UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T H.248.1

(05/2002)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE H: SISTEMAS AUDIOVISUALES Y MULTIMEDIOS

Infraestructura de los servicios audiovisuales – Procedimientos de comunicación

Protocolo de control de las pasarelas: Versión 2

Recomendación UIT-T H.248.1

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.399
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

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

Recomendación UIT-T H.248.1

Protocolo de control de las pasarelas: Versión 2

Resumen

Para obtener una mayor escalabilidad, 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, 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 Rec. UIT-T H.323.

En esta Recomendación se han hecho algunas mejoras con respecto a la Versión 1 de la misma Recomendación:

– auditoría específica de estadísticas, propiedades, señales y eventos;

– mejor tratamiento de multiplexación;

– topología de trenes;

– descripción más completa de los perfiles;

– modificación de ServiceChange.

NOTA – Esta Recomendación está formada por el texto de la Rec. UIT-T H.248 con su nuevo número, sus anexos A a E y su apéndice I.

Orígenes

La Recomendación UIT-T H.248.1, preparada por la Comisión de Estudio 16 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Rec. UIT-T A.8 el 22 de mayo de 2002.

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.

PROPIEDAD INTELECTUAL

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

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

ÍNDICE

Página

Página

Recomendación UIT-T H.248.1

Protocolo de control de las pasarelas: Versión 2

1 Alcance

La presente Recomendación define los protocolos que se utilizan entre los elementos de una pasarela multimedios 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 con 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 con esta Recomendación, dicho producto debe cumplir todas las condiciones obligatorias de la Rec. UIT-T H.248.1 (aprobada en 05/2002).

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) y '2' para referirse a la Rec. UIT-T H.248.1 (05/2002).

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.

2.1 Referencias normativas

– Recomendación UIT-T H.225.0 (2000), *Protocolos de señalización de llamada y paquetización de trenes de medios para sistemas de comunicación multimedios por paquetes*.

– Recomendación UIT-T H.235 (2000), *Seguridad y criptado para terminales multimedios de la serie H (basados en las Recomendaciones H.323 y H.245).*

– Recomendación UIT-T H.245 (2001), *Protocolo de control para comunicación multimedios*.

– Recomendación UIT-T H.246 (1998), *Interfuncionamiento de terminales multimedios de la serie H con terminales multimedios 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.*

– 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 (2002), *Protocolo de control de las pasarelas: Descripción de los códigos de error y de los motivos de cambio de servicio*.

– Recomendación UIT-T H.323 (2000), *Sistemas de comunicación multimedios 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*.

– Recomendación UIT-T I.371 (2000), *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*.

– Recomendación UIT-T Q.765.5 (2000), *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*.

– 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*.

– 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.

– Recomendación UIT-T Q.2965.1 (1999), *Sistema de señalización digital de abonado N.º 2 – Soporte de clases de calidad de servicios*.

– Recomendación UIT-T Q.2965.2 (1999), S*istema 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*.

– Recomendación UIT-T X.213 (2001) | ISO/CEI 8348:2002, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de red* más Enmienda 1 (1997), *Adición del identificador del formato de dirección del protocolo Internet*.

– Recomendación UIT-T X.680 (2002) | ISO/ CEI 8824-1:2002, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*

– 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*.

– 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 (1999), *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*.

– Recomendación UIT-T Q.764 (1999), *Sistema de señalización N.º 7 – Procedimientos de señalización de la parte usuario de la RDSI*.

– Recomendación UIT-T Q.1902.4 (2001), *Protocolo de control de llamada independiente del portador (conjunto de capacidades 2): Procedimientos de llamada básica.*

– 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 1889 (1996), *RTP: A Transport Protocol for Real-Time Applications.*

– IETF RFC 1890 (1996), *RTP Profile for Audio and Video Conferences with Minimal Control.*

– IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol.*

– IETF RFC 2543 (1999), *SIP: Session Initiation Protocol.*

– IETF RFC 2460 (1998), *Internet Protocol, Version 6 (IPv6) Specification.*

– IETF RFC 2805 (2000), *Media Gateway Control Protocol Architecture and Requirements.*

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) como los de la RDSI.

3.2 descriptor: Elemento sintáctico del protocolo que agrupa propiedades conexas. Por ejemplo, las propiedades de un flujo de medios en la pasarela pueden ser fijados por el controlador de pasarela de medios 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 multimedios T.120 solas o en combinación, y podrá efectuar traslaciones de medios dúplex. La MG puede también 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 típicamente 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 típicamente una o dos líneas analógicas y está situada en las instalaciones del cliente.

3.7 pasarela de señalización asociada a la facilidad, de red 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 llamadas y los canales portadores están coubicados en un mismo tramo físico.

3.8 pasarela de señalización no asociada a la facilidad, de red 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 señalización, como 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 termina típicamente un gran número de circuitos digitales.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

ALF	Entramado de nivel de aplicación (*application level framing*)
ATM	Modo de transferencia asíncrono (*asynchronous transfer mode*)
CAS	Señalización asociada al canal (*channel associated signalling*)
DTMF	Multifrecuencia bitono (*dual tone multi-frequency*)
FAS	Señalización asociada a la facilidad (*facility associated signalling*)
GSM	Sistema global para comunicaciones móviles (*global system for mobile communications*)
GW	Pasarela (*gateway*)
IANA	Autoridad de asignación de números Internet (*Internet assigned numbers authority*) reemplazada por ICANN, Corporación Internet para la asignación de nombres y números (*Internet corporation for assigned names and numbers*)
IP	Protocolo Internet (*Internet protocol*)
IVR	Respuesta vocal interactiva (*interactive voice response*)
MG	Pasarela de medios (*media gateway*)
MGC	Controlador de pasarela de medios (*media gateway controller*)
NFAS	Señalización no asociada a la facilidad (*non-facility associated signalling*)
PRI	Interfaz de velocidad primaria (*primary rate interface*)
PU-RDSI	Parte usuario de la red digital de servicios integrados
QoS	Calidad de servicio (*quality of service*)
RCC	Red con conmutación de circuitos
RTP	Protocolo de transporte en tiempo real (*real-time transport protocol*)
RTPC	Red telefónica pública conmutada
SG	Pasarela de señalización (*signalling gateway*)
SS7	Sistema de señalización N.º 7 (*signalling system N.º 7*)

5 Convenios

En esta Recomendación se utilizará el tiempo futuro (modo indicativo) de los verbos para hacer referencia a un requisito obligatorio, y se utilizará el verbo "DEBER" para hacer referencia a una característica o procedimiento sugeridos, pero que son facultativos. Se emplea el verbo "PODER" en el modo subjuntivo o en el modo potencial para hacer referencia una acción facultativa sin hacer alusión a una preferencia.

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 multimedios, una terminación puede ser multimedios 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 *nulo*, que contiene todas las terminaciones que no están asociadas con ninguna otra terminación. Por ejemplo, en una pasarela de acceso descompuesta en sus componentes, todos los canales en reposo se representan por terminaciones en el contexto nulo.

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.

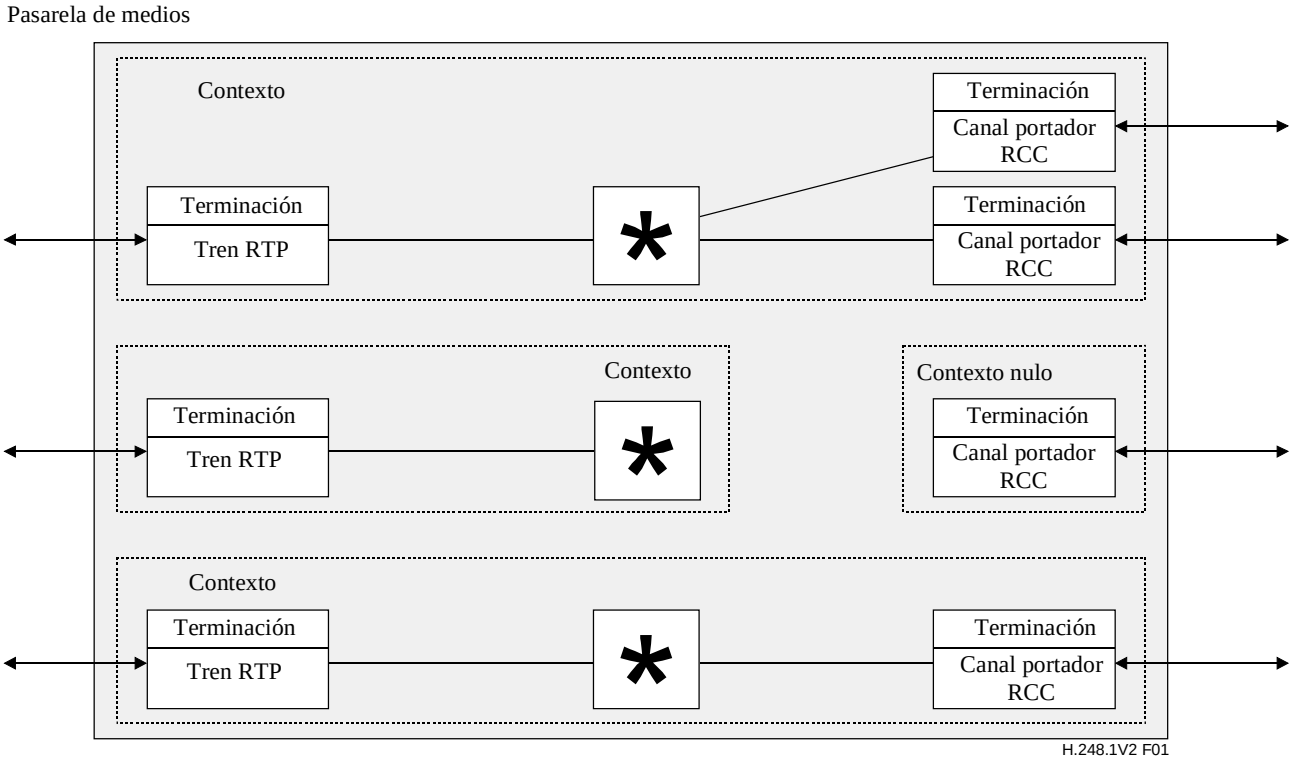


Figura 1/H.248.1 – Ejemplo de modelo de conexión H.248.1

El siguiente ejemplo de la figura 2 representa una forma de obtener un escenario de llamada en espera en una pasarela de acceso descompuesta en sus componentes, 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.

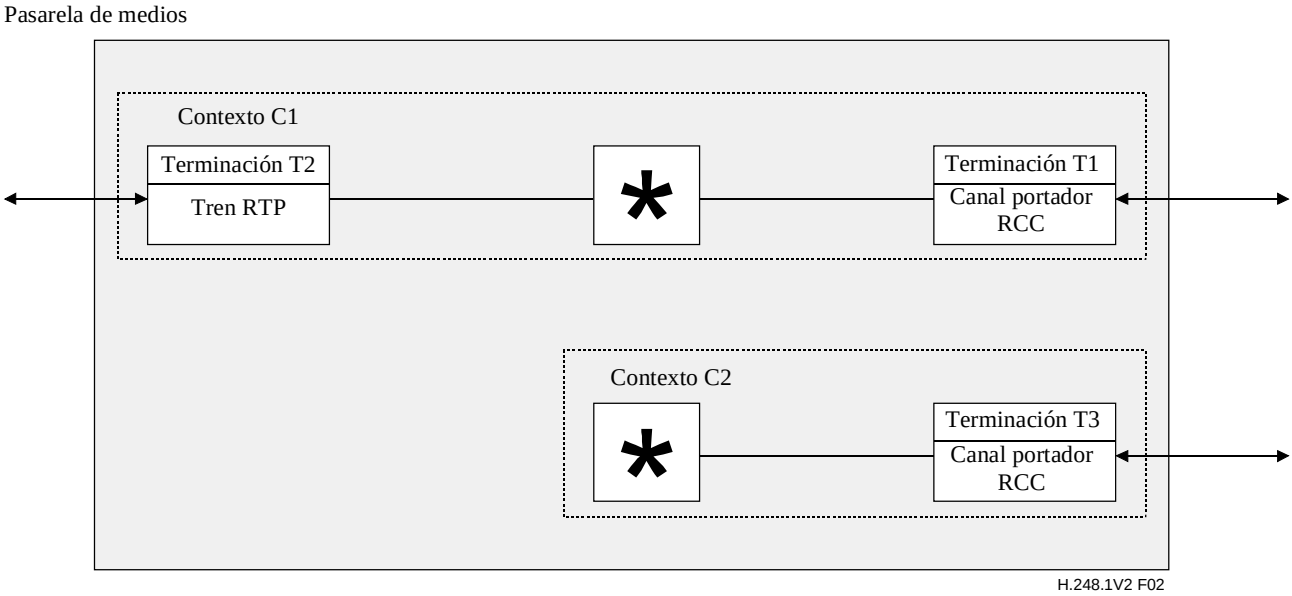


Figura 2/H.248.1 – Ejemplo de escenario de llamada en espera/aviso aplicado a T1

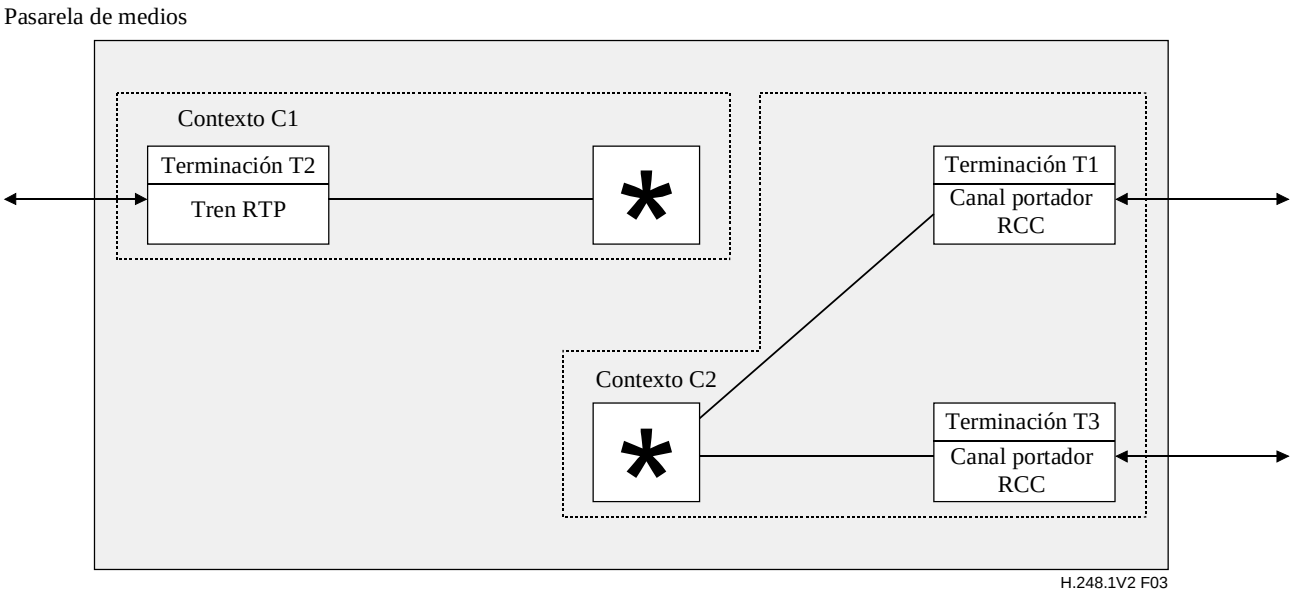


Figura 3/H.248.1 – Ejemplo de escenario de llamada en espera/contestación por T1

6.1 Contextos

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

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

En general, se utiliza una instrucción Añadir 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 y se puede desplazar una terminación de un contexto a otro con la instrucción Desplazar. Una terminación EXISTIRÁ en un solo contexto en cada momento.

