



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**T.125**

(02/98)

SERIE T: TERMINALES PARA SERVICIOS DE  
TELEMÁTICA

---

**Especificación de protocolo del servicio de  
comunicación multipunto**

Recomendación UIT-T T.125

(Anteriormente Recomendación del CCITT)

---

RECOMENDACIONES DE LA SERIE T DEL UIT-T  
**TERMINALES PARA SERVICIOS DE TELEMÁTICA**



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

## RECOMENDACIÓN UIT-T T.125

### ESPECIFICACIÓN DE PROTOCOLO DEL SERVICIO DE COMUNICACIÓN MULTIPUNTO

#### Resumen

Esta Recomendación define un protocolo que funciona en la estructura jerárquica de un dominio de comunicación multipunto. Especifica el formato de los mensajes del protocolo y los procedimientos que rigen su intercambio a través de un conjunto de conexiones de transporte. El objetivo del protocolo es implementar el servicio de comunicaciones multipunto definido en la Recomendación T.122.

El anexo A define un protocolo que permite la utilización de servicios de red multidistribución en una conferencia de tipo T.120. Para ello la Recomendación T.125 incluye el encaminamiento de datos. Asumida la responsabilidad del encaminamiento, el protocolo utiliza la multidifusión siempre que es posible. El anexo maneja redes subóptimas realizando pruebas de los trayectos multidifusión antes de que se dependa de ellos. Si la multidifusión no funciona en determinados puntos de la conferencia, en su lugar se utiliza la unidifusión. Este anexo es independiente de las redes en las que se utilice. Todos los aspectos que son específicos de una red dada se manejan en las pilas de protocolo de nivel inferior a éste.

Esta versión de la Recomendación introduce los cambios siguientes:

- el protocolo MAP se ha incluido como anexo A;
- se especifica la versión 3 del protocolo MCS;
- se han añadido las MCSPDU de *parámetros ampliados*;
- se han añadido las MCSPDU de *notificación de capacidades*;
- se han añadido campos a las MCSPDU de envío de datos y de envío de datos uniformes a fin de soportar cambios a la Recomendación T.122;
- se han realizado modificaciones de índole menor al formato del documento.

#### Orígenes

La Recomendación UIT-T T.125, ha sido revisada por la Comisión de Estudio 16 (1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 6 de febrero de 1998.

## PREFACIO

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

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

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

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

## NOTA

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

## PROPIEDAD INTELECTUAL

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

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

© UIT 1998

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

## ÍNDICE

### Página

1	Alcance .....	1
2	Referencias.....	1
3	Definiciones .....	2
4	Abreviaturas.....	4
5	Visión de conjunto del protocolo MCS .....	5
5.1	Modelo de la capa MCS.....	5
5.2	Servicios proporcionados por la capa MCS.....	5
5.2.1	Servicios de la capa MAP.....	5
5.3	Servicios tomados de la capa de transporte .....	6
5.4	Funciones de la capa MCS.....	6
5.4.1	Gestión de dominio.....	6
5.4.2	Gestión de canal.....	8
5.4.3	Transferencia de datos .....	8
5.4.4	Gestión de testigo .....	9
5.4.5	Notificación de capacidades .....	9
5.5	Procesamiento jerárquico.....	9
5.6	Parámetros de dominio .....	11
5.7	Parámetros ampliados .....	12
6	Utilización del servicio de transporte .....	12
6.1	Modelo del servicio de transporte.....	12
6.2	Utilización de múltiples conexiones.....	13
6.3	Liberación de la conexión de transporte .....	14
7	Estructura de las MCSPDU de versión 2.....	14
8	Estructura de las MCSPDUs de versión 3 .....	24
9	Codificación de las MCSPDU .....	37
10	Encaminamiento de las MCSPDU.....	38
10.1	MCSPDU de conexión y de parámetros ampliados.....	38
10.2	MCSPDU de dominio.....	40
11	Significado de las MCSPDU .....	43
11.1	Conexión-inicial.....	43
11.2	Conexión-respuesta.....	44
11.3	Conexión-adicional.....	45

	<b>Página</b>
11.4	Conexión-resultado..... 45
11.5	Propuesta-parámetros-ampliados..... 46
11.6	Aceptación-parámetros-ampliados..... 46
11.7	PlumbDomainIndication..... 47
11.8	ErectDomainRequest..... 47
11.9	MergeChannelsRequest..... 48
11.10	MergeChannelsConfirm..... 49
11.11	PurgeChannelsIndication..... 50
11.12	MergeTokensRequest..... 50
11.13	MergeTokensConfirm..... 52
11.14	PurgeTokensIndication..... 52
11.15	DisconnectProviderUltimatum..... 53
11.16	RejectMCSPDUUltimatum..... 53
11.17	AttachUserRequest..... 53
11.18	AttachUserConfirm..... 54
11.19	DetachUserRequest..... 54
11.20	DetachUserIndication..... 55
11.21	ChannelJoinRequest..... 56
11.22	ChannelJoinConfirm..... 57
11.23	ChannelLeaveRequest..... 57
11.24	ChannelConveneRequest..... 58
11.25	ChannelConveneConfirm..... 58
11.26	ChannelDisbandRequest..... 59
11.27	ChannelDisbandIndication..... 59
11.28	ChannelAdmitRequest..... 59
11.29	ChannelAdmitIndication..... 60
11.30	ChannelExpelRequest..... 60
11.31	ChannelExpelIndication..... 61
11.32	SendDataRequest..... 61
11.33	SendDataIndication..... 62
11.34	UniformSendDataRequest..... 63
11.35	UniformSendDataIndication..... 64
11.36	TokenGrabRequest..... 65
11.37	TokenGrabConfirm..... 65

	<b>Página</b>
11.38 TokenInhibitRequest.....	66
11.39 TokenInhibitConfirm.....	66
11.40 TokenGiveRequest.....	66
11.41 TokenGiveIndication.....	67
11.42 TokenGiveResponse.....	67
11.43 TokenGiveConfirm.....	68
11.44 TokenPleaseRequest.....	69
11.45 TokenPleaseIndication.....	69
11.46 TokenReleaseRequest.....	69
11.47 TokenReleaseConfirm.....	70
11.48 TokenTestRequest.....	70
11.49 TokenTestConfirm.....	70
11.50 CapabilitiesNotificationRequest.....	71
11.50.1 Petición de capacidades.....	71
11.51 CapabilitiesNotificationIndication.....	72
11.51.1 Indicación de capacidades.....	72
12 Base de información de proveedor MCS.....	73
12.1 Replicación de la jerarquía.....	73
12.2 Información de canal.....	74
12.3 Información de testigo.....	75
13 Elementos de procedimiento.....	78
13.1 Secuenciación de las MCSPDU.....	78
13.2 Control de flujo de entrada.....	78
13.3 Cumplimentación del caudal.....	79
13.4 Configuración de los dominios.....	80
13.5 Fusión de dominios.....	81
13.6 Desconexión de dominio.....	84
13.7 Atribución de identificadores de canal.....	84
13.8 Estado del testigo.....	85
13.9 Notificación de capacidades.....	85
13.9.1 Identificadores de capacidad.....	86
13.9.2 Cumbres de la versión 3.....	86
13.9.3 Reglas para la selección de una capacidad.....	86
13.9.4 Establecimiento del proveedor de cumbre V3.....	87
13.9.5 Ampliación de la cumbre V3.....	88

	<b>Página</b>
13.9.6	Procesamiento de las PDU en un nodo hoja..... 88
13.9.7	Procesamiento de las PDU en un nodo intermedio ..... 89
13.10	Arbitraje sobre la versión del protocolo..... 93
13.11	Interfuncionamiento de la versión de protocolo ..... 93
Anexo A – Protocolo de adaptación multidifusión ..... 94	
A.1	Campo de aplicación..... 94
A.2	Referencias normativas ..... 95
A.3	Definiciones ..... 96
A.4	Abreviaturas ..... 97
A.5	Visión de conjunto ..... 97
A.5.1	La utilización de la multidifusión ..... 98
A.5.2	Islas multidifusión ..... 99
A.5.3	Proveedores de grupo multidifusión ..... 101
A.5.4	Ordenamiento de los datos ..... 101
A.5.5	Utilización de unidifusión no fiable y de multidifusión ..... 101
A.6	Utilización del MAP ..... 102
A.6.1	Gestión del dominio..... 102
A.6.2	Gestión del canal ..... 103
A.6.3	Transferencia de datos ..... 103
A.6.4	Gestión del testigo ..... 106
A.7	Utilización de protocolos de transporte ..... 106
A.7.1	Protocolos de transporte unidifusión ..... 106
A.7.2	Protocolos de transporte multidifusión..... 108
A.7.3	Información local..... 110
A.8	Especificación de protocolo..... 111
A.8.1	Establecimiento de la conexión unidifusión fiable inicial..... 111
A.8.2	Arbitraje de protocolos de transporte ..... 112
A.8.3	Asignación y distribución de grupos multidifusión..... 115
A.8.4	Gestión del tráfico de datos multidifusión (autoajuste)..... 116
A.8.5	Abandono de la multidifusión ..... 118
A.8.6	Supresión de los grupos multidifusión ..... 119
A.9	Descripción de las MAPPDU ..... 120
A.9.1	MAPConnectRequest ..... 120
A.9.2	MAPConnectConfirm..... 121
A.9.3	MAPDisconnectRequest..... 122
A.9.4	MAPDisconnectConfirm ..... 122
A.9.5	MAPArbitrateProtocolsRequest ..... 122

	<b>Página</b>
A.9.6 MAPArbitrateProtocolsConfirm .....	125
A.9.7 MAPData .....	127
A.9.8 MAPAddGroupRequest.....	129
A.9.9 MAPRemoveGroupRequest .....	131
A.9.10 MAPDisableUnicastRequest .....	132
A.9.11 MAPEnableUnicastRequest .....	132
A.9.12 MAPEnableUnicastConfirm.....	132
A.9.13 MAPDisableMulticastRequest .....	133
A.9.14 MAPDisableMulticastConfirm.....	134
A.9.15 MAPEnableMulticastRequest .....	134
A.9.16 MAPSequenceNumber .....	134
A.10 Definición ASN.1 de las MAPPDU .....	135



## Recomendación T.125

# ESPECIFICACIÓN DE PROTOCOLO DEL SERVICIO DE COMUNICACIÓN MULTIPUNTO

(revisada en 1998)

## 1 Alcance

Esta Recomendación especifica:

- a) los procedimientos de un protocolo para la transferencia de información de datos y de control de un proveedor MCS a otro proveedor MCS par;
- b) la estructura y codificación de las unidades de datos de protocolo MCS utilizadas para la transferencia de información de datos y de control.

Los procedimientos se definen en función de:

- a) las interacciones entre proveedores MCS pares mediante el intercambio de unidades de datos de protocolo MCS;
- b) las interacciones entre un proveedor MCS y usuarios MCS mediante el intercambio de primitivas MCS;
- c) las interacciones entre un proveedor MCS y un proveedor del servicio de transporte mediante el intercambio de primitivas del servicio de transporte.

Estos procedimientos son aplicables a ejemplares (dícese instancias) de comunicación multipar entre sistemas que soportan MCS y desean interconectar en un entorno de sistemas abiertos.

## 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 del UIT-T actualmente vigentes.

- Recomendación UIT-T T.122 (1998), *Servicio de comunicación multipunto – Definición de los servicios.*
- Recomendación UIT-T T.123 (1996), *Pilas de protocolos de datos específicos de la red para conferencias multimedios.*
- Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: el modelo básico.*
- Recomendación UIT-T X.214 (1995) | ISO/CEI 8072:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de transporte.*
- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno (ASN.1) – Especificación de la notación básica.*

- Recomendación UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno: Especificación de las reglas de codificación básicas, de las reglas de codificación canónicas y de las reglas de codificación distinguidas.*
- Recomendación UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno – Especificación de las reglas de codificación compacta.*

### **3 Definiciones**

NOTA – En estas definiciones se utilizan las abreviaturas definidas en la cláusula 4.

Esta Recomendación se basa en los conceptos elaborados en la Recomendación X.200 y emplea los siguientes términos definidos en la misma:

- a) control de flujo;
- b) reensamblado;
- c) recombinación;
- d) segmentación;
- e) secuenciación;
- f) división;
- g) sintaxis de transferencia;
- h) conexión de transporte;
- i) identificación de punto extremo de conexión de transporte;
- j) servicio de transporte;
- k) punto de acceso al servicio de transporte;
- l) dirección de punto de acceso al servicio de transporte;
- m) unidad de datos del servicio de transporte.

Esta Recomendación se basa también en los conceptos elaborados en la Recomendación T.122 y emplea los siguientes términos definidos en la misma:

- a) MCSAP de control;
- b) anexión MCS;
- c) canal MCS;
- d) conexión MCS;
- e) dominio MCS;
- f) selector de dominio MCS;
- g) canal privado MCS;
- h) gestor de canal privado MCS;
- i) proveedor MCS;
- j) punto de acceso al servicio MCS;
- k) usuario MCS;
- l) identificador de usuario MCS;
- m) proveedor MCS superior.

En esta Recomendación se definen los términos siguientes.

**3.1 unidad de datos del servicio MCS:** Una cantidad de datos de usuario MCS cuya identidad queda preservada durante la transferencia de transmisor a receptores. Específicamente, el contenido de una petición MCS-ENVÍO-DATOS o de una petición MCS-ENVÍO-DATOS-UNIFORME.

**3.2 unidad de datos de interfaz MCS:** La unidad de información transferida a través de un MCSAP entre un usuario MCS y un proveedor MCS en una sola interacción. Cada unidad de datos de interfaz MCS contiene información de control de interfaz y puede también contener una unidad de datos del servicio MCS o parte de ella.

**3.3 unidad de datos de protocolo MCS:** Una unidad de información intercambiada en el protocolo MCS, constituida por información de control transferida entre proveedores MCS para coordinar su operación combinada y posiblemente datos transferidos a nombre de usuarios MCS a los cuales están prestando servicio.

**3.4 prioridad de transferencia de datos MCS:** Uno de cuatro niveles: superior, alto, medio, bajo. El valor se comunica sin modificación del transmisor a los receptores. En función de un parámetro de dominio MCS que da el número de prioridades distintas de datos implementadas, dos o más prioridades de las más bajas pueden recibir la misma calidad de servicio.

**3.5 unidad de datos de protocolo de servicio de comunicación multipunto válida:** Una MCSPDU cuya estructura y codificación se ajustan a esta Recomendación.

**3.6 unidad de datos de protocolo de servicio de comunicación multipunto no válida:** Una MCSPDU que no es válida.

**3.7 error de protocolo:** Utilización de una MCSPDU de una manera que no se ajusta a los procedimientos de esta Recomendación.

**3.8 unidad de datos de protocolo de servicio de comunicación multipunto de conexión:** Cualquiera de las siguientes: **conexión-inicial**, **conexión-respuesta**, **conexión-adicional**, **conexión-resultado**.

**3.9 unidad de datos de protocolo de servicio de comunicación multipunto de parámetros ampliados:** Puede ser **propuesta de parámetros ampliados** o **aceptación de parámetros ampliados**.

**3.10 unidad de datos de protocolo de servicio de comunicación multipunto de dominio:** Toda MCSPDU que no sea una MCSPDU de conexión.

**3.11 unidad de datos de protocolo de servicio de comunicación multipunto de datos:** Cualquiera de las siguientes: **SendDataRequest**, **SendDataIndication**, **UniformSendDataRequest**, **UniformSendDataIndication**.

**3.12 unidad de datos de protocolo de servicio de comunicación multipunto de control:** Cualquier MCSPDU dominio que no sea una MCSPDU de datos.

**3.13 conexión de transporte inicial:** La primera conexión de transporte de una conexión MCS, utilizada para intercambiar MCSPDU de control y MCSPDU de datos de la prioridad más alta.

**3.14 conexión de transporte adicional:** Una ulterior conexión de transporte que pertenece a una conexión MCS y se utiliza para intercambiar MCSPDU de datos de una prioridad más baja.

**3.15 subárbol de un proveedor de servicio de comunicación multipunto:** En el contexto de un dominio MCS, está constituido por el propio proveedor MCS y sus anexiones MCS, junto con sus proveedores MCS jerárquicamente subordinados y las anexiones MCS de éstos.

**3.16 altura de un proveedor de servicio de comunicación multipunto:** En el contexto de un dominio MCS, es un valor igual a la altura máxima de todos los proveedores MCS jerárquicamente subordinados aumentada en una unidad. Un proveedor MCS sin subordinados tiene una altura de uno.

**3.17 versión 3 o cumbre V3:** Proveedores MCS de un dominio jerárquico que utilizan la versión 3 o superior del protocolo MCS y que:

- se trata de un proveedor superior; o bien
- tienen un trayecto de conexión ascendente a un proveedor superior de versión 3 que sólo fluye a través de otros proveedores de versión 3.

**3.18 nodo hoja:** Un nodo con una conexión ascendente y sin conexiones descendentes.

**3.19 nodo intermedio:** Un nodo con una conexión ascendente y con al menos una conexión descendente.

**3.20 nodo superior:** Todo nodo con una conexión ascendente está conectado a su nodo superior.

**3.21 nodo interviniente:** Con respecto a un nodo específico, se llama nodo interviniente o partícipe a cualquier nodo situado en el trayecto desde dicho nodo específico hasta el proveedor cumbre.

**3.22 capacidad:** Comportamiento de alto nivel del sistema, que normalmente requiere la cooperación entre dos o más nodos.

**3.23 identificador de capacidad:** Identificador asignado a una capacidad. Si está normalizado, este identificador está documentado en la Recomendación T.125.

**3.24 indicador de participación:** Característica de una capacidad que indica el nivel de participación esperado de los nodos V3 en una cumbre V3 a fin de ejercitar dicha capacidad. El valor de un indicador de participación puede ser parcial (deben participar dos o más nodos) o global (deben participar todos los nodos).

**3.25 capacidad parcial:** Capacidad con un indicador de participación "parcial".

**3.26 capacidad global:** Capacidad con un indicador de participación "global".

**3.27 lista de capacidades cumbre:** Lista de capacidades disponibles en la cumbre V3.

**3.28 proveedor de cumbre V3:** Cualquier proveedor superior que sea nodo V3.

## 4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

MAP	Protocolo de adaptación de multidistribución ( <i>multicast adaptation protocol</i> ) tal como se describe en el anexo A
MAPSAP	Punto de acceso al servicio MAP ( <i>MAP service access point</i> )
MC	Conexión MAP ( <i>MAP connection</i> )
MCS	Servicio de comunicación multipunto ( <i>multipoint communication service</i> )
MCSAP	Punto de acceso al servicio MCS ( <i>MCS service access point</i> )
MCSPDU	Unidad de datos de protocolo MCS ( <i>MCS protocol data unit</i> )
PDU	Unidad de datos de protocolo ( <i>protocol data unit</i> )
TC	Conexión de transporte ( <i>transport connection</i> )
TS	Servicio de transporte ( <i>transport service</i> )

TSAP	Punto de acceso al servicio de transporte ( <i>transport service access point</i> )
TSDU	Unidad de datos del servicio de transporte ( <i>transport service data unit</i> )
V2	Versión 2
V3	Versión 3

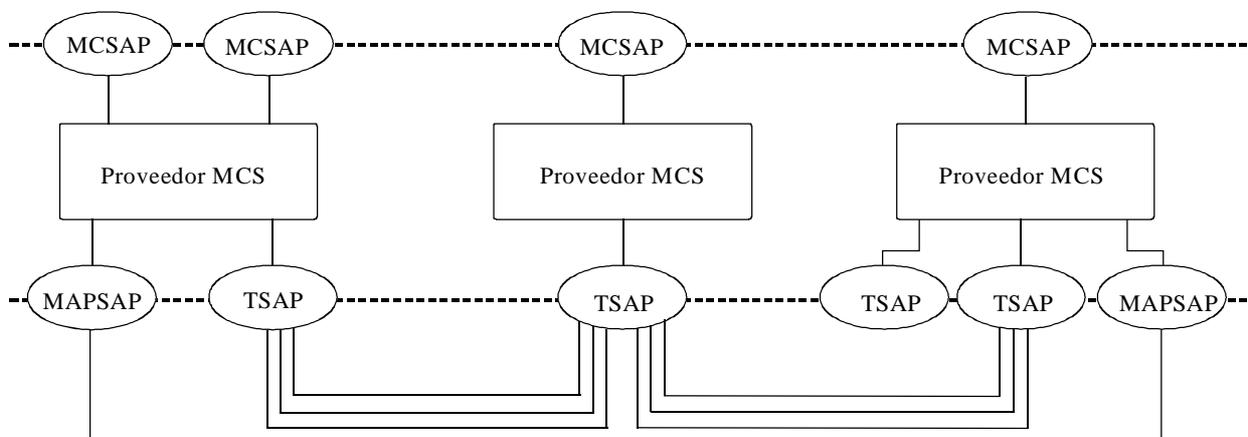
## 5 Visión de conjunto del protocolo MCS

### 5.1 Modelo de la capa MCS

Un proveedor MCS comunica con usuarios MCS a través de un MCSAP por medio de las primitivas MCS definidas en la Recomendación T.122. Estas primitivas pueden ser la causa o el resultado de intercambio de MCSPDU entre proveedores MCS pares que utilizan una conexión MCS, o pueden ser la causa o el resultado de acciones ejecutadas dentro de un solo proveedor MCS. Los intercambios de MCSPDU tienen lugar entre proveedores MCS que atienden el mismo dominio MCS.

Un proveedor MCS puede tener múltiples pares, a cada uno de los cuales se llega directamente por una conexión MCS o indirectamente a través de un proveedor MCS par. Una conexión MCS se compone de una conexión MAP o de una o más conexiones de transporte. Las conexiones basadas en MAP funcionan tal como se describe en el anexo A. En el caso de conexiones basadas en el transporte, el número de conexiones depende del número de prioridades de transferencia de datos implementadas en un dominio MCS. Los intercambios de protocolo se efectúan utilizando los servicios de la capa de transporte a través de un par de TSAP.

Este modelo de la capa MCS se ilustra en la figura 5-1.



T1602400-97

**Figura 5-1/T.125 – Modelo de la capa MCS**

### 5.2 Servicios proporcionados por la capa MCS

El protocolo MCS soporta los servicios definidos en la Recomendación T.122. La información se transfiere a y desde un usuario MCS mediante las primitivas indicadas en el cuadro 5-1.

#### 5.2.1 Servicios de la capa MAP

Cuando se utilizan conexiones MAP, el MCS funciona como se describe en el anexo A.

### 5.3 Servicios tomados de la capa de transporte

Para conexiones de transporte, el protocolo MCS utiliza un subconjunto del servicio de transporte orientado a conexión definido en la Recomendación X.214. Para transferir información a, o desde, un proveedor del servicio de transporte se utilizan las primitivas indicadas en el cuadro 5-2.

