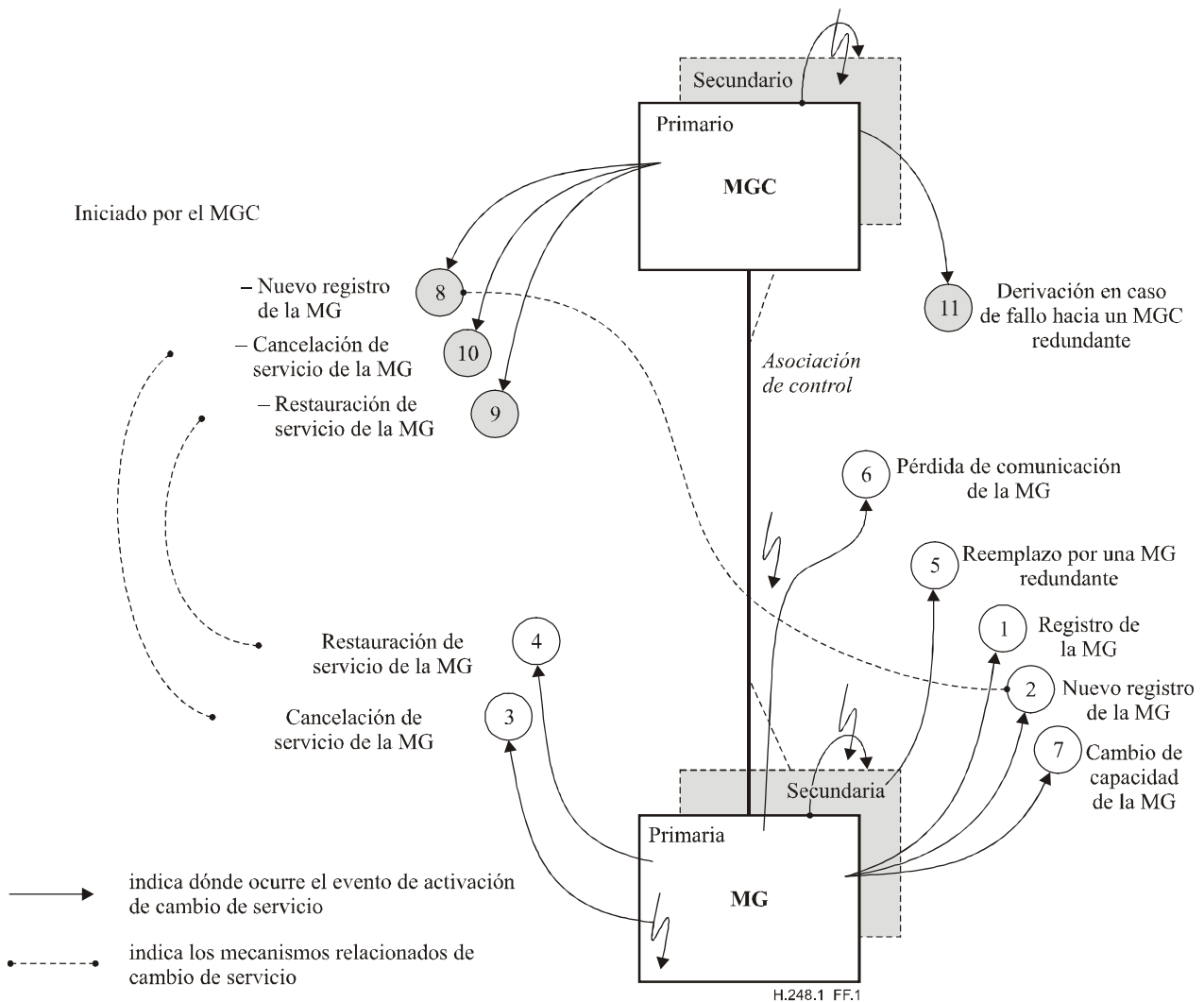


Figura F.1/H.248.1 – Eventos de activación de cambio de servicio

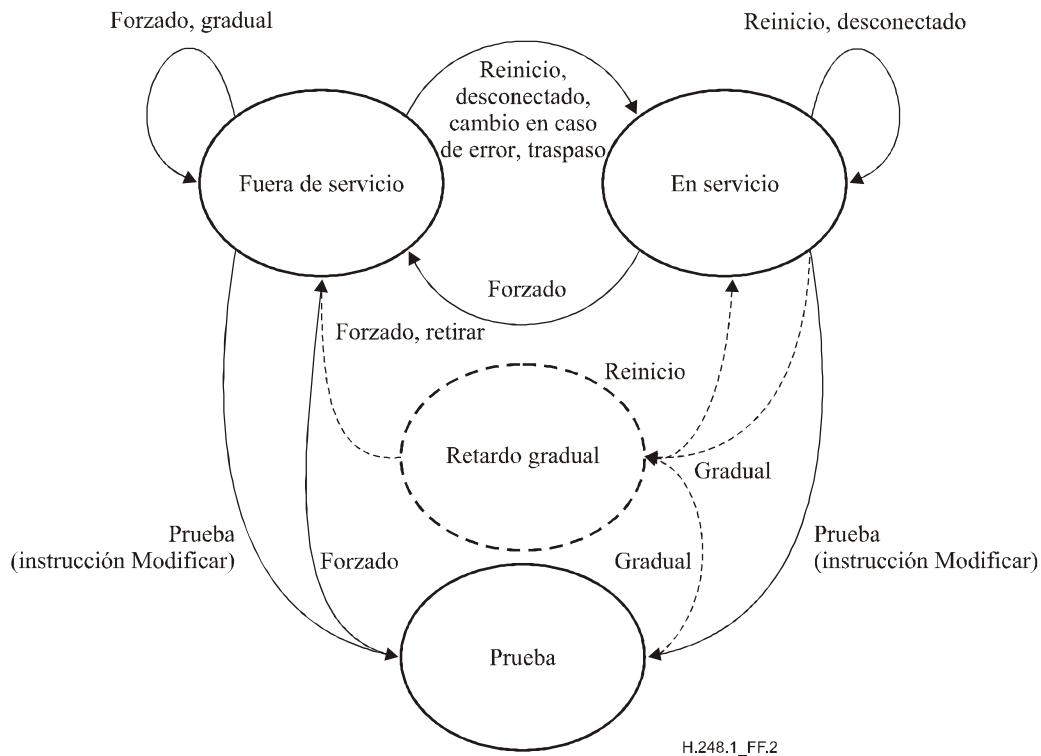


Figura F.2/H.248.1 – Modelo de estados simple de cambio de servicio

F.2 Definición de asociación de control

Una asociación de control es una relación de comunicación en la que un MGC controla una MG. Se crea en el registro y puede terminar de varias formas: la MG sale de servicio, ha sido transpasada con éxito a otro MGC o ha pasado a él tras un fallo. Una MG siempre tendrá a lo sumo una asociación de control en un momento dado, salvo cuando se divide una MG física en una o varias MG virtuales. En este último caso, cada MG virtual habrá de tener a lo sumo una asociación de control. Solamente las MG podrán crear ejemplares de asociaciones de control. Las asociaciones de control, tal como se definen en este anexo, se aplican solamente dentro del contexto de la señalización de control de pasarela de medios. Esta definición no debería confundirse con la definición de asociaciones de control en el contexto de transporte (TCP, UDP, etc.).

El empleo de instrucciones ServiceChange en la terminación Root afecta solamente la MG virtual a la cual pertenece dicha terminación. No se ven afectadas otras MG virtuales presentes en la misma MG física.

F.3 Eventos que provocan procedimientos de cambio de servicio

F.3.1 Registro de la MG

La MG puede registrarse:

- anunciando inicialmente su presencia, es decir, "registrándose" por el método de cambio de servicio (ServiceChangeMethod) "Reinicio", con el motivo de cambio de servicio (ServiceChangeReasons) 901 (Reinicio en frío o "Cold Boot") o 902 (Reinicio en caliente o "Warm Boot"),
- indicando que está modificando su asociación, ejecutando el método de cambio de servicio "Cambio por error" ("Failover") y con el ServiceChangeReason 909 "Fallo inminente del MGC" ("Impending Failure") o 908 "Fallo inminente de la MG" ("MG Impending Failure"), o

- indicando que recibió instrucciones para cambiar su asociación por el método de cambio de servicio "Traspaso" ("Handoff") y ServiceChangeReason 903 ("Cambio dirigido por el MGC") (véase F.3.11).

El proceso de registro se inicia cuando se envía la instrucción inicial ServiceChange de la MG al MGC, y queda finalizado cuando el MGC responde sin ofrecer otra dirección ni un error, y cuando ambas entidades se ponen de acuerdo en un ServiceChangeProfile. Al completarse el registro queda establecida la asociación de control. El registro siempre tiene lugar en la terminación Root. Al emprenderse un nuevo registro se terminan todas las asociaciones de control existentes entre el MGC y la MG. Se ignorarán todas las instrucciones de la asociación de control previa.

En el registro se utiliza el ServiceChangeProfile. Para más detalles, véase F.5.5.

En el registro se utiliza la ServiceChangeVersion. Para más detalles, véase F.5.6.

F.3.2 Nuevo registro de la MG

Hay dos casos de nuevo registro de la MG, a saber:

- 1) Cuando el MGC solicita a la MG que se registre nuevamente (F.3.8), en cuyo caso deberá enviar una instrucción ServiceChange con el ServiceChangeMethod "Traspaso", el motivo (ServiceChangeReason) 903 ("Cambio dirigido por el MGC") y el identificador "ServiceChangeMgcID" designado.
- 2) Cuando el MGC inicia la restauración de servicio (F.3.9). La MG se registrará nuevamente mediante el envío de una instrucción ServiceChange con el motivo ServiceChangeMethod "Reinicio".

El nuevo registro queda finalizado cuando el MGC responde a la instrucción sin ofrecer otra dirección ni un error, y cuando ambas entidades se ponen de acuerdo en un ServiceChangeProfile. Si no se recibe respuesta del MGC primario, la MG seguirá el procedimiento indicado en F.3.1. Se recomienda utilizar los motivos ServiceChangeReason 903 ("Cambio dirigido por el MGC") o 909 ("Fallo inminente del MGC").

Al completarse el registro queda establecida la asociación de control. El registro siempre tiene lugar en la terminación Root. Al emprenderse un nuevo registro se terminan todas las asociaciones de control existentes entre el MGC y la MG. Se ignorarán todas las instrucciones de la asociación de control previa.