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.
- La 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, el modo de una terminación (enviar/recibir/...) describe el flujo de los medios en el ingreso/egreso 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.

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 les substrae o retira 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 provisto 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 substrae de un contexto, se toma del contexto nulo, o se coloca en el contexto nulo, 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 valor), véase 7.2.5] y cuando la terminación se substrae del contexto.

Las pasarelas multimedios 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 MultiplexDescriptor (descriptor 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 StreamDescriptors (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.

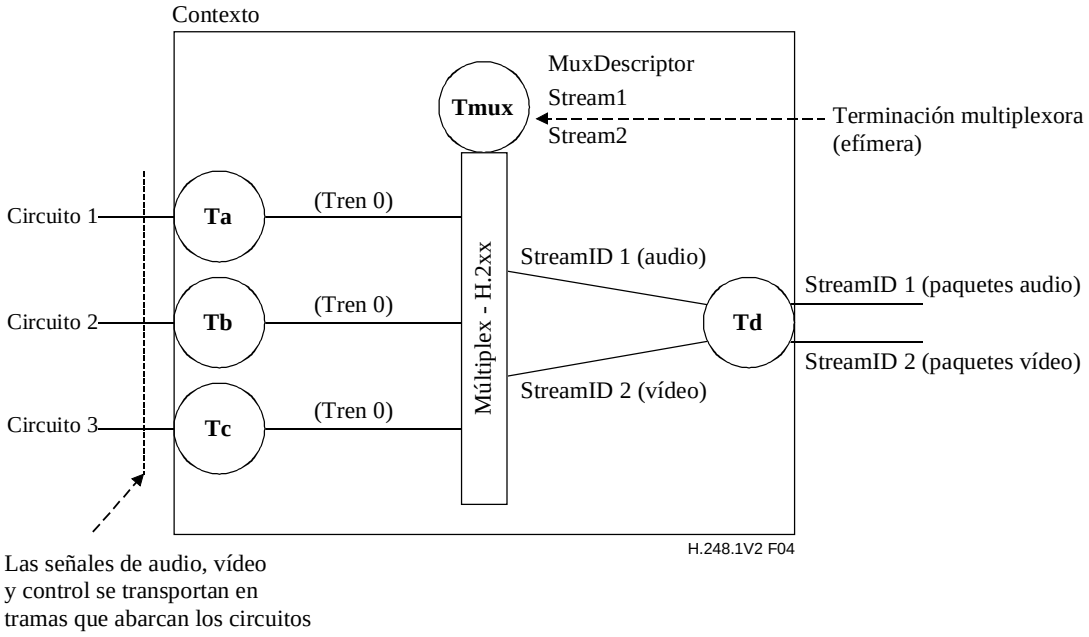


Figura 4/H.248.1 – Escenario de terminación multiplexada – Circuito a paquete

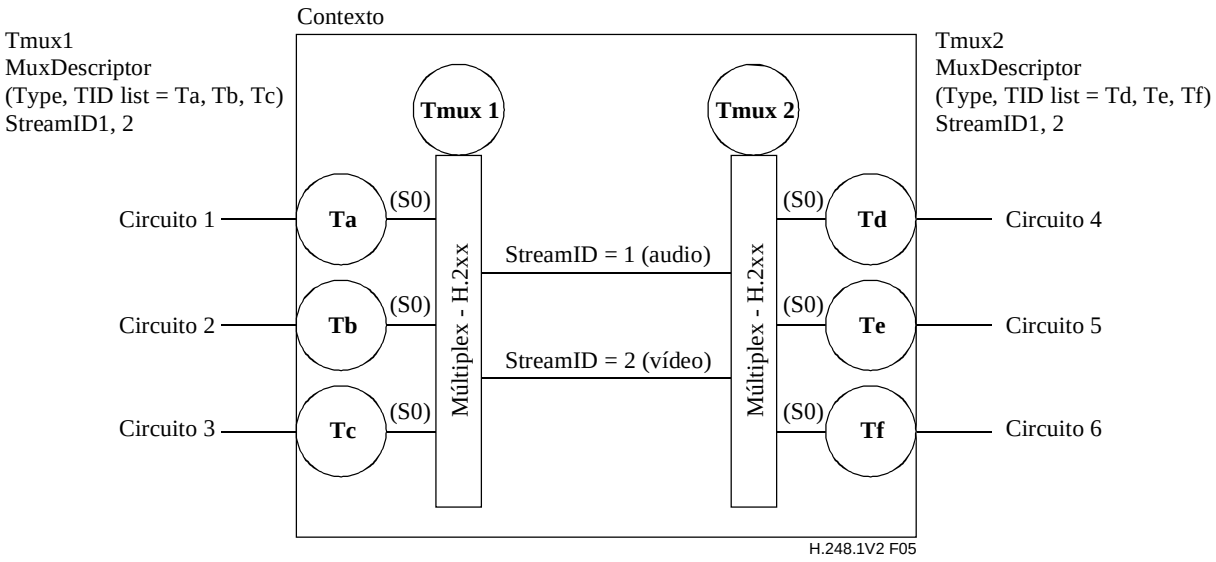


Figura 5/H.248.1 – Escenario de terminación multiplexada – Circuito a circuito

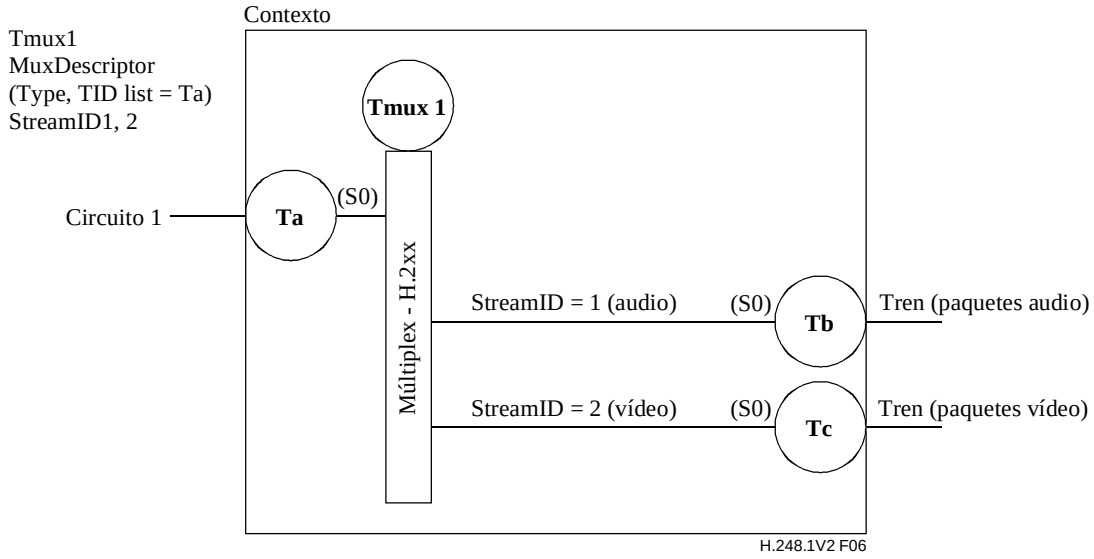


Figura 6/H.248.1 – Escenario de terminación multiplexada – Terminación simple a múltiple

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 substraída, 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 y que habían captado, por ejemplo mediante la instrucción Añadir.

6.2.2 TerminationID

Se hace referencia a las terminaciones por 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.

Cuando se utiliza ALL en el TerminationID de una instrucción, el efecto es idéntico al de repetir la instrucción con cada uno de los TerminationID concordantes. La utilización de ALL no direcciona la terminación ROOT. Puesto que cada una de estas instrucciones puede generar una respuesta, el tamaño de la respuesta completa puede ser grande. Si no se necesitan respuestas individuales, se puede pedir una respuesta comodín. En tal caso, se genera una sola respuesta, que contiene la UNION de todas las respuestas individuales que de otro modo se habrían generado, en la que se han suprimido los valores duplicados. Por ejemplo, dada una Termination Ta con propiedades p1 = a, p2 = b y Termination Tb con propiedades p2 = c, p3 = d, una respuesta UNION podría consistir en un TerminationId con comodines y la secuencia de propiedades p1 = a, p2 = b,c y p3 = d. La respuesta comodín puede ser particularmente útil en las instrucciones de auditoría.

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

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 comprobar una terminación para determinar los lotes que ésta 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 son visualizados. 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 pueden también tener el mismo identificador, etc.

Para soportar un determinado lote, la MG tiene que soportar todas las propiedades, señales, eventos y estadísticas definidos en ese lote. También tiene que soportar todos los parámetros de señal y de evento. La MG puede soportar un subconjunto de valores indicados en un lote para una determinada propiedad o parámetro.

Cuando los lotes son extendidos, 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 paquete 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 "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 nulo. 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 propiedad 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 del estado de la terminación. 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 PackageName 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.

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 (por ejemplo, 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 lote especifique otro comportamiento. Para más detalles, véase 7.1 que trata los distintos descriptores.

En el siguiente cuadro se indican todos los posibles descriptores y su utilización. No todos los descriptores son legales 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 multimedios (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/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 deseada 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.
Topología	Especifica sentidos de flujo entre terminaciones y un contexto.
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 ModemDescriptor no fue considerado en la Rec. UIT-T H.248.1, Versión 1 (03/2002).	

6.2.5 Terminación raíz

Ocasionalmente, 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 – para cambiar una propiedad, enviar una señal o establecer un evento;
- una instrucción Notificar – para informar un evento;
- un valor de retorno de AuditValue – para examinar los valores de propiedades y estadísticas implementadas en la raíz;

- una instrucción AuditCapability – para determinar qué propiedades de la raíz están implementadas;
- una instrucción ServiceChange – para declarar la pasarela como 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).

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 (o Mover)**: 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 uno de los descriptores requeridos, distinto del descriptor de Auditoría, (no se ha incluido), 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 TerminationIds concordantes.

7.1.2 Descriptor de módem

El descriptor de módem especifica el tipo de módem y sus parámetros, si procede, requeridos por ejemplo, en conversación con arreglo a H.324 y en conversación mediante 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 1 (03/2002) y las relaciones 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 LocalDescriptor y RemoteDescriptor.

7.1.3 Descriptor de múltiplex

En las llamadas multimedios se transportan varios trenes de medios en un número (posiblemente diferente) de portadores. 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 × 64K;
- posibles ampliaciones,

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

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

El tipo de múltiplex N × 64K implementa el servicio N × 64K (conforme con 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 × 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 tren de un solo medio.