### 5.4 Funciones de la capa MCS

El cuadro 5-1 identifica las unidades funcionales de MCS y las MCSPDU asociadas con cada primitiva MCS. Las MCSPDU se definen en la cláusula 7. Las primitivas y las MCSPDU pueden encontrarse en una simple relación de causa a efecto, en un sentido o en el otro. Por ejemplo, petición MCS-ANEXIÓN-USUARIO genera **AttachUserRequest**, y **AttachUserConfirm** genera confirmación MCS-ANEXIÓN-USUARIO. Otros casos pueden ser más complicados. Para la compleción de, por ejemplo, MCS-CONEXIÓN-PROVEEDOR es necesario el intercambio de MCSPDU adicionales como efectos secundarios de la primitiva de cuatro fases. Además, una cualquiera de cinco MCSPDU puede causar una indicación MCS-DESANEXIÓN-USUARIO, y una cualquiera de cuatro puede causar una indicación MCS-EXCLUSIÓN-CANAL.

#### 5.4.1 Gestión de dominio

La capa MCS mantiene la integridad de las conexiones MCS que constituyen un dominio MCS. Una conexión MCS es una conexión orientada, siendo uno de sus extremos el superior jerárquico del otro. Hay un solo proveedor MCS a la cabeza de cada dominio.

Al establecerse una conexión MCS, dos dominios se fusionan en uno. La capa MCS asegura que quede un proveedor superior. De ese modo se resuelve cualquier conflicto de identidad única o propiedad exclusiva que puedan surgir.

Al desconectarse una conexión MCS, un dominio se divide en dos porciones. La porción que contiene el proveedor superior subsiste. La porción inferior desaparece por sí misma.

La capa MCS identifica inequívocamente a los usuarios anexionados a un dominio. Los usuarios pueden tener conocimiento de cada uno de los demás mediante sus interacciones por primitivas MCS. La capa MCS notifica a todos los usuarios de un dominio cuando uno de ellos ha quedado desafectado. La capa MCS recupera los recursos de un usuario desanexionado.

**Cuadro 5-1/T.125 – Primitivas MCS**

<b>Unidad funcional</b>	<b>Primitivas</b>	<b>MCSPDU asociadas</b>
Gestión de dominio	Petición MCS-CONEXIÓN-PROVEEDOR Indicación MCS-CONEXIÓN-PROVEEDOR Respuesta MCS-CONEXIÓN-PROVEEDOR Confirmación MCS-CONEXIÓN-PROVEEDOR (efectos secundarios)	<b>Conexión-inicial</b> <b>Conexión-inicial</b> <b>Conexión-respuesta</b> <b>Conexión-respuesta</b> <b>Propuesta-parámetros-ampliados</b> <b>Aceptación-parámetros-ampliados</b> <b>Conexión-adicional</b> <b>Conexión-resultado</b> <b>PlumbDomainIndication</b> <b>ErectDomainRequest</b> <b>MergeChannelsRequest</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b> <b>MergeTokensRequest</b> <b>MergeTokensConfirm</b> <b>PurgeTokensIndication</b>

**Cuadro 5-1/T.125 – Primitivas MCS (fin)**

<b>Unidad funcional</b>	<b>Primitivas</b>	<b>MCSPDU asociadas</b>
	Petición MCS-DESCONEXIÓN-PROVEEDOR Indicación MCS-DESCONEXIÓN-PROVEEDOR	<b>DisconnectProviderUltimatum</b> <b>DisconnectProviderUltimatum</b> <b>RejectMCSPDUUltimatum</b>
	Petición MCS-ANEXIÓN-USUARIO Confirmación MCS-ANEXIÓN-USUARIO	<b>AttachUserRequest</b> <b>AttachUserConfirm</b>
	Petición MCS-DESANEXIÓN-USUARIO Indicación MCS-DESANEXIÓN-USUARIO	<b>DetachUserRequest</b> <b>DetachUserIndication</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b> <b>MergeTokensConfirm</b> <b>PurgeTokensIndication</b>
Gestión de canal	Petición MCS-INCORPORACIÓN-CANAL Confirmación MCS-INCORPORACIÓN-CANAL	<b>ChannelJoinRequest</b> <b>ChannelJoinConfirm</b>
	Petición MCS-ABANDONO-CANAL Indicación MCS-ABANDONO-CANAL	<b>ChannelLeaveRequest</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>
	Petición MCS-FORMACIÓN-CANAL Confirmación MCS-FORMACIÓN-CANAL	<b>ChannelConveneRequest</b> <b>ChannelConveneConfirm</b>
	Petición MCS-DISOLUCIÓN-CANAL Indicación MS-DISOLUCIÓN-CANAL	<b>ChannelDisbandRequest</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>
	Petición MCS-ADMISIÓN-CANAL Indicación MCS-ADMISIÓN-CANAL	<b>ChannelAdmitRequest</b> <b>ChannelAdmitIndication</b>
	Petición MCS-EXCLUSIÓN-CANAL Indicación MCS-EXCLUSIÓN-CANAL	<b>ChannelExpelRequest</b> <b>ChannelExpelIndication</b> <b>ChannelDisbandIndication</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>
Transferencia de datos	Petición MCS-ENVÍO-DATOS Indicación MCS-ENVÍO-DATOS	<b>SendDataRequest</b> <b>SendDataIndication</b>
	Petición MCS-ENVÍO-DATOS-UNIFORME Indicación MCS-ENVÍO-DATOS-UNIFORME	<b>UniformSendDataRequest</b> <b>UniformSendDataIndication</b>
Gestión de testigo	Petición MCS-TOMA-TESTIGO Confirmación MCS-TOMA-TESTIGO	<b>TokenGrabRequest</b> <b>TokenGrabConfirm</b>
	Petición MCS-INHIBICIÓN-TESTIGO Confirmación MCS-INHIBICIÓN-TESTIGO	<b>TokenInhibitRequest</b> <b>TokenInhibitConfirm</b>
	Petición MCS-CESIÓN-TESTIGO Indicación MCS-CESIÓN-TESTIGO Respuesta MCS-CESIÓN-TESTIGO Confirmación MCS-CESIÓN-TESTIGO	<b>TokenGiveRequest</b> <b>TokenGiveIndication</b> <b>TokenGiveResponse</b> <b>TokenGiveConfirm</b>
	Petición MCS-SOLICITUD-TESTIGO Indicación MCS-SOLICITUD-TESTIGO	<b>TokenPleaseRequest</b> <b>TokenPleaseIndication</b>
	Petición MCS-LIBERACIÓN-TESTIGO Confirmación MCS-LIBERACIÓN-TESTIGO	<b>TokenReleaseRequest</b> <b>TokenReleaseConfirm</b>
	Petición MCS-PRUEBA-TESTIGO Confirmación MCS-PRUEBA-TESTIGO	<b>TokenTestRequest</b> <b>TokenTestConfirm</b>
Notificación de capacidades		<b>CapabilitiesNotificationRequest</b> <b>CapabilitiesNotificationIndication</b>

**Cuadro 5-2/T.125 – Primitivas del servicio de transporte**

<b>Primitivas</b>	<b>Uso</b>	<b>Parámetros</b>	<b>Uso</b>
Petición T-CONEXIÓN	X	Dirección llamada	X
Indicación T-CONEXIÓN	X	Dirección llamante	X
		Opción datos acelerados	–
		Calidad de servicio	X
		Datos de usuario TS	–
Respuesta T-CONEXIÓN	X	Dirección respondedora	–
Confirmación T-CONEXIÓN	X	Opción datos acelerados	–
		Calidad de servicio	X
		Datos de usuario TS	–
Petición T-DATOS	X	Datos de usuario TS	X
Indicación T-DATOS	X		
Petición T-DATOS-ACELERADOS	–	Datos de usuario TS	–
Indicación T-DATOS-ACELERADOS	–		
Petición T-DESCONEXIÓN	X	Datos de usuario TS	–
Indicación T-DESCONEXIÓN	X	Motivo	–
		Datos de usuario TS	–
X El protocolo MCS presupone que esta prestación está siempre disponible.			
– El protocolo MCS no utiliza esta prestación.			

#### 5.4.2 Gestión de canal

La capa MCS registra qué partes de un dominio MCS contienen uno o más usuarios incorporados a un determinado canal, por lo que puede optimizar la transferencia de datos a destinos que deseen recibirlos.

La capa MCS trata los identificadores de usuario (id de usuario) como si fuesen canales de un solo miembro, a los cuales sólo el usuario designado está autorizado a incorporarse. A petición, puede crear canales privados a los cuales sólo tiene autorizado el acceso los usuarios admitidos, o asignar canales públicos que no tengan incorporados otros usuarios en ese momento.

#### 5.4.3 Transferencia de datos

La capa MCS mantiene un flujo secuenciado de datos a los usuarios que se han incorporado al canal. Un canal se convierte, en efecto, en una lista de distribución para multidifusión, con una gama comprendida entre ningún destino y la totalidad de los destinos en una difusión completa.

Por defecto, la capa MCS encamina datos a cada receptor a través del trayecto más corto de conexiones MCS. Facultativamente, encamina las unidades de datos de servicio MCS especificadas, a través del proveedor MCS superior, con lo que se garantiza su recepción uniforme por todos los receptores, entre los cuales puede estar también incluido el transmisor.

La capa MCS reconoce una o más prioridades de transferencia de datos y les concede un tratamiento preferencial. En el caso de datos completamente fiables, la capa MCS permite trabajar con unidades de datos de servicio MCS de tamaño ilimitado mediante la segmentación. Sin embargo, en el caso de datos no fiables, el tamaño de las unidades de datos de servicio MCS pueden ser de un tamaño inferior al de tamaño máximo de MCSPDU menos la cabecera de la PDU según determinan los parámetros del dominio.

La capa MCS regula el flujo global de datos dentro de un dominio. La inaptitud de un receptor para aceptar datos a la velocidad a que son ofrecidos repercute en la aparición de una presión hacia el

origen (o retropresión) que causa el bloqueo de los transmisores. Un usuario puede ser desanexionado involuntariamente si deja de mantener una velocidad mínima de recepción.

La capa MCS proporciona transferencia de datos completamente fiable (es decir, de entrega garantizada) y no fiable (es decir, de entrega no garantizada). Para una transferencia de datos completamente fiable, la capa MCS garantiza la recepción sin error de los datos transmitidos mientras que los usuarios de origen y de destino permanezcan anexionados y el usuario de destino permanezca incorporado al canal. Para una transferencia de datos no fiable, la capa MCS proporciona la recepción ordenada de los datos transmitidos, aunque puedan ocurrir interrupciones en el flujo de datos. Asimismo, tanto para la transferencia fiable como para la no fiable, los datos con prioridad más alta, tienen precedencia, por lo que un exceso de éstos puede demorar indefinidamente la entrega de datos con prioridad más baja.

#### 5.4.4 Gestión de testigo

La capa MCS implementa operaciones basadas en testigo en el proveedor MCS superior, con lo que se asegura la coherencia y la exclusividad.

#### 5.4.5 Notificación de capacidades

La capa MCS proporciona un mecanismo para el intercambio de información cualquiera que sean las capacidades. Por medio de este mecanismo, los nodos pueden determinar el estado del dominio en relación a una determinada capacidad. Ello permite a un nodo tomar la decisión sobre si utiliza dicha capacidad.

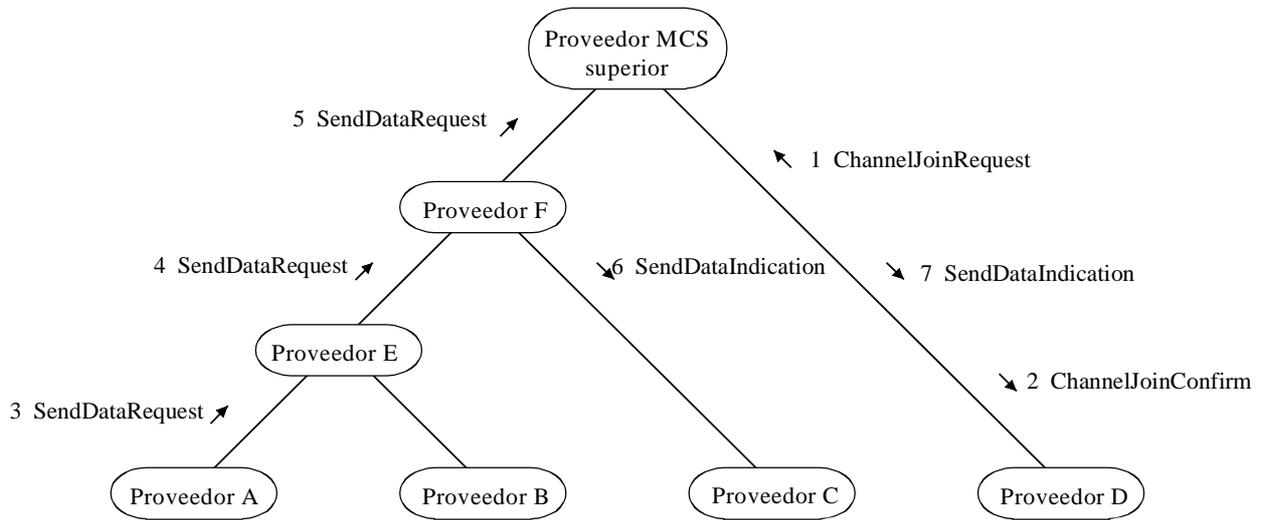
### 5.5 Procesamiento jerárquico

El procesamiento jerárquico en un dominio se ilustra en la figura 5-2.

Los nodos en la figura representan proveedores MCS y las flechas etiquetadas representan MCSPDU. Este ejemplo se refiere concretamente a un periodo de tiempo después de que el dominio se ha establecido mediante conexiones entre proveedores MCS y el uso de la transferencia de datos está comenzando a expandirse. En el paso 1 el proveedor D pide en nombre de un usuario incorporarse a un canal a través del cual se distribuirán datos, y en el paso 2 la petición es confirmada como exitosa. En el paso 3 un usuario anexionado al proveedor A envía datos y la correspondiente **SendDataRequest** comienza a fluir hacia arriba. Suponiendo que al canal por el que se están enviando los datos sólo se incorporan anexiones en A, C, y D, la MCSPDU se refleja hacia abajo en los pasos 6 y 7 como **SendDataIdentification**. El proveedor E, al ver que ningún otro subordinado necesita recibir datos, simplemente envía **SendDataRequest** hacia arriba en el paso 4. El proveedor F envía **SendDataRequest** hacia arriba en el paso 5, pero también la refleja hacia abajo en el paso 6, pues sabe que el proveedor C ha expresado su interés en el canal.

Los proveedores MCS no están especialmente preocupados de su nivel (o altura) en la jerarquía, salvo en lo tocante al papel que desempeñan en la imposición de un límite global a la altura del dominio y, en un amplio sentido, al conocimiento de si son el proveedor superior, o no lo son. El proveedor MCS superior no tiene conexión ascendente. Todos los demás tienen exactamente una.

Un proveedor MCS registra información sobre los canales y testigos utilizados en su subárbol de un dominio MCS.



T1602410-97

**Figura 5-2/T.125 – Procesamiento jerárquico en un dominio MCS**

Un proveedor registra los canales a los que se incorporan usuarios dentro del subárbol y, para cada uno de estos canales, dónde se origina la incorporación, es decir, desde qué anexiones MCS y desde qué conexiones MCS descendentes. Registra los identificadores de usuario que se asignan en el subárbol y dónde se originan. Registra canales privados que tienen un gestor o un usuario admitido en el subárbol y registra los identificadores de los usuarios asociados.

Un proveedor MCS registra los testigos que son tomados o inhibidos por usuarios dentro del subárbol, y registra los identificadores de los usuarios asociados.

Un proveedor MCS examina las peticiones que proceden de su subárbol para verificar que el identificador del usuario iniciador está legítimamente asignado a la anexión o conexión MCS descendente de origen. De esta forma se establecen anillos de protección alrededor del proveedor MCS superior y se limita la cantidad de perturbación que un participante malintencionado puede causar en un dominio que, por lo demás, es cooperativo.

En términos generales, abstracción hecha de varias excepciones, la operación de la capa MCS puede describirse como sigue:

- a) Una petición de primitiva MCS invocada en una anexión MCS genera una MCSPDU en el proveedor MCS correspondiente y la despacha en sentido ascendente hacia el proveedor MCS superior. Este proveedor, que tiene la información completa sobre el dominio MCS, actúa sobre la MCSPDU.
- b) Se puede generar una MCSPDU de confirmación en el proveedor MCS superior para retornar resultados a la anexión solicitante. Los proveedores MCS que le dan curso actualizan sus registros en función del efecto de la operación sobre el subárbol. Una confirmación se encamina al identificador del usuario iniciador mediante la consulta de registros locales en cada salto sucesivo hacia abajo.
- c) En lugar de esto se puede generar una MCSPDU de indicación para informar a otras anexiones sobre la acción ejecutada. Se puede hacer replicaciones de las indicaciones y enviarlas hacia abajo por varias conexiones que conducen a usuarios influidos. Los proveedores MCS pueden también actualizar sus registros, con el efecto de la operación como parte del procesamiento de una indicación.

La descripción precedente es un enunciado general destinado a crear un marco conceptual. Más adelante se dan detalles más completos sobre la información registrada en un proveedor MCS y la manera de procesar MCSPDU específicas.

Entre las excepciones están las siguientes: Algunas peticiones, sobre todo **ChannelJointRequest** y **ChannelLeaveRequest**, pueden dejar de surgir a un nivel a corta distancia por debajo del proveedor superior, y algunas indicaciones, sobre todo **SendDataIndication**, a un nivel inferior al del proveedor superior. Una MCSPDU, **TokenGiveResponse**, pertenece a la categoría de las respuestas. Y algunas MCSPDU pueden ser generadas por un proveedor MCS como una continuación del procesamiento de MCSPDU desiguales como en el caso de una **ChannelLeaveRequest** que sigue a una **DetachUserIndication**.

## 5.6 Parámetros de dominio

Los proveedores MCS que atienden sólo dominio MCS asignan recursos y ejecutan procedimientos de acuerdo con los siguientes parámetros. Los valores de estos parámetros son idénticos dentro de un dominio.

- a) Número máximo de canales MCS que pueden utilizarse simultáneamente. Incluye canales a que se incorporan usuarios, identificadores de usuario que han sido asignados, y canales privados que han sido creados.
- b) Número máximo de identificadores de usuario que pueden estar asignados simultáneamente. Este es un sublímite dentro de la restricción del parámetro precedente.
- c) Número máximo de los identificadores de testigo que pueden ser tomados o inhibidos simultáneamente.
- d) Número de prioridades de transferencia de datos implementadas. Es igual al número de TC en una conexión MCS. Un usuario MCS puede enviar y recibir datos aunque las prioridades estén fuera del límite. Sin embargo, en este caso esas prioridades pueden tratarse como la prioridad más baja que haya sido implementada.
- e) Caudal cumplimentado. Aunque el control de flujo global limita la transferencia de datos dentro de un dominio a la velocidad del receptor más lento, no debe permitirse que los receptores trabajen con una velocidad arbitrariamente baja. Si se permitiera esto, un participante en una conferencia podría obstaculizar a todos los demás. Este parámetro obliga a los proveedores MCS a observar una velocidad mínima de recepción en cada anexión MCS y en cada conexión MCS descendente. Los que no cumplen esta obligación corren el riesgo de ser desanexionados involuntariamente o desconectados, respectivamente.
- f) Altura máxima. Este parámetro limita la altura de todos los proveedores MCS, en particular la del proveedor superior.
- g) Tamaño máximo de las MCSPDU de dominio. El control de flujo global se basa en el almacenamiento de las MCSPDU de dominio (pero no las MCSPDU de conexión) en una memoria tampón dentro de un proveedor MCS. Por razones de sencillez, se supone que las memorias tampón tienen un tamaño fijo. Un proveedor MCS no generará MCSPDU más grandes. Esto limita la cantidad de información que puede estar contenida en una sola MCSPDU de control y sugiere los lugares donde datos de usuario ilimitados pueden segmentarse en varias MCSPDU de datos.
- h) Versión del protocolo. Está constituida por uno de tres posibles valores que definen diferentes codificaciones para las MCSPDU de dominio. En la versión tres del protocolo MCS este valor se establece en cada conexión. Ello permite que los proveedores MCS de versión dos puedan existir a niveles inferiores en una conferencia autorizada de versión tres.

NOTA – Una instancia dada de un proveedor MCS puede funcionar con limitaciones de recursos locales que también están parametrizadas. Entre ellas puede estar la cantidad de memoria tampón disponible para el almacenamiento de las MCSPDU que se encuentran en espera de ser transportadas, el número máximo de anexiones MCS, y el número máximo de conexiones MCS a otros proveedores. Estos parámetros se manejan localmente y no se comunican a través de un dominio MCS.

## **5.7 Parámetros ampliados**

Los proveedores MCS que son miembros de la cumbre V3 de un único dominio MCS asignan recursos y ejecutan los procedimientos de acuerdo con los parámetros siguientes.

- a) Se soportan datos no fiables. Si la cumbre V3 soporta datos no fiables, el valor booleano será verdadero. El valor de este parámetro es idéntico en toda la cumbre V3 de un dominio.
- b) Identificador de referencia del dominio. No es un valor negociado, pero lo genera un proveedor MCS a fin de identificar unívocamente un dominio específico. Aunque puede utilizarse el mismo valor a fin de identificar distintos dominios para distintos proveedores, debe representar inequívocamente un dominio específico para el proveedor que generó el valor.

## **6 Utilización del servicio de transporte**

### **6.1 Modelo del servicio de transporte**

Esta descripción se presenta en términos similares a los de las partes pertinentes de la Recomendación X.214, suponiendo que no se utilizan datos acelerados.

El servicio de transporte (TS) ofrece a un usuario TS las siguientes prestaciones:

- a) Medios para establecer una TC con otro usuario TS para el intercambio de TSDU. Entre dos mismos usuarios TS puede haber más de una TC.
- b) La oportunidad de pedir, negociar, y obtener por acuerdo con el proveedor una cierta calidad de servicio, asociada con cada TC en el momento de su establecimiento y especificada por parámetros que representan características tales como caudal, retardo de tránsito, tasa de error residual, y prioridad.
- c) Medio de transferir TSDU en una TC. La transferencia de TSDU, que están constituidas por un número entero de octetos, es transparente, entendiéndose por esto que los confines y el contenido de las TSDU quedan preservados sin modificación por el proveedor TS.
- d) Medio por el cual un usuario TS receptor puede controlar la velocidad a la que el usuario TS emisor puede enviar datos.
- e) La liberación incondicional, y por consiguiente posiblemente destructiva, de una TC.