En el nuevo registro se utiliza el ServiceChangeProfile. Para más detalles, véase F.5.5.

En el nuevo registro se utiliza la ServiceChangeVersion. Para más detalles, véase F.5.6.

F.3.3 Cancelación de servicio de la MG

Para poner la MG fuera de servicio ("OutOfService"), la MG envía una instrucción ServiceChange e indica el método (ServiceChangeMethod) "Forzado" ("Forced") o "Gradual" ("Graceful") en la terminación Root. Para más detalles, véase F.4.1.1.

Con el fin de poner fuera de servicio una terminación o un grupo de terminaciones, la MG envía una instrucción ServiceChange con un ServiceChangeMethod "Forzado" o "Gradual" en la terminación o terminaciones del caso. Para más detalles, véanse F.4.1.2 y F.4.1.3.

El retardo de cambio de servicio (ServiceChangeDelay) indica el tiempo que transcurre antes de que ocurra la cancelación del servicio. Para más detalles, véase F.5.3.

Para cancelar una instrucción Gradual anterior aplicable a toda la MG (de la que ya se acusó recibo), la MG envía una instrucción ServiceChange en la terminación Root e indica el método (ServiceChangeMethod) "Reinicio" y el motivo (ServiceChangeReason) 918, "Cancelación gradual". La MG seguirá en servicio y se pondrán de nuevo en servicio todas las terminaciones que habían sido desactivadas, a menos que la MG especifique otra cosa. Si la instrucción de

"Cancelación gradual" se recibe después de la expiración del temporizador de retardo, se considerará que la MG se ha registrado nuevamente igual que si hubiera enviado una instrucción Reinicio en frío.

Para cancelar una instrucción Gradual anterior aplicable a una terminación o un grupo de terminaciones (de la que ya se acusó recibo), la MG envía una instrucción ServiceChange e indica el método (ServiceChangeMethod) "Reinicio" y el motivo (ServiceChangeReason) "Cancelación de instrucción gradual" en la terminación o terminaciones en cuestión. La terminación seguirá en servicio. Si ésta ya fue desactivada, se pondrá nuevamente en servicio, como lo habría hecho con cualquier instrucción Reinicio de ServiceChange.

F.3.4 Restauración de servicio de la MG

Este evento ocurre cuando una MG se pone nuevamente en servicio tras un fallo o una intervención de mantenimiento.

Cuando lo envía la MG, ésta también envía una instrucción ServiceChange al MGC anunciándole que se ha reiniciado o que desea negociar la versión del protocolo o el perfil H.248. En este caso, se emplean el ServiceChangeMethod "Reinicio" y el ServiceChangeReason 900 ("Servicio restaurado"). El ServiceChangeReason indicará las acciones que tal vez tenga que emprender el MGC.

F.3.5 Reemplazo por una MG redundante

Hay dos casos posibles, a saber:

1) Reemplazo iniciado por la MG primaria:

Si una MG se desactiva (OutOfService) y desea ceder el procesamiento a una MG secundaria designada, envía una instrucción ServiceChange al MGC con el ServiceChangeMethod "Cambio por error" y un ServiceChangeReason 908 ("Fallo inminente de la MG"). El MGC dejará de enviar mensajes a la MG primaria y pondrá fin a la asociación de control. Luego, la MG secundaria enviará una instrucción ServiceChange con ServiceChangeMethod "Reinicio" y ServiceChangeReason 900 ("Servicio restaurado"), tras lo cual el MGC establecerá una nueva asociación de control con la MG secundaria.

2) Reemplazo iniciado por una MG secundaria:

Cuando una unidad secundaria alternativa de una MG detecte un problema en una MG primaria causado por un fallo o por una intervención de mantenimiento, y la MG primaria no pueda notificarlo al MGC, la MG secundaria enviará una instrucción ServiceChange al MGC con ServiceChangeMethod "Cambio por fallo" y ServiceChangeReason 919 "Cambio por fallo en caliente" ("Warm Failover") o 920 "Cambio por fallo en frío" ("Cold Failover"), según sea el caso. Se aplican los procedimientos de F.3.4.

El MGC no tendrá en cuenta ningún intento de reemplazo de MG desconocidas. Es necesario que el MGC sepa, bien sea por configuración o mediante otros medios, que la MG secundaria que anuncia un reemplazo está asociada con la primaria que ha fallado y que, por ende, está autorizada a reemplazarla.

Si la MG secundaria es conocida y puede reemplazar a la primaria, el MGC intentará comunicarse con ésta para saber si aún funciona, utilizando por ejemplo una instrucción AuditValue vacía u otro método conveniente (véase 11.5). Si la MG primaria responde, el MGC rechazará el intento de reemplazo. Si no, supondrá que ha fallado y aceptará el cambio por fallo, en caliente o en frío, tras lo cual se suprime la asociación de control con la MG primaria y se crea una nueva con la secundaria.

F.3.6 Pérdida de comunicación de la MG

Cuando la MG ha detectado la pérdida y el subsecuente reestablecimiento de la comunicación con el MGC, le envía una instrucción ServiceChange con ServiceChangeMethod "Desconectada" en la asociación de control en servicio. Si el MGC no responde, la MG envía entonces una instrucción ServiceChange con ServiceChangeMethod "Cambio por fallo" y ServiceChangeReason 909 ("Fallo inminente del MGC") a cada MGC en su lista, hasta establecer una nueva asociación de control o agotar la lista de MGC.

Si el MGC en la asociación de control original responde a la instrucción ServiceChange, dicha asociación continúa sin interrupción y se procesan todas las instrucciones como si no hubiera habido pérdida de comunicación. De lo contrario, se termina la asociación de control original cuando la MG envía una instrucción ServiceChange a un nuevo MGC y se ignoran todas las instrucciones originadas en las asociaciones de control previas. Una vez el nuevo MGC responde a la instrucción ServiceChange se establece una nueva asociación de control, completándose así el proceso de registro.

Si la MG agota su lista de MGC sin poder establecer una asociación de control, esperará durante un lapso de tiempo aleatorio e intentará registrarse de nuevo con los MGC de su lista, empezando por el de la asociación de control original. La MG enviará una instrucción ServiceChange con ServiceChangeMethod "Desconectada" al MGC de la asociación de control original cada vez que intente entrar en contacto con él, y una ServiceChange con ServiceChangeMethod "Cambio por fallo" a todos los otros MGC.

Conviene que los MGC que reciban instrucciones ServiceChange con ServiceChangeMethod "Desconectado" efectúen una auditoría de la MG para determinar si la pérdida de mensajes ha provocado una incongruencia de estados.

En estos casos, se recomienda utilizar el motivo ServiceChangeReason 900 ("Servicio restaurado"), aunque en ciertos casos pueda ser necesario emplear otros códigos de ServiceChangeReason.

F.3.7 Cambio de capacidad de la MG

Para anunciar un cambio en sus capacidades, la MG envía una instrucción ServiceChange e indica el método (ServiceChangeMethod) adecuado, y un ServiceChangeReason 916 ("Cambio de paquetes") o 917 ("Cambio de capacidades"). El MGC debería realizar una auditoría de la MG con el fin de establecer cuáles son sus nuevas capacidades. Si se trata de una MG en servicio que envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio", la asociación de control se mantiene durante el cambio de capacidades, y se considera que la MG no ha realizado un reinicio de sistema.

F.3.8 Nuevo registro de la MG iniciado por el MGC

El MGC puede solicitar a la MG que se registre nuevamente, emitiendo una instrucción ServiceChange en la terminación Root con ServiceChangeMethod "Traspaso", ServiceChangeReason 903 ("Cambio dirigido por el MGC") y su propio ServiceChangeMgcID (es decir, el del MGC actual).

En F.3.2 se explican las acciones que emprende la MG.

F.3.9 Restauración de servicio iniciada por el MGC

Con el fin de solicitar a la MG que se reinicie ella misma, el MGC le envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio" y un ServiceChangeReason adecuado, por ejemplo 900 ("Servicio restaurado") o 901 ("Reinicio en frío"). La MG debe establecer una nueva asociación de control conforme a F.3.2, utilizando el ServiceChangeReason indicado.

Para restaurar el servicio en una terminación o un grupo de terminaciones, el MGC envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio" en la terminación o terminaciones en cuestión. En F.4.1.2 y F.4.1.3 se explican las acciones que emprende el MGC.

ServiceChangeDelay indica el plazo de restauración del servicio. En F.5.3 se dan más detalles.

F.3.10 Cancelación de servicio iniciada por el MGC

F.3.10.1 Terminación Root

Para desactivar la MG ("OutOfService"), el MGC envía una instrucción ServiceChange con ServiceChangeMethod "Forzado" o "Gradual" en la terminación Root. Pueden invocarse varios motivos (ServiceChangeReasons), por ejemplo el 905 ("Terminación sacada del servicio"). En F.4.1.1 se explican las acciones que emprende el MGC.

ServiceChangeDelay indica el plazo para la cancelación de servicio. En F.5.3 se dan más detalles.

Para cancelar una instrucción Gradual anterior aplicable a toda la MG (de la que ya se acusó recibo), el MGC envía una instrucción ServiceChange en la terminación Root con un ServiceChangeMethod "Reinicio" y ServiceChangeReason 918, ("Cancelación de instrucción gradual"). La MG seguirá en servicio y se pondrán de nuevo en servicio todas las terminaciones que habían sido desactivadas, a menos que la MG especifique otra cosa. Cuando se reciba una "Cancelación de instrucción gradual" después de la expiración del temporizador de retardo, la MG informará un código de error 502 ("No está listo") al MGC. La MG tendrá que registrarse nuevamente para retornar al servicio.

F.3.10.2 Terminaciones físicas

Para desactivar ("OutOfService") una terminación o un grupo de terminaciones, el MGC envía una instrucción ServiceChange con ServiceChangeMethod "Forzado" o "Gradual" en la terminación o terminaciones en cuestión. Pueden invocarse varios motivos (ServiceChangeReasons), por ejemplo, 904 ("La terminación funciona mal"), 905 ("Terminación desactivada"), 906 ("Pérdida de conectividad de capa inferior"), o 907 ("Fallo de transmisión"). En F.4.2 y F.4.3 se explican las acciones que emprende el MGC.

ServiceChangeDelay indica el plazo para la cancelación de servicio. En F.5.3 se dan más detalles.