Un tren se identifica mediante un StreamID. El StreamID se utiliza para vincular los trenes en un contexto que corresponden unos con otros. Si existen múltiples trenes una terminación se sincronizará con cada uno de ellos. El descriptor de tren comprende hasta tres descriptores subsidiarios: el descriptor LocalControl, el descriptor local y el descriptor 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)
```

Por razones de conveniencia, los descriptores LocalControl, Local o Remote pueden incluirse en el descriptor de medios sin un descriptor Stream circundante. En este caso, se supone que el StreamID es 1.

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 lotes) 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: "prueba", "fuera de servicio", o "en servicio". El estado "prueba" indica que la terminación se está probando. El estado "fuera de servicio" indica que la terminación no puede utilizarse para tráfico. El estado "en servicio" indica que una terminación puede utilizarse para tráfico normal. El estado por defecto es "en servicio".

Los valores asignados a las propiedades pueden ser valores simples (entero/cadena/enumeración), o pueden ser valores subespecificados cuando se suministra más de un valor y la MG puede elegir:

- Valores alternativos: lista de valores, de los cuales hay que seleccionar uno.
- Gamas: 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 tampón después de la detección de un evento en el descriptor de eventos, o son procesados inmediatamente. Para más detalles véase 7.1.9.

7.1.6 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 LocalControl

El descriptor LocalControl contiene la propiedad modo, las propiedades ReserveGroup y ReserveValue y las propiedades de una terminación (definidas en lotes) que son específicas del tren, y son de interés para 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 sólo enviar, recibir solamente, enviar/recibir, inactivo y conexión en bucle. "Enviar" y "recibir" han de entenderse con respecto al exterior del contexto, de modo que, por ejemplo, un tren fijado a mode = sendOnly no pasa al contexto los medios recibidos. El valor por defecto para la propiedad modo es "Inactive". Las señales y los eventos no son afectados por el modo.

Las propiedades Reserve en valor booleano, ReserveValue y ReserveGroup de una terminación indican lo que se espera que haga la MG cuando recibe un descriptor local y/o distante.

Si el valor de una propiedad Reserve 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, RESPONDERÁ al MGC con 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 una propiedad Reserve 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 propiedades Reserve.

La nueva configuración del descriptor LocalControl 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.

7.1.8 Descriptor local y descriptor distante

El MGC utiliza el descriptor Local y el descriptor Distante (Remote) 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).

El descriptor Local se refiere a los medios recibidos por la MG y Distante (Remote) se refiere a los medios enviados por la MG.

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 con RFC 2327. 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 LocalControl.

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.

Cuando los descriptores han sido transferidos del MGC a la MG, son interpretados de conformidad con las reglas dadas en 7.1.1, haciéndose los siguientes comentarios adicionales para su clarificación:

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 alguna de las alternativas de grupo de propiedades solicitadas que puede actualmente soportar. Si ReserveValue es Verdadero, la MG reserva los recursos requeridos para soportar alguna de las alternativas de valores de propiedad solicitadas que puede 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 entonces 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 LocalControl 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 DEBE 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 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 EventsDescriptor contiene un RequestIdentifier y una lista de eventos que la pasarela de medios debe detectar e informar. El RequestIdentifier 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 RequestIdentifier se omite si el EventsDescriptor 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, y parámetros facultativos. El nombre de evento consiste en un 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 un evento de compleción de mapa de dígitos está presente o implícito en EventsDescriptor, 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). Cualquier otra acción queda en estudio. 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 EventsDescriptor nuevo sea como resultado de la recepción de una instrucción con un nuevo EventsDescriptor, sea mediante la activación de un EventsDescriptor insertado.

Si EventBufferControl está desactivado (Off), la MG continúa el procesamiento basado en el EventsDescriptor activo.

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

NOTA 1 – A los fines de tratamiento de EventBuffer, la activación de un EventsDescriptor insertado es equivalente a la recepción de un EventsDescriptor 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 EventsDescriptor. 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 EventsDescriptor, la MG comprobará la EventBuffer FIFO y emprenderá las siguientes acciones:

1) Si la EventBuffer está vacía, la MG espera la detección de eventos basados en el nuevo EventsDescriptor.
2) Si la EventBuffer no está vacía, la MG procesa la cola de espera FIFO comenzando por el primer evento:
    a) Si el evento en la cola se encuentra en la lista de eventos del nuevo EventsDescriptor, la MG actúa sobre el evento y la 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 EventsDescriptor. Mientras espera un nuevo EventsDescriptor, los eventos detectados que aparecen en el EventsBufferDescriptor se colocarán en la EventBuffer. Cuando se recibe un nuevo EventsDescriptor, se repetirá el procesamiento del evento desde el paso 1).
    b) Si el evento no se encuentra en el nuevo EventsDescriptor, 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 EventsDescriptor, la MG procesa los nuevos eventos con el nuevo EventsDescriptor.

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 Recomendación de la serie H.248.x.

El valor por defecto de EventBufferControl es Off.

NOTA 2 – Puesto que la propiedad EventBufferControl está en el TerminationStateDescriptor, la MG puede recibir una instrucción que cambie la propiedad EventBufferControl y no incluya un EventsDescriptor.

Normalmente, el reconocimiento de un evento producirá la parada de las señales que estén 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.

Un evento puede incluir un descriptor de señales insertadas y/o un descriptor de eventos insertados 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 EventsDescriptors cuando un evento marcó KeepActive y contiene un descriptor de señales insertadas.

Sólo se permite un nivel de inserción. Un EventsDescriptor insertado NO CONTENDRÁ otro EventsDescriptor insertado; un EventsDescriptor insertado PUEDE contener un SignalsDescriptor insertado.

Un EventsDescriptor 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 EventsDescriptor se procesarán de acuerdo con el nuevo EventsDescriptor.

Un descriptor de eventos vacío inhabilita todas las operaciones de reconocimiento y señalamiento de eventos. Un descriptor EventBuffer 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 EventBuffer

El descriptor EventBuffer 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

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

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

Las señales se definen en lotes. Las señales se denominarán con un nombre de lote (en el que se define la señal) y un SignalID. El SignalID no contendrá ningún comodín. Las señales que ocurren en un SignalsDescriptor 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 que ésta ha finalizado por su cuenta, de otra manera), la señal ha sido interrumpida por un evento, 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, la notificación se genera únicamente si la señal ha sido detenida o nunca ha sido activada por algún otro motivo. Para que tenga lugar la comunicación, el evento compleción de la señal (véase E.1.2) debe ser habilitado en el descriptor de eventos actualmente activo.

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

Hay tres tipos de señales:

- Activado/desactivado (*on/off*): la señal está presente hasta que es desactivada.
- Temporización: la señal está presente hasta que es desactivada o expira un determinado periodo de temporización.
- Breve: 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 SignalsDescriptor, reemplaza al tipo de señal por defecto (véase 12.1.4). Si se especifica la duración para una señal activado/desactivado, SERÁ IGNORADA.

Una lista de señales secuenciales consta de un identificador de lista de señales 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.

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

Por definición, las señales van desde la terminación hacia el exterior del contexto, salvo que se especifique otra cosa en un lote. Cuando una misma señal se aplica a múltiples 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 SignalsDescriptor reemplaza a cualquier SignalsDescriptor 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 que contengan la misma 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 contiene 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:

- el tipo de señal y la secuencia de señales de la lista de señales secuenciales del descriptor sustituto serán ignorados, 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 Audit especifica la información que se debe revisar. El descriptor Audit especifica la lista de descriptores y/o propiedades particulares que el sistema debe retornar. El descriptor Audit puede utilizarse en cualquier instrucción para obligar el retorno de 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 Audit:

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

La instrucción Audit puede estar vacía, 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 ServiceChange

ServiceChangeDescriptor contiene los siguientes parámetros:

- ServiceChangeMethod
- ServiceChangeReason
- ServiceChangeAddress
- ServiceChangeDelay
- ServiceChangeProfile

- ServiceChangeVersion
- ServiceChangeMGCId
- TimeStamp
- Extension

Véase 7.2.8.

7.1.14 Descriptor de DigitMap

7.1.14.1 Definición, creación, modificación y supresión de DigitMap

Un DigitMap (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 DigitMap contiene un nombre de DigitMap y el DigitMap que habrá de asignarse. El mapa de dígitos puede ser precargado en la MG por una acción de gestión, y se puede invocar mediante su nombre en un EventsDescriptor, puede ser definido dinámicamente, y se puede invocar después mediante su nombre, o el propio mapa de dígitos puede ser especificado en el EventsDescriptor. Se permite, para un evento de compleción de mapa de dígitos dentro de un descriptor de eventos, referirse por un nombre a un DigitMap que es definido por un descriptor de DigitMap dentro de la misma instrucción, con independencia de la petición transmitida de los descriptores respectivos.

Los DigitMaps definidos en un DigitMapDescriptor pueden aparecer en cualquiera de las instrucciones normalizadas de manipulación de terminación, del protocolo. Un DigitMap, una vez definido, puede utilizarse en todas las terminaciones especificadas por el TerminationID (posiblemente con comodines) en tal instrucción. Los DigitMaps 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 DigitMap con el mismo nombre. Cuando un DigitMap se define dinámicamente en un descriptor de DigitMap:

- 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 EventsDescriptors siguientes que especifican el nombre, incluido cualquier EventsDescriptor en la instrucción que contiene el descriptor DigitMap, utilizarán la definición nueva.
- Se suprime un DigitMap 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 DigitMap

La colección de dígitos conformes con un DigitMap puede 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 se haya marcado cualquier número. 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 que pueden configurarse con arreglo a un DigitMap. Los valores por defecto de estos temporizadores deben ser configurados en la MG pero pueden ser reemplazados por valores especificados dentro del DigitMap.

7.1.14.3 Sintaxis de DigitMap

La sintaxis formal del mapa de dígitos se describe por la regla DigitMap en la descripción de sintaxis formal del protocolo (véanse los anexos A y B). De acuerdo con esta sintaxis, un DigitMap 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 concernido, 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 puede también 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, la temporización entre eventos mientras se hace la concordancia de un símbolo punto terminal, 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 un especificador de temporización explícito se encuentra realmente en realidad en una secuencia de eventos alternativa, pero no se da ninguno en cualquier 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, la temporización regresa a las reglas por defecto dadas anteriormente. 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.

7.1.14.4 Evento de compleción DigitMap

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 termina 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 la compleción, se generará un evento de compleción del mapa de dígitos como el 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 ACTIVADO (ON), los siguientes eventos de dígitos serán tratados de la misma forma que cualquier otro evento.
- Si EventBufferControl está DESACTIVADO (OFF), el EventsDescriptor 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 compleción del mapa de dígitos, o hasta que el EventsDescriptor 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 compleción 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 EventsDescriptor interrumpe el proceso de almacenamiento intermedio, y este EventsDescriptor contiene un nuevo evento de compleción 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 EventsDescriptor 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 EventsDescriptor no permite un evento de compleción de mapa de dígitos ni la comunicación de eventos de dígitos particulares del lote correspondiente, se descartará el contenido de la memoria intermedia y se terminará el proceso de almacenamiento.

7.1.14.5 Procedimientos DigitMap

Hasta la compleción, 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 dígito largo (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 dígito permite las comunicaciones detalladas de fin de temporización, la cadena de marcación comunicada terminará en 'L', 'S' o 'T' según el caso.

3) Si un evento se detecta antes de que expire el temporizador, se corresponde con un símbolo de cadena de dígitos y añadido 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. Si, y sólo si, una secuencia que espera un evento de larga duración en esta posición es concordada (es decir, el evento tuvo una duración larga y cumplió la especificación para esta posición), entonces 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 secuencias de eventos alternativas que no especifican un evento de larga duración en la posición dada permanecen 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 compleción 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 compleción del mapa de dígitos puede especificar que se comunique el evento eliminado como un parámetro del evento de compleción. Se hará entonces independientemente del procesamiento ulterior del evento de dígitos.

   b) El evento de compleción 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 compleción como resultado del paso 5), el procesamiento retorna al paso 2).

7.1.14.6 Activación DigitMap

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 compleción de mapa de dígitos. El evento de compleción de mapa de dígitos 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 compleción de evento que no contiene el campo eventDM en su campo acciones solicitadas se considera un error. Al recibir tal evento en un EventsDescriptor, una MG contestará con una respuesta de error, que incluye el error 457 (Falta parámetro en señal o evento).

7.1.14.7 Interacción de DigitMap y procesamiento de evento

Mientras el mapa de dígitos es activado, se permite la detección de todos los eventos definidos en el lote que contiene el evento de compleción del mapa de dígitos especificado. El comportamiento de un evento normal (por ejemplo, la parada de señales a menos que el evento de compleción 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 compleción 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 compleción 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 compleción 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 ItemID con comodines (nombre de propiedad) 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
91xxxxxxxxxx	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|91xxxxxxxxxx|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. Para cada terminación se mantiene, cuando procede, un conjunto de estadísticas normalizadas (por ejemplo, el número de octetos enviados y recibidos). Las propiedades estadísticas concretas que son comunicadas para una terminación dada son determinadas por los lotes realizados por la terminación. Por defecto, las estadísticas se comunican cuando la terminación es substraída del contexto. Para hacerlo de otra forma se puede incluir un AuditDescriptor vacío en la instrucción Substraer. También puede retornar estadísticas la instrucción AuditValue, o cualquier instrucción Añadir/Desplazar/Modificar que utilice el descriptor Audit.

Las estadísticas son acumulativas; el hecho de comunicar estadísticas no las reinicia. Las estadísticas son reiniciadas cuando se substrae una terminación de un contexto.

7.1.16 Descriptor de lotes

El descriptor de lotes, que sólo se utiliza con la instrucción AuditValue, retorna una lista de lotes realizados por la terminación.

7.1.17 Descriptor ObservedEvents

El descriptor ObservedEvents se suministra con la instrucción Notificar para informar al MGC del evento o los eventos que fueron detectados. El ObservedEventsDescriptor se utiliza con la instrucción AuditValue; este descriptor retorna eventos que están almacenados en la memoria tampón de eventos y no han sido notificados. ObservedEvents contiene el RequestIdentifier del EventsDescriptor que activó la notificación, el(los) evento(s) detectados, facultativamente el(los) tiempo(s) de detección y cualesquiera parámetros del evento observado. Los tiempos de detección son comunicados con un 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 subcláusulas 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 es recibida por todas las demás terminaciones. La implementación del descriptor de topología es facultativa. Una MG que no soporta descriptores de topología, pero que recibe una instrucción que contienen uno, retorna el 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 de 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, la utilización del comodín ALL de tal modo que haya terminaciones que concuerden con *T1* y *T2* no está permitida.
- (*T1*, *T2*, ambos sentidos) 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. Sin embargo, si hay una terminación que concuerda con ambas, no se introduce una conexión en bucle.

Pueden utilizarse comodines CHOOSE en *T1* y *T2* también, 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 error: error 421 (Acción desconocida o combinación de acciones no permitida). Véase la figura 7, el cuadro siguiente y la figura 8, como ejemplos de los efectos que resultan de incluir descriptores de topología en acciones. En estos ejemplos se supone que los descriptores de topología se aplican secuencialmente.

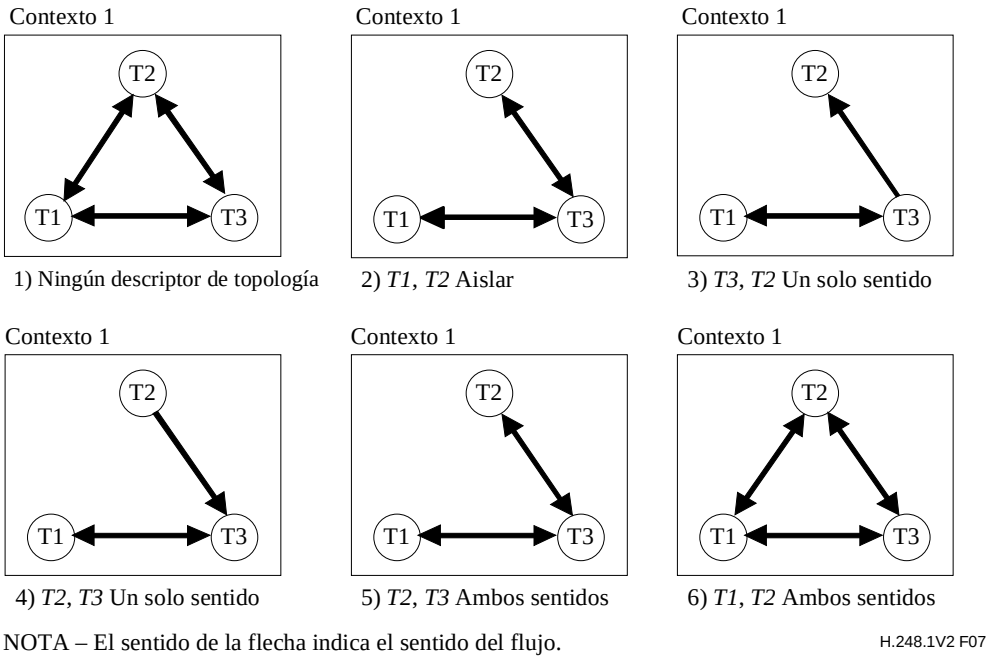
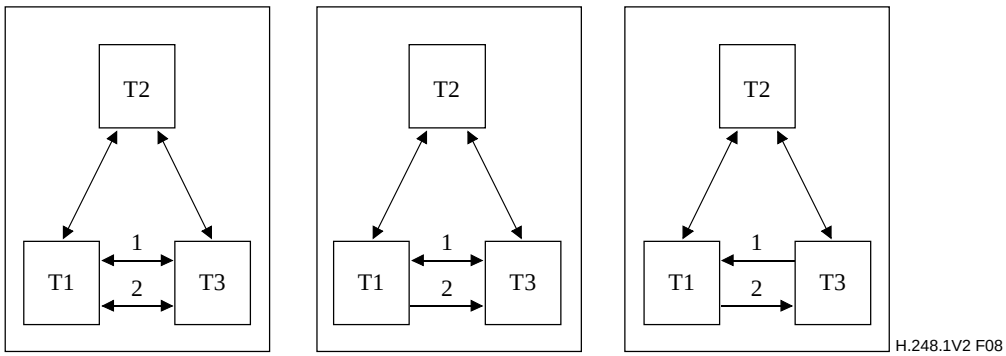


Figura 7/H.248.1 – Ejemplo de topologías

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	*T1*, *T2*, aislar Suprime la conexión entre *T1* y *T2*. *T3* tiene una conexión en ambos sentidos con *T1* y *T2*. *T1* y *T2* tienen conexión en ambos sentidos con *T3*.
3	*T3*, *T2*, un solo sentido Una conexión en un solo sentido de *T3* a *T2* (esto es, *T2* recibe flujo de medios de *T3*). Una conexión en ambos sentidos entre *T1* y *T3*.
4	*T2*, *T3*, un solo sentido Una conexión en un solo sentido de *T2* a *T3*. Se mantiene la conexión en ambos sentidos entre *T1* y *T3*
5	*T2*, *T3*, ambos sentidos *T2* y *T3* tienen conexión en ambos sentidos. Esta situación es la misma que 2.
6	*T1*, *T2*, ambos sentidos (*T2*, *T3*, ambos sentidos y *T1*, *T3* ambos sentidos pueden ser conexiones implícitas o explícitas). Todas las terminaciones tienen una conexión en ambos sentidos con cada una de las demás.

Una conexión en un solo sentido se debe implementar de tal manera que las demás terminaciones en el contexto no perciban el cambio de topología.



(a) Ningún descriptor de topología (b) *T1,T3*, un solo sentido, 2 (c) *T3,T1*, un solo sentido, 1

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

7.1.19 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). 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 una construcción sintáctica diferente de aquélla en la que aparece. Por ejemplo, el descriptor de error 422 (Error de sintaxis en acción), podría aparecer dentro de una instrucción aun cuando hiciera referencia a la construcción más amplia, 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.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationID

[, MediaDescriptor]

[, ModemDescriptor] (*)

[, MuxDescriptor]

[, EventsDescriptor]

[, EventBufferDescriptor]

[, SignalsDescriptor]

[, DigitMapDescriptor]

[, AuditDescriptor]

)

(*) El descriptor de módem no fue considerado en H.248.1, Versión 1 (03/2002).

El TerminationID especifica la terminación que habrá de añadirse al contexto. La terminación, bien es creada o bien es tomada del contexto nulo. Si se utiliza un comodín CHOOSE en el TerminationID, el sistema retornará el TerminationID seleccionado. Pueden utilizarse comodines 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 MediaDescriptor facultativo describe todos los trenes de medios.

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

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

El parámetro DigitMapDescriptor es facultativo. Si está presente da una definición de DigitMap que puede utilizarse en un EventsDescriptor.

El parámetro AuditDescriptor es facultativo. Si está presente, la instrucción retornará descriptores como se especifica en el AuditDescriptor.

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 AuditDescriptor.

NO SE UTILIZARÁ Añadir (Add) en una terminación que se encuentre "fuera de servicio" (Outofservice).

7.2.2 Modificar

La instrucción Modificar modifica las propiedades de una terminación.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
Modify( TerminationID
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
)
```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 1 (03/2002).