La operación de una TC se modela de forma abstracta por un par de colas que enlazan dos TSAP. Hay una cola para cada sentido de flujo de la información. Cada TC se modela mediante dos colas distintas.

Este modelo basado en colas se emplea para expresar la prestación de control de flujo. La capacidad de una cola es limitada, pero no tiene necesariamente que ser ni fija ni determinable. Los objetos de conexión, TSDU, y desconexión son introducidos y retirados de una cola como resultado de interacciones en los dos TSAP. La aptitud de un usuario TS para introducir objetos en una cola viene determinada por el comportamiento del usuario TS que retira objetos de esa cola y por el estado de la cola. Los únicos objetos que pueden ser introducidos en una cola por el proveedor TS son los objetos de desconexión. Los objetos se introducen en una cola bajo el control del proveedor TS. Los objetos normalmente se retiran de la cola bajo el control del usuario TS receptor. Normalmente, los objetos

se extraen en el mismo orden en que se introdujeron. La única excepción a la extracción normal es que un objeto puede ser suprimido por el proveedor TS única y exclusivamente si el siguiente es un objeto de desconexión.

Hay que proporcionar localmente un mecanismo de identificación de punto extremo de TC si el usuario TS y el proveedor TS necesitan distinguir entre varias TC en un TSAP. En tal caso, todas las primitivas tienen que utilizar este mecanismo de identificación para identificar la TC a que se aplican. Esta identificación implícita no se muestra como un parámetro de las primitivas TS y no debe confundirse con los parámetros de dirección de T-CONEXIÓN.

## 6.2 Utilización de múltiples conexiones

Una conexión MCS está constituida por una o más TC entre una misma pareja de proveedores MCS. La primera TC establecida se llama TC inicial; las establecidas posteriormente se llaman TC adicionales. Todas las TC pertenecientes a una conexión MCS las establece el mismo proveedor MCS, como reacción a una petición MCS-CONEXIÓN-PROVEEDOR. Esta petición contiene parámetros de dirección, que son las direcciones TSAP llamante y llamada. Estas direcciones se utilizan sin modificación alguna en las peticiones T-CONEXIÓN resultantes.

El número de TC por conexión MCS es uniforme en todo un dominio MCS. Este parámetro de dominio es igual al número de niveles de prioridad de transferencia de datos implementados. Se requieren TC separadas porque cada una de ellas es un vehículo para el control de flujo. Los bloqueos en los datos de prioridades más bajas no deben causar retrocesiones en perjuicio de datos de más alta prioridad. Para una implementación completa, los datos de prioridades más bajas y los de prioridades más altas deben ser vehiculados por TC diferentes.

La calidad de servicio solicitada para una TC puede variar según la prioridad de datos para la cual se estableció. Los objetivos de calidad de servicio no tienen que ser uniformes en la totalidad de un dominio MCS.

Entre los aspectos de la calidad de servicio que ofrecen interés están el caudal y el retardo de tránsito máximos y medios. Es posible que los datos de alta prioridad requieran un bajo retardo de tránsito para obtener una respuesta en tiempo real, pero que no requieran un caudal elevado. En cambio, es posible que los datos de baja prioridad requieran un caudal elevado, pero que las transferencias masivas no requieran un bajo retardo de tránsito.

La prioridad de la TC es otro aspecto de la calidad de servicio, aunque no se ajuste precisamente al concepto de la prioridad de transferencia de datos MCS. Dicha prioridad especifica el orden relativo en que, en caso necesario, las TC deben ver su calidad de servicio degradada. Una alta prioridad de la TC puede solicitarse junto con otras características, como un bajo retardo de tránsito, para asegurar que los datos prioritarios MCS reciban el tratamiento preferencial que merecen.

Las MCS-SPDU de conexión sólo aparecen como la primera TSDU transportada en uno u otro sentido de transmisión de una TC. **Conexión-inicial** y **conexión-respuesta** atraviesan la TC inicial de una conexión MCS. **Conexión-adicional** y **conexión-resultado** atraviesan TC adicionales, si existen.

Un proveedor MCS llamante, al emitir peticiones T-CONEXIÓN, controla mediante sus propias acciones qué TC forman parte de la misma conexión MCS y qué prioridades de transferencia de datos representan.

Un proveedor MCS llamado, al recibir indicaciones T-CONEXIÓN, tiene por lo general que aceptar una TC y leer su primera TSDU para saber lo que significa. **Conexión-inicial** identifica una TC entrante como el comienzo de una nueva conexión MCS. Una **conexión-adicional** identifica una TC entrante como parte de una conexión MCS en curso.

**Conexión-adicional** contiene un valor asignado por el proveedor MCS llamado y transportado al proveedor MCS llamante en una **conexión-respuesta** a través de la TC inicial que designa la TC adicional como perteneciente a la misma conexión MCS. **Conexión-adicional** da también explícitamente la prioridad de datos que la TC representa.

Las MCSPDU de conexión se intercambian inmediatamente después del establecimiento de la TC. Una conexión MCS, una vez confirmada, pasa a formar parte de un dominio MCS jerárquico. Después de esto, la conexión MCS transporta MCSPDU de dominio.

Con excepción de las MCSPDU de datos, las MCSPDU de dominio atraviesan la TC inicial de una conexión MCS. Las MCSPDU de datos atraviesan la TC que corresponde a su prioridad de datos. Si la prioridad especificada sobrepasa el número aplicado en un dominio MCS, una MCSPDU de datos atraviesa la TC de la más baja prioridad que ha sido aplicada.

Si sólo se ha aplicado una prioridad en un dominio MCS, cada una de sus conexiones MCS está constituida por una sola TC, no se utiliza **conexión-adicional** ni **conexión-resultado**, y todas las MCSPDU se transmiten en secuencia entre los proveedores.

### 6.3 Liberación de la conexión de transporte

Un proveedor TS da una mayor fiabilidad a las conexiones de extremo a extremo ejecutando protocolos en grado suficiente para compensar cualquier punto débil en servicios de red subyacentes. Un proveedor MCS no duplica esta funcionalidad. No intenta una ulterior recuperación automática en el caso de un fallo de transporte.

Los errores que no permiten una recuperación (errores "irrecuperables") se anuncian mediante una indicación T-DESCONEXIÓN. Si se desconecta cualquiera de las TC pertenecientes a un MCS, las otras se desconectan también inmediatamente. A menos que esto haya sido solicitado por el usuario, se genera una indicación MCS-DESCONEXIÓN-PROVEEDOR, y se da como motivo iniciada por el proveedor.

Por otra parte, una petición MCS-DESCONEXIÓN-PROVEEDOR debe aparecer como una indicación en el otro lado dándose como motivo solicitada por el usuario. Pese a las seguridades que se dan en la Recomendación X.214, la clase más simple de protocolo de transporte no permite pasar datos de usuario en T-DESCONEXIÓN. En consecuencia, el código de motivo de la desconexión se transfiere en una MCSPDU explícita. Al recibir esta MCSPDU, un proveedor MCS queda obligado a desconectar la conexión MCS que la transportó.

## 7 Estructura de las MCSPDU de versión 2

La estructura de las MCSPDU de versión 2 se especifica mediante la notación ASN.1 de la Recomendación X.208. La utilización y el significado de estas MCSPDU se describen más adelante en las cláusulas 9 y 10.

**MCS-PROTOCOL DEFINITIONS ::=**

**BEGIN**

*-- Part 1: Fundamental MCS types*

**ChannelId ::= INTEGER (0..65535)** *-- range is 16 bits*

**StaticChannelId ::= ChannelId (1..1000)** *-- those known permanently*

**DynamicChannelId ::= ChannelId (1001..65535)** *-- those created and deleted*

```

UserId ::= DynamicChannelId                -- created by Attach-User
                                           -- deleted by Detach-User

PrivateChannelId ::= DynamicChannelId      -- created by Channel-Convenc
                                           -- deleted by Channel-Disband

AssignedChannelId ::= DynamicChannelId     -- created by Channel-Join zero
                                           -- deleted by last Channel-Leave

TokenId ::= INTEGER (1..65535)           -- all are known permanently

TokenStatus ::= ENUMERATED
{
    notInUse                (0),
    selfGrabbed            (1),
    otherGrabbed          (2),
    selfInhibited        (3),
    otherInhibited      (4),
    selfRecipient       (5),
    selfGiving          (6),
    otherGiving         (7)
}

DataPriority ::= ENUMERATED
{
    top                    (0),
    high                  (1),
    medium                (2),
    low                   (3)
}

Segmentation ::= BIT STRING
{
    begin                 (0),
    end                   (1)
} (SIZE (2))

DomainParameters ::= SEQUENCE
{
    maxChannelIds        INTEGER (0..MAX),
                                           -- a limit on channel ids in use,
                                           -- static + user id + private + assigned

    maxUserIds          INTEGER (0..MAX),
                                           -- a sublimit on user id channels alone

    maxTokenIds         INTEGER (0..MAX),
                                           -- a limit on token ids in use
                                           -- grabbed + inhibited + giving + ungivable + given

    numPriorities       INTEGER (0..MAX),
                                           -- the number of TCs in an MCS connection

    minThroughput       INTEGER (0..MAX),
                                           -- the enforced number of octets per second

    maxHeight           INTEGER (0..MAX),
                                           -- a limit on the height of a provider

    maxMCSPDUsize       INTEGER (0..MAX),
                                           -- an octet limit on domain MCSPDUs

    protocolVersion     INTEGER (0..MAX)
}

```

-- Part 2: Connect provider

**Connect-Initial ::= [APPLICATION 101] IMPLICIT SEQUENCE**

```
{
  callingDomainSelector      OCTET STRING,
  calledDomainSelector       OCTET STRING,
  upwardFlag                 BOOLEAN,
                             -- TRUE if called provider is higher
  targetParameters           DomainParameters,
  minimumParameters          DomainParameters,
  maximumParameters          DomainParameters,
  userData                   OCTET STRING
}
```

**Connect-Response ::= [APPLICATION 102] IMPLICIT SEQUENCE**

```
{
  result                     Result,
  calledConnectId            INTEGER (0..MAX),
                             -- assigned by the called provider
                             -- to identify additional TCs of
                             -- the same MCS connection
  domainParameters          DomainParameters,
  userData                   OCTET STRING
}
```

**Connect-Additional ::= [APPLICATION 103] IMPLICIT SEQUENCE**

```
{
  calledConnectId            INTEGER (0..MAX),
  dataPriority                DataPriority
}
```

**Connect-Result ::= [APPLICATION 104] IMPLICIT SEQUENCE**

```
{
  result                     Result
}
```

-- Part 3: Merge domain

**PlumbDomainIndication ::= [APPLICATION 0] IMPLICIT SEQUENCE**

```
{
  heightLimit                INTEGER (0..MAX)
                             -- a restriction on the MCSPDU receiver
}
```

**ErectDomainRequest ::= [APPLICATION 1] IMPLICIT SEQUENCE**

```
{
  subHeight                  INTEGER (0..MAX),
                             -- height in domain of the MCSPDU transmitter
  subInterval                 INTEGER (0..MAX)
                             -- its throughput enforcement interval in milliseconds
}
```

**ChannelAttributes ::= CHOICE**

```
{
  static                      [0] IMPLICIT SEQUENCE
  {
    channelId                  StaticChannelId
                             -- joined is implicitly TRUE
  },
}
```

```

userId                                [1] IMPLICIT SEQUENCE
{
    joined                                BOOLEAN,
                                           -- TRUE if user is joined to its user id
    userId                                UserId
},

private                                [2] IMPLICIT SEQUENCE
{
    joined                                BOOLEAN,
                                           -- TRUE if channel id is joined below
    channelId                            PrivateChannelId,
    manager                               UserId,
    admitted                             SET OF UserId
                                           -- may span multiple MergeChannelsRequest
},

assigned                               [3] IMPLICIT SEQUENCE
{
    channelId                            AssignedChannelId
                                           -- joined is implicitly TRUE
}
}

MergeChannelsRequest ::= [APPLICATION 2] IMPLICIT SEQUENCE
{
    mergeChannels                        SET OF ChannelAttributes,
    purgeChannelIds                     SET OF ChannelId
}

MergeChannelsConfirm ::= [APPLICATION 3] IMPLICIT SEQUENCE
{
    mergeChannels                        SET OF ChannelAttributes,
    purgeChannelIds                     SET OF ChannelId
}

PurgeChannelsIndication ::= [APPLICATION 4] IMPLICIT SEQUENCE
{
    detachUserIds                        SET OF UserId,
                                           -- purge user id channels
    purgeChannelIds                     SET OF ChannelId
                                           -- purge other channels
}

TokenAttributes ::= CHOICE
{
    grabbed                              [0] IMPLICIT SEQUENCE
    {
        tokenId                          TokenId,
        grabber                           UserId
    },
    inhibited                            [1] IMPLICIT SEQUENCE
    {
        tokenId                          TokenId,
        inhibitors                       SET OF UserId
                                           -- may span multiple MergeTokensRequest
    },
    giving                               [2] IMPLICIT SEQUENCE
    {
        tokenId                          TokenId,

```

```

        grabber          UserId,
        recipient        UserId
    },

    ungivable            [3] IMPLICIT SEQUENCE
    {
        tokenId          TokenId,
        grabber          UserId
                        -- recipient has since detached
    },

    given                [4] IMPLICIT SEQUENCE
    {
        tokenId          TokenId,
        recipient        UserId
                        -- grabber released or detached
    }
}

```

**MergeTokensRequest ::= [APPLICATION 5] IMPLICIT SEQUENCE**

```

{
    mergeTokens          SET OF TokenAttributes,
    purgeTokenIds        SET OF TokenId
}

```

**MergeTokensConfirm ::= [APPLICATION 6] IMPLICIT SEQUENCE**

```

{
    mergeTokens          SET OF TokenAttributes,
    purgeTokenIds        SET OF TokenId
}

```

**PurgeTokensIndication ::= [APPLICATION 7] IMPLICIT SEQUENCE**

```

{
    purgeTokenIds        SET OF TokenId
}
-- Part 4: Disconnect provider

```

**DisconnectProviderUltimatum ::= [APPLICATION 8] IMPLICIT SEQUENCE**

```

{
    reason                Reason
}

```

**RejectMCSPDUUltimatum ::= [APPLICATION 9] IMPLICIT SEQUENCE**

```

{
    diagnostic            Diagnostic,
    initialOctets         OCTET STRING
}

```

*-- Part 5: Attach/Detach user*

**AttachUserRequest ::= [APPLICATION 10] IMPLICIT SEQUENCE**

```

{
}

```

**AttachUserConfirm ::= [APPLICATION 11] IMPLICIT SEQUENCE**

```

{
    result                Result,
    initiator             UserId OPTIONAL
}

```

**DetachUserRequest ::= [APPLICATION 12] IMPLICIT SEQUENCE**

```
{  
    reason                Reason,  
    userIds                SET OF UserId  
}
```

**DetachUserIndication ::= [APPLICATION 13] IMPLICIT SEQUENCE**

```
{  
    reason                Reason,  
    userIds                SET OF UserId  
}
```

*-- Part 6: Channel management*

**ChannelJoinRequest ::= [APPLICATION 14] IMPLICIT SEQUENCE**

```
{  
    initiator              UserId,  
    channelId              ChannelId  
                            -- may be zero  
}
```

**ChannelJoinConfirm ::= [APPLICATION 15] IMPLICIT SEQUENCE**

```
{  
    result                 Result,  
    initiator              UserId,  
    requested              ChannelId,  
                            -- may be zero  
    channelId              ChannelId OPTIONAL  
}
```

**ChannelLeaveRequest ::= [APPLICATION 16] IMPLICIT SEQUENCE**

```
{  
    channelIds             SET OF ChannelId  
}
```

**ChannelConveneRequest ::= [APPLICATION 17] IMPLICIT SEQUENCE**

```
{  
    initiator              UserId  
}
```

**ChannelConveneConfirm ::= [APPLICATION 18] IMPLICIT SEQUENCE**

```
{  
    result                 Result,  
    initiator              UserId,  
    channelId              PrivateChannelId OPTIONAL  
}
```

**ChannelDisbandRequest ::= [APPLICATION 19] IMPLICIT SEQUENCE**

```
{  
    initiator              UserId,  
    channelId              PrivateChannelId  
}
```

**ChannelDisbandIndication ::= [APPLICATION 20] IMPLICIT SEQUENCE**

```
{  
    channelId              PrivateChannelId  
}
```

**ChannelAdmitRequest ::= [APPLICATION 21] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           PrivateChannelId,  
    userIds             SET OF UserId  
}
```

**ChannelAdmitIndication ::= [APPLICATION 22] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           PrivateChannelId,  
    userIds             SET OF UserId  
}
```

**ChannelExpelRequest ::= [APPLICATION 23] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           PrivateChannelId,  
    userIds             SET OF UserId  
}
```

**ChannelExpelIndication ::= [APPLICATION 24] IMPLICIT SEQUENCE**

```
{  
    channelId           PrivateChannelId,  
    userIds             SET OF UserId  
}
```

-- Part 7: Data transfer

**SendDataRequest ::= [APPLICATION 25] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           ChannelId,  
    dataPriority         DataPriority,  
    segmentation        Segmentation,  
    userData             OCTET STRING  
}
```

**SendDataIndication ::= [APPLICATION 26] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           ChannelId,  
    dataPriority         DataPriority,  
    segmentation        Segmentation,  
    userData             OCTET STRING  
}
```

**UniformSendDataRequest ::= [APPLICATION 27] IMPLICIT SEQUENCE**

```
{  
    initiator           UserId,  
    channelId           ChannelId,  
    dataPriority         DataPriority,  
    segmentation        Segmentation,  
    userData             OCTET STRING  
}
```

**UniformSendDataIndication ::= [APPLICATION 28] IMPLICIT SEQUENCE**

```
{
    initiator          UserId,
    channelId          ChannelId,
    dataPriority       DataPriority,
    segmentation      Segmentation,
    userData          OCTET STRING
}
```

-- Part 8: Token management

**TokenGrabRequest ::= [APPLICATION 29] IMPLICIT SEQUENCE**

```
{
    initiator          UserId,
    tokenId            TokenId
}
```

**TokenGrabConfirm ::= [APPLICATION 30] IMPLICIT SEQUENCE**

```
{
    result Result,
    initiator          UserId,
    tokenId            TokenId,
    tokenStatus       TokenStatus
}
```

**TokenInhibitRequest ::= [APPLICATION 31] IMPLICIT SEQUENCE**

```
{
    initiator          UserId,
    tokenId            TokenId
}
```

**TokenInhibitConfirm ::= [APPLICATION 32] IMPLICIT SEQUENCE**

```
{
    result            Result,
    initiator          UserId,
    tokenId            TokenId,
    tokenStatus       TokenStatus
}
```

**TokenGiveRequest ::= [APPLICATION 33] IMPLICIT SEQUENCE**

```
{
    initiator          UserId,
    tokenId            TokenId,
    recipient          UserId
}
```

**TokenGiveIndication ::= [APPLICATION 34] IMPLICIT SEQUENCE**

```
{
    initiator          UserId,
    tokenId            TokenId,
    recipient          UserId
}
```

**TokenGiveResponse ::= [APPLICATION 35] IMPLICIT SEQUENCE**

```
{
    result            Result,
    recipient          UserId,
    tokenId            TokenId
}
```

**TokenGiveConfirm ::= [APPLICATION 36] IMPLICIT SEQUENCE**

```
{
    result          Result,
    initiator       UserId,
    tokenId         TokenId,
    tokenStatus    TokenStatus
}
```

**TokenPleaseRequest ::= [APPLICATION 37] IMPLICIT SEQUENCE**

```
{
    initiator       UserId,
    tokenId         TokenId
}
```

**TokenPleaseIndication ::= [APPLICATION 38] IMPLICIT SEQUENCE**

```
{
    initiator       UserId,
    tokenId         TokenId
}
```

**TokenReleaseRequest ::= [APPLICATION 39] IMPLICIT SEQUENCE**

```
{
    initiator       UserId,
    tokenId         TokenId
}
```

**TokenReleaseConfirm ::= [APPLICATION 40] IMPLICIT SEQUENCE**

```
{
    result Result,
    initiator       UserId,
    tokenId         TokenId,
    tokenStatus    TokenStatus
}
```

**TokenTestRequest ::= [APPLICATION 41] IMPLICIT SEQUENCE**

```
{
    initiator       UserId,
    tokenId         TokenId
}
```

**TokenTestConfirm ::= [APPLICATION 42] IMPLICIT SEQUENCE**

```
{
    initiator       UserId,
    tokenId         TokenId,
    tokenStatus    TokenStatus
}
```

-- Part 9: Status codes

**Reason ::= ENUMERATED** -- *in DisconnectProviderUltimatum, DetachUserRequest, DetachUserIndication*

```
{
    rn-domain-disconnected (0),
    rn-provider-initiated  (1),
    rn-token-purged        (2),
    rn-user-requested      (3),
    rn-channel-purged      (4)
}
```

**Result ::= ENUMERATED** -- *in Connect, response, confirm*

```
{  
    rt-successful                (0),  
    rt-domain-merging            (1),  
    rt-domain-not-hierarchical   (2),  
    rt-no-such-channel           (3),  
    rt-no-such-domain            (4),  
    rt-no-such-user              (5),  
    rt-not-admitted              (6),  
    rt-other-user-id             (7),  
    rt-parameters-unacceptable  (8),  
    rt-token-not-available       (9),  
    rt-token-not-possessed       (10),  
    rt-too-many-channels         (11),  
    rt-too-many-tokens           (12),  
    rt-too-many-users            (13),  
    rt-unspecified-failure       (14),  
    rt-user-rejected             (15)  
}
```

**Diagnostic ::= ENUMERATED** -- *in RejectMCSPDUUltimatum*

```
{  
    dc-inconsistent-merge        (0),  
    dc-forbidden-PDU-downward    (1),  
    dc-forbidden-PDU-upward      (2),  
    dc-invalid-BER-encoding       (3),  
    dc-invalid-PER-encoding       (4),  
    dc-misrouted-user            (5),  
    dc-unrequested-confirm        (6),  
    dc-wrong-transport-priority   (7),  
    dc-channel-id-conflict        (8),  
    dc-token-id-conflict          (9),  
    dc-not-user-id-channel        (10),  
    dc-too-many-channels          (11),  
    dc-too-many-tokens            (12),  
    dc-too-many-users             (13)  
}
```

-- *Part 10: MCSPDU repertoire*

**ConnectMCSPDU ::= CHOICE**