Para cancelar una instrucción Gradual anterior en una terminación o un grupo de terminaciones (de la que ya se acusó recibo) la MG envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio" y ServiceChangeReason 918 ("Cancelación de instrucción Gradual") en la terminación o terminaciones en cuestión. La terminación seguirá en servicio. Si ya se había desactivado, se pondrá nuevamente en servicio, igual que con cualquier instrucción ServiceChange Reinicio.

F.3.10.3 Terminaciones efímeras

El MGC no podrá utilizar ServiceChange para cancelar el servicio en terminaciones efímeras. Para eliminarla, bastará con retirar la terminación del contexto.

F.3.11 Derivación en caso de fallo hacia un MGC redundante

Cuando el MGC en una asociación de control deba salir de servicio por mantenimiento o fallo, puede dirigir la MG hacia un MGC secundario especificado, enviando una instrucción ServiceChange con ServiceChangeMethod "Traspaso", ServiceChangeReason 903 ("Cambio dirigido por el MGC") y la dirección del nuevo MGC en el parámetro ServiceChangeMgcID. Al recibir la respuesta de la MG a la instrucción se pone fin a la asociación de control. Luego, la MG envía una instrucción ServiceChange al MGC especificado, con ServiceChangeMethod "Traspaso" y ServiceChangeReason 903 ("Cambio dirigido por el MGC"). Al recibir la respuesta del MGC a esta instrucción se establece una nueva asociación de control.

En caso de que el MGC especificado rechace el intento de traspaso o no pueda enviar una instrucción ServiceChange con ServiceChangeMethod Traspaso debido a un fallo, la MG comenzará los procedimientos para pérdida de comunicación descritos en F.3.6, empezando con su MGC primario configurado.

F.4 Descripción del elemento de cambio de servicio (ServiceChange)

F.4.1 Método de cambio de servicio (ServiceChangeMethod)

En esta cláusula se describe el comportamiento del ServiceChangeMethod en los diversos tipos de terminación, es decir en:

- Las terminaciones Root.
- Las terminaciones físicas.
- Las terminaciones efímeras.

F.4.1.1 Comportamiento del método ServiceChange en las terminaciones Root

La instrucción ServiceChange en la terminación Root produce diferentes resultados, dependiendo del ServiceChangeMethod, como se describe a continuación:

- 1) Reinicio – Enviado por la MG para anunciar un rearranque o un cambio de capacidades, o que desea renegociar la versión del protocolo o el perfil. ServiceChangeReason indica las acciones que posiblemente tendrá que realizar el MGC. Cuando este último lo envíe, la MG se reiniciará ella misma utilizando el ServiceChangeReason indicado.
- 2) Forzado – Si es enviado por la MG, ésta indica que está saliendo de servicio inmediatamente. Al recibir la respuesta del MGC a la instrucción, la MG termina la asociación de control. Si es enviado por el MGC, la MG se pondrá ella misma fuera de servicio y terminará la asociación de control después de haber enviado la respuesta a la instrucción. Obsérvese que un MGC no puede poner de nuevo en servicio una MG, aunque sí puede enviar una petición para su desactivación. La MG empieza el registro con el MGC para establecer una nueva asociación de control e indica la restauración de servicio, tal como lo haría para cualquier otro registro. La indicación de plazo (ServiceChangeDelay) no vale para este ServiceChangeMethod "Forced".
- 3) Gradual – Si es enviado por la MG, indica que está saliendo de servicio al expirar el plazo ServiceChangeDelay, tras el cual se termina la asociación de control. Si es enviado por el MGC, la MG se pondrá ella misma fuera de servicio y terminará la asociación de control al final del periodo ServiceChangeDelay. Cuando se emplee un ServiceChangeDelay igual a cero, o cuando éste no esté presente, significa que la MG saldrá de servicio y terminará la asociación de control al suprimirse el último contexto retirando sus terminaciones, y que el MGC no añadirá nuevas conexiones. La MG debería atribuir el valor desactivado propiedad ("OutOfService") a la ServiceStates en la terminación Root, al expirar el periodo ServiceChangeDelay o tras la supresión de todas las terminaciones de los contextos activos (lo que ocurra primero). Para cancelar una instrucción ServiceChange con ServiceChangeMethod "Gradual" ya enviada (de la que ya se acusó recibo), la entidad que inició el ServiceChangeMethod Gradual envía una instrucción ServiceChange con ServiceChangeMethod Reinicio y ServiceChangeReason 918 ("Cancelación de instrucción gradual").

- 4) Cambio por fallo – Si es enviado por la MG a un MGC que no está en la asociación de control actual, indica que ha fallado su MGC actual e intenta registrarse con el receptor de la instrucción. Si es enviado por la MG a su MGC en la asociación de control actual, indica que la MG primaria ha fallado y ha sido reemplazada por una secundaria. En ambos casos se termina la asociación de control previa tras el envío de la instrucción ServiceChange, y se establece la nueva entre la MG y el nuevo MGC, o entre el MGC y la nueva MG luego de recibir la respuesta a la instrucción. Los MGC no enviarán instrucciones ServiceChange con ServiceChangeMethod "Cambio por fallo".
- 5) Traspaso – Si es enviado por el MGC, indica que se está transfiriendo la MG a un nuevo MGC. Termina la asociación de control existente, tras la recepción de la respuesta a la instrucción. Si es enviado por la MG al MGC, indica que la MG está intentando establecer una nueva asociación de control, conforme a una instrucción de traspaso recibida del MGC en su anterior asociación de control. Un traspaso enviado por la MG al MGC es un registro y se establece una nueva asociación de control después de haber recibido la respuesta a la instrucción. La MG no utilizará Traspaso sin que antes no se lo haya pedido un MGC.
- 6) Desconectado – Si lo envía la MG, indica que se ha perdido la comunicación en la asociación de control existente, pero se ha reestablecido de nuevo. Se renueva la asociación de control existente. Los MGC no enviarán instrucciones ServiceChange con ServiceChangeMethod "Desconectado".

F.4.1.2 Comportamiento del método ServiceChange en las terminaciones físicas

Mientras la MG está en una asociación de control activa, una instrucción ServiceChange en una terminación física produce diferentes resultados, dependiendo del ServiceChangeMethod, como se describe a continuación:

- 1) Reinicio – Si es enviado por la MG, anuncia que se ha(n) reiniciado la(s) terminación(es) o que hay un cambio de capacidades. ServiceChangeReason indica las acciones que posiblemente tendrá que realizar el MGC. Si lo envía este último, la MG reiniciará la terminación o terminaciones utilizando el ServiceChangeReason indicado.
- 2) Forzado – Si es enviado por la MG, indica que la terminación se desactiva inmediatamente. Si es enviado por el MGC, la MG desactivará inmediatamente la terminación (OutOfService). En ambos casos se atribuirá el valor desactivado ("OutOfService"), al parámetro ServiceStates y el MGC debe suprimir todos los contextos o recursos asociados con la terminación. ServiceChangeDelay no tiene ningún efecto en el ServiceChangeMethod "Forzado".
- 3) Gradual – Si es enviado por la MG, indica que tras el ServiceChangeDelay la terminación o terminaciones están saliendo del servicio. Si es enviado por el MGC, la MG desactivará la terminación al final del periodo ServiceChangeDelay. Se atribuirá el valor Desactivado ("OutOfService") a la propiedad ServiceStates después de haber expirado dicho periodo o cuando se retiren la terminación o terminaciones de un contexto activo (lo que ocurra primero). El MGC debe suprimir todos los contextos o recursos asociados con la terminación. Cuando se emplee un ServiceChangeDelay igual a cero, o cuando éste no esté presente, significa que la terminación saldrá de servicio al suprimirse del contexto mediante la instrucción Substract. El MGC no empleará la terminación o terminaciones indicadas para la conexión si no se ha cancelado la instrucción Gradual ni se ha puesto de nuevo en servicio la terminación por una instrucción ServiceChange ulterior. Para cancelar una instrucción ServiceChange ya enviada (de la que se acusó recibo) con ServiceChangeMethod "Gradual", la entidad que inicia el método Gradual envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio" y ServiceChangeReason 918 ("Cancelación de instrucción gradual").
- 4) Cambio por error – No se utilizará ServiceChangeMethod "Cambio por error" con terminaciones que no sean Root.

- 5) Traspaso – No se utilizará ServiceChangeMethod "Traspaso" con terminaciones que no sean Root.
- 6) Desconectado – No se utilizará ServiceChangeMethod "Desconectado" con terminaciones que no sean Root.

F.4.1.3 Comportamiento del método ServiceChange en las terminaciones efímeras

Mientras la MG está en una asociación de control activa, una instrucción ServiceChange en una terminación efímera produce diferentes resultados, dependiendo del ServiceChangeMethod, como se describe a continuación:

- 1) Reinicio – Si es enviado por la MG, anuncia que se ha(n) reiniciado la(s) terminación(es) o que hay un cambio de capacidades. ServiceChangeReason indica qué acciones podría tener que emprender el MGC. Éste no enviará ServiceChangeMethod "Reinicio" para terminaciones efímeras.
- 2) Forzado – Si es enviado por la MG, indica que la terminación o terminaciones están saliendo de servicio inmediatamente. El MGC debe retirar la terminación. El MGC no enviará ServiceChangeMethod "Forzado" para terminaciones efímeras. ServiceChangeDelay no tiene ningún efecto en el ServiceChangeMethod "Forzado".
- 3) Gradual – Si es enviado por el MG, indica que tras el periodo ServiceChangeDelay la terminación o terminaciones están saliendo del servicio. El MGC debe retirar la terminación o terminaciones al expirar dicho periodo. El MGC no enviará ServiceChangeMethod "Gradual" para terminaciones efímeras. Cuando se emplee un ServiceChangeDelay igual a cero, se indica que se ha de destruir la terminación al retirarla del contexto mediante la instrucción Substract. La MG debería atribuir el valor Desactivado "Out of Service" a la propiedad ServiceStates de la terminación al expirar ServiceChangeDelay, o cuando se suprime la terminación de un contexto activo (lo que ocurra primero). Para cancelar una instrucción ServiceChange ya enviada (de la que se acusó recibo) con ServiceChangeMethod "Gradual", la entidad que inicia el método Gradual envía una instrucción ServiceChange con ServiceChangeMethod "Reinicio" y ServiceChangeReason 918 ("Cancelación de instrucción gradual").
- 4) Cambio por error – No se utilizará ServiceChangeMethod "Cambio por error" con terminaciones que no sean Root.
- 5) Traspaso – No se utilizará ServiceChangeMethod "Traspaso" con terminaciones que no sean Root.
- 6) Desconectado – No se utilizará ServiceChangeMethod "Desconectado" con terminaciones que no sean Root.