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. El orden de los intentos cuando múltiples TerminationID concuerdan no se especifica. La opción 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

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

TerminationID

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Subtract(TerminationID

[, AuditDescriptor]

)

(*) El descriptor de módem no fue considerado en H.248.1, Versión 1 (03/2002).

TerminationID en los parámetros de entrada representa la terminación que es substraída. El TerminationID puede ser específico o puede ser un valor de comodín que indica que todas las terminaciones (o un conjunto de terminaciones conexas) en el contexto de la instrucción Substraer habrán de ser substraídas. 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. El orden de los intentos cuando múltiples TerminationID concuerdan no se especifica.

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 Nulo. No está permitido Substraer una terminación del contexto nulo.

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

Por defecto, el sistema retorna el parámetro estadísticas para comunicar información tomada en la terminación o las terminaciones especificadas en la instrucción. La información comunicada es aplicable a la existencia de la terminación o de las terminaciones en el contexto del cual se substraen.

El AuditDescriptor es facultativo. Si está presente, la instrucción retornará solamente los descriptores especificados en el AuditDescriptor, los cuales pueden estar vacíos. Si se omite, se retorna por defecto el descriptor de estadísticas. Los valores de retorno posibles son los mismos que los de Añadir.

Si se substrae una terminación configurada de un contexto, sus valores de propiedad volverán a:

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

7.2.4 Desplazar

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 nulo, o desde dicho contexto.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
Move( TerminationID
    [, MediaDescriptor]
    [, ModemDescriptor] (*)
    [, MuxDescriptor]
    [, EventsDescriptor]
    [, EventBufferDescriptor]
    [, SignalsDescriptor]
    [, DigitMapDescriptor]
    [, AuditDescriptor]
)
```

(*) El descriptor de módem no fue considerado en H.248.1, Versión 1 (03/2002).

El TerminationID especifica la terminación que habrá de desplazarse. Esto puede hacerse mediante comodines, pero CHOOSE no será utilizado en el TerminationID. 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 AuditDescriptor con la opción estadísticas, por ejemplo, retornaría estadísticas en la terminación justamente antes del desplazamiento. Los posibles descriptores retornados desde 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".

7.2.5 AuditValue

La instrucción AuditValue retorna los valores actuales de propiedades, eventos, señales y estadísticas asociadas con terminaciones. AuditValue puede pedir el contenido de un descriptor o sólo de una propiedad, un evento, una señal o estadísticas.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

AuditValue(TerminationID,

AuditDescriptor

)

(*) El descriptor de módem no fue considerado en H.248.1, Versión 1 (03/2002).

El TerminationID puede ser específico o estar representado 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.

- Se puede solicitar la auditoría de un descriptor especificando ese descriptor en AuditDescriptor sin ninguna otra información.
- Para hacer la auditoría de una determinada propiedad, se incluyen los identificadores correspondientes de tren (facultativo), de grupo (facultativo) y de propiedad en el descriptor de medios. El sistema retorna al 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 de lista de señales y/o de señal. El sistema retorna los valores de todos los parámetros de la señal, únicamente si la señal está activa: indicación de actividad, tipo de señal, duración, indicación de compleción de la señal y propiedades definidas por el paquete.

- Para hacer la auditoría de un evento, se suministran las identidades stream id (facultativo), eventID, requestID (facultativo). El sistema retorna los valores de todos los parámetros del evento: acciones del evento y parámetros definidos en el paquete.
- Para hacer la auditoría de una estadística, se proporciona la identidad de la estadística. El sistema retorna el valor actual de esta estadística. No se reinicia la estadística.
- Para hacer la auditoría de un paquete, se proporciona la identidad y la versión del paquete. El sistema retorna todas las propiedades, las señales, los eventos y las estadísticas definidas en ese paquete con su valor actual.

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 retorne 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.

ObservedEvents retorna una lista de eventos en EventBuffer. Si se hace una auditoría de ObservedEventsDescriptor mientras está activo un DigitMap, el descriptor ObservedEvents retornado incluye también un evento de compleción de mapa de dígitos que muestra la cadena de marcación actual pero no un método de terminación.

EventBuffer retorna el conjunto de eventos y valores de parámetros asociados permitidos actualmente en EventBufferDescriptor. PackagesDescriptor retorna una lista de lotes realizados por la terminación. DigitMapDescriptor retorna el nombre o el valor del DigitMap actual para la terminación. Si el valor de terminación ID del DigitMap solicitado en una instrucción AuditValue es ALL, el sistema retorna todos los DigitMaps en la pasarela. Estadísticas retorna los valores actuales de todas las estadísticas que se llevan en la terminación. Si se especifica un descriptor Audit 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, esto es, específico, nulo o con comodines. (Obsérvese que ContextID All no incluye el Contexto nulo.) El TerminationID puede ser específico, o incluir comodines.

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

Supóngase que la pasarela tiene 4 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}}}

Retorna:

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

La instrucción:

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

Retorna:

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

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

La instrucción:

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

Retorna:

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

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

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

ContextID	TerminationID	Información obtenida
Específico	Comodín	Auditoría de terminaciones concordantes en un contexto
Específico	Específico	Auditoría de una terminación individual en un contexto
Nulo	Raíz	Auditoría del estado y eventos de una pasarela de medios
Nulo	Comodín	Auditoría de todas las terminaciones concordantes en el contexto nulo
Nulo	Específico	Auditoría de una terminación individual fuera de todo contexto
Todos	Comodín	Auditoría de todas las terminaciones concordantes y del contexto al que están asociadas
Todos	Raíz	Lista de todos los ContextId (la lista de ContextID debe retornarse utilizando múltiples respuestas de acción, cada una e las cuales contendrá un ContextID tomado de la lista)
Todos	Específico	ContextID (no nulo) en el que la terminación existe en ese momento

7.2.6 AuditCapabilities

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.

TerminationID

[,MediaDescriptor]

[,ModemDescriptor](*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

AuditCapabilities(TerminationID,

AuditDescriptor)

(*) El descriptor de módem no se consideró en H.248.1, Versión 1 (03/2002).

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

- Puede hacerse una petición de auditoría de todo el descriptor identificando ese descriptor en AuditDescriptor sin ninguna otra información.

- Para hacer la auditoría de una determinada propiedad en el descriptor de medios, se indican las identidades stream ID (facultativo) y propertyID. El sistema retorna una lista de valores posibles de esta propiedad.
- Para hacer la auditoría de una señal, se suministran las identidades de lista de señales y/o señal. El sistema retorna una lista de valores posibles asociados con cada parámetro de señal (indicación de actividad, tipo de señal, duración, indicación de compleción de señal y propiedades definidas en el paquete).
- Para hacer la auditoría de un evento, se suministran los indicadores pertinentes stream ID (facultativo), eventID, requestID (facultativo). El sistema retorna una lista de valores posibles asociados a cada parámetro de evento (acciones de evento y parámetros definidos en el paquete).
- Para hacer la auditoría de una estadística se suministra la identidad de esa estadística. El sistema retorna a los valores posibles de la estadística. La estadística no reinicia.

Cuando se pide la auditoría de un descriptor, la instrucción AuditCapabilities retorna 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 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 apropiadas 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.

EventsDescriptor retorna la lista de posibles eventos en la terminación junto con la lista de todos los valores posibles de los parámetros de EventsDescriptor. EventBufferDescriptor retorna la misma información que EventsDescriptor. SignalsDescriptor retorna 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. StatisticsDescriptor retorna los valores de las estadísticas que se llevan en la terminación. ObservedEventsDescriptor retorna los nombres de los eventos activos en la terminación. DigitMap y lotes no legales 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
Nulo	Raíz	Auditoría de estados y eventos de MG
Nulo	Comodín	Auditoría de todas las terminaciones concordantes en el contexto nulo
Nulo	Específico	Auditoría de una sola terminación fuera de todo contexto
Todos	Comodín	Auditoría de todas las terminaciones concordantes y del contexto a que están asociadas
Todos	Raíz	Lo mismo que para AuditValue
Todos	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.

ObservedEventsDescriptor 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 0 quedan en estudio.

Las instrucciones Notificar con el RequestID distinto de 0 aparecerán solamente como resultado de la detección de un evento especificado por un descriptor de eventos activo en la terminación considerada.

RequestID retorna el parámetro RequestID del EventsDescriptor 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 EventsDescriptor que provocó la instrucción o insertados en el descriptor de eventos, a menos que el RequestID sea 0 (lo que queda en estudio).

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

7.2.8 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 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.

TerminationID,

[ServiceChangeDescriptor]

ServiceChange(TerminationID,

ServiceChangeDescriptor

)

El parámetro TerminationID 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.

El ServiceChangeDescriptor contiene los siguientes parámetros obligatorios:

- ServiceChangeMethod;
- ServiceChangeReason;
- ServiceChangeDelay;
- ServiceChangeAddress;

- ServiceChangeProfile;
- ServiceChangeVersion;
- ServiceChangeMgcId;
- TimeStamp;
- ServiceChangeInfo.

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 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 trasladar la terminación de un contexto activo (lo que sea primero) a fuera de servicio.

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 serviceState debe estar "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 todas las terminaciones han sido sustraídas.

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 mutuamente por la MG y el MGC.

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.

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 ServiceChangeDescriptor o el ServiceChangeResultDescriptor. El parámetro

ServiceChangeAddress proporciona una dirección que se utilizará dentro del contexto de la asociación que se está actualmente negociando, 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.

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", 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" hasta que la terminación esté en el contexto nulo.

El parámetro facultativo ServiceChangeProfile especifica el perfil (si existe) del protocolo soportado. ServiceChangeProfile incluye la versión del perfil soportado.

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).

El parámetro facultativo TimeStamp especifica el tiempo real registrado por el emisor. Como tal, no es necesariamente un tiempo absoluto, por ejemplo en el huso horario local, sino que, simplemente, establece un tiempo de comienzo arbitrario con respecto al cual se compararán todas las futuras indicaciones de tiempo transmitidas por un emisor durante su asociación. Puede ser utilizado por el respondedor para determinar la diferencia de su tiempo con el de su correspondiente. TimeStamp es enviado con una precisión de centésimas de segundo.

El parámetro facultativo Extension puede contener cualquier valor cuyo significado sea mutuamente acordado entre la MG y el MGC. El parámetro facultativo ServiceChangeInfo puede contener el paquete/propiedad/señal/evento/estadística del motivo para el cambio de servicio.

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" para el TerminationID y ServiceChangeMethod igual a cambio-en-caso-de-fallo cuando la MG detecta fallos del MGC. 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 ServiceChangeDescriptor 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, las respuestas de transacción y las indicaciones pendientes deben enviarse a la dirección desde la cual se han originado las peticiones correspondientes. Se multiplican los mensajes, pero es necesario porque las instrucciones y respuestas no pueden ser empaquetadas juntas. El parámetro TimeStamp será enviado con una instrucción de registro y su respuesta.

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.

El retorno de ServiceChange está vacío, 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 no hay indicación de perfil, el MGC utilizará las capacidades especificadas por el perfil que aparece en la petición de modificación de servicio.
- ServiceChangeVersion, si el respondedor desea negociar la versión del protocolo que se utilizará para la asociación.

Se han definido los siguientes ServiceChangeReasons (motivos de cambio de servicio). Esta lista puede ampliarse por un registro en la IANA como se describe en 14.3.