```
{  
    connect-initial              Connect-Initial,  
    connect-response             Connect-Response,  
    connect-additional           Connect-Additional,  
    connect-result               Connect-Result  
}
```

**DomainMCSPDU ::= CHOICE**

```
{  
    plumbDomainIndication        PlumbDomainIndication,  
    erectDomainRequest           ErectDomainRequest,  
    mergeChannelsRequest         MergeChannelsRequest,  
    mergeChannelsConfirm        MergeChannelsConfirm,  
    purgeChannelsIndication      PurgeChannelsIndication,  
    mergeTokensRequest          MergeTokensRequest,  
    mergeTokensConfirm          MergeTokensConfirm,  
    purgeTokensIndication        PurgeTokensIndication,  
    disconnectProviderUltimatum  DisconnectProviderUltimatum,  
    rejectMCSPDUUltimatum       RejectMCSPDUUltimatum,  
    attachUserRequest            AttachUserRequest,  
}
```

attachUserConfirm	AttachUserConfirm,
detachUserRequest	DetachUserRequest,
detachUserIndication	DetachUserIndication,
channelJoinRequest	ChannelJoinRequest,
channelJoinConfirm	ChannelJoinConfirm,
channelLeaveRequest	ChannelLeaveRequest,
channelConveneRequest	ChannelConveneRequest,
channelConveneConfirm	ChannelConveneConfirm,
channelDisbandRequest	ChannelDisbandRequest,
channelDisbandIndication	ChannelDisbandIndication,
channelAdmitRequest	ChannelAdmitRequest,
channelAdmitIndication	ChannelAdmitIndication,
channelExpelRequest	ChannelExpelRequest,
channelExpelIndication	ChannelExpelIndication,
sendDataRequest	SendDataRequest,
sendDataIndication	SendDataIndication,
uniformSendDataRequest	UniformSendDataRequest,
uniformSendDataIndication	UniformSendDataIndication,
tokenGrabRequest	TokenGrabRequest,
tokenGrabConfirm	TokenGrabConfirm,
tokenInhibitRequest	TokenInhibitRequest,
tokenInhibitConfirm	TokenInhibitConfirm,
tokenGiveRequest	TokenGiveRequest,
tokenGiveIndication	TokenGiveIndication,
tokenGiveResponse	TokenGiveResponse,
tokenGiveConfirm	TokenGiveConfirm,
tokenPleaseRequest	TokenPleaseRequest,
tokenPleaseIndication	TokenPleaseIndication,
tokenReleaseRequest	TokenReleaseRequest,
tokenReleaseConfirm	TokenReleaseConfirm,
tokenTestRequest	TokenTestRequest,
tokenTestConfirm	TokenTestConfirm

}

END

## 8 Estructura de las MCSPDUs de versión 3

La estructura de las MCSPDU de versión 3 se especifica utilizando la notación ASN.1 de la Recomendación X.208. La utilización y el significado de estas MCSPDU se describe ulteriormente en las cláusulas 9 y 10.

**MCS-PROTOCOL-3 DEFINITIONS AUTOMATIC TAGS::=**

**BEGIN**

*-- Part 1: Fundamental MCS types*

**H221NonStandardIdentifier ::= OCTET STRING (SIZE (4..255))**

*-- First four octets shall be country  
-- code and Manufacturer code, assigned  
-- as specified in Annex A/H.221 for  
-- NS-cap and NS-comm*

**Key ::= CHOICE**

*-- Identifier of a standard or non-standard object*

```
{
  object          OBJECT IDENTIFIER,
  h221NonStandard H221NonStandardIdentifier
}
```

```

NonStandardParameter ::= SEQUENCE
{
    key          Key,
    data         OCTET STRING
}

ChannelId ::= INTEGER (0..65535)           -- range is 16 bits

StaticChannelId ::= ChannelId (1..1000)    -- those known permanently

DynamicChannelId ::= ChannelId (1001..65535) -- those created and deleted

UserId ::= DynamicChannelId               -- created by Attach-User
                                              -- deleted by Detach-User

PrivateChannelId ::= DynamicChannelId     -- created by Channel-Convene
                                              -- deleted by Channel-Disband

AssignedChannelId ::= DynamicChannelId    -- created by Channel-Join zero
                                              -- deleted by last Channel-Leave

TokenId ::= INTEGER (1..65535)           -- all are known permanently

TokenStatus ::= CHOICE
{
    notInUse          NULL,
    selfGrabbed       NULL,
    otherGrabbed      NULL,
    selfInhibited     NULL,
    otherInhibited    NULL,
    selfRecipient     NULL,
    selfGiving        NULL,
    otherGiving       NULL,
    ...
}

DataPriority ::= CHOICE
{
    top          NULL,
    high         NULL,
    medium       NULL,
    low          NULL,
    ...
}

Segmentation ::= BIT STRING
{
    begin          (0),
    end            (1)
} (SIZE (2))

-- Part 2: Extended parameter

ExtendedParameters ::= SEQUENCE
{
    unreliableDataSupported    BOOLEAN,
    domainReferenceID         INTEGER (0 .. 65535),
    nonStandard                SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

ExtendedParameterPropose ::= SEQUENCE
{
    targetExtendedParameters      ExtendedParameters,
    minimumExtendedParameters     ExtendedParameters,
    maximumExtendedParameters     ExtendedParameters,
    nonStandard                   SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

ExtendedParameterAccept ::= SEQUENCE
{
    extendedParameters           ExtendedParameters,
    nonStandard                   SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

-- Part 3: Merge domain

```

PlumbDomainIndication ::= SEQUENCE
{
    heightLimit                   INTEGER (0..MAX),
                                   -- a restriction on the MCSPDU receiver
    nonStandard                   SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

ErectDomainRequest ::= SEQUENCE
{
    subHeight                     INTEGER (0..MAX),
                                   -- height in domain of the MCSPDU transmitter
    subInterval                   INTEGER (0..MAX),
                                   -- its throughput enforcement interval in milliseconds
    nonStandard                   SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

ChannelAttributes ::= CHOICE
{
    static                        SEQUENCE
    {
        channelId                 StaticChannelId,
                                   -- joined is implicitly TRUE
        nonStandard               SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },

    userId                        SEQUENCE
    {
        joined                     BOOLEAN,
                                   -- TRUE if user is joined to its user id
        userId                     UserId,
        nonStandard               SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },

    private                       SEQUENCE
    {
        joined                     BOOLEAN,
                                   -- TRUE if channel id is joined below
        channelId                 PrivateChannelId,
        manager                    UserId,
    }
}

```

```

        admitted          SET OF UserId,
                           -- may span multiple MergeChannelsRequest
        nonStandard        SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },
    assigned                SEQUENCE
    {
        channelId          AssignedChannelId,
                           -- joined is implicitly TRUE
        nonStandard        SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },
    ...
}

MergeChannelsRequest ::= SEQUENCE
{
    mergeChannels          SET OF ChannelAttributes,
    purgeChannelIds        SET OF ChannelId,
    nonStandard            SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

MergeChannelsConfirm ::= SEQUENCE
{
    mergeChannels          SET OF ChannelAttributes,
    purgeChannelIds        SET OF ChannelId,
    nonStandard            SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

PurgeChannelsIndication ::= SEQUENCE
{
    detachChannelIds       SET OF ChannelId,
                           -- purge user id channels
    purgeChannelIds        SET OF ChannelId,
                           -- purge other channels
    nonStandard            SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

TokenAttributes ::= CHOICE
{
    grabbed                SEQUENCE
    {
        tokenId            TokenId,
        grabber            UserId,
        nonStandard        SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },
    inhibited              SEQUENCE
    {
        tokenId            TokenId,
        inhibitors         SET OF UserId,
                           -- may span multiple MergeTokensRequest
        nonStandard        SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },
}

```

```

giving                               SEQUENCE
{
    tokenId                           TokenId,
    grabber                           UserId,
    recipient                          UserId,
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
},

ungivable                             SEQUENCE
{
    tokenId                           TokenId,
    grabber                           UserId,
                                     -- recipient has since detached
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
},

given                                 SEQUENCE
{
    tokenId                           TokenId,
    recipient                          UserId,
                                     -- grabber released or detached
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
},
...
}

```

**MergeTokensRequest ::= SEQUENCE**

```

{
    mergeTokens                       SET OF TokenAttributes,
    purgeTokenIds                     SET OF TokenId,
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**MergeTokensConfirm ::= SEQUENCE**

```

{
    mergeTokens                       SET OF TokenAttributes,
    purgeTokenIds                     SET OF TokenId,
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**PurgeTokensIndication ::= SEQUENCE**

```

{
    purgeTokenIds                     SET OF TokenId,
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

*-- Part 4: Disconnect provider*

**DisconnectProviderUltimatum ::= SEQUENCE**

```

{
    reason                            Reason,
    nonStandard                        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

RejectMCSPDUUltimatum ::= SEQUENCE
{
    diagnostic          Diagnostic,
    initialOctets       OCTET STRING,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

*-- Part 5: Attach/Detach user*

```

AttachUserRequest ::= SEQUENCE
{
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

AttachUserConfirm ::= SEQUENCE
{
    result              Result,
    initiator           UserId OPTIONAL,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

DetachUserRequest ::= SEQUENCE
{
    reason              Reason,
    userIds             SET OF UserId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

DetachUserIndication ::= SEQUENCE
{
    reason              Reason,
    userIds             SET OF UserId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

*-- Part 6: Channel management*

```

ChannelJoinRequest ::= SEQUENCE
{
    initiator           UserId,
    channelId           ChannelId,          -- may be zero
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

ChannelJoinConfirm ::= SEQUENCE
{
    result              Result,
    initiator           UserId,
    requested           ChannelId,          -- may be zero
    channelId           ChannelId OPTIONAL,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**ChannelLeaveRequest ::= SEQUENCE**  
 {  
   **channelIds**                  **SET OF ChannelId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelConveneRequest ::= SEQUENCE**  
 {  
   **initiator**                  **UserId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelConveneConfirm ::= SEQUENCE**  
 {  
   **result**                      **Result,**  
   **initiator**                  **UserId,**  
   **channelId**                  **PrivateChannelId OPTIONAL,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelDisbandRequest ::= SEQUENCE**  
 {  
   **initiator**                  **UserId,**  
   **channelId**                  **PrivateChannelId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelDisbandIndication ::= SEQUENCE**  
 {  
   **channelId**                  **PrivateChannelId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelAdmitRequest ::= SEQUENCE**  
 {  
   **initiator**                  **UserId,**  
   **channelId**                  **PrivateChannelId,**  
   **userIds**                    **SET OF UserId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelAdmitIndication ::= SEQUENCE**  
 {  
   **initiator**                  **UserId,**  
   **channelId**                  **PrivateChannelId,**  
   **userIds**                    **SET OF UserId,**  
   **nonStandard**              **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**ChannelExpelRequest ::= SEQUENCE**  
 {  
   **initiator**                  **UserId,**  
   **channelId**                  **PrivateChannelId,**

```

        userIds                SET OF UserId,
        nonStandard            SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    }

```

**ChannelExpelIndication ::= SEQUENCE**

```

{
    channelId                PrivateChannelId,
    userIds                  SET OF UserId,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

-- Part 7: Data transfer

**SendDataRequest ::= SEQUENCE**

```

{
    initiator                UserId,
    channelId                ChannelId,
    reliability               BOOLEAN,
    domainReferenceID        INTEGER (0 .. 65535)    OPTIONAL,
    dataPriority              DataPriority,
    segmentation             Segmentation,
    userData                  OCTET STRING,
    totalDataSize            INTEGER                OPTIONAL,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**SendDataIndication ::= SEQUENCE**

```

{
    initiator                UserId,
    channelId                ChannelId,
    reliability               BOOLEAN,
    domainReferenceID        INTEGER (0 .. 65535)    OPTIONAL,
    dataPriority              DataPriority,
    segmentation             Segmentation,
    userData                  OCTET STRING,
    totalDataSize            INTEGER                OPTIONAL,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**UniformSendDataRequest ::= SEQUENCE**

```

{
    initiator                UserId,
    channelId                ChannelId,
    reliability               BOOLEAN,
    domainReferenceID        INTEGER (0 .. 65535)    OPTIONAL,
    dataPriority              DataPriority,
    segmentation             Segmentation,
    userData                  OCTET STRING,
    totalDataSize            INTEGER                OPTIONAL,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

UniformSendDataIndication ::= SEQUENCE
{
    initiator                UserId,
    channelId                ChannelId,
    reliability              BOOLEAN,
    domainReferenceID       INTEGER (0 .. 65535)    OPTIONAL,
    dataPriority             DataPriority,
    segmentation            Segmentation,
    userData                OCTET STRING,
    totalDataSize           INTEGER                OPTIONAL,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

-- Part 8: Token management

```

TokenGrabRequest ::= SEQUENCE
{
    initiator                UserId,
    tokenId                 TokenId,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenGrabConfirm ::= SEQUENCE
{
    result                  Result,
    initiator               UserId,
    tokenId                 TokenId,
    tokenStatus             TokenStatus,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenInhibitRequest ::= SEQUENCE
{
    initiator                UserId,
    tokenId                 TokenId,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenInhibitConfirm ::= SEQUENCE
{
    result                  Result,
    initiator               UserId,
    tokenId                 TokenId,
    tokenStatus             TokenStatus,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenGiveRequest ::= SEQUENCE
{
    initiator                UserId,
    tokenId                 TokenId,
    recipient               UserId,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenGiveIndication ::= SEQUENCE
{
    initiator           UserId,
    tokenId             TokenId,
    recipient           UserId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenGiveResponse ::= SEQUENCE
{
    result              Result,
    recipient           UserId,
    tokenId             TokenId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenGiveConfirm ::= SEQUENCE
{
    result              Result,
    initiator           UserId,
    tokenId             TokenId,
    tokenStatus        TokenStatus,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenPleaseRequest ::= SEQUENCE
{
    initiator           UserId,
    tokenId             TokenId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenPleaseIndication ::= SEQUENCE
{
    initiator           UserId,
    tokenId             TokenId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenReleaseRequest ::= SEQUENCE
{
    initiator           UserId,
    tokenId             TokenId,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenReleaseConfirm ::= SEQUENCE
{
    result              Result,
    initiator           UserId,
    tokenId             TokenId,
    tokenStatus        TokenStatus,
    nonStandard         SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TokenTestRequest ::= SEQUENCE
{
    initiator                UserId,
    tokenId                  TokenId,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

TokenTestConfirm ::= SEQUENCE
{
    initiator                UserId,
    tokenId                  TokenId,
    tokenStatus              TokenStatus,
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}
-- Part 9: Capabilities notification

CapabilityID ::= CHOICE
{
    standardID              INTEGER (0 .. 65535),
    nonstandardID           Key
}

CapabilityClass ::= CHOICE
{
    null                    NULL,
    unsignedMin             INTEGER (0 .. MAX),
    unsignedMax             INTEGER (0 .. MAX)
}

ParticipationIndicator ::= CHOICE
{
    global                  NULL,
    partial                 INTEGER (1 .. 2)
}

RequestCapability ::= SEQUENCE
{
    capabilityID            CapabilityID,
    capabilityClass         CapabilityClass,
    participationIndicator  ParticipationIndicator,
    nonStandard             SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

SeqOfRequestCapabilities ::= SEQUENCE OF RequestCapability

IndicationCapability ::= SEQUENCE
{
    capabilityID            CapabilityID,
    capabilityClass         CapabilityClass,
    summitProviderSupported BOOLEAN,
    intermediateNodeSupported BOOLEAN,
    nonStandard             SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**SeqOfIndicationCapabilities ::= SEQUENCE OF IndicationCapability**

**CapabilitiesNotificationRequest ::= SEQUENCE**

```
{
    v2NodePresent          BOOLEAN,
    addList                 SeqOfRequestCapabilities    OPTIONAL,
    removeList             SeqOfRequestCapabilities    OPTIONAL,
    nonStandard            SEQUENCE OF NonStandardParameter  OPTIONAL,
    ...
}
```

**CapabilitiesNotificationIndication ::= SEQUENCE**

```
{
    v2NodePresent          BOOLEAN,
    addList                 SeqOfIndicationCapabilities  OPTIONAL,
    removeList             SeqOfIndicationCapabilities  OPTIONAL,
    nonStandard            SEQUENCE OF NonStandardParameter  OPTIONAL,
    ...
}
```

*-- Part 10: Status codes*

**Reason ::= CHOICE** *-- in DisconnectProviderUltimatum, DetachUserRequest, DetachUserIndication*

```
{
    rn-domain-disconnected    NULL,
    rn-provider-initiated     NULL,
    rn-token-purged           NULL,
    rn-user-requested         NULL,
    rn-channel-purged         NULL,
    ...
}
```

**Result ::= CHOICE** *-- in Connect, response, confirm*

```
{
    rt-successful             NULL,
    rt-domain-merging         NULL,
    rt-domain-not-hierarchical NULL,
    rt-no-such-channel        NULL,
    rt-no-such-domain         NULL,
    rt-no-such-user           NULL,
    rt-not-admitted           NULL,
    rt-other-user-id          NULL,
    rt-parameters-unacceptable NULL,
    rt-token-not-available     NULL,
    rt-token-not-possessed    NULL,
    rt-too-many-channels      NULL,
    rt-too-many-tokens        NULL,
    rt-too-many-users         NULL,
    rt-unspecified-failure    NULL,
    rt-user-rejected          NULL,
    ...
}
```

**Diagnostic ::= CHOICE** *-- in RejectMCSPDUUltimatum*

```
{
    dc-inconsistent-merge     NULL,
    dc-forbidden-PDU-downward NULL,
    dc-forbidden-PDU-upward   NULL,
    dc-invalid-BER-encoding    NULL,
    dc-invalid-PER-encoding    NULL,
    dc-misrouted-user         NULL,

```

```

    dc-unrequested-confirm          NULL,
    dc-wrong-transport-priority     NULL,
    dc-channel-id-conflict          NULL,
    dc-token-id-conflict            NULL,
    dc-not-user-id-channel          NULL,
    dc-too-many-channels            NULL,
    dc-too-many-tokens              NULL,
    dc-too-many-users               NULL,
    ...
}

-- Part 11: MCSPDU repertoire

NonStandardPDU ::= SEQUENCE
{
    data                            NonStandardParameter,
    ...
}

ExtendedParameterMCSPDU ::= CHOICE
{
    extendedParameterPropose        ExtendedParameterPropose,
    extendedParameterAccept          ExtendedParameterAccept,
    nonStandard                      NonStandardPDU,
    ...
}