F.5 Utilización de los parámetros de cambio de servicio

F.5.1 Método de cambio de servicio (ServiceChangeMethod)

Véase la descripción de F.4.

F.5.2 Motivo de cambio de servicio (ServiceChangeReason)

Este parámetro permite a la parte receptora adoptar el comportamiento adecuado en determinadas situaciones. Por ejemplo, si una MG envía un ServiceChangeReason 901 ("Reinicio en frío") al MGC reiniciar, el MGC podría suponer que la MG ha perdido todo su estado y, por ende, no realizaría las acciones de mantenimiento y auditoría para evaluarla y hacerla pasar a un estado útil.

En el cuadro F.1 se señala con cuáles ServiceChangeMethods se pueden enviar determinados ServiceChangeReasons.

Cuadro F.1/H.248.1 – Los métodos (ServiceChangeMethod) y los motivos (ServiceChangeReason) que pueden invocarse

Motivo de cambio de servicio	Método de cambio de servicio						Descripción
	Reinicio	Forzado	Gradual	Desconectado	Cambio por error	Traspaso	
900	X			Root solamente MG solamente			Servicio restaurado
901	Root solamente						Reinicio en frío
902	Root solamente						Reinicio en caliente
903						Root solamente	Cambio dirigido por el MGC
904		X	X				La terminación funciona mal
905		X	X				Terminación fuera del servicio
906		X	X				Pérdida de conectividad de capa inferior
907		X	X				Fallo de transmisión
908		Root solamente MG solamente	Root solamente MG solamente		Root solamente MG solamente		Fallo inminente de la MG
909					Root solamente MG solamente		Fallo inminente del MGC
910	MG solamente	X	X				Fallo de capacidad de medios
911	MG solamente	X	X				Fallo de capacidad de módem
912	MG solamente	X	X				Fallo de capacidad de multiplexación
913	MG solamente	X	X				Fallo de capacidad de señal
914	MG solamente	X	X				Fallo de capacidad de evento
915		X	X				Pérdida de estado

Cuadro F.1/H.248.1 – Los métodos (ServiceChangeMethod) y los motivos (ServiceChangeReason) que pueden invocarse

Motivo de cambio de servicio	Método de cambio de servicio						Descripción
	Reinicio	Forzado	Gradual	Desconectado	Cambio por error	Traspaso	
916	X			Root solamente MG solamente	Root solamente MG solamente		Cambio de paquetes
917	X			Root solamente MG solamente	Root solamente MG solamente		Cambio de capacidad
918	X						Cancelación de gradual
919					Root solamente MG solamente		Cambio por error, en caliente
920					Root solamente MG solamente		Cambio por error, en frío

F.5.3 Retardo de cambio de servicio (ServiceChangeDelay)

Tiempo que transcurre antes de que tenga efecto la instrucción ServiceChange y se cambie la propiedad ServiceStates de la MG o las terminaciones. Se responde a esta instrucción como si se hubiera actuado al recibir el mensaje, pero la MG o la(s) terminación(es) sólo cambian de estado después de este retardo. ServiceChangeDelay cero equivale a no especificar ningún retardo, salvo si se emplea en combinación con el ServiceChangeMethod "Gradual". En F.4 se describen los efectos ServiceChangeDelay en los distintos métodos (ServiceChangeMethods).

F.5.4 Dirección de cambio de servicio (ServiceChangeAddress)

La utilización de este parámetro se describe en 7.2.8. No se recomienda emplearlo. Si lo hubiere, se puede utilizar solamente con instrucciones ServiceChange en la terminación Root, y se debe enviar toda nueva transacción a la nueva dirección y/o número de puerto especificados. La respuesta se enviará a la dirección desde la cual proviene esta petición.

F.5.5 Perfil de cambio de servicio (ServiceChangeProfile)

Este parámetro permite que el MGC y la MG negocien el perfil H.248 que se ha de utilizar en la asociación de control. El MGC también puede efectuar una auditoría de la MG para establecer cuáles otras capacidades soporta. ServiceChangeProfile se enviará solamente en las instrucciones de registro y nuevo registro.

F.5.6 Versión de cambio de servicio (ServiceChangeVersion)

Se emplea para negociar la versión de protocolo H.248 que se va a utilizar entre el MGC y la MG. ServiceChangeVersion es obligatorio en el registro inicial, y es conveniente enviarlo en cualquier otra instrucción de registro que implique una negociación de protocolo. ServiceChangeVersion no se enviará en instrucciones que no sean de registro. En F.3.1 se describen los procedimientos de registro.

F.5.7 Identificador de MGC de cambio de servicio (ServiceChangeMgcID)

El MGC puede enviar este parámetro en una instrucción ServiceChange hacia la terminación Root. Si lo recibe cuando intenta registrarse, la MG tratará de registrarse con el MGC en la dirección especificada. Cuando se recibe en una instrucción de Traspaso del MGC primario de la MG, la MG se servirá de los procedimientos descritos en 11.5.

F.5.8 Indicación de tiempo (TimeStamp)

En 7.2.8 se describe la utilización de este parámetro facultativo. Si bien TimeStamp no tiene ningún efecto en las instrucciones ServiceChange, puede ser útil para otros fines al receptor de éstas, por ejemplo para la coordinación del proceso de facturación o de la temporización.

F.6 ServiceChange comparado con TerminationState

La propiedad ServiceStates en el descriptor de estado de terminación equivale al valor actual del estado de la terminación. ServiceStates puede tener tres valores, a saber en servicio ("InService"), fuera de servicio ("OutOfService") y prueba ("Test").

El MGC sólo podrá utilizar la instrucción Modificar cuando el estado original sea Prueba o cuando la modificación produzca dicho estado. Cualquier otra modificación de la propiedad ServiceStates constituye una violación del protocolo y producirá un código de error 401 ("Error de protocolo"). Para cambiar del estado "en servicio" al estado "fuera de servicio" o viceversa, se debe utilizar la instrucción ServiceChange.

En el caso de terminaciones en Prueba, la MG sólo enviará al MGC ServiceChanges con ServiceChangeMethod "Forzado" o "Gradual". Entonces podrán enviarse otras instrucciones ServiceChange para poner la propiedad ServiceState de la terminación en el estado Activado (InService).

Aunque conviene que los receptores hagan todos los esfuerzos razonables para aceptar instrucciones ServiceChange, en algunos casos se justifica el rechazo de éstas. Por ejemplo, si una MG envía una instrucción ServiceChange para poner en servicio una terminación, pero el MGC no dispone de recursos atribuidos para ella, bien sea por configuración u otra razón, el MGC puede rechazarla. Cuando la entidad receptora rechaza la instrucción ServiceChange, el estado de la terminación no cambia. El remitente puede ya sea dejar pasar un cierto tiempo y tratar de nuevo o esperar a que el receptor intente modificar el estado de la terminación. Las peticiones de desactivación de una terminación deberían ser satisfechas.

En el cuadro F.2 se indican las instrucciones que pueden servir para efectuar un cambio de estado en una terminación.

Cuadro F.2/H.248 – Instrucciones de transición de estado y sus resultados

Estado actual	Nuevo estado	Instrucción	¿Permitida por el MGC?	¿Permitida por la MG?
InService	Test	ServiceChange	No es posible	No es posible
		Modificar	Sí	No es posible
InService	OutOfService	ServiceChange	Sí, sólo en Root y física	Sí
		Modificar	No	No es posible
OutOfService	InService	ServiceChange	Sí, pero no en Root	Sí
		Modificar	No	No es posible
OutOfService	Test	ServiceChange	No es posible	No es posible
		Modificar	Sí	No es posible
Test	InService	ServiceChange	No	No
		Modificar	Sí	No es posible
Test	OutOfService	ServiceChange	No	Sí
		Modificar	Sí	No es posible

Apéndice I

Ejemplo de flujos de llamadas

Todos los implementadores de sistemas H.248.1 deben leer atentamente la parte normativa de esta Recomendación antes de utilizarlo en una implementación. No se deben utilizar los ejemplos de este apéndice como explicaciones completas sobre la manera de crear mensajes de protocolo.

En los ejemplos de este apéndice se utiliza el protocolo SDP para codificar los descriptores de tren local y distante. SDP se define en RFC 2327. Si hay una discrepancia entre el SDP utilizado en estos ejemplos y RFC 2327, se debe consultar RFC para precisar lo que es correcto. Los perfiles de audio utilizados son los definidos en RFC 1890, y otros registrados en la IANA. Por ejemplo, la codificación con modulación por impulsos codificados según la ley A, G.711, se designa por PCMA en SDP, y se le asigna el perfil 0. La codificación según G.723.1 se designa por G723 y se le asigna el perfil 4, la codificación según H.263 se designa por H263 y se le asigna el perfil 34. Véase también <http://www.iana.org/assignments/rtp-parameters>.

I.1 Llamada entre dos pasarelas residenciales

Este ejemplo ilustra la utilización de los elementos del protocolo para establecer una comunicación entre dos pasarelas residenciales a través de una red basada en IP. Para simplificar la explicación, en este ejemplo se supone que las dos pasarelas residenciales que intervienen en la llamada están controladas por el mismo controlador de pasarela de medios.

I.1.1 Programación de terminaciones de línea analógica de pasarela residencial para comportamiento en reposo

A continuación se presentan las invocaciones de API por el controlador de pasarela de medios y por las pasarelas de medios para programar las terminaciones de este ejemplo para comportamiento en reposo. Tanto la pasarela de medios de origen como la de terminación tienen terminaciones de línea analógica en reposo programadas para que busquen eventos de iniciación de llamada (es decir, eventos de descolgado), utilizando la instrucción modificar con los parámetros apropiados. Se

utiliza el contexto NULL para indicar que las terminaciones todavía no participan en un contexto. Se utiliza la terminación Root para indicar que se trata de la MG completa y no de una terminación dentro de la MG.

En este ejemplo, MG1 tiene la dirección IP 124.124.124.222, MG2 tiene la dirección IP 125.125.125.111, y el MGC tiene la dirección IP 123.123.123.4. El puerto Megaco es 55555 para los tres.