```
900 Service Restored (servicio restablecido)
901 Cold Boot (arranque en frío)
902 Warm Boot (arranque en caliente)
903 MGC Directed Change (cambio dirigido por el MGC)
904 Termination malfunctioning (funcionamiento defectuoso de la
    terminación)
905 Termination taken out of service (terminación puesta fuera de
    servicio)
906 Loss of lower layer connectivity (e.g. downstream sync) pérdida de la
    conectividad de capa inferior (por ejemplo,sincronismo de flujo
    descendente)
907 Transmission Failure (fallo de transmisión)
908 MG Impending Failure (fallo inminente de MG)
909 MGC Impending Failure (fallo inminente de MGC)
910 Media Capability Failure (fallo de la capacidad de medios)
911 Modem Capability Failure (fallo de la capacidad de módem)
912 Mux Capability Failure (fallo de la capacidad de múltiplex)
913 Signal Capability Failure (fallo de la capacidad de señales)
914 Event Capability Failure (fallo de la capacidad de eventos)
915 State Loss (pérdida del estado)
916 Packages Change (cambio de lotes)
917 Capability Change (cambio de capacidades)
```

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 especifica el modo en que se manipulan y se efectúa una auditoría de los 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.

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 NULOS) 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.

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

ContextID	TerminationID	Audit
Específico	No es aplicable	Valor del atributo en el contexto especificado.
Nulo	No es aplicable	No autorizado.
Todos	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.

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. Esta información puede ser solicitada por una auditoría de los atributos de contexto o puede ser una información detallada del 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 ContexID 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 9 es una representación gráfica de las relaciones entre transacción, acción e instrucción.

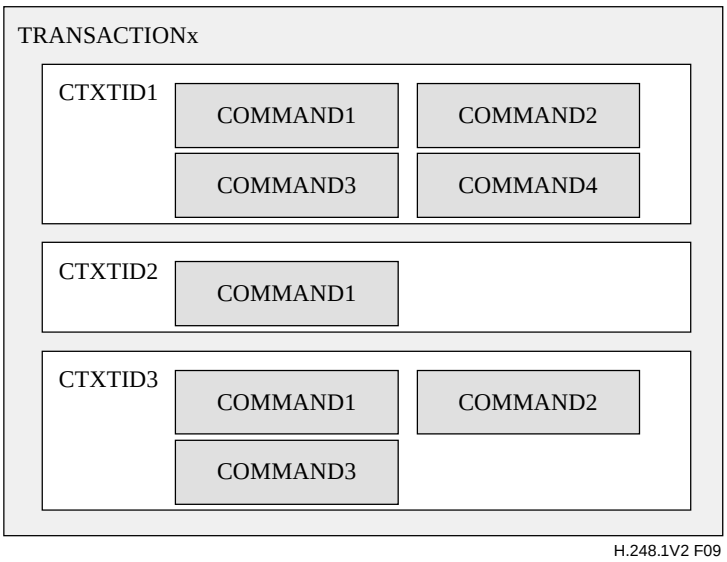


Figura 9/H.248.1 – Transacciones, acciones e instrucciones

Las transacciones son 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.

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.

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. Se utiliza TransactionPending para notificar periódicamente al receptor que una transacción continúa procesándose pero que todavía no ha finalizado.

Las aplicaciones DEBEN implementar un temporizador en el nivel de aplicación para cada transacción. La expiración del temporizador debe 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 por 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 0 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 *nulo*.

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 nulo cuando se utiliza el comodín AAL.

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 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 y una o más instrucciones por contexto.

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 empiece la nueva especificación de un parámetro ContextID, sea 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 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. 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.

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

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

Se considera que un EventsDescriptor o un EventBufferDescriptor ha sido procesado cuando se ha determinado que el descriptor 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, todos o nulo.

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 petición de transacción. 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 7.1.19 (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 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 nulo y un solo descriptor de error 403 (Error de sintaxis en TransactionRequest).

Si el final de una transacción no puede ser determinado de manera fiable y una o más acciones pueden ser analizadas gramaticalmente, 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 última 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 TransactionPending la invoca el receptor. TransactionPending indica 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. Una propiedad de la raíz (normalMGExecutionTime) puede ser fijada por el MGC para indicar el intervalo dentro del cual el MGC espera una respuesta a cualquier transacción de la MG. Otra propiedad (normalMGCExecutionTime) puede ser fijada por el MGC para indicar el intervalo dentro del cual la MG debe esperar una respuesta a cualquier transacción del MGC. 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 (MGOriginatedPendingLimit) 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 de raíz (MGCOriginatedPendingLimit) 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 xxxOriginatedPendingLimit (MGOriginatedPending Limit o MGCOriginatedPendingLimit) 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 (MGOriginatedPendingLimit o MGCOriginatedPendingLimit) 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 2 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 serie H.248.x (por ejemplo, H.248.4 y H.248.5). Podrán definirse transportes adicionales como Recomendaciones adicionales de la serie H.248.x. 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.

Se suministra a la MG un nombre o dirección [tal como el nombre de servidor de nombre de dominio (DNS, *domain name server*) o una dirección IP] de un MGC primario y de cero o más MGC secundarios (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 por otra parte 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 EventsDescriptor.
- 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) En una terminación, normalmente 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 rearranque. 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 rearranques

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 rearranque que se fija a un valor aleatorio distribuido uniformemente entre 0 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 rearranque.

El procedimiento de rearranque 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 rearranque.

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.

Los controladores de pasarelas de medios se dimensionan típicamente 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 que típicamente es de 3 minutos. El procesamiento de una llamada suele consistir en 5 a 6 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 5 a 6 transacciones para cada terminación cada 30 minutos en promedio, o, dicho de otra manera, que trata aproximadamente una transacción por cada terminación cada 5 ó 6 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 rearrancan. 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.

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 de 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 cláusula 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 cláusula 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 DEBEN 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 cláusula 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 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 Remote. 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. 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 paquete/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 EventsDescriptor 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 se suministra previamente a 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 alterno 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 trata de obtener sus MGC secundarios previamente suministrados, 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 la contestación a una instrucción ServiceChange con Rearranque se pierda, y que una instrucción sea recibida por la MG 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 error 406 (Versión no soportada).

Si el MGC soporta una versión que es superior a la versión de la MG, pero puede soportar la versión inferior propuesta por la MG, enviará 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 el MGC no puede cumplimentar lo anterior, rechazará la asociación mediante el error 406 (Versión no soportada).

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

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 (paulatino o forzado) y especifica el TerminationID raíz. Cuando retorna al servicio, envía un ServiceChange con un método "rearranque".

El hecho de que el MGC pueda enviar mensajes duplicados a ambas MG es conveniente para pares de MG capaces de cambio-en-caso-de-fallo redundante de una de las MG. 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 método "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 MGC 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 mantienen la información de estado; se considera que esta materia cae fuera del alcance de la presente Recomendación. El MGC y la MG pueden dar los siguientes pasos cuando ocurre el 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 es 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.

12 Definición de lotes

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

Los lotes definidos por Grupo IETF aparecerán en distintas normas RFC.

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

1) Se debe especificar un documento público o un documento de un foro normalizado, al cual se pueda hacer referencia como el documento que describe el lote según la directriz antes mencionada.
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: togen/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:

12.1.1 Lote

Descripción global del lote; se especifica:

Nombre del lote: descriptivo solamente

PackageID: es un identificador

Descripción:

Versión:

Una nueva versión de un lote sólo puede añadir propiedades, eventos, señales, estadísticas adicionales y 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 DEBERÍA publicar este PackageID cuando informe lotes.

Extiende (facultativo): Descriptor 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, 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 paquetes 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.

12.1.2 Propiedades

Propiedades definidas por el lote; se especifica:

Nombre de propiedad: descriptivo solamente

PropertyID: es un identificador

Descripción:

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 (véase 12.3)

Sublista: una lista de varios valores de una lista. El tipo de sublista SE ESPECIFICARÁ también. 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.3.

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

Definido en:

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. Se prevé que éstos sean los casos más comunes, pero es posible que las propiedades se definan en otros descriptores.

Características: Lectura/escritura o ambas, 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.

12.1.3 Eventos

Eventos definidos por el lote; se especifica:

Nombre de evento: solamente descriptivo

EventID: es un identificador

Descripción:

Parámetros de EventsDescriptor:

Parámetros utilizados por el MGC para configurar el evento, y encontrados en el EventsDescriptor. Véase 12.2.

Parámetros de ObservedEventsDescriptor:

Parámetros retornados al MGC en peticiones Notificación y en contestaciones a peticiones de instrucción procedentes del MGC que efectúa una auditoría de ObservedEventsDescriptor, y encontrados en el ObservedEventsDescriptor. Véase 12.2.

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 SignalsDescriptor

Descripción:

SignalType: uno de los siguientes:

OO (On/Off)

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 StatisticsDescriptor

Descripción:

Unidades: unidad de medida, por ejemplo milisegundos, paquetes

12.1.6 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" o "DigitMap".

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 (véase 12.3)

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.3.

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

Descripción:

12.3 Listas

Entre los posibles valores para parámetros están las enumeraciones. Las enumeraciones pueden definirse en una lista. Se recomienda que la lista esté registrada en la IANA, de manera que los lotes que amplían la lista puedan definirse sin la preocupación de que haya nombres en conflicto.

12.4 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.

12.5 Registro de lotes

Un lote puede registrarse en la IANA por razones de interoperabilidad. Para las consideraciones relativas a la IANA véase la cláusula 13.

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. 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 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.

De los siguientes puntos, sólo los dos primeros son secciones obligatorias, pero también deberían incluirse los siguientes:

- Identificación del perfil: Nombre y versión del perfil que se envía en una instrucción cambio de servicio.
- Resumen: Descripción del perfil.
- Convenios para la formación de nombres:
  - Convenios para la formación de nombres de MGC/MG: Direccionamiento asociado a los nombres del MGC y la MG.
  - Nombres de terminaciones: Estructura de la identidad de una terminación.
  - Nombres de mapas de dígitos: Los nombres de los distintos mapas de dígitos.
- Descriptor de topología: ¿Utiliza este perfil el descriptor de topología?
- Indicación de tiempo: Se indica si habrá una indicación de tiempo en las instrucciones ServiceChange y/o Notificar.
- Temporizadores de transacción: Especifica los valores de los temporizadores de transacción.
- Transporte: Especifica cuáles son los transportes de la subserie H.248 soportados por el perfil.
- Codificación: Especifica cuál es la codificación soportada por el perfil.
- Soporte obligatorio de elemento de información SDP y anexo C: Especifica cuáles son los atributos SDP y los elementos de información anexo C que debe soportar.
- Paquetes: Especifica los paquetes soportados en este perfil.

  Obligatorio: Especifica los paquetes que deberá soportar este perfil.

  Facultativo: Especifica los paquetes que podrá soportar el perfil.

  Información de configuración del paquete: Especifica los valores de propiedades que se consideran configuradas (por ejemplo, nombres y números de ciclos de un anuncio H.248.7).
- Seguridad: Especifica los mecanismos de seguridad utilizados.
- Procedimiento: Especifica los procedimientos asociados al perfil.

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 que describe el lote, que será público:

   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 números de error de 3 ó 4 caracteres.

5) Los números de error registrados por todas las demás organizaciones o individuos tendrán número de error de 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, basic encoding rules), de las reglas de codificación canónica (CER, canonical encoding rules) y de las reglas de codificación distinguida.*

A.1 Codificación de comodines

La utilización de los comodines ALL y CHOOSE está permitida en el protocolo. Un MGC puede especificar parcialmente identificadores de terminaciones y dejar que la MG escoja entre los valores que son conformes con la especificación parcial. Los identificadores de terminaciones pueden codificar una jerarquía de nombres, que se configura. Por ejemplo, un TerminationID puede consistir en un grupo de troncales, una troncal perteneciente a un grupo, y un circuito. La utilización de comodines será posible en todos los niveles. En los párrafos siguientes se explica la forma de conseguir esto.

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: 0000001 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 a los 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; SE UTILIZARÁ el ID de contexto 0xFFFFFFFE para indicar un comodín CHOOSE; y SE UTILIZARÁ el ID de contexto 0xFFFFFFFF para indicar un comodín ALL.

SE UTILIZARÁ el ID de terminación 0xFFFFFFFFFFFFFFFF para indicar la terminación ROOT.

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 – La especificación ASN.1 a continuación contiene una cláusula que define la `TerminationIDList` como una secuencia de los TerminationID. La longitud de esta secuencia SERÁ uno, excepto posiblemente cuando se utiliza en contextAuditResult.

NOTA 3 – 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 4 – El módulo ASN.1 en este anexo utiliza tipos cadena de octetos para codificar valores de parámetro de propiedad, parámetro de señal y parámetro de evento. Los tipos reales de estos valores varían y se especifican en el anexo C o en la definición de lote 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 lotes se relaciona con la codificación BER de acuerdo con el siguiente cuadro:

Tipo especificado en el lote	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) version2(2)}
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 2.
     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
    ...
}

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
     },
     ...
}

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

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

ErrorCode ::= INTEGER(0..65535)
-- See clause 13 for IANA considerations with respect to error codes

ErrorText ::= IA5String

ContextID ::= INTEGER(0..4294967295)

-- Context NULL Value: 0
-- Context CHOOSE Value: 4294967294 (0xFFFFFFFE)
-- 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,
    ...
}

ContextAttrAuditRequest ::= SEQUENCE
{
    topology NULL OPTIONAL,
    emergency NULL OPTIONAL,
    priority NULL OPTIONAL,
    ...
}

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
}

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,
...
}


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

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

AuditRequest ::= SEQUENCE
{
     terminationID          TerminationID,
     auditDescriptor        AuditDescriptor,
     ...
}

AuditReply ::= CHOICE
{
     contextAuditResult     TerminationIDList,
     error     `            ErrorDescriptor,
     auditResult            AuditResult,
     ...
}
AuditResult ::= SEQUENCE
{

     terminationID          TerminationID,
     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,
     indauddigitMapDescriptor       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,
     ...
}
```

```
IndAudLocalControlDescriptor ::= SEQUENCE
{
     streamMode          NULL OPTIONAL,
     reserveValue        NULL OPTIONAL,
     reserveGroup        NULL OPTIONAL,
     propertyParms       SEQUENCE OF IndAudPropertyParm OPTIONAL,
     ...
}

IndAudPropertyParm ::= SEQUENCE
{
     name           PkgdName,
     ...
}

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,
     ...
}

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,
```

```
    ...
}

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 extraInfo 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.
-- Termination ID 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,
     ...
}
```

```
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 Statistics ID. (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 Property, Event, Signal, or Statistics ID, use
-- 0xFFFF for last two octets, choose is not allowed.
```

```
-- Wildcarding of Package Name is permitted only if Property,
-- Event, Signal, or Statistics ID 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,
     ...
}

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

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,
     ...
}

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

SignalName ::= PkgdName

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

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
{
     mtl  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 digit map 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
}

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 verison 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
}

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)),      -- yyyymmdd format
    time    IA5String(SIZE(8))       -- hhmmssss format
    -- per ISO 8601:1988
}

Value ::= SEQUENCE OF OCTET STRING


END
```

A.3 Mapas de dígitos y nombres de trayectos

Desde el punto de vista sintáctico, mapas de dígitos son cadenas que están sometidas a restricciones sintácticas. La sintaxis de mapas de dígitos válidos se especifica en ABNF (RFC 2234). La sintaxis de los mapas de dígitos presentados en esta cláusula tiene una finalidad exclusivamente ilustrativa. 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              ;digits 0-9
             / %x41-4B / %x61-6B    ;a-k and A-K
             / "L"     / "S" /     "T"   ;Inter-event timers
                                          ;(long, short, start)
             / "Z"                        ;Long duration event
DOT = %x2E ; "."
LWSP = *(WSP / COMMENT / EOL)
WSP = SP / HTAB
```

```
COMMENT = ";" *(SafeChar / RestChar / WSP) EOL
EOL = (CR [LF]) / LF
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
SafeChar = DIGIT / ALPHA / "+" / "-" / "&" / "!" / "_" / "/" /
          "'" / "?" / "@" / "^" / "`" / "~" / "*" / "$" / "\" /
          "(" / ")" / "%" / "."
RestChar = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
          "<" / ">" / "=" / %x22
DIGIT = %x30-39         ; digits 0 through 9
ALPHA = %x41-5A / %x61-7A     ; A-Z, a-z
```

Un nombre de trayecto es también una cadena sometida a restricciones sintácticas. La producción 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 "*"designado por ALL. El TerminationID "$" designado por 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 impone 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 es independiente del contenido. Por ejemplo, "Add" puede ser el AddToken o un NAME que depende del 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, 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 2.

messageBody          = ( errorDescriptor / transactionList )

transactionList      = 1*( transactionRequest / transactionReply /
                       transactionPending / transactionResponseAck )
;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
contextProperty      = (topologyDescriptor / priority / EmergencyToken)

contextAudit         = ContextAuditToken LBRKT
                       contextAuditProperties *(COMMA
                       contextAuditProperties) RBRKT

; at-most-once
contextAuditProperties = ( TopologyToken / EmergencyToken /
                           PriorityToken )

; "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 LBRKT
                     [ ImmAckRequiredToken COMMA]
               ( errorDescriptor / actionReplyList ) RBRKT

actionReplyList      = actionReply *(COMMA actionReply )

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

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

```
commandReplyList     = commandReplys *(COMMA commandReplys )

commandReplys        = (serviceChangeReply / auditReply / ammsReply /
                        notifyReply )

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

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

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

subtractRequest      =  SubtractToken EQUAL TerminationID
                        [ LBRKT auditDescriptor RBRKT]

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

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

auditOther           = EQUAL TerminationID [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 / auditItem)

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

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

notifyReply          = NotifyToken EQUAL TerminationID
                       [ LBRKT errorDescriptor RBRKT ]

serviceChangeRequest = ServiceChangeToken EQUAL TerminationID
                       LBRKT serviceChangeDescriptor RBRKT

serviceChangeReply   = ServiceChangeToken EQUAL TerminationID
                       [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, 0xFFFFFFFE and 0xFFFFFFFF are reserved.
ContextID             = (UINT32 / "*" / "-" / "$")

domainAddress         = "[" (IPv4address / IPv6address) "]"
;RFC 2373 contains the definition of IP6Addresses.
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

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 / "-" / "*" / ".")