DomainMCSPDU ::= CHOICE
{
    plumbDomainIndication            PlumbDomainIndication,
    erectDomainRequest               ErectDomainRequest,
    mergeChannelsRequest             MergeChannelsRequest,
    mergeChannelsConfirm             MergeChannelsConfirm,
    purgeChannelsIndication          PurgeChannelsIndication,
    mergeTokensRequest               MergeTokensRequest,
    mergeTokensConfirm               MergeTokensConfirm,
    purgeTokensIndication            PurgeTokensIndication,
    disconnectProviderUltimatum      DisconnectProviderUltimatum,
    rejectMCSPDUUltimatum            RejectMCSPDUUltimatum,
    attachUserRequest                AttachUserRequest,
    attachUserConfirm                AttachUserConfirm,
    detachUserRequest                DetachUserRequest,
    detachUserIndication              DetachUserIndication,
    channelJoinRequest               ChannelJoinRequest,
    channelJoinConfirm               ChannelJoinConfirm,
    channelLeaveRequest               ChannelLeaveRequest,
    channelConveneRequest             ChannelConveneRequest,
    channelConveneConfirm            ChannelConveneConfirm,
    channelDisbandRequest            ChannelDisbandRequest,
    channelDisbandIndication          ChannelDisbandIndication,
    channelAdmitRequest              ChannelAdmitRequest,
    channelAdmitIndication            ChannelAdmitIndication,
    channelExpelRequest               ChannelExpelRequest,
    channelExpelIndication            ChannelExpelIndication,
    sendDataRequest                  SendDataRequest,
    sendDataIndication                SendDataIndication,
    uniformSendDataRequest            UniformSendDataRequest,
    uniformSendDataIndication          UniformSendDataIndication,
    tokenGrabRequest                 TokenGrabRequest,
    tokenGrabConfirm                  TokenGrabConfirm,
    tokenInhibitRequest               TokenInhibitRequest,
}

```

<b>tokenInhibitConfirm</b>	<b>TokenInhibitConfirm,</b>
<b>tokenGiveRequest</b>	<b>TokenGiveRequest,</b>
<b>tokenGiveIndication</b>	<b>TokenGiveIndication,</b>
<b>tokenGiveResponse</b>	<b>TokenGiveResponse,</b>
<b>tokenGiveConfirm</b>	<b>TokenGiveConfirm,</b>
<b>tokenPleaseRequest</b>	<b>TokenPleaseRequest,</b>
<b>tokenPleaseIndication</b>	<b>TokenPleaseIndication,</b>
<b>tokenReleaseRequest</b>	<b>TokenReleaseRequest,</b>
<b>tokenReleaseConfirm</b>	<b>TokenReleaseConfirm,</b>
<b>tokenTestRequest</b>	<b>TokenTestRequest,</b>
<b>tokenTestConfirm</b>	<b>TokenTestConfirm,</b>
<b>nonStandard</b>	<b>NonStandardPDU,</b>
...	
}	

**END**

## **9 Codificación de las MCSPDU**

Cada MCSPDU se transporta como una TSDU a través de una TC perteneciente a una conexión MCS. Las MCSPDU de conexión tienen un tamaño ilimitado. Las MCSPDU de dominio tienen su tamaño limitado por un parámetro del dominio MCS.

Para transferir MCSPDU entre proveedores MCS pares se utiliza una codificación normalizada de valor de datos ASN.1. La negociación de la versión del protocolo incluye la selección de las reglas de codificación. Esta negociación implica el intercambio de una MCSPDU **conexión-inicial** y de **conexión-respuesta** sobre la TC inicial. Se definen tres versiones de este protocolo:

- *Versión 1* – Todas las MCSPDU se describen en la cláusula 7 y utilizan las reglas de codificación básicas de la Recomendación X.690.
- *Versión 2* – Todas las MCSPDU se describen en la cláusula 7. Esta versión utiliza las reglas de codificación básica para las MCSPDU de conexión y las reglas de codificación empaquetada de la Recomendación X.691 para todas las MCSPDU de dominio subsiguientes. Específicamente, la variante **ALIGNED** de **BASIC-PER** se aplicará al tipo ASN.1 **DomainMCSPDU**. La cadena de bits producida se transportará como un número entero de octetos. El bit inicial de esta cadena de bits coincidirá con el bit más significativo del primer octeto.
- *Versión 3* – Las MCSPDU de conexión se describen en la cláusula 7 y utilizan las reglas de codificación básica para retrocompatibilidad con la versión 2. Las MCSPDU de parámetros ampliados y de dominio se describen en la cláusula 8 y utilizan las reglas de codificación empaquetada que se especifican para la versión 2. La versión 3 permite que los nodos puedan traducir la versión 3 del protocolo en la versión 2 del mismo para la transmisión sobre una conexión con un nodo que haya especificado la versión 2 del protocolo en los parámetros de dominio. Esta traducción se describe en la subcláusula 13.11.

NOTA 1 – Las reglas de codificación empaquetada permiten obtener encabezamientos de MCSPDU más compactos.

NOTA 2 – Tanto las reglas de codificación básica (BER) como las reglas de codificación empaquetada (PER) son autolimitativas, en el sentido de que contienen información suficiente para localizar el final de cada MCSPDU codificada. Pudiera alegarse que esto hace innecesario el uso de TSDU y que este protocolo podría implementarse a través de servicios de transporte no normalizados que transporten trenes de octetos sin preservar los confines de las TSDU. No obstante, ese método es más vulnerable a errores de implementación. Si en cualquier circunstancia se perdieran las demarcaciones entre las MCSPDU, una recuperación sería difícil.

## 10 Encaminamiento de las MCSPDU

### 10.1 MCSPDU de conexión y de parámetros ampliados

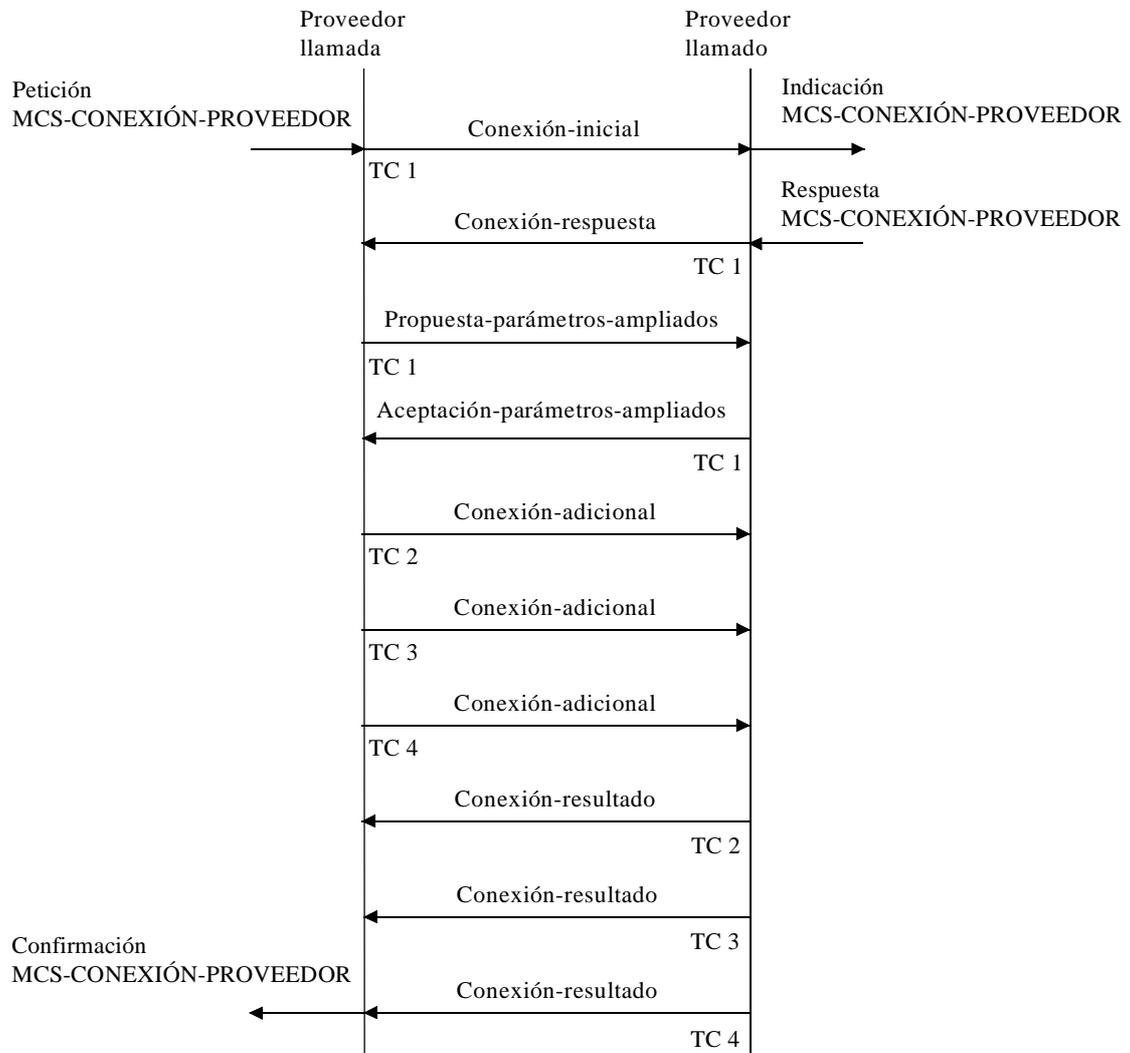
Las figuras 10-1 y 10-2 especifican el intercambio de MCSPDU de conexión y de parámetros ampliados (el número y orden relativo de las TC adicionales puede variar).

Al recibir una **conexión-respuesta**, el proveedor MCS llamante determina si la nueva conexión se establece entre dos miembros de la cumbre V3 examinando el número de versión negociado. Si así ocurre, inicia el intercambio de MCSPDU de **parámetros-ampliados**. Si ninguno de los nodos es miembro de una cumbre V3, el proveedor llamante no emitirá una **propuesta-parámetros-ampliados**, sino que procederá con MCSPDU de **conexión-adicional**.

Una vez que recibe una **conexión-respuesta**, el proveedor MCS llamante conoce el valor negociado para el número de prioridades de transferencia de datos aplicadas al dominio. Con fines de ilustración, las MCSPDU de conexión en TC adicionales se han representado siguiendo la secuencia precisa de 2, 3, 4 en el proveedor MCS llamado. En realidad, las conexiones de transporte puede que no se establezcan en el mismo orden en que fueron pedidas. En efecto, una **conexión-adicional** puede llegar en una posición diferente de aquella en que fue enviada, lo que hace que se retorne una **conexión-resultado** también fuera de secuencia. Puede suceder asimismo que esta última, aunque haya sido enviada en secuencia, aparezca en un orden diferente en tránsito. Un proveedor MCS llamante no tiene que esperar a que se establezcan todas las TC adicionales para enviar la primera **conexión-adicional**. El proveedor MCS llamado no tiene que esperar a que llegue un conjunto completo de MCSPDU de **conexión-adicional** para retornar la primera **conexión-resultado**. La llegada de un conjunto completo de resultados de éxito al proveedor MCS llamante, cualquiera que sea el orden en que se reciban, provoca la generación de una confirmación MCS-CONEXIÓN-PROVEEDOR de éxito.

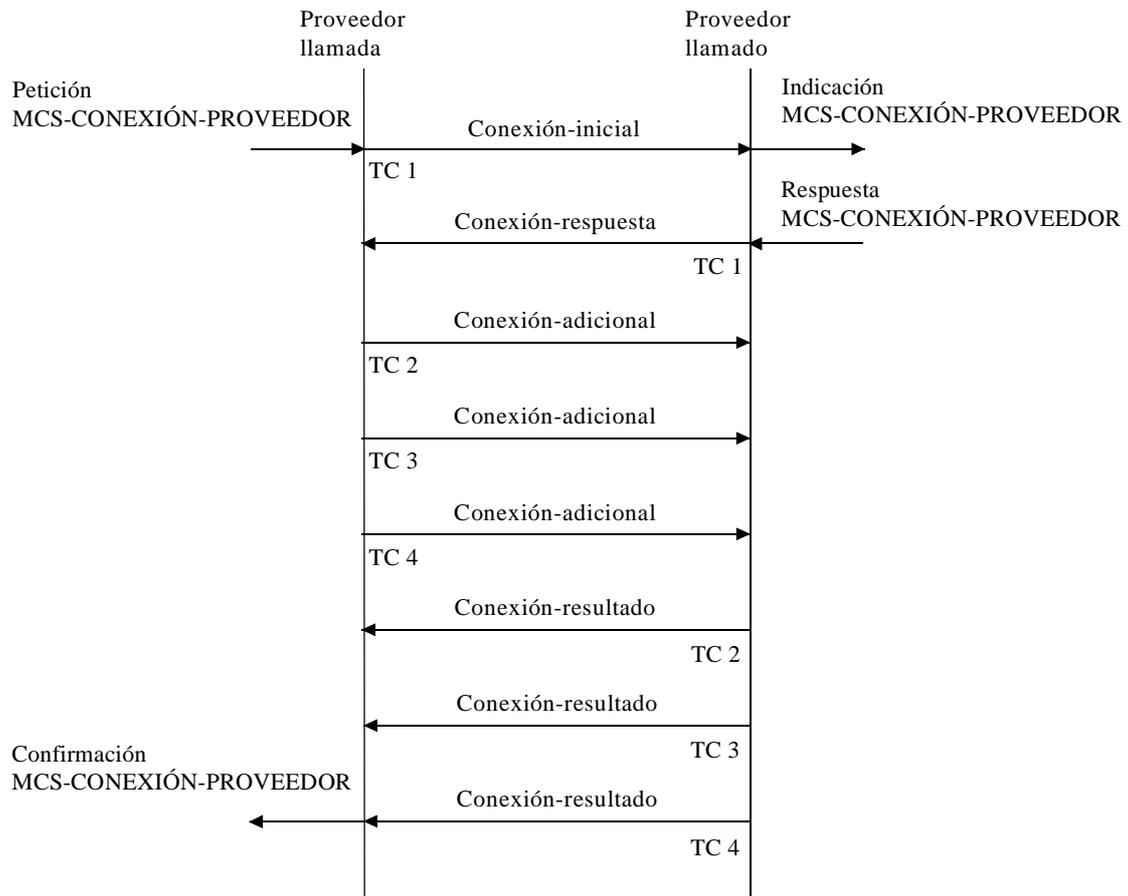
Una **conexión-respuesta** o **conexión-resultado** de fracaso o una indicación T-DESCONEXIÓN en algún punto intermedio traen consigo la desconexión de todas las TC que hasta ese momento pertenezcan a la conexión MCS y la generación de una confirmación MCS-CONEXIÓN-PROVEEDOR de fracaso.

Una petición MCS-CONEXIÓN-PROVEEDOR especifica cuál de dos proveedores MCS es el más alto. Esta relación jerárquica determina el ulterior encaminamiento de las MCSPDU de dominio, después de lo cual la distinción entre proveedor MCS llamante y llamado es intrascendente. Por ejemplo, como paso siguiente la capa MCS deberá fusionar los recursos de dos dominios que antes eran independientes. El proveedor MCS más bajo genera **MergeChannelsRequest** y **MergeTokensRequest**, las que se transmiten al proveedor MCS más alto a través de la nueva conexión MCS. Estas MCSPDU pueden transmitirse en el sentido llamante a llamado o en el opuesto, lo que depende de cómo se haya fijado la bandera ascendente.



T1602430-97

**Figura 10-1/T.125 – Flujo de mensajes de las MCSPDU de conexión y de parámetros ampliados entre nodos miembros de una cumbre V3**



T1602440-97

**Figura 10-2/T.125 – Flujo de mensajes de las MCSPDU de conexión entre nodos que no son miembros de la cumbre V3**

## 10.2 MCSPDU de dominio

El cuadro 10-1 especifica el encaminamiento de las MCSPDU de dominio.

Si un proveedor MCS genera o reenvía una MCSPDU de categoría *petición*, dicha unidad se transmite en sentido ascendente a través de la conexión MCS única. **ErectDomainRequest**, **ChannelJointRequest**, **ChannelLeaveRequest** y **CapabilitiesNotificationRequest** pueden ser consumidas en algún proveedor MCS intermedio. Otras peticiones suben para ser tratadas por el proveedor MCS superior, a menos que se determine que el contenido de una petición no es válido, en cuyo caso puede ignorarse su MCSPDU, sin confirmación.

**Cuadro 10-1/T.125 – Encaminamiento de las MCSPDU de dominio**

<b>Categoría</b>	<b>MCSPDU</b>	<b>TC</b>	<b>Sentido</b>
Petición	<b>ErectDomainRequest</b> <b>MergeChannelsRequest</b> <b>MergeTokensRequest</b> <b>AttachUserRequest</b> <b>DetachUserRequest</b> <b>ChannelJoinRequest</b> <b>ChannelLeaveRequest</b> <b>ChannelConveneRequest</b> <b>ChannelDisbandRequest</b> <b>ChannelsAdmitRequest</b> <b>ChannelExpelRequest</b> <b>TokenGrabRequest</b> <b>TokenInhibitRequest</b> <b>TokenGiveRequest</b> <b>TokenPleaseRequest</b> <b>TokenReleaseRequest</b> <b>TokenTestRequest</b> <b>CapabilitiesNotificationRequest</b>	I	Hacia arriba
	<b>SendDataRequest</b> <b>UniformSendDataRequest</b>	A	
Indicación	<b>PlumbDomainIndication</b> <b>PurgeChannelsIndication</b> <b>PurgeTokensIndication</b> <b>DetachUserIndication</b> <b>ChannelDisbandIndication</b> <b>ChannelAdmitIndication</b> <b>ChannelExpelIndication</b> <b>TokenGiveIndication</b> <b>TokenPleaseIndication</b> <b>CapabilitiesNotificationIndication</b>	I	Hacia abajo
	<b>SendDataIndication</b> <b>UniformSendDataIndication</b>	A	

Si un proveedor MCS genera o reenvía una MCSPDU de categoría *indicación*, copias de la misma, posiblemente con un contenido modificado, se transmiten en sentido descendente a través de cero o más conexiones MCS, de acuerdo con las siguientes reglas:

- a) **PlumbDomainIndication** se envía hacia abajo en todas las conexiones MCS. El límite de altura que ella contiene se decrementa en una unidad. Un proveedor MCS que reciba esta MCSPDU con un límite de altura de cero deberá desconectar.
- b) **PurgeChannelsIndication** se envía hacia abajo en todas las conexiones MCS. El conjunto de identificadores de usuario reenviados no se modifica, de modo que todos los usuarios

desanexionados serán anunciados a los que permanecen. El conjunto de otros identificadores de canal reenviados puede ser restringido a los utilizados en un subárbol. En un antiguo proveedor superior que está todavía fusionándose con un dominio más alto, ambos identificadores de usuario y otros identificadores de canal están restringidos a aquellos cuya aceptación en el dominio superior ha sido confirmada.

- c) **PurgeTokensIndication** se envía hacia abajo en todas las conexiones MCS. El conjunto de identificadores de testigo reenviados puede restringirse a los utilizados en un subárbol. En un proveedor superior anterior que está todavía fusionando con un dominio superior, los identificadores de testigo están restringidos a aquellos cuya aceptación en el dominio superior ha sido confirmada.
- d) **DetachUserIndication** se envía hacia abajo en todas las conexiones MCS. El conjunto de identificadores de usuario reenviados se mantiene sin modificación, de modo que todos los usuarios desanexionados serán anunciados a los usuarios que quedan. En un proveedor superior anterior que está aún fusionando con un dominio superior, los identificadores de usuario están restringidos a aquellos cuya aceptación en el dominio superior ha sido confirmada.
- e) **ChannelDisbandIndication** se envía hacia abajo en todas las conexiones MCS que contienen en su subárbol el gestor del canal privado o cualquier usuario admitido.
- f) **ChannelAdmitIndication** y **ChannelExpelIndication** se envían hacia abajo en todas las conexiones MCS que contienen uno o más de los usuarios influidos en su subárbol. El conjunto de los identificadores de usuario puede limitarse a los residentes en un subárbol.
- g) **TokenGiveIndication** se envía hacia abajo en una sola conexión MCS que contiene el receptor designado en su subárbol.
- h) **TokenPleaseIndication** se envía hacia abajo en todas las conexiones MCS que contienen en su subárbol un usuario que ha tomado o inhibido el testigo, o al que se le está cediendo el testigo.
- i) **CapabilitiesNotificationIndication** se envía hacia abajo en todas las conexiones MCS en la cumbre V3. Cada nodo que recibe esta MCSPDU modifica su copia local de la lista de capacidades de la cumbre V3 en base al contenido de lista de adición (Add List) y lista de supresión (Remove List).
- j) **SendDataIndication** y **UniformSendDataIndication** se envían hacia abajo en todas las conexiones MCS por las cuales se efectúa una incorporación al canal especificado, excepto que, cuando se genera **SendDataIndication**, dicha unidad no se devuelve en la conexión por la que llegó **SendDataRequest**.

No es necesario reenviar las indicaciones **PurgeChannelsIndication**, **PurgeTokensIndication**, **DetachUserIndication**, **ChannelAdmitIndication** y **ChannelExpelIndication** si los conjuntos de identificadores que contienen están vacíos.

Si un proveedor MCS genera o reenvía una MCSPDU de categoría *respuesta*, dicha unidad se desplaza hacia arriba a través de la conexión MCS única. La unidad sube para ser tratada por el proveedor MCS superior, a menos que se determine que su contenido es inválido.

Si un proveedor MCS genera o reenvía una MCSPDU de categoría *confirmación*, dicha unidad desciende por una sola conexión MCS, de acuerdo con las siguientes reglas.

- a) **MergeChannelsConfirm** recorre, en sentido opuesto, el mismo camino de la **MergeChannelsRequest** más antigua que no haya sido contestada por una confirmación. Para esto se requiere que cada proveedor MCS mantenga una cola de tipo primero en entrar primero en salir para las peticiones pendientes.

- b) **MergeTokensConfirm** recorre, en sentido opuesto, el mismo camino de la **MergeTokensRequest** más antigua que no haya sido contestada por una confirmación. Para esto se requiere que cada proveedor MCS mantenga una cola de tipo primero en entrar primero en salir para las peticiones pendientes.
- c) **AttachUserConfirm** recorre, en sentido opuesto, el mismo camino de la **AttachUserRequest** más antigua que no haya sido contestada por una confirmación. Si hay pendientes más de una **AttachUserRequest**, no es crítico el que sea una u otra; no obstante, por razones de equidad, cada proveedor MCS debe mantener una cola de tipo primero en entrar primero en salir. Después de enviar **AttachUserConfirm**, un proveedor MCS registrará a qué subárbol se asigna de esa forma el identificador de usuario contenido en ella.
- d) Otras MCSPDU de categoría confirmación contienen un identificador de usuario iniciador que fue anteriormente asignado por la acción de **AttachUserConfirm** como se ha explicado antes. Estas MCSPDU se envían hacia abajo por la conexión MCS que conduce al subárbol donde fue asignado el identificador de usuario. Estas unidades continúan de esta manera y llegan finalmente al proveedor que atiende la afectación MCS solicitante.

Se generan confirmaciones en el curso del procesamiento de peticiones iguales. Todas las confirmaciones con excepción de **ChannelJointConfirm** las genera el proveedor MCS superior.

Si un proveedor MCS genera una MCSPDU de categoría *ultimátum*, ésta se transmite por una conexión MCS única en sentido ascendente o descendente. **DisconnectProviderUltimatum** ordena al proveedor MCS receptor que desconecte la conexión que la transporta. **RejectMCSPDUUltimatum** rechaza una MCSPDU errónea con un código de diagnóstico e invita al proveedor MCS que la transmitió a desconectar. Los ultimátum no se reenvían.

## 11 Significado de las MCSPDU

En los cuadros 11-1 a 11-51 se reitera el contenido de las MCSPDU definidas en la cláusula 8.

### 11.1 Conexión-inicial

Una **conexión-inicial** es generada por una petición MCS-CONEXIÓN-PROVEEDOR. Se envía como la primera TSDU a través de la TC inicial de una nueva conexión MCS. En el receptor, genera una indicación MCS-CONEXIÓN-PROVEEDOR.

**Cuadro 11-1/T.125 – MCSPDU de conexión inicial**

Contenido	Fuente	Sumidero
Selector del dominio llamante	Petición	Indicación
Selector del dominio llamado	Petición	Indicación
Bandera ascendente	Petición	Indicación
Parámetros de dominio deseados	Petición	Indicación
Parámetros de dominio mínimos	Petición	Indicación
Parámetros de dominio máximos	Petición	Indicación
Datos de usuario	Petición	Indicación

La dirección de transporte llamante y la dirección de transporte llamada son parámetros adicionales de la petición y de la indicación MCS-CONEXIÓN-PROVEEDOR. Estas direcciones se convierten en parámetros de T-CONEXIÓN y no se pasan explícitamente en ninguna MCSPDU. El mismo par

de direcciones de transporte se utilizará para pedir todas las TC pertenecientes a la misma conexión MCS.

La calidad de servicio de transporte es un parámetro adicional de la petición MCS-CONEXIÓN-PROVEEDOR pero no de la indicación. La calidad de servicio puede variar de una TC a otra, y la calidad disponible sólo viene a conocerse en el curso del establecimiento de una TC dada. Como el número de TC adicionales que se necesitan no se conoce hasta que MCS-CONEXIÓN-PROVEEDOR haya negociado el parámetro de dominio para el número de prioridades de datos implementadas, esta primitiva no puede negociar completamente, al mismo tiempo, su calidad de servicio de transporte. Un proveedor MCS llamado deberá por tanto aceptar automáticamente las TC entrantes con la calidad de servicio ofrecida mientras ésta satisfaga cualquier valor mínimo especificado facultativamente por el proveedor MCS llamante e indicado a través de cada una de las T-CONEXIÓN.

La interpretación de valores de selector de dominio es un asunto local y la determina cada proveedor MCS. Dichos valores están constituidos por cadenas de octetos que tienen las características de una dirección. Se puede determinar valores aceptables mediante el proceso de configuración de un proveedor MCS. Más de un valor pueden seleccionar el mismo dominio. Un selector de dominio no especificado es una cadena de octetos de longitud cero. Por convenio local, puede representarse por algún valor explícito.

La bandera ascendente especifica el sentido de transmisión de una nueva conexión MCS: se pone a verdadero si el proveedor llamado ha de considerarse más alto que el proveedor llamante y se pone a falso en los demás casos. Un proveedor MCS desempeña un cometido en la jerarquía de un dominio basado en el sentido de las conexiones MCS en las que participa. Ningún proveedor permitirá dos conexiones a proveedores más altos. Un proveedor sin una conexión a un proveedor más alto actuará como el proveedor MCS superior.

Los parámetros de dominio deseados de **conexión-inicial** deberán, cada uno de ellos, estar comprendidos entre los valores mínimo y máximo especificados. Un proveedor MCS modificará los parámetros de dominio solicitados para reflejar límites de su implementación o para imponer valores ya convenidos entre los miembros existentes de un dominio. Puede aumentar los mínimos y reducir los máximos. Deberá cambiar los valores deseados con el solo objeto de mantenerlos dentro del intervalo. Un proveedor MCS tiene que estar preparado para satisfacer cualquier respuesta comprendida en la gama de valores que propone.

Los datos de usuario están constituidos por una cadena de octetos de longitud arbitraria. La longitud puede ser cero.

Un proveedor MCS acepta automáticamente cada TC entrante hasta el límite de su capacidad. Los datos de un usuario en T-CONEXIÓN no se utilizan. La primera TSDU recibida en una transferencia de datos, sea una MCSPDU de **conexión-inicial**, o de **conexión-adicional**, determina la naturaleza de la TC. Si el contenido es inaceptable, un proveedor MCS llamado puede desconectar la TC inmediatamente. La reacción preferida es retornar una **conexión-respuesta** o una **conexión-resultado**, según el caso, con la explicación de la causa por la cual fracasó la conexión MCS. Después de esto, el proveedor MCS llamante desconectará.

## 11.2 Conexión-respuesta

Una **conexión-respuesta** se genera por una respuesta MCS-CONEXIÓN-PROVEEDOR. Es la primera TSDU enviada en sentido de retorno a través de la TC inicial de una nueva conexión MCS. Transporta la aceptación de una conexión MCS al proveedor MCS llamante, quien procede entonces a establecer las TC adicionales que se necesiten.

**Cuadro 11-2/T.125 – MCSPDU de conexión-respuesta**

Contenido	Fuente	Sumidero
Resultado	Respuesta	Confirmación
Parámetros de dominio	Respuesta	Confirmación
Identificador de conexión llamada	Proveedor llamado	Proveedor llamante
Datos de usuario	Respuesta	Confirmación

Si el resultado es de éxito, esta MCSPDU fija los parámetros de dominio en vigor. Entre éstos se encuentra el número de prioridades de transferencia de datos implementadas, que es igual al número de TC en una conexión MCS. Si este número es mayor que uno, hay que crear TC adicionales y vincularlas a la conexión MCS mediante el intercambio de MCSPDU de **conexión-adicional** y de **conexión-resultado**.

El identificador de conexión llamada sirve de medio para asociar TC entrantes adicionales en el proveedor MCS llamado con esta TC inicial. Su valor se elige exclusivamente para esta finalidad. Deberá identificar unívocamente una conexión MCS en curso en el proveedor llamado. Este identificador no tiene un significado duradero tras la compleción de MCS-CONEXIÓN-PROVEEDOR.