1) Una MG se registra en un MGC utilizando la instrucción ServiceChange:

MG1 to MGC:

```
MEGACO/1 [124.124.124.222]
Transaction = 9998 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Restart, Version=3,
      ServiceChangeAddress=55555, Profile=ResGW/1}
    }
  }
}
```

2) El MGC envía una contestación:

MGC to MG1:

```
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
  Context = - {ServiceChange = ROOT {
    Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}
```

3) El MGC programa una terminación en el contexto NULL. El TerminationID es A4444; el streamID es 1; el requestId en el descriptor de eventos es 2222. El MID es el identificador del emisor de este mensaje; en este caso, es la dirección IP y puerto [123.123.123.4]:55555. La propiedad modo para este tren se fija a SendRecv. "al" es el paquete de supervisión de línea analógica. Se supone que se han configurado los descriptores Local y Remote.

MGC to MG1:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 9999 {
  Context = - {
    Modify = A4444 {
      Media { Stream = 1 {
        LocalControl {
          Mode = SendRecv,
          tdmc/gain=2, ; in dB,
          tdmc/ec=on
        },
      },
    },
    Events = 2222 {al/of {strict=state}}
  }
}
```

El guión del plan de marcación podría haberse cargado previamente en la MG. Su función sería estar en espera de descolgado, activar el tono de invitación a marcar y comenzar a tomar los dígitos DTMF. Sin embargo, en este ejemplo se ha utilizado el mapa de dígitos, que se coloca en posición después al detectarse el evento de descolgado (paso 5) más adelante).

Obsérvese que el descriptor de eventos insertado podría haberse utilizado para combinar los pasos 3) y 4) con los pasos 8) y 9), y eliminar así los pasos 6) y 7).

- 4) El MG1 acepta la instrucción Modificar enviando esta contestación:

```
MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 9999 {
  Context = - {Modify = A4444}
}
```

- 5) Un intercambio similar tiene lugar entre la MG2 y el MGC, obteniéndose como resultado una terminación en reposo denominada A5555.

I.1.2 Toma de dígitos del originador e iniciación de la terminación

Lo siguiente se basa en las condiciones anteriormente mostradas. Se presentan las transacciones procedentes del controlador de pasarela de medios y de la pasarela de medios de origen (MG1, *originating media gateway*) para hacer que la terminación de origen (A4444) pase a través de las etapas de toma de dígitos requeridas para iniciar una conexión con la pasarela de medios de terminación (MG2, *terminating media gateway*).

- 6) La MG1 detecta un evento de descolgado procedente del usuario 1 y lo informa al controlador de pasarela de medios mediante una instrucción Notificar.

```
MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10000 {
  Context = - {
    Notify = A4444 {ObservedEvents =2222 {
      19990729T22000000:al/of(init=OFF)}}
  }
}
```

- 7) Y se acusa recibo de la instrucción Notificar.

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10000 {
  Context = - {Notify = A4444}
}
```

- 8) El MGC modifica la terminación para reproducir el tono de marcar, buscar dígitos de conformidad con Dialplan0 y buscar el evento colgado ahora.

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10001 {
  Context = - {
    Modify = A4444 {
      Events = 2223 {
        al/on(strict=state), dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
(0| 00| [1-7]xxx| 8xxxxxxx| Fxxxxxxx| Exx| 91xxxxxxxxxxx| 9011x.)}
      }
    }
  }
}
```

- 9) Y se acusa recibo de la instrucción Modificar.

```
MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10001 {
  Context = - {Modify = A4444}
}
```

- 10) Seguidamente, la MG1 acumula los dígitos a medida que son marcados por el usuario 1. Cuando se obtiene una concordancia apropiada de los dígitos tomados con el plan de marcación de números para A4444, se envía una nueva instrucción Notificar al controlador de pasarela de medios.

```
MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10002 {
  Context = - {
    Notify = A4444 {ObservedEvents =2223 {
      19990729T22010001:dd/ce{ds="916135551212",Meth=UM}}}
  }
}
```

- 11) Y se acusa recibo de la instrucción Notificar.

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10002 {
  Context = - {Notify = A4444}
}
```

- 12) El controlador analiza entonces los dígitos y determina que es necesario establecer una conexión de MG1 a MG2. Tanto la terminación TDM A4444 como la terminación RTP se añaden a un nuevo contexto en MG1. Todavía está en modo RecvOnly porque no se han especificado valores de descriptor distante. Los códecs preferidos están en el orden de preferencia por el MGC.

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10003 {
  Context = $ {
    Add = A4444,
    Add = $ {
      Media {
        Stream = 1 {
          LocalControl {
            Mode = RecvOnly,

            nt/jit=40 ; in ms
          },
          Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
          }
        }
      }
    }
  }
}
```

NOTA – El MGC expresa sus valores de parámetros preferidos como una serie de bloques SDP en Local. La MG llena el descriptor Local en la contestación.

- 13) La MG1 acusa recibo de la nueva terminación y llena la dirección IP local en el puerto UDP. También elige un códec basándose en las preferencias del MGC en local. La MG1 fija el puerto RTP a 2222.

```
MEGACO/3 [124.124.124.222]:55555
Reply = 10003 {
  Context = 2000 {
    Add = A4444,
    Add=A4445{
      Media {
        Stream = 1 {
          Local {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
a=recvonly
          } ; RTP profile for G.723.1 is 4
        }
      }
    }
  }
}
```

- 14) El MGC asocia ahora A5555 con el nuevo contexto en MG2, y establece un tren RTP (esto es, se reasigna A5556), quedando establecida la conexión SendRecv hasta el usuario de origen, usuario 1. El MGC también arregla el timbre en A5555.

```
MGC to MG2:
MEGACO/3 [123.123.123.4]:55555
Transaction = 50003 {
  Context = $ {
    Add = A5555 { Media {
      Stream = 1 {
        LocalControl {Mode = SendRecv} }},
      Events=1234{al/of(strict=state)},
      Signals {al/ri}
    },
    Add = $ {Media {
      Stream = 1 {
        LocalControl {
          Mode = SendRecv,
          nt/jit=40 ; in ms
        },
        Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
        },
        Remote {
v=0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
        } ; RTP profile for G.723.1 is 4
      }
    }
  }
}
```


- 15) Se acusa recibo. El número de puerto de tren es diferente del número del puerto de control. En este caso es 1111 (en SDP).

```
MG2 to MGC:
MEGACO/3 [125.125.125.111]:55555
Reply = 50003 {
  Context = 5000 {
    Add = A5555,
    Add = A5556{
      Media {
        Stream = 1 {
          Local {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
}
      } ; RTP profile for G.723.1 is 4
    }
  }
}
```

- 16) Ahora es necesario comunicar a MG1 la dirección IP y el puerto UDP antes mencionados.

```
MGC to MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {cg/rt}
    },
    Modify = A4445 {
      Media {
        Stream = 1 {
          Remote {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
}
      } ; RTP profile for G.723.1 is 4
    }
  }
}

MG1 to MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10005 {
  Context = 2000 {Modify = A4444, Modify = A4445}
}
```

- 17) Las dos pasarelas están ahora conectadas y el usuario 1 oye el tono de llamada. La MG2 espera ahora hasta que el usuario 2 descuelgue, en cuyo momento quedará establecida la comunicación en ambos sentidos.

From MG2 to MGC:

```
MEGACO/3 [125.125.125.111]:55555
Transaction = 50005 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1234 {
      19990729T22020002:al/of(init=off)}}
  }
}
```

From MGC to MG2:

```
MEGACO/3 [123.123.123.4]:55555
Reply = 50005 {
  Context = - {Notify = A5555}
}
```

From MGC to MG2:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 50006 {
  Context = 5000 {
    Modify = A5555 {
      Events = 1235 {al/on(strict=state)},
      Signals ; to turn off ringing
    }
  }
}
```

From MG2 to MGC:

```
MEGACO/3 [125.125.125.111]:55555
Reply = 50006 {
  Context = 5000 {Modify = A4445}
}
```

- 18) Se cambia el modo en MG1 a SendRecv y se detiene la señal de llamada de retorno.

MGC to MG1:

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 10006 {
  Context = 2000 {
    Modify = A4445 {
      Media {
        Stream = 1 {
          LocalControl {
            Mode=SendRecv
          }
        }
      }
    }
  },
  Modify = A4444 {
    Signals
  }
}
```

from MG1 to MGC:

```
MEGACO/3 [124.124.124.222]:55555
Reply = 10006 {
  Context = 2000 {Modify = A4445, Modify = A4444}}
```

19) El MGC decide efectuar una auditoría en la terminación RTP en MG2.

```
MEGACO/3 [123.123.123.4]:55555
Transaction = 50007 {
  Context = 5000 {AuditValue = A5556{
    Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
  }
}
```

20) El MG2 contesta.

```
MEGACO/3 [125.125.125.111]:55555
Reply = 50007 {
  Context = 5000 {
    AuditValue = A5556 {
      Media {
        TerminationState { ServiceStates = InService,
          Buffer = OFF },
        Stream = 1 {
          LocalControl { Mode = SendRecv,
            nt/jit=40 },
          Local {

```

v=0

```
o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
a=ptime:30
      },
      Remote {

```

v=0

```
o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
      } } },
      Events,
      Signals,
      DigitMap,
      Packages {nt-1, rtp-1},
      Statistics { rtp/ps=1200, ; packets sent
        nt/os=62300, ; octets sent
        rtp/pr=700, ; packets received
        nt/or=45100, ; octets received
        rtp/pl=0.2, ; % packet loss
        rtp/jit=20,
        rtp/delay=40 } ; avg latency
    }
  }
}
```

```

}
}
```

21) Cuando el MGC recibe una señal de colgado de una de las MG, suprime la llamada. En este ejemplo, el primero que cuelga es el usuario en MG2.

From MG2 to MGC:

```
MEGACO/3 [125.125.125.111]:55555
```

```

Transaction = 50008 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1235 {
      19990729T24020002:al/on(init=off)}
    }
  }
}

```

From MGC to MG2:

```

MEGACO/3 [123.123.123.4]:55555
Reply = 50008 {
  Context = - {Notify = A5555}
}

```

- 22) El MGC envía ahora a ambas MG una instrucción Substraer para suprimir la llamada. En este ejemplo sólo se muestra la instrucción Substraer enviada a MG2. Cada terminación reúne su propio conjunto de estadísticas. Es posible que el MGC no tenga que pedir que se le envíen ambos conjuntos. A5555 es una terminación física, y A5556 es una terminación RTP.

From MGC to MG2:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 50009 {
  Context = 5000 {
    Subtract = A5555 {Audit{Statistics}},
    Subtract = A5556 {Audit{Statistics}}
  }
}

```

From MG2 to MGC:

```

MEGACO/3 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
    Subtract = A5555 {
      Statistics {
        nt/os=45123, ; Octets Sent
        nt/or=45123, ; Octets Received
        nt/dur=40000 ; in milliseconds
      }
    },
    Subtract = A5556 {
      Statistics {
        rtp/ps=1245, ; packets sent
        nt/os=62345, ; octets sent
        rtp/pr=780, ; packets received
        nt/or=45123, ; octets received
        rtp/pl=10, ; % packets lost
        rtp/jit=27,
        rtp/delay=48, ; average latency
        nt/dur=38000 ; in millisec
      }
    }
  }
}

```

- 23) El MGC configura ahora ambas pasarelas, MG1 y MG2, de modo que estén listas para la detección del próximo evento de descolgado (véase el paso 1). Obsérvese que éste podría ser el estado por defecto de una terminación en el contexto NULL, en cuyo caso el MGC no tendría que enviar ningún mensaje a la MG. La terminación que retorna al contexto NULL toma de nuevo sus valores por defecto.