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 )
```

```
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
parmValue            = (EQUAL alternativeValue/ INEQUAL VALUE)
alternativeValue     = ( VALUE
          / LSBRKT VALUE *(COMMA VALUE) RSBRKT
                         ; sublist (i.e. A AND B AND ...)
                / LBRKT VALUE *(COMMA VALUE) RBRKT
               ; alternatives (i.e. A OR B OR ...)

          /  LSBRKT VALUE COLON VALUE RSBRKT )
           ; range


INEQUAL              = LWSP (">" / "<" / "#" ) LWSP
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 ( "OFF" / LockStepToken )

terminationStateDescriptor = TerminationStateToken LBRKT
            terminationStateParm *( COMMA terminationStateParm ) RBRKT

; at-most-once per item except for propertyParm
terminationStateParm =(propertyParm / serviceStates / eventBufferControl )

serviceStates        = ServiceStatesToken EQUAL ( 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 ]

; at-most-once each of KeepActiveToken , eventDM 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 )

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, eventDM or
; eventStream
; KeepActiveToken and embedSig must not both be present
secondEventParameter = ( embedSig / KeepActiveToken / eventDM /
                         eventStream / eventOther )

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, every signalParameterName at most once
sigParameter       = sigStream / sigSignalType / sigDuration / sigOther
                   / notifyCompletion / KeepActiveToken
sigStream            = StreamToken EQUAL StreamID
sigOther             = sigParameterName parmValue
sigParameterName     = NAME
sigSignalType        = SignalTypeToken EQUAL signalType
signalType           = (OnOffToken / TimeOutToken / BriefToken)
sigDuration          = DurationToken EQUAL UINT16
notifyCompletion     = NotifyCompletionToken EQUAL (LBRKT
          notificationReason *(COMMA notificationReason) RBRKT)

notificationReason   = ( TimeOutToken / InterruptByEventToken
                       / InterruptByNewSignalsDescrToken
                       / OtherReasonToken )
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) RSBRKT))
                       [ 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            = ( MuxToken / ModemToken / MediaToken /
                         SignalsToken / EventBufferToken /
                         DigitMapToken / StatsToken / EventsToken /
                         ObservedEventsToken / PackagesToken ) /
                         indAudterminationAudit)

indAudterminationAudit   = indAudauditReturnParameter
                            *(COMMA indAudauditReturnParameter)

indAudauditReturnParameter = (indAudmediaDescriptor / /
                              indAudeventsDescriptor /
                              indAudsignalsDescriptor /
                              indAuddigitMapDescriptor /
                              indAudeventBufferDescriptor /
                              indAudstatisticsDescriptor /
                              indAudpackagesDescriptor)


indAudmediaDescriptor   = MediaToken LBRKT indAudmediaParm RBRKT

; at-most-once per item
; and either streamParm or streamDescriptor but not both
indAudmediaParm            = (indAudstreamParm / indAudstreamDescriptor /
                                 indAudterminationStateDescriptor)

; at-most-once
indAudstreamParm           = ( indAudlocalControlDescriptor )
; SDP too complex to pull out individual pieces for audit,
; hence no individual audit for Local and Remote

indAudstreamDescriptor     = StreamToken EQUAL StreamID
                                 LBRKT indAudstreamParm RBRKT

indAudlocalControlDescriptor = LocalControlToken LBRKT indAudlocalParm RBRKT

; at-most-once per item
indAudlocalParm            = ( ModeToken / pkgdName /
                               ReservedValueToken /
                               ReservedGroupToken )


indAudterminationStateDescriptor = TerminationStateToken LBRKT
                                indAudterminationStateParm RBRKT

; at-most-once per item
indAudterminationStateParm =(pkgdName / ServiceStatesToken / BufferToken)

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
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
; 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 /
                        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 8601:1988
; Date = yyyymmdd
Date                 = 8(DIGIT)
; Time = hhmmssss
Time                 = 8(DIGIT)
statisticsDescriptor = StatsToken LBRKT statisticsParameter
                      *(COMMA statisticsParameter ) RBRKT

;at-most-once per item
statisticsParameter  = pkgdName [EQUAL VALUE]

topologyDescriptor   = TopologyToken LBRKT topologyTriple
                       *(COMMA topologyTriple) RBRKT
topologyTriple       = terminationA COMMA
                       terminationB COMMA topologyDirection
                       [COMMA eventStream ]
terminationA         = TerminationID
terminationB         = TerminationID
topologyDirection    = BothwayToken / IsolateToken / OnewayToken

priority             = PriorityToken EQUAL UINT16

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 / RestChar/ WSP) DQUOTE

UINT16               = 1*5(DIGIT)  ; %x0-FFFF
UINT32               = 1*10(DIGIT) ; %x0-FFFFFFFF

NAME                 = ALPHA *63(ALPHA / DIGIT / "_" )
VALUE                = quotedString / 1*(SafeChar)
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")
AuditToken                = ("Audit"                / "AT")
AuditCapToken             = ("AuditCapability"      / "AC")
AuditValueToken           = ("AuditValue"           / "AV")
AuthToken                 = ("Authentication"       / "AU")
BothwayToken              = ("Bothway"              / "BW")
BriefToken                = ("Brief"                / "BR")
BufferToken               = ("Buffer"               / "BF")
CtxToken                  = ("Context"              / "C")
ContextAuditToken         = ("ContextAudit"         / "CA")
DigitMapToken             = ("DigitMap"             / "DM")
DisconnectedToken         = ("Disconnected"         / "DC")
DelayToken                = ("Delay"                / "DL")
DurationToken             = ("Duration"             / "DR")
EmbedToken                = ("Embed"                / "EM")
EmergencyToken            = ("Emergency"            / "EG")
ErrorToken                = ("Error"                / "ER")
EventBufferToken          = ("EventBuffer"          / "EB")
EventsToken               = ("Events"               / "E")
FailoverToken             = ("Failover"             / "FL")
ForcedToken               = ("Forced"               / "FO")
GracefulToken             = ("Graceful"             / "GR")
H221Token                 = ("H221" )
H223Token                 = ("H223" )
H226Token                 = ("H226" )
HandOffToken              = ("HandOff"              / "HO")
ImmAckRequiredToken       = ("ImmAckRequired"       / "IA")
InactiveToken             = ("Inactive"             / "IN")
IsolateToken              = ("Isolate"              / "IS")
InSvcToken                = ("InService"            / "IV")
InterruptByEventToken     = ("IntByEvent"           / "IBE")
InterruptByNewSignalsDescrToken
                          = ("IntBySigDescr"        / "IBS")
KeepActiveToken           = ("KeepActive"           / "KA")
LocalToken                = ("Local"                / "L")
LocalControlToken         = ("LocalControl"         / "O")
LockStepToken             = ("LockStep"             / "SP")
LoopbackToken             = ("Loopback"             / "LB")
MediaToken                = ("Media"                / "M")
MegacopToken              = ("MEGACO"               / "!")
MethodToken               = ("Method"               / "MT")
MgcIdToken                = ("MgcIdToTry"           / "MG")
ModeToken                 = ("Mode"                 / "MO")
ModifyToken               = ("Modify"               / "MF")
ModemToken                = ("Modem"                / "MD")
MoveToken                 = ("Move"                 / "MV")
MTPToken                  = ("MTP")
MuxToken                  = ("Mux"                  / "MX")
NotifyToken               = ("Notify"               / "N")
NotifyCompletionToken     = ("NotifyCompletion"/ "NC")
Nx64kToken                = ("Nx64Kservice"         / "N64")
ObservedEventsToken       = ("ObservedEvents"       / "OE")
OnewayToken               = ("Oneway"               / "OW")
OnOffToken                = ("OnOff"                / "OO")
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")
RestartToken               = ("Restart"               / "RS")
RemoteToken                = ("Remote"                / "R")
ReservedGroupToken         = ("ReservedGroup"         / "RG")
ReservedValueToken         = ("ReservedValue"         / "RV")
SendonlyToken              = ("SendOnly"              / "SO")
SendrecvToken              = ("SendReceive"           / "SR")
ServicesToken              = ("Services"              / "SV")
ServiceStatesToken         = ("ServiceStates"         / "SI")
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.

Los bits 7-4 se codifican como 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 se codifican como 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 las propiedades de los trenes de medios

Los parámetros para descriptores Local (local), Remote (distante) y LocalControl (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), 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".

Los rótulos se indican como números hexadecimales en este anexo. Cuando se da valor a 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 que se implementaran 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. "cadena" se tratará 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 de octetos de tamaño 1.

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.

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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
Echocanc	100B		No se utiliza. Véase E.13 para un ejemplo de posibles propiedades de control de eco.
Ganancia	100C	Entero sin signo	Ganancia en dB: 0..65535
Jitterbuff	100D	Entero sin signo	Tamaño de la memoria tampón para fluctuación de fase 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.
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

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 2 1 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 2 1 0 0 punto a punto 0 1 punto a multipunto Ref.: Rec. UIT-T Q.2931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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
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

<table>
<tr><th>Identificador de propiedad</th><th>Rótulo de propiedad</th><th>Tipo</th><th colspan="2">Valor</th></tr>
<tr><td rowspan="9">QosClass</td><td rowspan="9">4019</td><td rowspan="9">Entero 0-5</td><td colspan="2">Clase de calidad de servicio (QoS)</td></tr>
<tr><td>Clase de QoS</td><td>Significado</td></tr>
<tr><td>0</td><td>QoS por defecto asociada con la capacidad ATC definida en la Rec. UIT-T Q.2961.2</td></tr>
<tr><td>1</td><td>Restrictiva</td></tr>
<tr><td>2</td><td>Tolerante</td></tr>
<tr><td>3</td><td>Binivel</td></tr>
<tr><td>4</td><td>No limitada</td></tr>
<tr><td>5</td><td>Restrictiva binivel</td></tr>
<tr><td colspan="2">Ref.: Rec. UIT-T Q.2965.1</td></tr>
<tr><td>AALtype</td><td>401A</td><td>1 octeto</td><td colspan="2">Tipo AAL
Bits
8 7 6 5 4 3 2 1
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</td></tr>
</table>

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	32 bits Ipv4Address	Dirección Ipv4 Ref.: IETF RFC 791
IPv6	6002	128 bits	Dirección IPv6 Ref.: IETF RFC 2460
Port	6003	Entero sin signo	0..65535
Porttype	6004	Enumerado	TCP(0), UDP(1), SCTP(2)

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: SAR asegurada (14 octetos); SAR no asegurada (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.: Recomendaciones 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, *generic identifier transport*) Ref.: Rec. UIT-T Q.2941.1
AAL1ST	8001	1 octeto	Subtipo AAL 1 Bits 8 7 6 5 4 3 2 1 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 8 7 6 5 4 3 2 1 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 8 7 6 5 4 3 2 1 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 8 7 6 5 4 3 2 1 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
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

Las entradas del cuadro relativas 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 8 7 6 5 4 3 2 1 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 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

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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 5 4 3 2 1 0 0 0 0 0 habla 0 1 0 0 0 información digital sin restricciones 0 1 0 0 1 información digital con restricciones 1 0 0 0 0 audio 3,1 kHz 1 0 0 0 1 información digital sin restricciones con tonos/anuncios 1 1 0 0 0 vídeo Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.763
TransMode	9005	2 bits	Modo transferencia Bits 2 1 0 0 Modo circuito 1 0 Modo paquete Ref.: Rec. UIT-T Q.931
TransRate	9006	5 bits	Velocidad de transferencia Bits 5 4 3 2 1 0 0 0 0 0 Se utilizará este código para llamadas en modo paquete 1 0 0 0 0 64 kbit/s 1 0 0 0 1 2 × 64 kbit/s 1 0 0 1 1 384 kbit/s 1 0 1 0 1 1536 kbit/s 1 0 1 1 1 1920 kbit/s 1 1 0 0 0 Multivelocidad (velocidad de base: 64 kbit/s) 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 5 4 3 2 1 0 0 0 0 1 Adaptación de velocidad según las Recomendaciones UIT-T 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 Recomendaciones UIT-T H.221 y H.242 0 0 1 1 0 Recomendaciones 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

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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 5 4 3 2 1 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 Recomendaciones 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 Recomendaciones 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 Recomendaciones 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 Recomendaciones 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 Recomendaciones UIT-T V.6 y X.1 0 1 1 1 1 56 kbit/s Rec. UIT-T V.6 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 Recomendaciones UIT-T V.6 y X.1 1 1 0 0 0 1,2/0,075 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 0 0 1 0,050 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 0 1 0 0,075 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 0 1 1 0,110 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 1 0 0 0,150 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 1 0 1 0,200 kbit/s Recomendaciones UIT-T V.6 y X.1 1 1 1 1 0 0,300 kbit/s Recomendaciones 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

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
INTRATE	900C	2 bits	Velocidad intermedia Bits 2 1 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, *network independent clock*) 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 enviante no soporta este procedimiento facultativo) 1 Puede aceptar datos con reloj independiente de la red (es decir, el enviante 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
flowcontrx	9010	Booleano	Control de flujo en recepción (Rx) 0 No puede aceptar datos con mecanismo de control de flujo (es decir, el enviante no soporta este mecanismo facultativo) 1 Puede aceptar datos con mecanismo de control de flujo (es decir, el enviante 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 El establecimiento de múltiples tramas no está soportado. Sólo se permiten tramas UI 1 El establecimiento de múltiples tramas está soportado 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

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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 2 1 0 0 No se utilizan 0 1 1 bit 1 0 1,5 bits 1 1 2 bits Ref.: Rec. UIT-T Q.931
databits	9018	2 bits	Número de bits de datos, sin contar el bit de paridad, si está presente Bits 2 1 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 3 2 1 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

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
modem	901B	6 bits	Tipo de módem Bits 6 5 4 3 2 1 0 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 *bis* 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 *bis* 0 1 0 1 1 1 Rec. UIT-T V.26 *ter* 0 1 1 0 0 0 Rec. UIT-T V.27 0 1 1 0 0 1 Rec. UIT-T V.27 *bis* 0 1 1 0 1 0 Rec. UIT-T V.27 *ter* 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 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
layer2prot	901C	5 bits	Protocolo de capa 2 de información de usuario Bits 5 4 3 2 1 0 0 0 1 0 Rec. UIT-T Q.921/I.441 0 0 1 1 0 Rec. UIT-T X.25, capa enlace 0 1 1 0 0 Control de enlace lógico LAN (ISO/CEI 8802-2) Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.931
layer3prot	901D	5 bits	Protocolo de capa 3 de información de usuario Bits 5 4 3 2 1 0 0 0 1 0 Rec. UIT-T Q.931 0 0 1 1 0 Rec. UIT-T X.25, capa paquetes 0 1 0 1 1 ISO/CEI TR 9577 (Identificación de protocolo en la capa de red) Todos los demás valores están reservados. Ref.: Rec. UIT-T Q.931
addlayer3prot	901E	Octeto	Protocolo de capa 3 de información de usuario adicional Bits Bits 4 3 2 1 4 3 2 1 1 1 0 0 1 1 0 0 Protocolo Internet (RFC 791) (ISO/CEI TR 9577) 1 1 0 0 1 1 1 1 Protocolo punto a punto (RFC 1661) Ref.: Rec. UIT-T Q.931

Identificador de propiedad	Rótulo de propiedad	Tipo	Valor
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	Indicadores de la naturaleza de la conexión Bits 2 1 *Indicador de satélite* 0 0 ningún circuito de satélite en la conexión 0 1 un circuito de satélite en la conexión 1 0 dos circuitos de satélite en la conexión 1 1 reserva Bits 4 3 *Indicador de prueba de continuidad* 0 0 no se requiere prueba de continuidad 0 1 se requiere prueba de continuidad en este circuito 1 0 prueba de continuidad efectuada en un circuito anterior 1 1 reserva Bit 5 *Indicador de dispositivo de control de eco* 0 dispositivo de control de eco de salida no incluido 1 dispositivo de control de eco de salida incluido Bits 8 7 6 Reserva Ref.: Rec. UIT-T Q.763
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 de SDP

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
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

Anexo D

Transporte por IP

D.1 Transporte por IP/UDP mediante el empleo de entramado a 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.