La mayor parte de los parámetros de la confirmación MCS-CONEXIÓN-PROVEEDOR se transportan en **conexión-respuesta**. Si el resultado es de fracaso o no se necesitan TC adicionales, la confirmación se genera inmediatamente. En otro caso, se difiere hasta que se conozcan los resultados para vincular TC adicionales a la conexión MCS.

### 11.3 Conexión-adicional

Si el proveedor MCS llamante determina que ambos proveedores son miembros de la cumbre V3, se genera **conexión-adicional** tras la recepción de la **aceptación-parámetros-ampliados**, en cualquier otro caso aquélla se genera después de la recepción de **conexión-respuesta**. Se envía como la primera TSDU a través de una TC adicional o de una nueva conexión MCS.

La prioridad de datos toma los valores *alta*, *media* y *baja*, en secuencia, hasta el número de TC adicionales requeridas.

**Cuadro 11-3/T.125 – MCSPDU de conexión-adicional**

Contenido	Fuente	Sumidero
Id de conexión llamada	Proveedor llamante	Proveedor llamado
Prioridad de datos	Proveedor llamante	Proveedor llamado

### 11.4 Conexión-resultado

Se genera **conexión-resultado** tras la recepción de **conexión-adicional**. Es la primera TSDU enviada en sentido de retorno por una TC adicional o una nueva conexión MCS.

**Cuadro 11-4/T.125 – MCSPDU de conexión-resultado**

Contenido	Fuente	Sumidero
Resultado	Proveedor llamado	Confirmación

Si cualquier resultado es de fracaso, se generará una confirmación MCS-CONEXIÓN-PROVEEDOR inmediatamente. Todas las TC asociadas con la conexión MCS serán desconectadas y se ignorarán las MCSPDU que ellas transporten.

En otro caso, se esperarán resultados de éxito para cada TC adicional. Estos resultados podrán retornar fuera de secuencia. Una vez que se hayan obtenido todos, se generará una confirmación MCS-CONEXIÓN-PROVEEDOR de éxito.

Tras una confirmación MCS-CONEXIÓN-PROVEEDOR, podrán pasar MCSPDU de dominio a través de una conexión MCS. Cada conexión MCS pertenece a un solo dominio. En configuraciones donde un proveedor MCS atiende más de un dominio, la conexión MCS que conduce MCSPDU determina el dominio a que se aplican. Las descripciones de MCSPDU de dominio que se tratan en el resto de esta subcláusula se sitúan en el contexto de un dominio único.

### 11.5 Propuesta-parámetros-ampliados

La **propuesta-parámetros-ampliados** se genera después de la recepción de una **conexión-respuesta** si el proveedor MCS llamante determina que los dos proveedores son miembros de la cumbre V3. Se envía a través de la TC inicial de una nueva conexión MCS.

**Cuadro 11-5/T.125 – MCSPDU de propuesta-parámetros-ampliados**

Contenido	Fuente	Sumidero
Parámetros ampliados objetivo	Proveedor llamante	Proveedor llamado
Parámetros ampliados mínimos	Proveedor llamante	Proveedor llamado
Parámetros ampliados máximos	Proveedor llamante	Proveedor llamado

En el caso de parámetros ampliados con una gama de posible valores, los parámetros deseados de la **propuesta-parámetros-ampliados** estarán comprendidos entre los valores mínimo y máximo especificados. Un proveedor MCS revisará los parámetros ampliados solicitados con el fin de reflejar cuales son los límites de su implementación o para imponer valores acordados entre los miembros de un dominio. Puede aumentar los mínimos y disminuir los máximos. Sólo modificará los valores deseados a fin de mantenerlos dentro del intervalo. Un proveedor MCS debe estar preparado para admitir cualquier valor aceptado que esté comprendido en la gama de valores por él propuesta. Si los parámetros ampliados son inaceptables, un proveedor MCS llamado desconectará la TC inmediatamente.

En el caso de *soporte de datos no fiables*, si el proveedor llamante permite que se negocie el valor, el valor mínimo será falso, el valor máximo será verdadero y el valor deseado indica la preferencia del llamante. Si no es negociable el soporte de datos no fiables, los valores deseado, mínimo y máximo se fijarán según la preferencia del llamante.

El valor para el *identificador de referencia del dominio* será idéntico para el valor deseado, mínimo y máximo e identificará de forma inequívoca el dominio al proveedor llamante. El proveedor llamado puede utilizar este valor para referenciar este dominio al proveedor llamante.

### 11.6 Aceptación-parámetros-ampliados

La **aceptación-parámetros-ampliados** se genera después de recibir la **propuesta-parámetros-ampliados**. Transporta la aceptación de los valores de los parámetros ampliados hasta el proveedor MCS llamante, que procede entonces a establecer las TC adicionales que sean requeridas.

**Cuadro 11-6/T.125 – MCSPDU de aceptación-parámetros-ampliados**

Contenido	Fuente	Sumidero
Parámetros ampliados	Proveedor llamado	Proveedor llamante

El valor del *identificador de referencia del dominio* es idéntico para los valores deseado, mínimo y máximo e identifica inequívocamente el dominio al proveedor llamado. El proveedor llamante puede utilizar este valor para hacer referencia de este dominio al proveedor llamado.

### 11.7 PlumbDomainIndication

Se genera **PlumbDomainIndication** tras la compleción exitosa de MCS-CONEXIÓN-PROVEEDOR. Esta unidad "hermetiza" la jerarquía de proveedores MCS por debajo de una nueva conexión MCS para asegurar que no se haya creado ningún ciclo. **PlumbDomainIndication** la genera también el proveedor MCS superior para imponer la máxima altura de un dominio.

**Cuadro 11-7/T.125 – MCSPDU PlumbDomainIndication**

Contenido	Fuente	Sumidero
Límite de altura	Superior antiguo o actual	Subordinados

**PlumbDomainIndication** la genera en el extremo inferior de una nueva conexión MCS el proveedor que ha dejado de ser el superior de un dominio. Su contenido se inicializa al parámetro de dominio para la altura máxima del dominio. **PlumbDomainIndication** se transmite en sentido descendente a través de todas las conexiones MCS.

Según se necesite, **PlumbDomainIndication** se genera de la misma manera en el proveedor MCS superior.

Dondequiera que se reciba **PlumbDomainIndication**, se examina el límite de altura en ella contenido. Si el límite es mayor que cero se decrementa en una unidad, y se reenvía **PlumbDomainIndication** en sentido descendente por todas las conexiones MCS. Por otra parte, un valor cero significa que el receptor está demasiado lejos del proveedor superior. Deberá reaccionar desconectando la conexión MCS ascendente. Con esto se suprime un subárbol en su totalidad y se ayuda a reparar la altura del dominio.

En presencia de un ciclo, el límite de altura de **PlumbDomainIndication** tiene que disminuir hasta que llegue a cero. El proveedor que detecta esto rompe el ciclo, para lo cual deberá suprimir todos los proveedores que intervienen en el ciclo, y sus subordinados, desde el dominio.

NOTA – Un proveedor MCS que sólo cuente con el conocimiento local de las conexiones MCS no puede impedir la creación de ciclos. Puede asegurar que haya en todo momento, cuando más, una conexión ascendente, pero no puede asegurar que la conexión ascendente no forme bucle con algún proveedor inferior. Cuando se crea un ciclo, la causa inmediata de ello es una conexión ascendente defectuosa desde el proveedor MCS superior. Las aplicaciones de controlador, que especifican las conexiones MCS que habrán de crearse, deberán tratar de evitar esos errores.

### 11.8 ErectDomainRequest

Se genera **ErectDomainRequest** tras la compleción exitosa de MCS-CONEXIÓN-PROVEEDOR. Esta unidad se transmite en sentido ascendente y comunica cambios en la altura de proveedores y sus intervalos de cumplimentación del caudal. Un proveedor MCS genera **ErectDomainRequest** cada vez que su altura o intervalo cambian.

La altura de un proveedor MCS puede cambiar cuando se añade o se retira una conexión MCS, o cuando un proveedor subordinado comunica un cambio mediante una **ErectDomainRequest**. Su intervalo de supervisión para cumplimentar el caudal mínimo especificado como un parámetro de dominio puede cambiar al adaptarse a los intervalos comunicados por subordinados, o por otras razones. Si cualquiera de estos valores cambia, un proveedor MCS transmitirá **ErectDomainRequest** a su superior inmediato.

**Cuadro 11-8/T.125 – MCSPDU ErectDomainRequest**

Contenido	Fuente	Sumidero
Altura en el dominio	Subordinado	Proveedor más alto
Intervalo de cumplimentación del caudal	Subordinado	Proveedor más alto

### 11.9 MergeChannelsRequest

Se genera **MergeChannelsRequest** tras la compleción exitosa de MCS-CONEXIÓN-PROVEEDOR. Esta unidad comunica en sentido ascendente los atributos de canales que tenían un antiguo proveedor superior, para poder incorporarlos a un dominio fusionado.

**Cuadro 11-9/T.125 – MCSPDU MergeChannelsRequest**

Contenido	Fuente	Sumidero
Fusión de canales	Antiguo superior	Proveedor superior
Purga de identificadores de canal	Intermedios	Proveedor superior

La **MergeChannelsRequest** puede ser rellena con los atributos de múltiples canales, hasta el límite impuesto por el dominio al tamaño de las MCSPDU. Como se indica detalladamente en las definiciones ASN.1 de la cláusula 7, cada uno de los cuatro tipos de canales que se están utilizando (estático, identificador de usuario, privado, asignado) tiene su correspondiente conjunto de atributos. Estos atributos están contenidos en la base de información del proveedor MCS superior y están parcialmente replicados en los subárboles donde se utiliza un canal. Cuando se fusionan dos dominios, mediante MCS-CONEXIÓN-PROVEEDOR, los canales que se están utilizando en el dominio más bajo tienen que ser incorporados a la base de información del proveedor más alto, o eliminados (purgados) del proveedor más bajo. Esta decisión incumbe al proveedor superior del dominio fusionado.

Cada canal deberá considerarse individualmente. Si los límites impuestos por el dominio a los canales que se están utilizando lo permiten y no se está dando ya al identificador de canal un uso conflictivo, el dominio más alto deberá expandirse hasta incluirlo. La utilización de un identificador de canal estático nunca da lugar a conflicto. La utilización de un identificador de canal privado en el dominio más bajo no da lugar a conflicto si se utiliza también como identificador de canal privado en el dominio más alto y tiene el mismo identificador de usuario como gestor. Todas las demás combinaciones de uso simultáneo están desautorizadas, y el identificador de canal deberá ser purgado del dominio más bajo.

Si un canal privado tiene un extenso conjunto de usuarios admitidos, sus atributos pudieran no caber en una sola **MergeChannelsRequest** y deberán enviarse hacia arriba en múltiples MCSPDU. Sin embargo, la segunda petición de fusionar el mismo canal privado, y las siguientes, deberán aplazarse hasta que se haya recibido una **MergeChannelsConfirm** en respuesta a la primera. Sólo entonces se sabrá si los límites impuestos por el dominio han permitido que se ponga en uso el canal en el

dominio más alto. Si la primera petición fracasa, no se repetirá con un subconjunto restante de usuarios admitidos.

Cada **MergeChannelsRequest** provoca una contestación del proveedor MCS superior mediante **MergeChannelsConfirm**, en la misma secuencia. Una **MergeChannelsConfirm** no contiene nada que identifica explícitamente la **MergeChannelsRequest** precedente. Las contestaciones se encaminarán basándose exclusivamente en el orden en que se recibieron estas MCSPDU. Los proveedores MCS por encima del proveedor superior tomarán nota de cada **MergeChannelsRequest** no contestada, y de su procedencia, de manera que se pueda retornar la correspondiente **MergeChannelsConfirm** a través de la misma conexión MCS.

Los proveedores MCS intermedios validarán los identificadores de usuario enunciados en atributos de canales privados, para asegurar que se asignen legítimamente al subárbol en que origina la **MergeChannelsRequest**. Se suprimirán los identificadores de usuario inválidos. Si se suprime el gestor de un canal privado, todos los atributos de canal deberán suprimirse en la petición de fusión y solamente el identificador de canal se incluirá en el conjunto que habrá de purgarse. Salvo esta validación de identificadores de usuario, los proveedores MCS intermedios no modificarán el contenido de una **MergeChannelsRequest**.

Un antiguo proveedor superior esperará la confirmación individual de que todos los identificadores de usuario y de todos los identificadores de testigo hayan sido incorporados al dominio fusionado, o purgados, antes de empezar a someter atributos de canal estático, asignado o privado, para la fusión.

#### 11.10 MergeChannelsConfirm

Una **MergeChannelsConfirm** contesta a una **MergeChannelsRequest** precedente. Refleja el mismo conjunto de identificadores de canal y un subconjunto de los atributos. Los atributos de canal no incorporados en el dominio fusionado se comunican como identificadores de canal para ser purgados.

Los identificadores de canal aceptados se reflejan con los atributos que se introdujeron en la base de información del proveedor MCS superior. Los proveedores intermedios actualizarán su base de información para hacerla conforme.

**Cuadro 11-10/T.125 – MCSPDU MergeChannelsConfirm**

Contenido	Fuente	Sumidero
Fusión de canales	Proveedor superior	Intermedios
Purga de identificadores de canal	Proveedor superior	Antiguo proveedor superior

Los canales que van a ser purgados del dominio más bajo se listan por los identificadores solamente. Si los mismos identificadores de canal se utilizaron en el dominio más alto, se mantienen sin modificación. Los proveedores intermedios reenviarán los identificadores de canal purgados sin tratarlos.

Los proveedores MCS encaminarán **MergeChannelsConfirm** a la fuente de la **MergeChannelsRequest** precedente basándose en el conocimiento de que hay una contestación de uno a uno. **MergeChannelsConfirm** retorna al antiguo proveedor superior que generó **MergeChannelsRequest**. Allí los canales fusionados pueden ser ignorados, ya que han permanecido en la base de información mientras está pendiente una contestación. Los identificadores de canal purgados se suprimirán, ya que estaban destinados a **PurgeChannelsIndication**.

Los proveedores MCS intermedios confirmarán que los identificadores de usuario anunciados en atributos de canal privado se asignan al subárbol al cual se encamina **MergeChannelsConfirm**. Si un gestor de canal privado ha sido desanexionado y reasignado en algún otro lugar en el tiempo transcurrido desde que la **MergeChannelsRequest** precedente fue validada, un proveedor intermedio generará una **ChannelDisbandRequest** haciendo del canal privado una víctima de la fusión de dominio y trasladará el identificador de canal al conjunto purgado. Si cualesquiera usuarios admitidos han sido reasignados en algún otro lugar, los excluirá del canal.

### 11.11 PurgeChannelsIndication

**PurgeChannelsIndication** se genera en un antiguo proveedor superior tras la recepción de **MergeChannelsConfirm**. Se difunde en sentido descendente y purga el uso de identificadores de canal especificados eliminándolos de los proveedores subordinados.

**Cuadro 11-11/T.125 – MCSPDU PurgeChannelsIndication**

Contenido	Fuente	Sumidero
Desanexar identificadores de usuario	Antiguo superior	Subordinados
Purga de identificadores de canal	Antiguo superior	Subordinados

De acuerdo con el uso que se esté haciendo de un identificador de canal, el efecto de su purga es el siguiente: indicación MCS-DESANEXIÓN-USUARIO a todos los usuarios de un canal de identificador de usuario; indicación MCS-ABANDONO-CANAL a los usuarios incorporados de un identificador de canal estático o asignado; indicación MCS-DISOLUCIÓN-CANAL al gestor, e indicación MCS-EXCLUSIÓN-CANAL a los usuarios admitidos de un identificador de canal privado.

Un proveedor superior anterior que conoce la utilización de todos los canales en su dominio más bajo, puede generar las indicaciones apropiadas a partir solamente de sus identificadores de canal. Sin embargo, sus subordinados sólo pueden tener un conocimiento parcial. Hay que informarles qué identificadores de canal representan a usuarios desanexionados, para los cuales se genera siempre una indicación, y cuáles representan a otras clases de canales, para los cuales se genera solamente una indicación si el canal está en uso en el proveedor subordinado. Por tanto, **PurgeChannelsIndication** divide los identificadores de canal que han de purgarse en estas dos categorías.

La purga de un identificador de usuario mediante **PurgeChannelsIndication** tendrá las mismas consecuencias que su supresión mediante **DetachUserIndication**, excepto que, como el receptor de **PurgeChannelsIndication** ya no es el proveedor superior, no tiene necesidad de generar **ChannelDisbandIndication** o **TokenGiveConfirm** como efecto secundario.

NOTA – Un proveedor puede recibir en **PurgeChannelsIndication** identificadores de canal estático y de canal asignado a los que no se ha incorporado o identificadores de canales privados para los cuales sus afectaciones no son ni gestores ni usuarios admitidos. Registros de utilización mantenidos en la base de información permiten a un proveedor suprimir indicaciones de primitivas para esos identificadores de canal.

### 11.12 MergeTokensRequest

Se genera una **MergeTokensRequest** tras la compleción exitosa de MCS-CONEXIÓN-PROVEEDOR. Dicha unidad comunica en sentido ascendente los atributos de testigos poseídos por un antiguo proveedor superior de modo que puedan ser incorporados al dominio fusionado.

### Cuadro 11-12/T.125 – MCSPDU MergeTokensRequest

Contenido	Fuente	Sumidero
Fusión de testigos	Antiguo superior	Proveedor superior
Purga de identificadores de testigo	Intermedios	Proveedor superior

La **MergeTokensRequest** puede ser rellena con los atributos de múltiples testigos, hasta el límite impuesto por el dominio al tamaño de las MCSPDU. Como se indica detalladamente en las definiciones ASN.1 de la cláusula 7, cada uno de los tipos de testigos que se están utilizando (tomado, inhibido, en cesión, no cedible, cedido) tiene su correspondiente conjunto de atributos. Estos atributos están contenidos en la base de información del proveedor MCS superior y están parcialmente replicados en los subárboles donde se utiliza un testigo. Cuando se fusionan dos dominios, mediante MCS-CONEXIÓN-PROVEEDOR, los testigos que se están utilizando en el dominio más bajo tienen que ser incorporados a la base de información del dominio más alto, o purgados del dominio más bajo. Esta decisión incumbe al proveedor superior del dominio fusionado.

Cada testigo deberá considerarse individualmente. Si los límites impuestos por el dominio a los testigos que se están utilizando lo permiten y no se está dando al identificador de testigo un uso conflictivo, el dominio más alto deberá expandirse hasta incluirlo. La inhibición de un identificador de testigo en el dominio más bajo no da lugar a conflicto si se inhibe también en el dominio más alto. Todas las demás combinaciones de uso simultáneo están desautorizadas, y el identificador de testigo deberá ser purgado del dominio más bajo.

Si un testigo tiene un extenso conjunto de usuarios inhibidores, sus atributos pudieran no caber en una sola **MergeTokensRequest** y deberán enviarse hacia arriba en múltiples MCSPDU. Sin embargo, la segunda petición para el mismo testigo inhibido, y las siguientes, deberán aplazarse hasta que se haya recibido una **MergeTokensConfirm** como contestación a la primera. Sólo entonces se sabrá si los límites impuestos por el dominio han permitido que se ponga en uso el testigo en el dominio más alto. Si la primera petición fracasa, no se repetirá con un subconjunto restante de inhibidores.

Cada **MergeTokensRequest** provoca que el proveedor MCS superior conteste con una **MergeTokensConfirm** en la misma secuencia. Una **MergeTokensConfirm** no contiene nada que identifica explícitamente la **MergeTokensRequest** precedente. Las contestaciones se encaminarán basándose exclusivamente en el orden en que se recibieron estas MCSPDU. Los proveedores MCS por encima del antiguo proveedor superior tomarán nota de cada **MergeTokensRequest** no contestada, y de su procedencia, de manera que se pueda retornar la correspondiente **MergeTokensConfirm** a través de la misma conexión MCS.

Los proveedores MCS intermedios validarán los identificadores de usuario anunciados en atributos de testigo, para asegurar que se asignan legítimamente al subárbol en que se origina la **MergeTokensRequest**. Se suprimirán los identificadores de usuario inválidos. Un testigo que está siendo cedido permanecerá tomado si se suprime quien lo tomó, o quien lo recibió, pero no ambos. Si como consecuencia de esta supresión de identificadores de usuario se libera un testigo, todos sus atributos deberán suprimirse de la petición de fusión y sólo el identificador de testigo se incluirá en el conjunto que va a ser purgado. Un testigo inhibido permanecerá inhibido en **MergeTokensRequest** incluso si se suprimen todos los inhibidores, dejando un conjunto vacío en los atributos, porque pueden sobrevivir inhibidores de otras MCSPDU. Salvo para esta validación de identificadores de usuario, los proveedores MCS no modificarán el contenido de una **MergeTokensRequest**.

Un antiguo proveedor superior esperará la confirmación individual de que todos los identificadores de usuario hayan sido incorporados al dominio fusionado, o purgados, antes de empezar a someter atributos de testigo para la fusión.

### 11.13 MergeTokensConfirm

Una **MergeTokensConfirm** contesta a una **MergeTokensRequest** precedente. Refleja el mismo conjunto de identificadores de testigo y un subconjunto de los atributos. Los atributos de testigo no incorporados en el dominio fusionado se comunican como identificadores de testigo para ser purgados.

**Cuadro 11-13/T.125 – MCSPDU MergeTokensConfirm**

Contenido	Fuente	Sumidero
Fusión de testigos	Proveedor superior	Intermedios
Identificadores de testigo purgados	Proveedor superior	Antiguo superior

Los identificadores de testigo aceptados se reflejan con los atributos que se introdujeron en la base de información del proveedor MCS superior. Los proveedores intermedios actualizarán su base de información para hacerla conforme.