Apéndice II

Modelo de lote H.248

Se recomienda definir los lotes H.248 nuevos y actualizados utilizando el siguiente modelo (plantilla), que es un formato de Recomendación de la UIT. Los editores de organizaciones que no sigan el formato de la UIT deberían utilizar por lo menos las estructuras de la cláusula 6. Se escriben en negritas las palabras clave de los lotes H.248. Las partes que hay que rellenar se indican mediante "< >". En la cláusula 12 se proporciona información más detallada acerca de cómo crear un nuevo paquete.

NOTA – Para los códigos de error específicos de lotes es necesario el registro IANA. La atribución de los códigos de error ha de obedecer las reglas descritas en 4.1/H.248.8, "Asignación de códigos de error".

Recomendación UIT-T H.248.<xxx>

Protocolo de control de pasarela: Lotes <xxx>

1 Alcance

<El alcance o limitaciones del lote>

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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

- Recomendación UIT-T H.248.1 (<xx/xxxx>), *Protocolo de control de pasarela: Versión <x>*
- <Otras referencias>

3 Términos y definiciones

En esta Recomendación se utilizan los siguientes términos y definiciones.

<Términos y definiciones>

4 Abreviaturas, siglas o acrónimos

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

<Abreviaturas, siglas o acrónimos>

5 Convenios

<Convenios>

- 6** **<Título de lote>**
Nombre de lote: <nombre>
PackageID: <PackageID "texto", binario <0x????>>
Descripción: <descripción>
Versión: <versión>
Extiende: <PackageID y versión extendidos>
- 6.1** **Propiedades**
- 6.1.1** **<Título de propiedad>**
Nombre de propiedad: <nombre>
PropertyID: <ID texto, ID binario (0x????)>
Descripción: <descripción>
Tipo: <tipo>
Valores posibles: <valores>
Valor por defecto: <valor>
Definido en: <Local, Remote, LocalControl, TerminationState, ContextAttribute>
Características: <Lectura solamente, Lectura/escritura>
- 6.2** **Eventos**
- 6.2.1** **<Título de evento>**
Nombre de evento: <nombre de evento>
EventID: <(ID texto), (ID binario (0x????))>
Descripción: <descripción>
- 6.2.1.1** **Parámetros del descriptor de eventos**
- 6.2.1.1.1** **<Título de parámetro>**
Nombre de parámetro: <nombre>
ParameterID: <(ID texto), (ID binario (0x????))>
Descripción: <descripción>
Tipo: <tipos>
Facultativo: <Sí/no>
Valores posibles: <valores>
Valor por defecto: <valor>
- 6.2.1.2** **Parámetros del descriptor de eventos observados**
- 6.2.1.2.1** **<Título de parámetro>**
Nombre de parámetro: <nombre>
ParameterID: <(ID texto), (ID binario (0x????))>
Descripción: <descripción>
Tipo: <tipos>
Facultativo: <Sí/no>

Valores posibles: <valores>

Valor por defecto: <valor>

6.3 Señales

6.3.1 <Título de señal>

Nombre de señal: <nombre>

SignalID: <(ID texto), (ID binario (0x????))>

Descripción: <descripción>

Tipo de señal: <tipo>

Duración: <duración>

6.3.1.1 Parámetros adicionales

6.3.1.1.1 <Título de parámetro>

Nombre de parámetro: <nombre>

ParameterID: <(ID texto), (ID binario (0x????))>

Descripción: <descripción>

Tipo: <tipos>

Facultativo: <Sí/no>

Valores posibles: <valores>

Valor por defecto: <valor>

6.4 Estadísticas

6.4.1 <Título de estadística>

Nombre de estadística: <nombre>

Statistic ID: <(ID texto), (ID binario (0x????))>

Descripción: <descripción>

Tipo: <tipo>

Valores posibles: <valores>

Nivel: <Terminación, Tren, Ambos>

6.5 Códigos de error

6.5.1 <Título de código de error>

Código de error #: <número>

Nombre de código de error: <nombre>

Definición: <definición>

Mensaje de texto en el descriptor de error: <el mensaje que se produce en retorno>

Observaciones: <observaciones>

6.6 Procedimientos

<Procedimientos asociados con el paquete>

Apéndice III

Plantilla para la definición de un perfil H.248

Se recomienda definir los nuevos perfiles H.248 utilizando el siguiente modelo de perfil, que es un formato de Recomendación de la UIT. Los editores de organizaciones que no sigan el formato de la UIT deberían utilizar por lo menos las estructuras de la cláusula 6. Cuando no se utilice esta plantilla, los editores deberían garantizar que sus perfiles incluyen los títulos y los puntos del presente apéndice.

Algunos de los títulos de esta estructura son elementos facultativos. Puede ocurrir que en determinada definición de perfil no se utilice algún elemento H.248.1, a pesar de ser obligatorio en dicha Recomendación. Estos elementos también se incluyen en la estructura de este apéndice, y los que no se enumeran han de ser considerados obligatorios, conforme al protocolo H.248.1. En esta plantilla de protocolo los elementos pueden calificarse como "facultativos" u "obligatorios", indicando en el primer caso que es potestad del emisor o del receptor incluirlos en el mensaje. Si la entidad receptora recibe un elemento facultativo que no ha implementado conforme a 6.2.3, debería enviar el código de error 501 ("No implementado").

Es recomendable que el editor añada una descripción escrita adicional a algunas de las siguientes subcláusulas, cuando esto permita aclarar aún más su comportamiento H.248.1. Por ejemplo, si la instrucción Desplazar (véase 6.8.4) se circunscribe a determinados tipos de terminaciones, conviene indicarlo.

Hay que suprimir el texto que va en cursivas.

El texto a incluir debe ir entre los paréntesis angulares (<texto>).

Recomendación UIT-T <xxx>

<Título de perfil>

1 Alcance

<El alcance o limitaciones del este perfil>

2 Referencias

<Referencias>

3 Términos y definiciones

En esta Recomendación se utilizan los siguientes términos y definiciones.

<Términos y definiciones>

4 Abreviaturas, siglas o acrónimos

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

<Abreviaturas, siglas o acrónimos>

5 Convenios

<Convenios>

6 Descripción de perfil

6.1 Identificación de perfil

Nombre de perfil:	<nombre 1-64 caracteres>
Versión:	<versión 1-99>

El nombre y la versión de perfil que se envían en la instrucción de cambio de servicio.

6.2 Resumen

<Descripción>

Descripción del perfil.

6.3 Versión de protocolo de control de pasarela

<número de versión>

La versión mínima de H.248 que se requiere para soportar este perfil, basándose en la sintaxis de base y no en una atribución arbitraria. Tiene que ver con la ServiceChangeVersion de 6.8.8.

6.4 Modelo de conexión

<Utilizar texto y diagramas>

Descripción de las configuraciones permitidas en un contexto.

Número máximo de contextos:	<Entero>
Número máximo de terminaciones por contexto:	<Entero>
Combinaciones de tipo de terminación permitidas en un contexto:	<por ejemplo Context[a](IP,TDM), Context[b](TDM,AAL2), etc.>

6.5 Atributos de contexto

Atributo de contexto	Soportado	Valores soportados
Topología	<Sí/No>	Véase 6.7.8
Indicador de prioridad	<Sí/No>	<1-15>
Indicador de emergencia	<Sí/No>	NA
Indicador IEPS	<Sí/No>	NA
Descriptor de atributo de contexto	<Sí/No>	Si es "Sí", véase 6.8.9 para detalles sobre los atributos soportados.
Parámetro ContextIDList	<Sí/No>	NA

¿Soporta el atributo de contexto Y/O (AND/OR) en la operación Escoger (Select)?

Atributo de contexto Y/O (AND/OR)	<Sí/No>	<Y/O/AMBOS (AND/OR/BOTH)>
--	---------	---------------------------

6.6 Terminaciones

6.6.1 Nombres de terminación

<Estructura del TerminationID>

Identifica las identidades de terminación asociadas con terminaciones físicas, efímeras y multiplex.

6.6.2 Terminaciones Multiplexadas

¿Soporta terminaciones multiplexadas?	<Sí/No>
---------------------------------------	---------

Si la respuesta es sí:

Soporta tipos múltiplex:	<H.221, H.223, H.226, V.76, N × 64K>
Número máximo de terminaciones conectadas al múltiplex:	<Entero>

¿Se utilizan terminaciones multiplexadas? Si es el caso, hay que describirlo.

6.7 Descriptores

6.7.1 Descriptor de tren

Número máximo de trenes por tipo de terminación	<Tipo de terminación>	<Entero>
---	-----------------------	----------

Si hay más de 1:

Configuración de tren:	<Describir las configuraciones permitidas. ¿Se permite más de un tren de audio? etc.>
------------------------	---

6.7.1.1 Descriptor de control local (LocalControl)

¿Se utilizan las propiedades ReserveGroup y ReserveValue?

<i>Si no, tipos de terminación y tren adecuado de la lista genérica.</i>		Tipo de terminación	Tipo de tren
ReserveGroup utilizado:	<Sí/No>	<Tipo>	<Tipo>
ReserveValue utilizado:	<Sí/No>	<Tipo>	<Tipo>

¿Cuáles valores de modo StreamMode se utilizan?