ALF es un conjunto de técnicas que permite a una aplicación, por oposición 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 el anexo 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 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 par 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 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 por 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 de haberse emitido la respuesta, sea 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

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 por 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" 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 DEBE hacer lo siguiente:

- Debe multiplicar por 2 el valor estimado del retardo promedio, AAD.
- Debe calcular un valor aleatorio uniformemente distribuido entre 0,5 AAD y AAD.
- Debe 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. Esto puede tener por consecuencia o bien un número irregular de retransmisiones, o valores de temporizador demasiado largos para ser eficaces. Las entidades que prevean que una transacción requerirá un largo tiempo de ejecución pueden enviar una respuesta provisional "transacción pendiente". DEBEN enviar esta respuesta si reciben 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. La terminación raíz tiene una propiedad (ProvisionalResponseTimerValue) que puede fijarse al número máximo solicitado de milisegundos entre la recepción de una instrucción y la transmisión de la respuesta TransactionPending. Tras la recepción de una respuesta final que sigue a la recepción de respuestas provisionales, se enviará una respuesta inmediata, y se utilizarán temporizadores de repetición normales más tarde. Una entidad que envía una respuesta provisional, INCLUIRÁ el campo immAckRequired en la respuesta final que sigue, indicando que se espera una confirmación inmediata. La recepción de una transacción pendiente después de la recepción de una contestación será ignorada.

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, a la que suele hacerse referencia como un 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 actividades 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 cuando, por ejemplo, una interfaz con una entidad se torna indisponible.
- Fallo de una entidad cuando, por ejemplo, toda una entidad se torna indisponible.

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 5 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 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"

Puede haber pérdidas de transporte en los mensajes transportados por TCP, 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

Con una entrega fiable, se espera que la incidencia de pérdida de una petición de transacción o de una contestación a la transacción sea muy baja. 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. Las entidades que pueden prever que una transacción requerirá un largo tiempo de ejecución pueden enviar una respuesta provisional, "transacción pendiente". Pueden enviar estas respuestas si reciben 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 lotes para uso con la presente Recomendación.

E.1 Genérico

PackageID: g (0x0001)

Versión: 1

Extiende: Ninguno

Descripción:

Lote genérico para los elementos frecuentes

E.1.1 Propiedades

Ninguna.

E.1.2 Eventos

Causa

EventID: causa (0x0001)

Evento de error genérico

Parámetros EventsDescriptor: Ninguno

Parámetros ObservedEventsDescriptor:

Causa general

ParameterID: Generalcause (0x0001)

Este parámetro agrupa los fallos en seis grupos, sobre los cuales el MGC puede actuar.

Tipo: enumeración

Valores posibles:

"NR" Liberación normal (Normal Release) (0x0001)

"UR" Recursos no disponibles (Unavailable Resources) (0x0002)

"FT" Fallo, Temporal (Failure, Temporary) (0x0003)

"FP" Fallo, Permanente (Failure, Permanent) (0x0004)

"IW" Error de interfuncionamiento (Interworking Error) (0x0005)

"UN" No soportado (Unsupported) (0x0006)

Causa del fallo

ParameterID: Failurecause (0x0002)

Valores posibles: CADENA DE OCTETOS

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.

Compleción de la señal

EventID: sc (0x0002)

Indica la terminación de señales para las cuales el parámetro notifyCompletion se fijó para permitir informes de evento de compleción. Para una descripción más detallada del procedimiento, véanse 7.1.1, 7.1.17 y 7.2.7.

Parámetros EventsDescriptor: Ninguno

Parámetros ObservedEventsDescriptor:

Identidad de la señal

ParameterID: SigID (0x0001)

Este parámetro identifica la señal que ha terminado. En el caso de una señal que aparece en una lista de señales, el parámetro identidad de lista de señales también deberá retornarse e indicará la lista pertinente.

Tipo: Binario: (cadena de) octeto, Texto: cadena

Valores posibles: una señal que ha terminado. Una señal se identificará utilizando la sintaxis pkgdName sin comodines.

Método de terminación

ParameterID: Meth (0x0002)

Indica los medios por los cuales la señal terminó.

Tipo: enumeración

Valores posibles:

"TO" (0x0001) Expiró el periodo de temporización de la señal, o está concluyó por sí misma

"EV" (0x0002) Interrumpida por evento

"SD" (0x0003) Detenida por nuevo descriptor de señales

"NC" (0x0004) No completada, otra causa

ID de lista de señales

ParameterID: SLID (0x0003)

Indica a que lista de señales pertenece la señal en cuestión. El ID SignalList sólo se retorna cuando la señal está contenida en una lista de señales.

Tipo: entero

Valores posibles: cualquier entero

E.1.3 Señales

Ninguna.

E.1.4 Estadísticas

Ninguna.

E.2 Lote raíz base

Lote raíz base

PackageID: root (0x0002)

Versión: 2

Extiende: ninguno

Descripción:

Este lote define propiedades amplias de la pasarela.

E.2.1 Propiedades

MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

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

Definido en: TerminationState

Características: lectura solamente

MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

Número máximo de terminaciones permitido en un contexto; véase 6.1.

Tipo: entero

Valores posibles: cualquier entero

Definido en: TerminationState

Características: lectura solamente

normalMGExecutionTime

PropertyId: normalMGExecutionTime (0x0003)

Fijado por el MGC para indicar el intervalo dentro del cual el MGC espera una respuesta a cualquier transacción del MG (sin tener en cuenta el retardo de red).

Tipo: entero

Valores posibles: cualquier entero; representa milisegundos

Definido en: TerminationState

Características: lectura/escritura

normalMGCExecutionTime

PropertyId: normalMGCExecutionTime (0x0004)

Fijado por el MGC para indicar el intervalo dentro del cual la MG debe esperar una respuesta a cualquier transacción del MGC (sin tener en cuenta el retardo de red).

Tipo: entero

Valores posibles: cualquier entero; representa milisegundos

Definido en: TerminationState

Características: lectura/escritura

MGProvisionalResponseTimerValue

PropertyId: MGProvisionalResponseTimerValue (0x0005)

Indica cuánto tiempo debe esperar el MGC una respuesta pendiente de la MG si una transacción no puede ser completada. Inicialmente se fija a normalMGExecutionTime, más el retardo de red, pero puede reducirse.

Tipo: entero

Valores posibles: cualquier entero; representa milisegundos

Definido en: TerminationState

Características: lectura/escritura

MGCProvisionalResponseTimerValue

PropertyId: MGCProvisionalResponseTimerValue (0x0006)

Indica cuánto tiempo debe esperar la MG una respuesta pendiente del MGC si una transacción no puede ser completada. Inicialmente se fija a normalMGCExecutionTime, más el retardo de red, pero puede reducirse.

Tipo: entero

Valores posibles: cualquier entero; representa milisegundos

Definido en: TerminationState

Características: lectura/escritura

MGCOriginatedPendingLimit

PropertyId: MGCOriginatedPendingLimit (0x0007)

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

Definido en: TerminationState

Características: lectura/escritura

MGOriginatedPendingLimit

PropertyId: MGOriginatedPendingLimit (0x0008)

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

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

PackageID: tonegen (0x0003)

Versión: 1

Extiende: Ninguno

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 se van a configurar en las MG las características de los tonos apropiados para el país en que están ubicadas.

Previsto para ser extendido solamente: Sí

E.3.1 Propiedades

Ninguna.

E.3.2 Eventos

Ninguno.

E.3.3 Señales

Tono para la reproducción

SignalID: pt (0x0001)

Reproduce un tono de audio por un canal de audio

Tipo de señal: Breve

Duración: proporcionada

Parámetros adicionales:

Lista de Ids de tonos

ParameterID: tl (0x0001)

Tipo: lista de ids de tonos

Lista de tonos que habrán de ser reproducidos en secuencia. La lista CONTENDRÁ uno o más ids de tonos

Duración entre señales

ParameterID: ind (0x0002)

Tipo: entero

Periodo de temporización entre dos tonos consecutivos en milisegundos

No se especifican ids 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.3.4 Estadísticas

Ninguna.

E.3.5 Procedimientos

Ninguno.

E.4 Lote de detección de tonos

PackageID: tonedet (0x0004)

Versión: 1

Previsto para ser extendido solamente: Sí

Extiende: Ninguno

Este lote define eventos para la detección de tonos de audio Los tonos se seleccionan por sus nombres (id de tono). Cabe esperar que se van a configurar en las MG 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.

E.4.1 Propiedades

Ninguna.

E.4.2 Eventos

Comienzo de tono detectado

EventID: std, 0x0001

Detecta el comienzo de un tono. Las características de la detección positiva de tonos dependen de la implementación.

Parámetros EventDescriptor:

Lista de ids de tonos

ParameterID: tl (0x0001)

Tipo: lista de ids de tonos

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 valores posibles para id de tono. Si tl es "comodín", se detecta cualquier tono.

Parámetros ObservedEventsDescriptor:

id de tono

ParameterID: tid (0x0003)

Tipo: enumeración

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían valores posibles para id de tono.

Fin de tono detectado

EventID: etd, 0x0002

Detecta la terminación de un tono.

Parámetros EventDescriptor:

Lista de id de tonos

ParameterID: tl (0x0001)

Tipo: enumeración o lista de tipos enumerados

Valores posibles: No se especifican valores posibles en este lote. Las extensiones a este lote añadirían valores posibles para id de tono.

Parámetros ObservedEventsDescriptor:

Id de tono

ParameterID: tid (0x0003)

Tipo: enumeración

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían valores posibles para id de tono.

Duración

ParameterId: dur (0x0002)

Tipo: entero, en milisegundos

Este parámetro contiene la duración del tono desde la primera detección hasta que se para.

Tono largo detectado

EventID: ltd, 0x0003

Detecta que un tono ha estado reproduciéndose durante un determinado tiempo.

Parámetros EventDescriptor:

Lista de id de tonos

ParameterID: tl (0x0001)

Tipo: enumeración o lista

Valores posibles: "comodín" como se ha definido antes es el único valor definido en este lote. Las extensiones a este lote añadirían valores posibles para id de tono.

Duración

ParameterID: dur (0x0002)

Tipo: entero, duración con respecto a la cual se comprueba

Valores posibles: cualquier entero lícito, expresado en milisegundos

Parámetros ObservedEventsDescriptor:

Id de tono

ParameterID: tid (0x0003)

Tipo: Enumeración

Valores posibles: No se especifican valores posibles en este lote. Las extensiones a este lote añadirían valores posibles para id de tono.

E.4.3 Señales

Ninguna.

E.4.4 Estadísticas

Ninguna.

E.4.5 Procedimientos

Ninguno.

E.5 Lote generador de DTMF básico

PackageID: dg (0x0005)

Versión: 1

Extiende: tonegen versión 1

Este lote define los tonos DTMF básicos como señales y extiende los valores permitidos del parámetro tl de tono reproducido en tonegen.

E.5.1 Propiedades

Ninguna.

E.5.2 Eventos

Ninguno.

E.5.3 Señales

Carácter DTMF 0

SignalID: d0 (0x0010)

Generar tono DTMF 0. Las características físicas del tono DTMF 0 se definen en la pasarela.

Tipo de señal: Breve

Duración: proporcionada

Parámetros adicionales:

Ninguno

Valores adicionales:

d0 (0x0010) se define como u id de tono para tono reproducido

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. 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 SignalIds 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

PackageID: dd (0x0006)

Versión: 1

Extiende: tonedet versión 1

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, ids 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"

E.6.1 Propiedades

Ninguna.

E.6.2 Eventos

Dígitos DTMF

Los EventIds se definen con los mismos nombres que los SignalIds definidos en el cuadro de E.5.3.

Evento de compleción de DigitMap

EventID: ce, 0x0004

Generado cuando se completa un mapa de dígitos como se describe en 7.1.14.

Parámetros de EventsDescriptor: Ninguno.

Parámetros de ObservedEventsDescriptor:

DigitString

ParameterID: ds (0x0001)

Tipo: cadena de símbolos de mapa de dígitos (posiblemente vacía) retornada como una quotedString

Valores posibles: una secuencia de los caracteres "0" a "9", "A" a "F", y el modificador "Z" de larga duración.

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.

Método de terminación

ParameterID: Meth (0x0003)

Tipo: enumeración

Valores posibles:

"UM" (0x0001) Concordancia inequívoca

"PM" (0x0002) Concordancia parcial, compleción por expiración del temporizador o evento no concordante

"FM" (0x0003) Concordancia total, compleción por expiración del temporización o evento no concordante

Descripción: indica la razón de la generación del evento. Véanse los procedimientos de 7.1.14.

E.6.3 Señales

Ninguna.

E.6.4 Estadísticas

Ninguna.

E.6.5 Procedimientos

El procesamiento de mapa de dígitos se activa solamente si se activa un descriptor de eventos que contiene un evento de compleción de mapa de dígitos como se define en E.6.2 y ese evento de compleción de mapa de dígitos contiene un campo eventDM en las acciones solicitadas como se define en 7.1.9. Otros parámetros como KeepActive o eventos insertados de descriptores de señales pueden también estar presentes en el descriptor de eventos y no influyen en la activación del procesamiento de mapa de dígitos.

E.7 Lote generador de tonos de progresión de la llamada

PackageID: cg, 0x0007

Versión: 1

Extiende: tonegen versión 1

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.

E.7.1 Propiedades

Ninguna.

E.7.2 Eventos

Ninguno.

E.7.3 Señales

Tono de invitación a marcar

SignalID: dt (0x0030)

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: proporcionada

Parámetros adicionales:

Ninguno

Valores adicionales:

dt (0x0030) se define como un id de tono para tono reproducido

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. 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 ids de tonos corresponde a los definidos en la Rec. UIT-T E.180/Q.35. Véase la Rec. UIT-T E.180/Q.35 para la definición del significado de estos tonos.