Los testigos que han de ser purgados del dominio más bajo se listan por los identificadores solamente. Si los mismos identificadores de testigo se utilizaron en el dominio más alto, se mantienen sin modificación. Los proveedores intermedios reenviarán los identificadores de testigo purgados sin tratarlos.

Los proveedores MCS encaminarán **MergeTokensConfirm** a la fuente de la **MergeTokensRequest** precedente basándose en el conocimiento de que hay una contestación de uno a uno. **MergeTokensConfirm** retorna al antiguo proveedor superior que generó **MergeTokensRequest**. Allí los testigos fusionados pueden ser ignorados, ya que han seguido en la base de información mientras está pendiente una contestación. Los identificadores de testigo purgados se suprimirán, ya que están destinados a **PurgeTokensIndication**.

Los proveedores MCS intermedios confirmarán que los identificadores de usuario anunciados en atributos de testigo se asignan al subárbol al cual se encamina **MergeTokensConfirm**. Si algún identificador de usuario ha sido desanexionado y reasignado en algún otro lugar en el tiempo transcurrido desde que se validó la **MergeTokensRequest** precedente, un proveedor intermedio generará para ellos una **DetachUserRequest** con el código de motivo *canal purgado* haciéndolos víctimas de la fusión de dominio. Si un identificador de testigo no inhibido es liberado mediante esta supresión de identificadores de usuario, será trasladado al conjunto purgado.

### 11.14 PurgeTokensIndication

**PurgeTokensIndication** se genera en un antiguo proveedor superior tras la recepción de **MergeTokensConfirm**. Se difunde en sentido descendente y purga el uso de identificadores de testigo especificados eliminándolos de los proveedores subordinados.

**Cuadro 11-14/T.125 – MCSPDU PurgeTokensIndication**

Contenido	Fuente	Sumidero
Purga de identificadores de testigo	Antiguo superior	Subordinados

El efecto de purgar un testigo es grave: indicación MCS-DESANEXIÓN-USUARIO a todo usuario que haya tomado, inhibido, o a que se esté cediendo un identificador de testigo. Un proveedor deberá implementar esto generando **DetachUserRequest** a nombre de los usuarios influidos con el motivo *testigo purgado*.

NOTA – Se prevé que la Recomendación T.122 se revisará en el futuro para prever una indicación MCS-LIBERACIÓN-TESTIGO en esta situación. Esto permitiría al usuario influido permanecer afectado aunque se le revoque el derecho de utilizar el testigo.

### 11.15 DisconnectProviderUltimatum

**DisconnectProviderUltimatum** se genera por una petición MCS-DESCONEXIÓN-PROVEEDOR. Esta unidad genera a su vez una indicación MCS-DESCONEXIÓN-PROVEEDOR en el otro extremo de una conexión MCS. **DisconnectProviderUltimatum** obliga al receptor a desconectar la conexión MCS que la transportó.

**DisconnectProviderUltimatum** la genera también un proveedor MCS cuando detecta una condición de error como la existencia de un ciclo en la jerarquía del dominio. En tales casos el motivo es uno que no sea *solicitado por el usuario*.

**Cuadro 11-15/T.125 – MCSPDU DisconnectProviderUltimatum**

Contenido	Fuente	Sumidero
Motivo (o razón)	Proveedor solicitante	Indicación

### 11.16 RejectMCSPDUUltimatum

Se genera **RejectMCSPDUUltimatum** cuando un proveedor MCS recibe una MCSPDU inválida o detecta un error de protocolo MCS. Esta unidad invita al proveedor par en el otro extremo de una conexión MCS a desconectar, ya que una recuperación tras una situación que no debería ocurrir sería incierta.

**RejectMCSPDUUltimatum** diagnostica el error y retorna una porción inicial de la TSDU errónea, por lo general tantos octetos como quepan en una MCSPDU de tamaño máximo. El proveedor receptor puede optar entre desconectar y perseverar.

**Cuadro 11-16/T.125 – MCSPDU RejectMCSPDUUltimatum**

Contenido	Fuente	Sumidero
Diagnóstico	Proveedor rechazante	Proveedor rechazado
Octetos iniciales	Proveedor rechazante	Proveedor rechazado

### 11.17 AttachUserRequest

**AttachUserRequest** se genera por una respuesta MCS-ANEXIÓN-USUARIO. Esta unidad sube al proveedor MCS superior, que contesta con una **AttachUserConfirm**. Si el límite impuesto por el dominio al número de los identificadores de usuario lo permite, se genera un nuevo identificador de usuario.

**Cuadro 11-17/T.125 – MCSPDU AttachUserRequest**

Contenido	Fuente	Sumidero
(Ninguno)	–	–

**AttachUserRequest** sólo contiene una información: el tipo de MCSPDU. El dominio a que se afecta el usuario viene determinado por la conexión MCS que transporta la MCSPDU. La única característica inicial del identificador de usuario generado es su unicidad.

Un proveedor MCS deberá tomar nota de cada **AttachUserRequest** no contestada recibida y de la conexión MCS por la que llegó, a fin de poder contestar con una **AttachUserConfirm** dirigida a la misma fuente. Para distribuir equitativamente las contestaciones, cada proveedor deberá mantener una cola de tipo primero en entrar primero en salir, con este fin.

### 11.18 AttachUserConfirm

Se genera **AttachUserConfirm** en el proveedor MCS superior al recibir **AttachUserRequest**. Cuando se transmite en retorno al proveedor solicitante, esta unidad genera una confirmación MCS-ANEXIÓN-USUARIO.

**Cuadro 11-18/T.125 – MCSPDU AttachUserConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador (facultativo)	Proveedor superior	Confirmación

**AttachUserConfirm** contiene un identificador de usuario únicamente si el resultado es de éxito. Los proveedores que reciban una **AttachUserConfirm** de éxito introducirán el identificador de usuario en su base de información.

Los proveedores MCS encaminarán **AttachUserConfirm** a la fuente de una **AttachUserRequest** precedente basándose en el conocimiento de que hay una contestación de uno a uno. Un proveedor que transmita **AttachUserConfirm** tomará nota de la conexión MCS descendente a la que se asignó de esa forma el nuevo identificador de usuario, de modo que pueda validar el identificador de usuario cuando aparezca después en otras peticiones.

### 11.19 DetachUserRequest

**DetachUserRequest** se genera por una petición MCS-DESANEXIÓN-USUARIO. Si es válida, asciende al proveedor MCS superior, el cual suprime al usuario en su base de información y difunde **DetachUserIndication** para informar a otros proveedores sobre el cambio.

**Cuadro 11-19/T.125 – MCSPDU DetachUserRequest**

Contenido	Fuente	Sumidero
Motivo (o razón)	Proveedor solicitante	Proveedor superior
Identificadores de usuario	Proveedor solicitante	Proveedor superior

Una petición MCS-DESANEXIÓN-USUARIO genera una **DetachUserRequest** que contiene el motivo *solicitado por usuario* y un solo identificador de usuario.

**DetachUserRequest** la genera también un proveedor MCS cuando se desconecta una conexión MCS descendente. En ese punto, todos los usuarios en el subárbol influido se pierden y se informará que

han sido desanexionados dándose como motivo *dominio desconectado*. Si están pendientes asignaciones de identificadores de usuario, una ulterior contestación, sea **MergeChannelsConfirm** o **AttachUserConfirm**, podrá entonces quedar sin ruta de retorno a su fuente en el subárbol desconectado. Un proveedor en tal situación deberá también generar **DetachUserRequest** para suprimir los identificadores de usuario no asignables.

Los proveedores que reciben **DetachUserRequest** validarán los identificadores de usuario contenidos en dicha unidad para asegurar que son legítimamente asignados al subárbol de origen. Los identificadores de usuario inválidos se suprimirán. Si no quedan identificadores de usuario, la **DetachUserRequest** deberá ignorarse.

Los identificadores de usuario contenidos en **DetachUserRequest** no se suprimirán en la base de información hasta que el proveedor reciba **DetachUserIndication**. De esta forma se mantiene la coherencia con el proveedor MCS superior.

NOTA – Si en un dominio MCS se aplica más de una prioridad de datos, **DetachUserIndication** podría llegar a un proveedor dado antes que datos enviados con anterioridad al mismo usuario, pero con una prioridad más baja. Este protocolo no impide que se entreguen datos a una anexión incluso después de que se haya comunicado mediante una **DetachUserIndication** que el emisor se había desanexionado.

### 11.20 DetachUserIndication

Se genera **DetachUserIndication** en el proveedor MCS superior tras la recepción de **DetachUserRequest**. Dicha unidad se difunde hacia abajo a todos los demás proveedores y genera indicaciones MCS-DESANEXIÓN-USUARIO en todas las anexiones.

En una anexión subsistente, **DetachUserIndication** genera una MCS-DESANEXIÓN-USUARIO para cada identificador de usuario que ella contiene. No importa que el usuario notificado haya tenido o no conocimiento previo de la existencia de un usuario desanexionado.

Al recibir una **DetachUserIndication** que contiene su propio identificador de usuario, una anexión MCS deja de existir. Cualesquiera canales que dejen de estar incorporados como resultado de la desanexión de un usuario, con excepción del propio canal de identificador de usuario, será abandonado mediante **ChannelLeaveRequest**.

**Cuadro 11-20/T.125 – MCSPDU DetachUserIndication**

Contenido	Fuente	Sumidero
Motivo (o razón)	Proveedor superior	Indicación
Identificadores de usuario	Proveedor superior	Indicación

Los proveedores que reciben **DetachUserIndication** suprimirán los identificadores de usuario especificados en su base de información, con independencia de que sea gestor o usuario admitido de un canal probado.

Los canales gestionados por un usuario desanexionado se suprimirán si no hay otros usuarios admitidos. Si quedan otros usuarios, la supresión del gestor tendrá como consecuencia que el proveedor superior les envíe una **ChannelDisbandIndication** en multidifusión. Si el conjunto de usuarios admitidos en un canal privado es nulo y el gestor no reside en el subárbol, el identificador de canal debe ser suprimido de la base de información. En cualquier otro caso, si se desincorpora un canal privado como consecuencia de la desanexión de usuarios, un proveedor generará el correspondiente **ChannelLeaveRequest**.

Un proveedor difunda **DetachUserIndication** calculará, para cada uno de los canales privados afectados y para cada subárbol de destino, si el subárbol siguiente contiene cualquiera de los anexos

admitidos en el canal privado. Si no contiene ninguno, el proveedor concluirá que el correspondiente proveedor subordinado no se incorporará en lo sucesivo al canal privado y a tal efecto actualizará su base de información de forma inmediata, sin esperar un **ChannelLeaveRequest**.

El estado de todo testigo que haya sido tomado o inhibido por un usuario desanexionado, o que se le esté cediendo, deberá ajustarse correspondientemente. La supresión de un receptor previsto de testigo provocará que el proveedor superior envíe una **TokenGivenConfirm** de fracaso al cedente del testigo, a menos que haya liberado el testigo o se haya desanexionado a sí mismo.

### 11.21 ChannelJoinRequest

**ChannelJoinRequest** se genera por una petición MCS-INCORPORACIÓN-CANAL. Si es válida, sube hasta llegar a un proveedor MCS con información suficiente para generar una contestación **ChannelJoinConfirm**. Este puede ser el proveedor MCS superior.

**Cuadro 11-21/T.125 – MCSPDU ChannelJoinRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor más alto
Identificador de canal	Petición	Proveedor más alto

El identificador de usuario de la anexión MCS iniciadora la suministra el proveedor MCS que recibe la petición primitiva. Los proveedores que reciben **ChannelJoinRequest** ulteriormente validarán el identificador de usuario para asegurar que sea legítimamente asignado al subárbol de origen. Si el identificador de usuario es inválido se ignorará la MCSPDU.

NOTA – Con esto se tiene en cuenta la posibilidad de que **ChannelJoinRequest** compita en una especie de carrera hacia arriba con una purga de identificador de usuario iniciador que estuviera bajando. Un proveedor que recibe **PurgeChannelsIndication** primero podría recibir poco después una **ChannelJoinRequest** que contiene un identificador de usuario inválido. Este es un suceso normal y no es motivo para rechazar la MCSPDU.

**ChannelJoinRequest** puede subir hasta un proveedor MCS que tiene en su base de información el identificador de canal solicitado. Cualquier proveedor en tal situación, al actuar conformemente con el proveedor MCS superior, deberá convenir en que la petición debe o no tener éxito. Si la petición debe fracasar, el proveedor generará una **ChannelJoinConfirm** de fracaso. Si debe tener éxito y el proveedor ya se ha incorporado al mismo canal, generará una **ChannelJoinConfirm** de éxito. En estos dos casos, MCS-INCORPORACIÓN-CANAL se completa sin que sea necesario llegar al proveedor MCS superior. En otro caso, si la petición debe tener éxito pero todavía no se ha efectuado la incorporación al canal, un proveedor deberá reenviar **ChannelJoinRequest** hacia arriba.

Si **ChannelJoinRequest** sube hasta el proveedor MCS superior, el identificador del canal solicitado podría ser cero, valor que no se encuentra en ninguna base de información porque corresponde a un identificador inválido. Si lo permite el límite impuesto por el dominio al número de canales en uso, deberá generarse un nuevo identificador de canal asignado, y retornarse en una **ChannelJoinConfirm** de éxito. Si el identificador del canal solicitado está en la gama estática y el límite impuesto por el dominio al número de canales en uso lo permite, el identificador de canal se introducirá en la base de información y se retornará asimismo en una **ChannelJoinConfirm** de éxito.

En los demás casos, la petición sólo tendrá éxito si el identificador del canal se encuentra ya en la base de información del proveedor MCS superior. A un identificador de canal de usuario sólo puede incorporarse el mismo usuario. A un identificador de canal privado sólo pueden incorporarse los

usuarios que hayan sido previamente admitidos por su gestor. A un identificador de canal asignado puede incorporarse cualquier usuario.

### 11.22 ChannelJoinConfirm

**ChannelJoinConfirm** se genera en un proveedor MCS más alto al recibirse **ChannelJoinRequest**. Cuando se devuelve al proveedor solicitante genera una confirmación MCS-INCORPORACIÓN-CANAL.

**Cuadro 11-22/T.125 – MCSPDU ChannelJoinConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor más alto	Confirmación
Iniciador	Proveedor más alto	Encaminamiento MCSPDU
Solicitado	Proveedor más alto	Confirmación
Identificador de canal (facultativo)	Proveedor más alto	Confirmación

**ChannelJoinConfirm** contiene un identificador de canal incorporado solamente si el resultado es éxito.

La petición de identificador de canal es igual que en **ChannelJoinRequest**. Esto ayuda a la anexión iniciadora a relacionar la confirmación MCS-INCORPORACIÓN-CANAL con una petición precedente. Como **ChannelJoinRequest** no tiene que subir al proveedor superior, las confirmaciones se pueden producir fuera de orden.

Si el resultado es éxito, **ChannelJoinConfirm** incorpora el proveedor MCS receptor al canal especificado. De allí en adelante, los proveedores más altos le encaminarán todo dato que los usuarios envíen por el canal. Un proveedor permanecerá incorporado al canal mientras lo esté cualquiera de sus anexionaciones o proveedores subordinados. Para abandonar el canal un proveedor deberá generar **ChannelLeaveRequest**.

Los proveedores que reciben una **ChannelJoinConfirm** de éxito introducirán el identificador de canal en su base de información. Si no está ya allí, se dará al identificador de canal el tipo estático o asignado, de acuerdo con su rango.

**ChannelJoinConfirm** se reenviará en el sentido de transmisión del identificador del usuario iniciador. Si el identificador de usuario no puede obtenerse porque ya no existe una conexión MCS, el proveedor decidirá si hay o no motivo para permanecer incorporado al canal. Si no lo hay, generará **ChannelLeaveRequest**.

### 11.23 ChannelLeaveRequest

**ChannelLeaveRequest** la genera un proveedor para retirarse de un conjunto de canales. Esto puede ser motivado por una petición MCS-ABANDONO-CANAL procedente de la última anexión incorporada al canal. **ChannelLeaveRequest** continúa subiendo si los proveedores más altos, en consecuencia, dejan también de tener motivo para estar incorporados.

Los proveedores que reciben **ChannelLeaveRequest** detendrán los encaminamientos a la conexión MCS que le transportó cualesquiera datos enviados por usuarios a través de los canales especificados. Cuando la última anexión o proveedor subordinado abandona un canal, un proveedor MCS generará una **ChannelLeaveRequest** correspondiente.

**Cuadro 11-23/T.125 – MCSPDU ChannelLeaveRequest**

Contenido	Fuente	Sumidero
Identificadores de canal	Proveedor solicitante	Proveedor más alto

#### 11.24 ChannelConveneRequest

**ChannelConveneRequest** se genera por una petición MCS-FORMACIÓN-CANAL. Si es válida, sube al proveedor MCS superior, el cual contesta con una **ChannelConveneConfirm**. Si el límite impuesto por el dominio al número de identificadores de canal lo permite, se genera un nuevo identificador de canal privado.

**ChannelConveneRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

El solicitante deviene gestor del canal privado. Inicialmente nadie está incorporado al canal y su gestor es el único usuario admitido.

**Cuadro 11-24/T.125 – MCSPDU ChannelConveneRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior

#### 11.25 ChannelConveneConfirm

**ChannelConveneConfirm** se genera en el proveedor MCS superior al recibirse **ChannelConveneRequest**. Cuando se devuelve al proveedor solicitante, genera una confirmación MCS-FORMACIÓN-CANAL.

**Cuadro 11-25/T.125 – MCSPDU ChannelConveneConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador	Proveedor superior	Encaminamiento MCSPDU
Identificador de canal (facultativo)	Proveedor superior	Confirmación

**ChannelConveneConfirm** contiene un identificador de canal privado sólo si el resultado es satisfactorio.

Los proveedores que reciben una **ChannelConveneConfirm** de éxito introducirán el identificador de canal en su base de información como un canal privado con el usuario iniciador como su gestor.

**ChannelConveneConfirm** se reenviará en el sentido de ida del identificador del usuario iniciador. Si el identificador de usuario no puede alcanzarse porque ya no existe una conexión MCS, no es necesario ejecutar acciones especiales, ya que, de todas formas, tiene que llegar posteriormente una **DetachUserIndication** para comunicar que el iniciador se ha desanexionado. Dado que el iniciador es su gestor, queda con esto suprimido el identificador de canal en la base de información.

## 11.26 ChannelDisbandRequest

**ChannelDisbandRequest** se genera por una petición MCS-DISOLUCIÓN-CANAL. Si es válida sube al proveedor MCS superior, el cual suprime el identificador de canal y genera **ChannelDisbandIndication**.

**Cuadro 11-26/T.125 – MCSPDU ChannelDisbandRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de canal	Petición	Proveedor superior

**ChannelDisbandRequest** puede ser generada también por un proveedor MCS por iniciativa propia para disolver un canal.

**ChannelDisbandRequest** contiene el identificador del usuario iniciador, que debe ser validado para asegurar que será legítimamente asignado al subárbol de origen. Si el iniciador no coincide con el gestor del canal privado, tal como está registrado en la base de información, se ignorará la MCSPDU.

## 11.27 ChannelDisbandIndication

**ChannelDisbandIndication** se genera en el proveedor MCS superior al recibirse **ChannelDisbandRequest**. Esta unidad se envía hacia abajo, por multidifusión, a los proveedores que contienen el gestor o un usuario admitido en su subárbol. Genera indicaciones MCS-EXCLUSIÓN-CANAL a los usuarios admitidos con el motivo *canal disuelto*.

**Cuadro 11-27/T.125 – MCSPDU ChannelDisbandIndication**

Contenido	Fuente	Sumidero
Identificador de canal	Proveedor superior	Indicación

**ChannelDisbandIndication** deberá ser generada también por el proveedor MCS superior cuando se desanexiona el gestor de un canal privado.

Los proveedores que reciben **ChannelDisbandIndication** suprimirán el canal en su base de información.

## 11.28 ChannelAdmitRequest

**ChannelAdmitRequest** se genera por una petición MCS-ADMISIÓN-CANAL. Si es válida, sube al proveedor MCS superior, el cual admite en el canal privado los usuarios especificados y envía **ChannelAdmitIndication** en multidifusión para avisar a los proveedores en cuyo subárbol residen.

**ChannelAdmitRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelDisbandRequest**.

Los otros identificadores de usuario de **ChannelAdmitRequest**, que representan usuarios que van a admitirse, se validarán en el proveedor MCS superior, que es el único que conoce el conjunto completo de los usuarios. Los que no sean válidos se excluirán de la **ChannelAdmitIndication** resultante.

**Cuadro 11-28/T.215 – MCSPDU ChannelAdmitRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de canal	Petición	Proveedor superior
Identificadores de usuario	Petición	Proveedor superior

Los identificadores de usuario contenidos en **ChannelAdmitRequest** no serán admitidos al canal privado hasta que un proveedor reciba **ChannelAdmitIndication**. De este modo se mantiene la coherencia con el proveedor MCS superior.

### 11.29 ChannelAdmitIndication

**ChannelAdmitIndication** se genera en el proveedor MCS superior tras la recepción de **ChannelAdmitRequest**. Esta unidad se envía hacia abajo por multidifusión a los proveedores que contienen en su subárbol un usuario recientemente admitido. Genera indicaciones MCS-ADMISIÓN-CANAL en las anexiones influidas.

Los proveedores que reciben **ChannelAdmitIndication** actualizarán normalmente el canal en su base de información, admitiendo a los usuarios especificados que residen en su subárbol. Sin embargo, si un proveedor es el superior anterior de un dominio más bajo que se está fusionando como resultado de MCS-CONEXIÓN-PROVEEDOR, puede rechazar la admisión generando **DetachUserRequest** para los identificadores de usuario anexionados con el motivo *canal purgado*.