Tipo de terminación	Tipo de tren	Valores de StreamMode permitidos
<Tipo>	<Tipo>	<SendOnly, RecvOnly, SendRecv, Loopback>

6.7.2 Descriptor de eventos

Es posible validar todos los eventos contenidos en este perfil en cualquier terminación/tren [salvo los siguientes].

NOTA – El texto entre [] es facultativo y se incluye solamente de haber dichas excepciones.

Los eventos se pueden validar en tipos de terminación y tipos de tren:	<Sí/No>		
<i>Si la opción es sí</i>	EventID	Tipo de terminación	Tipo de tren
	<Nombre e identificador de evento, por ejemplo Generic Error Event (g/causa, 0x0001/0x0001)>	<Tipo>	<Tipo de tren, por ejemplo Audio/Vídeo/Datos o StreamID>

¿Se utiliza el control de memoria intermedia de eventos (EventBuffer)?

Utiliza control de memoria intermedia de eventos:	<Sí/No>
--	---------

¿Se utiliza KeepActive (mantienen activo) en eventos?

Utiliza mantienen activo en eventos:	<Sí/No>
---	---------

¿Se utiliza la incorporación en eventos?

Eventos incorporados en un descriptor de eventos:	<Sí/No>
Señales incorporadas en un descriptor de eventos:	<Sí/No>

¿Soporta eventos incorporados (EmbeddedEvents) regulados?

Activación de eventos incorporados en estos casos:	<Ninguno / Un evento particular específico>
---	---

¿Se utiliza la bandera descriptor de restablecimiento (ResetEventsDescriptor)?

ResetEventsDescriptor se utiliza con estos eventos:	<TODOS / Ninguno / Un evento particular específico>
--	---

¿Cuál comportamiento de notificación se soporta?

Notificación inmediata (NotifyImmediate) en:	<TODOS / Ninguno / Un evento particular específico>
Notificación regulada (NotifyRegulated) en:	<TODOS / Ninguno / Un evento particular específico>
Nunca se notifica (NeverNotify) en:	<TODOS / Ninguno / Un evento particular específico>

6.7.3 Descriptor de memoria intermedia de eventos (EventBuffer)

¿Se soporta?

Utiliza descriptor de memoria intermedia de eventos:	<Sí/No>	
<i>Si la opción es sí</i>	Los EventID	<Nombre e identificador de evento, por ejemplo Generic Error Event (g/causa, 0x0001/0x0001) o ALL>

6.7.4 Descriptor de señales

Es posible validar todas las señales contenidas en este perfil en cualquier terminación/tren [salvo las siguientes].

NOTA – El texto entre [] es facultativo y se incluye solamente de haber dichas excepciones.

La validación de señales depende del tipo de terminación y tipo de tren:	<Sí/No> NOTA – "No" significa que se pueden validar todas las señales en cualquier terminación o tren. Con la opción "Sí", cualquier señal que no se incluya en esta lista podrá ser validada en cualquier terminación o tren.		
<i>Si la opción es sí</i>	EventID	Tipo de terminación	Tipo de tren/ID
	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001)>	<Tipo>	<Tipo de tren, por ejemplo Audio/Vídeo/Datos o StreamID>

¿Soporta listas de señales? Si es el caso, ¿cuál es el número máximo de señales en una lista para cada terminación/tren que soporta las listas?

Soporta listas de señales:	<Sí/No>		
<i>Si la opción es sí</i>	Tipo de terminación que soporta listas:	<Tipo/ALL>	
	Tipo de tren que soporta listas:	<Tipo/ALL>	
	Número máximo de señales en una lista de señales	<Entero>	
	Soporta parámetro de retardo entre señales:	<Sí/No>	

¿Soporta tipo y duración de señal prioritarios?

Soporta tipo y duración de señal:	<Sí/No>	
<i>Si la opción es sí</i>	SignalID	Tipo o duración prioritarios
	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001) o "ALL">	<Tipo, Duración, Ambos>

¿Soporta el sentido de señal?

Soporta sentido de señal:	<Sí/No>
----------------------------------	---------

¿Soporta la opción "Notificar al finalizar" ("notifyCompletion")? ¿Cuáles tipos? ¿Se utiliza el identificador de petición (RequestID) con la opción "Notificar al finalizar"?

Soporta NotifyCompletion:	<Sí/No>	
<i>Si la opción es sí</i>	SignalID	Tipo de ejecución completa soportado
	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001) o ALL>	<ALL, TO, EV, ED, NC, PI>
Soporta el parámetro RequestID:	<Sí/No>	

¿Es posible emitir simultáneamente varias señales?

Señales emitidas simultáneamente:	<Sí/No>	
<i>Si la opción es sí</i>	Los SignalID que se pueden emitir simultáneamente:	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001) o ALL>

¿Soporta la opción "Mantener activo" ("KeepActive") para las señales?

Utiliza KeepActive en las señales:	<Sí/No>
---	---------

6.7.5 Descriptor de mapa de dígitos (DigitMap)

¿Soporta mapas de dígitos? Si es el caso, describir los nombres, estructuras y temporizadores.

Soporta mapas de dígitos:	<Sí/No>		
<i>Si la opción es sí</i>	Nombre de mapa de dígitos	Estructura	Temporizadores
	<nombre>	<Describir>	<temporizadores>

6.7.6 Descriptor de estadísticas

¿Soporta las estadísticas en terminaciones, trenes o ambos?

Soporta estadísticas en:	<Terminaciones / Trenes/ Ambos>
---------------------------------	---------------------------------

¿Hay que notificar las estadísticas?

Se notifican las estadísticas en Substraer:	<Sí/No>	
<i>Si la opción es sí</i>	StatisticID notificados:	<Nombre e identificador de estadísticas, por ejemplo Paquetes enviados (rtp/ps, 0x00c/0x0004) o ALL>

6.7.7 Descriptor de eventos observados (ObservedEvents)

¿Soporta tiempo de detección?

Soporta tiempo de detección de evento:	<Sí/No>
---	---------

6.7.8 Descriptor de topología

Si se utiliza ¿cuáles son las configuraciones permitidas?

Tripletas permitidas:	<(T1, T2, unidireccional) etc.>
------------------------------	---------------------------------

6.7.9 Descriptor de error

¿Cuáles son los códigos de error H.248.8 y definidos en lotes que soporta?

Códigos de error enviados por el MGC:

Códigos de error H.248.8 que soporta:	<ALL H.248.8, lista de números>
Códigos de error definidos en lotes que soporta:	La lista de códigos de error se puede encontrar en 6.14.x <Hacer referencia a la subcláusula pertinente>

Códigos de error enviados por la MG:

Códigos de error H.248.8 que soporta:	<ALL H.248.8, lista de números>
Códigos de error definidos en lotes que soporta:	La lista de códigos de error se puede encontrar en 6.14.x <Hacer referencia a la subcláusula pertinente>

6.8 API de instrucciones

NOTA – Se supone que se puede producir en retorno un Descriptor de error en cualquier respuesta a una instrucción.

6.8.1 Añadir (Add)

¿Cuáles descriptores se pueden utilizar en una petición Añadir?

Descriptores que puede incluir una petición Añadir:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
--	---

¿Cuáles descriptores se pueden utilizar en una respuesta Añadir?

Descriptores que puede incluir una respuesta Añadir:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
---	---

6.8.2 Modificar (Modify)

¿Cuáles descriptores se pueden utilizar en una petición Modificar?

Descriptores que puede incluir una petición Modificar:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
---	---

¿Cuáles descriptores se pueden utilizar en una respuesta Modificar?

Descriptores que puede incluir una respuesta Modificar:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
--	---

6.8.3 Substraer (Subtract)

¿Es posible utilizar un Descriptor de auditoría en una petición Substraer?

Descriptores que puede incluir una petición Substraer:	<Audit>
---	---------

¿Es posible utilizar un Descriptor de estadísticas en una respuesta Substraer?

Descriptores que puede incluir una respuesta Substraer:	<Statistics>
--	--------------

6.8.4 Desplazar (Move)

¿Se utiliza la instrucción Desplazar? Es posible que no se utilice en algunas configuraciones de contexto.

Se emplea la instrucción Desplazar:	<Sí/No>
--	---------

Si la opción es sí:

Descriptores que puede incluir una petición Desplazar:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
Descriptores que puede incluir una respuesta Desplazar:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>

6.8.5 Auditoría de valor (AuditValue)

¿Cuáles son los descriptores y/o propiedades, señales, eventos o estadísticas que pueden ser objeto de una auditoría?

Propiedades objeto de auditoría:	<Nombre e identificador de propiedad, por ejemplo Número máximo de contextos (Root/maxNumberOfContexts, 0x0002/0x0001), ALL o Ninguno>	<Descriptores: Local, Remote, LocalControl, TerminationState>
Estadísticas objeto de auditoría:	<Nombre e identificador de estadística, por ejemplo Lotes enviados (rtp/ps, 0x00c/0x0004), ALL o Ninguno>	
Señales objeto de auditoría:	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001), ALL o Ninguno>	
Eventos objeto de auditoría:	<Nombre e identificador de evento, por ejemplo Evento de error genérico (g/cause, 0x0001/0x0001), ALL o Ninguno>	
Posible auditoría de lotes:	¿Se puede efectuar la auditoría del Descriptor de lote? <Sí/No>	

6.8.6 Auditoría de capacidades (AuditCapabilities)

¿Cuáles son los descriptores y/o propiedades, señales, eventos o estadísticas que pueden ser objeto de una auditoría?

Propiedades objeto de auditoría:	<Nombre e identificador de propiedad, por ejemplo Número máximo de contextos (Root/maxNumberOfContexts, 0x0002/0x0001), ALL o Ninguno>	<Descriptores: Local, Remote, LocalControl, TerminationState>
Estadísticas objeto de auditoría:	<Nombre e identificador de estadística, por ejemplo Lotes enviados (rtp/ps, 0x00c/0x0004), ALL o Ninguno>	
Señales objeto de auditoría:	<Nombre e identificador de señal, por ejemplo Playtone (tonegen/pt, 0x0003/0x0001), ALL o Ninguno>	
Eventos objeto de auditoría:	<Nombre e identificador de evento, por ejemplo Evento de error genérico (g/cause, 0x0001/0x0001), ALL o Ninguno>	

¿Es posible una auditoría delimitada?