E.8 Lote de detección de tonos de progresión de la llamada

PackageID: cd (0x0008)

Versión: 1

Extiende: tonedet versión 1

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 ids 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 ids de tonos corresponde a la Rec. UIT-T E.180/Q.35. Véase la Rec. UIT-T E.180/Q.35 para la definición del significado de estos tonos.

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

PackageID: al, 0x0009

Versión: 1

Extiende: Ninguno

Este lote define eventos y señales para una línea analógica.

E.9.1 Propiedades

Ninguna.

E.9.2 Eventos

colgado

EventID: on (0x0004)

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 EventDescriptor:

Transición estricta

ParameterID: estricto (0x0001)

Tipo: enumeración

Valores posibles: "exacto" (0x00), "estado" (0x01), "error por fallo" (0x02)

"exacto" significa que únicamente se reconocerá una transición efectiva del conmutador a la situación de colgado;

"estado" significa que 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" significa que si el conmutador ya está colgado, la instrucción falla y se comunica un error.

Parámetros ObservedEventsDescriptor:

Estado inicial

ParameterID: init (0x0002)

Tipo: Booleano

Valores posibles:

"Verdadero" significa que se comunicó el evento porque la línea ya estaba en situación de colgado cuando el descriptor de eventos que contiene dicho evento fue activado.

"Falso" significa que el evento representa una transición efectiva a la situación de colgado.

descolgado

EventID: of (0x0005)

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 EventDescriptor:

Transición estricta

ParameterID: strict (0x0001)

Tipo: enumeración

Valores posibles: "exacto" (0x00), "estado" (0x01), "error por fallo" (0x02)

"exacto" significa que únicamente se reconocerá una transición efectiva del conmutador a la situación de descolgado;

"estado" 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" significa que si el conmutador ya está descolgado, la instrucción falla y se comunica un error.

Parámetros ObservedEventsDescriptor:

Estado inicial

ParameterID: init (0x0002)

Tipo: Booleano

Valores posibles:

"Verdadero" significa que se comunicó el evento porque 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.

señal de gancho conmutador

EventID: fl (0x0006)

Detecta señales producidas por el gancho conmutador del microteléfono. Se produce una señal de gancho conmutador cuando un colgado va seguido de un descolgado en un lapso de tiempo comprendido entre un máximo y un mínimo.

Parámetros EventDescriptor:

Duración mínima

ParameterID: mindur (0x0004)

Tipo: entero en milisegundos

Se proporciona valor por defecto

Duración máxima

ParameterID: maxdur (0x0005)

Tipo: entero en milisegundos

Se proporciona valor por defecto

Parámetros ObservedEventsDescriptor:

Ninguno.

E.9.3 Señales

timbre

SignalID: ri, 0x0002

Aplica corriente de timbre a la línea

Tipo de señal: temporización

Duración: proporcionada

Parámetros adicionales:

Cadencia

ParameterID: cad (0x0006)

Tipo: lista de enteros que representan duraciones de segmentos alternos activado y desactivado, que constituyen un ciclo completo de señales de timbre que comienzan con un segmento activado. Las unidades son milisegundos.

El valor por defecto es un valor fijo o un valor proporcionado. Las MG de función restringida pueden no tener en cuenta valores de cadencia que ellas sean incapaces de generar.

Frecuencia

ParameterID: freq (0x0007)

Tipo: entero en Hz

El valor por defecto es un valor fijo o un valor proporcionado. Las MG de función restringida pueden no tener en cuenta valores de cadencia que ellas sean incapaces de generar.

E.9.4 Estadísticas

Ninguna.

E.9.5 Procedimientos

Si el MGC establece un EventsDescriptor que contiene un evento de transición del estado del conmutador (colgado o descolgado) cuyo parámetro "estricto" (0x0001) está configurado en "error por fallo" y el estado del conmutador ya es el que implica la transición, la ejecución de la instrucción que contiene ese EventsDescriptor falla. La MG INCLUIRÁ el código de error 540 (Estado inicial no esperado del gancho) en su respuesta.

E.9.6 Código de error

Este lote define un nuevo código de error:

540 (Estado inicial no esperado del gancho)

En E.9.5 figura el procedimiento de utilización de este código.

E.10 Lote de continuidad básica

PackageID: ct (0x000a)

Versión: 1

Extiende: Ninguno

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 transceptor.

E.10.1 Propiedades

Ninguna.

E.10.2 Eventos

Compleción

EventID: cmp, 0x0005

Este evento detecta la compleción de la prueba de continuidad.

Parámetros EventDescriptor:

Ninguno

Parámetros ObservedEventsDescriptor:

Resultado

ParameterID: res (0x0008)

Tipo: enumeración

Valores posibles: éxito (0x0001), fracaso (0x0000).

E.10.3 Señales

Prueba de continuidad

SignalID: ct (0x0003)

Inicia el envío del tono de prueba de continuidad en la terminación a la cual se aplica.

Tipo de señal: temporización

Se proporciona valor por defecto

Parámetros adicionales:

Ninguno

Responder

SignalID: rsp (0x0004)

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

Se proporciona duración por defecto

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 campo 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, el evento cmp será generado con el valor éxito para el parámetro resultado. Si así no fuera, el evento cmp será generado con el valor fallo para el 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

PackageID: nt (0x000b)

Versión: 1

Extiende: Ninguno

Este lote define propiedades de terminaciones de red independientes del tipo de red.

E.11.1 Propiedades

Tamaño máximo de la memoria tampón para la fluctuación

PropertyID: jit (0x0007)

Esta propiedad establece un tamaño máximo de la memoria tampón para la fluctuación.

Tipo: entero en milisegundos

Valores posibles: esta propiedad se especifica en milisegundos.

Definido en: LocalControlDescriptor

Características: lectura/escritura

E.11.2 Eventos

fallo de red

EventID: netfail, 0x0005

La terminación genera este evento cuando detecta un fallo por un motivo relacionado con la red interna o con una red externa.

Parámetros EventDescriptor:

Ninguno

Parámetros ObservedEventsDescriptor:

causa

ParameterID: cs (0x0001)

Tipo: cadena

Valores posibles: cualquier cadena de texto

Este parámetro puede estar incluido con el evento de fallo para proporcionar información de diagnóstico sobre el motivo de fallo.

aviso de calidad

EventID: qualert, 0x0006

Esta propiedad permite a la MG indicar una pérdida de calidad de la conexión de red. La MG puede efectuar esto midiendo la pérdida de paquetes, la fluctuación entre llegadas, y el tiempo de propagación, e indicando seguidamente esto como un porcentaje de la pérdida de calidad.

Parámetros EventDescriptor:

Umbral

ParameterId: th (0x0001)

Tipo: entero

Valores posibles: 0 a 99

Descripción: umbral para el porcentaje de pérdida de calidad medida, calculado sobre la base de un método proporcionado que podría tomar en consideración la pérdida de paquetes, la fluctuación, y el retardo, por ejemplo. Se activa este evento cuando el valor calculado rebasa el umbral.

Parámetros ObservedEventsDescriptor:

Umbral

ParameterId: th (0x0001)

Tipo: entero

Valores posibles: 0 a 99

Descripción: umbral para el porcentaje de pérdida de calidad medida, calculado sobre la base de un método proporcionado que podría tomar en consideración la pérdida de paquetes, la fluctuación, y el retardo, por ejemplo.

E.11.3 Señales

Ninguna.

E.11.4 Estadísticas

Duración

StatisticsID: dur (0x0001)

Descripción: proporciona el lapso de tiempo durante el cual la terminación ha estado en el contexto.

Tipo: doble, en milisegundos

Octetos enviados

StatisticID: os (0x0002)

Tipo: doble

Valores posibles: cualquier entero de 64 bits

Octetos recibidos

StatisticID: or (0x0003)

Tipo: doble

Valores posibles: cualquier entero de 64 bits

E.11.5 Procedimientos

Ninguno.

E.12 Lote RTP

PackageID: rtp (0x000c)

Versión: 1

Extiende: Lote red versión 1

Este lote se utiliza para soportar la transferencia de datos multimedios basada en paquetes por medio de protocolo de transporte en tiempo real (RTP) (RFC 1889).

E.12.1 Propiedades

Ninguna.

E.12.2 Eventos

Transición de cabida útil

EventID: pltrans, 0x0001

Este evento detecta y notifica cuándo hay una transición de un formato de cabida útil RTP a otro formato.

Parámetros de EventDescriptor:

Ninguno

Parámetros de ObservedEventsDescriptor:

ParameterName: rtppayload

ParameterID: rtppltype, 0x01

Tipo: lista de tipos enumerados

Valores posibles: el método de codificación se especificará utilizando uno o más nombres de codificación válidos, como los definidos RTP AV o registrados en la IANA.

E.12.3 Señales

Ninguna.

E.12.4 Estadísticas

Paquetes enviados

StatisticID: ps (0x0004)

Tipo: doble

Valores posibles: cualquier entero de 64 bits

Paquetes recibidos

StatisticID: pr (0x0005)

Tipo: doble

Valores posibles: cualquier entero de 64 bits

Pérdida de paquetes

StatisticID: pl (0x0006)

Describe la tasa actual de pérdida de paquetes en un tren RTP, como se define en RFC 1889. La pérdida de paquetes se expresa como un porcentaje: número de paquetes perdidos en el intervalo entre dos informes de recepción, dividido por el número de paquetes que se esperan en ese intervalo.

Tipo: doble

Valores posibles: número entero de 32 bits y una fracción de 32 bits

Fluctuación

StatisticID: jit (0x0007)

Pide el valor actual de la fluctuación entre llegadas en un tren RTP como se define en RFC 1889. La fluctuación mide la variación del tiempo entre llegadas para paquetes de datos RTP.

Retardo

StatisticID: delay (0x0008)

Pide el valor actual del tiempo de propagación de paquete en unidades de indicación de tiempo. Equivale a la latencia promedio.

E.12.5 Procedimientos

Ninguno.

E.13 Lote de circuitos TDM

PackageID: tdmc (0x000d)

Versión: 1

Extiende: Lote red versión 1

Este lote puede ser utilizado por 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 utilizaciones diferentes de TDM.

E.13.1 Propiedades

Compensación de eco

PropertyID: ec (0x0008)

Tipo: booleano

Valores posibles:

"on" (cuando se pide compensación de eco) y

"off" (cuando se desactiva)

El valor por defecto se suministrará

Definido en: LocalControlDescriptor

Características: lectura/escritura

Control de ganancia

PropertyID: gain (0x000a)

El control de ganancia o la adaptación del nivel de señal y la reducción del nivel de ruido se utilizan para adaptar el nivel de la señal. Sin embargo es necesario desactivar esta función, por ejemplo en llamadas en que intervienen módems.

Tipo: entero

Valores posibles:

El parámetro control de ganancia puede especificarse, sea como "automático" (0xffffffff), sea como un número explícito de decibelios de ganancia (cualquier otro valor entero). El valor por defecto es proporcionado por la MG.

Definido en: LocalControlDescriptor

Características: lectura/escritura.

E.13.2 Eventos

Ninguno.

E.13.3 Señales

Ninguna.

E.13.4 Estadísticas

Ninguna.

E.13.5 Procedimientos

Ninguno.

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 AnalogLine 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 nulo 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 por defecto 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,
            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. El modo para este tren se fija a SendReceive. "al" es el lote de supervisión de línea analógica. Se supone que se han configurado Local y Remote.

```
MGC to MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 9999 {
    Context = - {
        Modify = A4444 {
            Media { Stream = 1 {
                     LocalControl {
                         Mode = SendReceive,
                         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 EventsDescriptor 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/1 [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/1 [124.124.124.222]:55555
Transaction = 10000 {
   Context = - {
       Notify = A4444 {ObservedEvents =2222 {
         19990729T22000000:al/of(init=false)}}
   }
}
```

7) Y se acusa recibo de la instrucción Notificar.

```
MGC to MG1:
MEGACO/1 [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/1 [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|91xxxxxxxxx|9011x.)}
        }
    }
}
```

9) Y se acusa recibo de la instrucción Modificar.

```
MG1 to MGC:
MEGACO/1 [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/1 [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/1 [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 ReceiveOnly 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/1 [123.123.123.4]:55555
Transaction = 10003 {
    Context = $ {
       Add = A4444,
       Add = $ {
           Media {
             Stream = 1 {
                  LocalControl {
                      Mode = ReceiveOnly,

                      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/1 [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 SendReceive hasta el usuario de origen, usuario 1. El MGC también arregla el timbre en A5555.

```
MGC to MG2:
MEGACO/1 [123.123.123.4]:55555
Transaction = 50003 {
    Context = $ {
       Add = A5555  { Media {
            Stream = 1 {
                 LocalControl {Mode = SendReceive} }},
       Events=1234{al/of(strict=state)},
            Signals {al/ri}
            },
       Add  = $ {Media {
            Stream = 1 {
                 LocalControl {
                    Mode = SendReceive,
                    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/1 [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/1 [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/1 [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/1 [125.125.125.111]:55555
Transaction = 50005 {
   Context = 5000 {
       Notify = A5555 {ObservedEvents =1234 {
         19990729T22020002:al/of(init=false)}}
   }
}

From MGC to MG2:

MEGACO/1 [123.123.123.4]:55555
Reply = 50005 {
    Context = - {Notify = A5555}
}

From MGC to MG2:
```

```
MEGACO/1 [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/1 [125.125.125.111]:55555
Reply = 50006 {
 Context = 5000 {Modify = A4445}
}
```

18) Se cambia el modo en MG1 a SendReceive, y se detiene la señal de llamada de retorno.

```
MGC to MG1:
MEGACO/1 [123.123.123.4]:55555
Transaction = 10006 {
   Context = 2000 {
      Modify = A4445 {
         Media {
            Stream = 1 {
               LocalControl {
                  Mode=SendReceive
               }
            }
         }
      },
      Modify = A4444 {
         Signals { }
      }
   }
}

from MG1 to MGC:
MEGACO/1 [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/1 [123.123.123.4]:55555
Transaction = 50007 {
   Context = - {AuditValue = A5556{
      Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
   }
}
```

20) El MG2 contesta.

```
MEGACO/1 [125.125.125.111]:55555
Reply = 50007 {
   Context = - {
AuditValue = A5556 {
          Media {
             TerminationState { ServiceStates = InService,
                 Buffer = OFF },
             Stream = 1 {
                 LocalControl { Mode = SendReceive,
                    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/1 [125.125.125.111]:55555
Transaction = 50008 {
   Context = 5000 {
       Notify = A5555 {ObservedEvents =1235 {
          19990729T24020002:al/on(init=false)}
       }
   }
}

From MGC to MG2:

MEGACO/1 [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/1 [123.123.123.4]:55555
Transaction = 50009 {
   Context = 5000 {
      Subtract = A5555 {Audit{Statistics}},
      Subtract = A5556 {Audit{Statistics}}
   }
}

From MG2 to MGC:

MEGACO/1 [125.125.125.111]:55555
Reply = 50009 {
   Context = 5000 {
     Subtract = A5555 {
          Statistics {
             nt/os=45123, ; Octets Sent
             nt/dur=40 ; in seconds
             }
       },
       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
          }
       }
   }
}
```

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 nulo, en cuyo caso el MGC no tendría que enviar ningún mensaje a la MG. La terminación que retorna al contexto nulo toma de nuevo sus valores por defecto.

SERIES DE RECOMENDACIONES DEL UIT-T

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

23026

Impreso en Suiza
Ginebra, 2003