Propiedades objeto de auditoría/Atributos de contexto que se emplean para la auditoría delimitada:	<Ninguna / ALL / especificar cuáles>
---	--------------------------------------

6.8.7 Notificar (Notify)

¿Cuáles descriptores se pueden utilizar en una instrucción Notificar?

Descriptores que emplea una petición o una respuesta Notificar:	<ObservedEvents, Error>
--	-------------------------

6.8.8 Cambio de servicio (ServiceChange)

¿Cuáles son los métodos y los motivos de cambio de servicio (ServiceChangeMethods y ServiceChangeReasons) que soporta?

Métodos y motivos de cambio de servicio enviados por el MGC:

Métodos que soporta:	Motivos que soporta:
<Gradual, Forzado, Reinicio, Traspaso, Cambio por fallo, TODOS, ¿Otro?>	<900-920>

Métodos y motivos de cambio de servicio enviados por la MG:

Métodos que soporta:	Motivos que soporta:
<Gradual, Forzado, Reinicio, Desconectado, Traspaso, Cambio por fallo, TODOS, ¿Otro?>	<900-920>

¿Se utiliza la dirección de cambio de servicio (*ServiceChangeAddress*)?

Se emplea la dirección de cambio de servicio:	<Sí/No>
--	---------

¿Se utiliza el retardo de cambio de servicio (*ServiceChangeDelay*)?

Se emplea el retardo de cambio de servicio:	<Sí/No>	
<i>Si la opción es sí</i>	Periodo de tiempo válido:	<0-x> ms

¿Se utiliza la bandera cambio de servicio incompleto (*ServiceChangeIncomplete*)?

Se emplea la bandera cambio de servicio incompleto:	<Sí/No>
--	---------

¿Qué versión de H.248.1 utiliza la versión de cambio de servicio (*ServiceChangeVersion*)? El valor mínimo debería coincidir con el número de versión más bajo que se identifica en 6.3.

Versión que se emplea en la <i>ServiceChangeVersion</i>:	<1, 2, 3>
---	-----------

¿Es posible soportar varios perfiles conforme a H.248.18?

Negociación de perfil conforme a H.248.18:	<Sí/No>
---	---------

6.8.9 Manipulación y auditoría de atributos de contexto

¿Cuáles atributos de contexto pueden ser manipulados y/o sujetos a una auditoría?

Se manipulan los atributos de contexto:	<Topología, Emergencia, Prioridad, Indicador IEPS, Descriptor de atributo de contexto (hacer la lista de nombres de atributos), TODOS>
Se realiza una auditoría a los atributos de contexto:	<Topología, Emergencia, Prioridad, Indicador IEPS, Descriptor de atributo de contexto (hacer la lista de nombres de atributos), TODOS>

6.9 Sintaxis y codificación de instrucciones genéricas

Especifica cuáles son las codificaciones que soporta el perfil.

Soporta las codificaciones:	<Texto y Binaria, Binaria, Texto>
------------------------------------	-----------------------------------

6.10 Transacciones

¿Cuál es el máximo de peticiones/respuestas (*TransactionRequest/TransactionReplies*) por mensaje?

Número máximo de <i>TransactionRequests</i> / <i>TransactionReplies</i> / <i>TransResponseAcks</i> / Respuestas de segmento por mensaje:	<Entero>
---	----------

¿Cuál es el máximo de instrucciones en cada petición de transacción (TransactionRequest)?

Número máximo de instrucciones por petición de transacción:	<Entero>
--	----------

¿Cuál es el máximo de instrucciones en cada respuesta de transacción (TransactionReply)?

Número máximo de instrucciones por respuesta de transacción:	<Entero>
---	----------

¿Se pueden calificar las instrucciones como "Facultativas"? Hay que dar detalles.

Instrucciones que se pueden calificar como "Facultativas":	<Añadir, Modificar, Desplazar, Substraer, Auditvalue, Auditcapability, Servicechange, Todas, Ninguna>
---	---

Especificar los valores de los temporizadores de transacción

Temporizador de transacción:	Valor
normalMGExecutionTime	<Entero o "Configurado">
normalMGCEExecutionTime	<Entero o "Configurado">
MGOriantedPendingLimit	<Entero o "Configurado">
MGCOriantedPendingLimit	<Entero o "Configurado">
MGProvisionalResponseTimerValor	<Entero o "Configurado">
MGCProvisionalResponseTimerValor	<Entero o "Configurado">

6.11 Mensajes

Convenios de denominación de MGC/MG: direccionamiento de MID asociado con los nombres de MGC/MG.

<Describirlos aquí>

Indica el número máximo de transacciones por mensaje (este cuadro se puede omitir, si no es importante)

Número máximo de transacciones por mensaje:	<Entero>
--	----------

6.12 Transporte

Especifica qué transportes de la subserie H.248 soporta este perfil.

Transportes que soporta:	<UDP, TCP, SCTP, MTP3B, SSCOP/AAL 5, ALF/AAL 5>
---------------------------------	---

¿Soporta la segmentación y, si así fuere, con qué método?

Soporta la segmentación:	<No, inherente en el transporte, segmentación H.248>
---------------------------------	--

¿Se emplea la supervisión de asociación de control (véase 11.6) y, si es el caso, con qué método?

Soporta la supervisión de asociación de control:	<No, inherente en el transporte, AuditValue vacío en la Root, H.248.14>
---	---

6.13 Seguridad

Especifica los mecanismos de seguridad utilizados.

Soporta la seguridad:	<Ninguno, modelo AH "interim", IPSec>
------------------------------	---------------------------------------

6.14 Lotes

Especifica los lotes que se soportan en este perfil.

Obligatorio: los lotes que este perfil tiene que soportar.

Lotes obligatorios:		
Nombre de lote	PackageID	Versión
<nombre>	<xxxx, (0x00xx)>	<1, 2, 3, ...>

Facultativo: los lotes que este perfil podría soportar.

Lotes facultativos:			
Nombre de lote	PackageID	Versión	Condición para soportarlo
<nombre>	<xxx, 0x00??>	<1, 2, 3, ...>	<Describir>

Información de utilización de lote

Este cuadro especifica cómo se van a utilizar los anteriores lotes. Por ejemplo:

- Enumera las propiedades/señales/eventos/estadísticas que son facultativos u obligatorios
- Si se configura el valor de propiedad/señal/evento, conviene especificarlo (por ejemplo, nombres y número de ciclos para un aviso H.248.7).

Valores de las propiedades que son especificadas como configuradas.

Información de utilización de lote:

6.14.x <Nombre de lote>

Propiedades	Obligatorio/facultativo	Se utiliza en la instrucción:	Valores que soporta:	Valor configurado:
<nombre e identidad, por ejemplo, lotes enviados (rtp/ps, 0x000c/0x0004), ALL o Ninguno>	<M/O>	<Añadir, Modificar, Desplazar, AUDITVALUE, AUDITCAP>	<Valores/ALL>	<Valor/no aplicable>
Señales	Obligatorio/facultativo	Se utiliza en la instrucción:		Valor configurado de duración:
<nombre e identidad>	<M/O>	<Añadir, Modificar, Desplazar, AUDITVALUE, AUDITCAP>		<Valor/no aplicable>
	Parámetros de señal	Obligatorio/facultativo	Valores que soporta:	Valor configurado de duración:
	<nombre e identidad>	<M/O>	<Valores/ALL>	<Valor/no aplicable>

Eventos	Obligatorio/ facultativo	Se utiliza en la instrucción:		
<nombre e identidad>	<M/O>	<Añadir, Modificar, Desplazar, Notificar, AUDITVALUE, AUDITCAP>		
	Parámetros de eventos	Obligatorio/ facultativo	Valores que soporta:	Valor configurado:
	<nombre e identidad>	<M/O>	<Valores / ALL>	<Valor / no aplicable>
	Parámetros de eventos observados	Obligatorio/ facultativo	Valores que soporta:	Valor configurado:
<nombre e identidad>	<M/O>	<Valores/ALL>	<Valor/no aplicable>	
Estadísticas	Obligatorio/ facultativo	Se utiliza en la instrucción:		Valores que soporta:
<nombre e identidad>	<M/O>	<Añadir, Modificar, Desplazar, Substraer, AUDITVALUE, AUDITCAP>		<Valores/ALL>
Códigos de error	Obligatorio/Facultativo			
<número>	<M/O>			

El usuario puede incluir otras restricciones en el cuadro.

6.15 Soporte obligatorio de elementos de información SDP y del anexo C

Especifica qué atributos SDP y elementos de información del anexo C se han de soportar.

Elementos de información SDP y del anexo C que soporta:		
Elemento de información	Soporta el anexo C	Soporta el SDP
<nombre>	<propiedad del anexo C>	<Describir>

6.16 Soporte facultativo de elementos de información SDP y del anexo C

Especifica qué atributos SDP y elementos de información del anexo C se pueden soportar.

Elementos de información SDP y del anexo C facultativos:			
Elemento de información	Soporta el anexo C	Soporta el SDP	Condición para soportarlo:
<nombre>	<propiedad del anexo C>	<Describir>	<Describir>

6.17 Procedimientos

Especifica los procedimientos que se asocian al perfil.

Se recomienda que los procedimientos tengan el siguiente formato:

6.17.1 <Nombre de procedimiento>

Acciones que se realizan cuando sea necesario el <nombre> de procedimiento:

Envío de una instrucción <ADD.req, MOD.req, MOV.req, SUB.req, AuditValue.req, AuditCapability.req, ServiceChange.req, Notify.req> desde el/la <MGC/MG> hasta la/el <MG/MGC> que contenga la siguiente información.

<Poner la información en el formato adecuado. Por ejemplo, los nombres de descriptores, propiedad, señal, evento o estadística, o una abstracción de los parámetros reales, expresados como elementos de información>.

Al recibir la instrucción, la/el <MG/MGC> deberá:

- <Indicar las acciones del caso>

Al finalizar la instrucción de procesamiento (1) se envía una instrucción <ADD.reply, MOD.reply, MOV.reply, SUB.reply, AuditValue.reply, AuditCapability.reply, ServiceChange.reply, Notify.reply> de la/el <MG/MGC> a el/la <MGC/MG> con la siguiente información.

<Poner la información en el formato adecuado. Por ejemplo, los nombres de descriptores, propiedad, señal, evento o estadística, o una abstracción de los parámetros reales, expresados como elementos de información>

SERIES DE RECOMENDACIONES DEL UIT-T

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