**Cuadro 11-29/T.125 – MCSPDU ChannelAdmitIndication**

Contenido	Fuente	Sumidero
Iniciador	Proveedor superior	Indicación
Identificador de canal	Proveedor superior	Indicación
Identificadores de usuario	Proveedor superior	Encaminamiento MCSPDU

### 11.30 ChannelExpelRequest

**ChannelExpelRequest** se genera por una petición MCS-EXCLUSIÓN-CANAL. Si es válida sube al proveedor MCS superior, quien excluye del canal los usuarios especificados y multidifunde **ChannelExpelIndication** para avisar a los proveedores en cuyo subárbol residen.

**ChannelExpelRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelDisbandRequest**.

Los otros identificadores de usuario de **ChannelExpelRequest**, que representan usuarios que habrán de ser excluidos, serán validados en el proveedor MCS superior, que es el único que conoce el conjunto completo de los usuarios admitidos. Los que hayan sido admitidos no serán incluidos en la **ChannelExpelIndication** resultante.

Los identificadores de usuario contenidos en **ChannelExpelRequest** no serán excluidos del canal privado hasta que un proveedor reciba **ChannelExpelIndication**. De esta forma se mantiene la coherencia con el proveedor MCS superior.

**Cuadro 11-30/T.125 – MCSPDU ChannelExpelRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de canal	Petición	Proveedor superior
Identificadores de usuario	Petición	Proveedor superior

### 11.31 ChannelExpelIndication

**ChannelExpelIndication** se genera en el proveedor MCS superior al recibirse **ChannelExpelRequest**. Se envía hacia abajo por multidifusión a los proveedores que contienen en su subárbol un usuario excluido. Genera indicaciones MCS-EXCLUSIÓN-CANAL en las anexiones influidas, con el motivo solicitado *por usuario*.

**Cuadro 11-31/T.125 – MCSPDU ChannelExpelIndication**

Contenido	Fuente	Sumidero
Identificador de canal	Proveedor superior	Indicación
Identificadores de usuario	Proveedor superior	Encaminamiento MCSPDU

Los proveedores que reciben **ChannelExpelIndication** actualizarán el canal en su base de información, suprimiendo los usuarios especificados en el conjunto de usuarios admitidos en el canal. Si el conjunto de usuarios admitidos en un canal privado se vacía y el gestor no reside en el subárbol, se suprimirá el identificador de canal en la base de información. En otro caso, si como resultado de las exclusiones no queda nadie incorporado al canal, un proveedor generará una **ChannelLeaveRequest** correspondiente.

Un proveedor que reenvía **ChannelExpelIndication** determinará, para cada subárbol de destino, si, después de ello, contiene cualesquiera anexiones admitidas en el canal privado. Si no hay ninguna, el proveedor deberá llegar a la conclusión de que el proveedor subordinado correspondiente ya no está incorporado al canal privado y actualizará su base de información inmediatamente a este efecto, sin esperar **ChannelLeaveRequest**.

### 11.32 SendDataRequest

**SendDataRequest** se genera por una petición MCS-ENVÍO-DATOS. Si es válida, sube hasta el proveedor MCS superior. A lo largo del camino, los proveedores pueden generar a partir de ella una **SendDataIndication** con idéntico contenido, y multidifundirla hacia abajo.

**SendDataRequest** contiene el identificador del usuario iniciador, que será validado como se ha explicado para **ChannelJoinRequest**.

Si el identificador de canal figura en la base de información del proveedor MCS superior como un canal privado y el iniciador de **SendDataRequest** no es un usuario admitido, se ignorará la MCSPDU.

**SendDataRequest** contiene el nivel de fiabilidad que ha sido especificado en la petición original MCS-ENVIO-DATOS. Si los datos no son fiables, también se especifica el identificador de referencia del dominio.

La TC inicial o adicional que transporta **SendDataRequest** cotejará su prioridad de datos, teniendo en cuenta el número de prioridades implementadas en el dominio. Las MCSPDU que llegan a través de una conexión MCS por la TC errónea deberán rechazarse.

**Cuadro 11-32/T.125 – MCSPDU SendDataRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor más alto
Identificador de canal	Petición	Proveedor más alto
Fiabilidad	Petición	Proveedor más alto
ID de referencia del dominio	Petición	Proveedor más alto
Prioridad de datos	Petición	Proveedor más alto
Segmentación	Proveedor solicitante	Proveedor más alto
Tamaño total de datos	Proveedor solicitante	Proveedor más alto
Datos de usuario	Petición	Proveedor más alto

Un proveedor fijará las banderas de segmentación *comienzo* y *fin* para mostrar la relación de los datos de usuario en la **SendDataRequest** con los confines de una unidad de datos de servicio MCS. Los proveedores están en libertad de fragmentar y reensamblar las MCSPDU que forman parte de la misma unidad de datos de servicio MCS, mientras esto no perturbe la integridad de datos de usuario. Sin embargo, será poco el beneficio que se obtenga de tal manipulación, pues el tamaño máximo de una MCSPDU es constante en la totalidad de un dominio.

Para una determinada **SendDataRequest**, si la bandera de *comienzo* de segmentación es verdadera y si la bandera de *fin* es falsa, se proporciona el tamaño total de datos de los datos de usuario segmentados. Este campo sólo estará presente si los nodos fuente y sumidero son miembros de la cumbre V3.

Un proveedor generará, a partir de **SendDataRequest**, una **SendDataIndication** con el mismo contenido y la transmitirá a todos los proveedores que se hayan incorporado al canal especificado, excepto al proveedor subordinado que transmitió **SendDataRequest** hacia arriba. A menos que el canal esté listado en la base de información del proveedor como un identificador de usuario residente en su subárbol, reenviará también **SendDataRequest** hacia arriba.

### 11.33 SendDataIndication

Se genera **SendDataIndication** en un proveedor MCS más alto al recibirse **SendDataRequest**. Esta unidad se multidifunde hacia abajo y genera indicaciones MCS-ENVÍO-DATOS en todas las anexiones que están incorporadas al canal.

**Cuadro 11-33/T.125 – MCSPDU SendDataIndication**

Contenido	Fuente	Sumidero
Iniciador	Proveedor más alto	Indicación
Identificador de canal	Proveedor más alto	Indicación
Fiabilidad	Proveedor más alto	Indicación
Identificador de referencia del dominio	Proveedor más alto	Indicación
Prioridad de datos	Proveedor más alto	Indicación
Segmentación	Proveedor más alto	Proveedor indicante
Tamaño total de datos	Proveedor más alto	Proveedor indicante
Datos de usuario	Proveedor más alto	Indicación

La TC inicial o adicional que transporta **SendDataIndication** cotejará su prioridad de datos, teniendo en cuenta el número de prioridades implementadas en el dominio. Las MCSPDU que llegan a través de una conexión MCS por la TC errónea deberán rechazarse.

Las banderas de segmentación *comienzo* y *fin* permiten reensamblar datos de usuario para formar una unidad de datos de servicio MCS completa. Estas banderas se interpretarán en el contexto de las MCSPDU **SendDataIndication** que llegan del mismo usuario a través del mismo canal y con la misma prioridad. Un flujo de fragmentos a reensamblar pueden estar entrelazados con otras MCSPDU y datos de otros usuarios transmitidos por otros canales y con otras prioridades.

Para una **SendDataIndication** dada, se proporciona el tamaño total de datos de los datos segmentados del usuario siempre que la bandera de *comienzo* de segmentación sea verdadera y la bandera de *fin* sea falsa. Este campo sólo estará presente si los nodos fuente y sumidero son miembros de la cumbre V3.

La manera de indicar las unidades de datos de servicio a los usuarios MCS anexionados es una cuestión de implementación local. Una posible solución consiste en entregar cada MCSPDU como una unidad de datos de interfaz separada, con banderas de segmentación incluidas. En otras soluciones, se puede tratar de reensamblar dentro del proveedor receptor, deben prever el caso de las unidades de datos de servicio de gran tamaño y dar a conocer al usuario el orden relativo en que comienzan a llegar las unidades de datos de servicio.

Los proveedores que reciben **SendDataIndication** la reenviarán a todos los subordinados que están incorporados al canal.

### 11.34 UniformSendDataRequest

**UniformSendDataRequest** se genera por una petición MCS-ENVÍO-DATOS-UNIFORME. Si es válida, sube al proveedor MCS superior, el cual, basándose en ella, genera una **UniformSendDataIndication** de contenido idéntico y la multidifunde hacia abajo.

**Cuadro 11-34/T.125 – MCSPDU UniformSendDataRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de canal	Petición	Proveedor superior
Fiabilidad	Petición	Proveedor más alto
ID de referencia del dominio	Petición	Proveedor más alto
Prioridad de datos	Petición	Proveedor superior
Segmentación	Proveedor solicitante	Proveedor superior
Tamaño total de datos	Proveedor solicitante	Proveedor más alto
Datos de usuario	Petición	Proveedor superior

**UniformSendDataRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

Si el identificador de canal figura en la base de información del proveedor MCS receptor como un canal privado y el iniciador de **UniformSendDataRequest** no es un usuario admitido, se ignorará la MCSPDU.

La TC inicial o adicional que transporta **UniformSendDataRequest** cotejará su prioridad de datos, teniendo en cuenta el número de prioridades implementadas en el dominio. Las MCSPDU que llegan a través de una conexión MCS por la TC errónea deberán rechazarse.

Un proveedor fijará las banderas de segmentación *comienzo* y *fin* para mostrar la relación de datos de usuario en la **UniformSendDataRequest** con los confines de una unidad de datos de servicio MCS. Los proveedores están en libertad de segmentar y reensamblar las MCSPDU que forman parte de la misma unidad de datos de servicio MCS mientras esto no perturbe la integridad de datos de usuario. Sin embargo, será poco el beneficio que se obtenga de tal manipulación, pues el tamaño máximo de una MCSPDU es constante en la totalidad de un dominio.

Para un determinado **UniformSendDataRequest**, se proporciona el tamaño total de datos de los datos segmentados del usuario siempre que la bandera de *comienzo* de segmentación sea verdadera y la bandera de *fin* sea falsa. Este campo sólo estará presente si los nodos fuente y sumidero son miembros de la cumbre V3

Basándose en **UniformSendDataRequest**, el proveedor MCS superior generará una **UniformSendDataIndication** con el mismo contenido.

### 11.35 UniformSendDataIndication

**UniformSendDataIndication** se genera en el proveedor MCS superior al recibirse **UniformSendDataRequest**. Esta unidad se multidifunde hacia abajo y genera indicaciones MCS-ENVÍO-DATOS-UNIFORME en todas las anexiones que están incorporadas al canal.

**Cuadro 11-35/T.125 – MCSPDU UniformSendDataIndication**

Contenido	Fuente	Sumidero
Iniciador	Proveedor superior	Indicación
Identificador de canal	Proveedor superior	Indicación
Fiabilidad	Proveedor más alto	Indicación
Identificador de referencia del dominio	Proveedor más alto	Indicación
Prioridad de datos	Proveedor superior	Indicación
Segmentación	Proveedor superior	Proveedor indicante
Tamaño total de datos	Proveedor más alto	Proveedor indicante
Datos de usuario	Proveedor superior	Indicación

La TC inicial o adicional que transporta **UniformSendDataIndication** cotejará su prioridad de datos, teniendo en cuenta el número de prioridades implementadas en el dominio. Las MCSPDU que llegan a través de una conexión MCS por la TC errónea deberán rechazarse.

Las banderas de segmentación *comienzo* y *fin* permiten reensamblar datos de usuario para formar una unidad de datos de servicio MCS completa. Estas banderas se interpretarán en el contexto de las MCSPDU **UniformSendDataIndication** que llegan del mismo usuario a través del mismo canal y con la misma prioridad. Un flujo de fragmentos a reensamblar pueden entrelazarse con otras MCSPDU y datos de otros usuarios transmitidos por otros canales y con otras prioridades.

Para un determinado **UniformSendDataIndication**, se proporciona el tamaño total de datos de los datos segmentados del usuario siempre que la bandera de *comienzo* de segmentación sea verdadera y la bandera de *fin* sea falsa. Este campo sólo estará presente si los nodos fuente y sumidero son miembros de la cumbre V3.

La manera de indicar las unidades de datos a los usuarios MCS anexionados es una cuestión de implementación local.

Los proveedores que reciben **UniformSendDataIndication** la reenviarán a todos los subordinados que estén incorporados al canal.

### 11.36 TokenGrabRequest

**TokenGrabRequest** se genera por una petición MCS-TOMA-TESTIGO. Si es válida, sube al proveedor MCS, el que contesta con una **TokenGrabConfirm**.

**Cuadro 11-36/T.125 – MCSPDU TokenGrabRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior

**TokenGrabRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

Si el testigo está libre y el límite impuesto por el dominio al número de testigos lo autoriza, será tomado. Si el testigo está inhibido por el usuario solicitante solamente, será tomado. En otro caso, el estado del testigo no cambiará.

### 11.37 TokenGrabConfirm

**TokenGrabConfirm** se genera en el proveedor MCS superior al recibirse **TokenGrabRequest**. Cuando ha sido devuelta al proveedor MCS solicitante genera una confirmación MCS-TOMA-TESTIGO.

**Cuadro 11-37/T.125 – MCSPDU TokenGrabConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador	Proveedor superior	Encaminamiento MCSPDU
Identificador de testigo	Proveedor superior	Confirmación
Estado de testigo	Proveedor superior	Confirmación

El resultado será *éxito* si el testigo ya estaba libre o si el mismo usuario cambió su estado de inhibido a tomado. Otros resultados son *demasiados testigos* y *testigo no disponible*. Este último se aplica a un testigo ya tomado por el solicitante; esto se determina examinando el estado del testigo.

Los proveedores que reciben **TokenGrabConfirm** actualizarán el estado del testigo en la base información para que concuerde con el estado retornado.

Se reenviará **TokenGrabConfirm** en el mismo sentido que el identificador del usuario iniciador. Si el identificador del usuario no puede obtenerse porque ya no existe una conexión MCS, no es necesario ejecutar acciones especiales, ya que tiene que llegar posteriormente una **DetachUserIndication** para informar que el iniciador se ha desanexionado. Este liberará su control sobre el identificador de testigo en su base de información.

### 11.38 TokenInhibitRequest

**TokenInhibitRequest** se genera por una petición MCS-INHIBICIÓN-TESTIGO. Si es válida, sube al proveedor MCS superior, el cual contesta con una **TokenInhibitConfirm**.

**Cuadro 11-38/T.125 – MCSPDU TokenInhibitRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior

**TokenInhibitRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

Si el testigo está libre y el límite impuesto por el dominio al número de testigos lo autoriza, será inhibido. Si el testigo está tomado por el usuario solicitante, será inhibido. Si el testigo está ya inhibido, se añadirá el solicitante al conjunto de inhibidores. En otro caso, el estado del testigo no cambiará.

### 11.39 TokenInhibitConfirm

**TokenInhibitConfirm** se genera en el proveedor MCS superior al recibirse **TokenInhibitRequest**. Cuando ha sido devuelta al proveedor solicitante, genera una confirmación MCS-INHIBICIÓN-TESTIGO.

El resultado será *éxito* si el testigo ya estaba libre o inhibido, o si el mismo usuario cambió su estado de tomado a inhibido. Otros resultados son *demasiados testigos* y *testigo no disponible*.

**Cuadro 11-39/T.125 – MCSPDU TokenInhibitConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador	Proveedor superior	Encaminamiento MCSPDU
Identificador de testigo	Proveedor superior	Confirmación
Estado del testigo	Proveedor superior	Proveedor intermedio

Los proveedores que reciben **TokenInhibitConfirm** actualizarán el estado del testigo en la base información para que concuerde con el estado retornado.

Esta MCSPDU se encamina de la misma forma que **TokenGrabConfirm**.

### 11.40 TokenGiveRequest

**TokenGiveRequest** se genera por una petición MCS-CESIÓN-TESTIGO. Si es válida sube al proveedor MCS superior, el cual genera **TokenGiveIndication** o una **TokenGiveConfirm** de fracaso.

**Cuadro 11-40/T.125 – MCSPDU TokenGiveRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior
Receptor	Petición	Proveedor superior

**TokenGiveRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

Si el solicitante ha tomado el testigo y el receptor deseado existe, se transmite **TokenGiveIndication** hacia el receptor. De no ser así, la petición fracasará, el estado del testigo se mantendrá sin cambiar, y se transmitirá **TokenGiveConfirm** hacia el solicitante con el resultado *testigo no poseído* o *no hay tal usuario*.

#### 11.41 TokenGiveIndication

Se genera **TokenGiveIndication** en el proveedor MCS superior al recibirse **TokenGiveRequest**. Cuando se ha encaminado al receptor deseado, genera una indicación MCS-CESIÓN-TESTIGO.

Los proveedores que reciben **TokenGiveIndication** por lo general actualizan el identificador del testigo en su base de información como en curso de cesión del iniciador al receptor. Si embargo, si un proveedor es el antiguo superior de un dominio más bajo que se está fusionando como resultado de MCS-CONEXIÓN-PROVEEDOR, puede rechazar el testigo ofrecido generando **TokenGiveResponse** con el motivo *fusión de dominio*.

**Cuadro 11-41/T.125 – MCSPDU TokenGiveIndication**

Contenido	Fuente	Sumidero
Iniciador	Proveedor superior	Indicación
Identificador de testigo	Proveedor superior	Indicación
Receptor	Proveedor superior	Encaminamiento MCSPDU

Se reenviará **TokenGiveIndication** en el sentido del identificador del usuario receptor. Si el identificador de usuario no puede obtenerse porque ya no existe una conexión MCS, no es necesario ejecutar acciones especiales, ya que ulteriormente tiene que llegar una **DetachUserIndication** para informar que el receptor se ha desanexionado. Este liberará su control sobre el identificador de testigo en la base de información.

#### 11.42 TokenGiveResponse

Se genera **TokenGiveResponse** por una respuesta MCS-CESIÓN-TESTIGO. Si es válida sube al proveedor MCS superior, el cual genera una **TokenGiveConfirm** para comunicar el resultado al cedente del testigo.

**Cuadro 11-42/T.125 – MCSPDU TokenGiveResponse**

Contenido	Fuente	Sumidero
Resultado	Respuesta	Proveedor superior
Receptor	Proveedor respondedor	Proveedor superior
Identificador de testigo	Proveedor respondedor	Proveedor superior

Un resultado de *éxito* significa que el receptor ha aceptado el testigo ofrecido.

El identificador de usuario de la anexión MCS respondedora lo suministra el proveedor MCS que recibe la respuesta de primitiva. Los proveedores que reciben **TokenGiveResponse** subsiguientemente validarán el identificador de usuario para asegurar que se asigna legítimamente al subárbol de origen. Si el identificador de usuario no es válido se ignorará la MCSPDU.

Si el identificador de testigo no figura en la base de información del proveedor como en curso de cesión al receptor, se ignorará la MCSPDU. Si el identificador de testigo está todavía tomado por el cedente, su estado se actualizará como tomado por el receptor si el resultado es de éxito; en otro caso, el estado volverá a ser el de tomado por el cedente, o se suprimirá en la base de información, lo que dependerá de que el cedente resida o no en el subárbol del proveedor. Si, después de eso, el identificador de testigo ha sido liberado por el cedente y el resultado no es de éxito, se suprimirá el testigo en la base de información del proveedor.

Si la MCSPDU es válida y no ha sido ignorada, se reenviará hacia arriba. El proveedor MCS superior tratará la **TokenGiveResponse** como se ha especificado antes. Además, si el cedente no ha liberado ya el testigo, el proveedor superior generará una **TokenGiveConfirm** que contendrá el mismo resultado que **TokenGiveResponse**.

### 11.43 TokenGiveConfirm

Se genera **TokenGiveConfirm** en el proveedor MCS superior al recibirse **TokenGiveResponse**. Cuando se devuelve al proveedor solicitante, genera una confirmación MCS-CESIÓN-TESTIGO.

**Cuadro 11-43/T.125 – MCSPDU TokenGiveConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador	Proveedor superior	Encaminamiento MCSPDU
Identificador de testigo	Proveedor superior	Confirmación
Estado del testigo	Proveedor superior	Proveedores intermedios

**TokenGiveConfirm** la genera también el proveedor MCS superior al recibir **TokenGiveRequest** si no se puede ofrecer un testigo al receptor deseado. Esto se hace en lugar de generar **TokenGiveIndication**. **TokenGiveConfirm** se generará asimismo con el resultado *no hay tal usuario* si se desanexiona el receptor antes de recibirse **TokenGiveResponse**.

Los proveedores que reciben **TokenGiveConfirm** actualizarán el estado del testigo en su base de información para que concuerde con el estado retornado.

Esta MCSPDU se encamina de la misma manera que **TokenGrabConfirm**.

#### 11.44 TokenPleaseRequest

**TokenPleaseRequest** se genera por una petición MCS-SOLICITUD-TESTIGO. Si es válida, sube al proveedor MCS superior, el cual multidifunde **TokenPleaseIndication** para avisar a los usuarios actuales con relación al testigo.

**TokenPleaseRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

**Cuadro 11-44/T.125 – MCSPDU TokenPleaseRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior

#### 11.45 TokenPleaseIndication

**TokenPleaseIndication** se genera en el proveedor MCS superior al recibir **TokenPleaseRequest**. Esta unidad se multidifunde hacia abajo y genera indicaciones MCS-SOLICITUD-TESTIGO.

Los proveedores que reciben **TokenPleaseIndication** la reenviarán a todos los subordinados que contienen en su subárbol un usuario que ha tomado o inhibido el testigo especificado, o al que se le está cediendo éste.

**Cuadro 11-45/T.125 – MCSPDU TokenPleaseIndication**

Contenido	Fuente	Sumidero
Iniciador	Proveedor superior	Indicación
Identificador de testigo	Proveedor superior	Indicación

#### 11.46 TokenReleaseRequest

**TokenReleaseRequest** se genera por una petición MCS-LIBERACIÓN-TESTIGO. Si es válida, sube al proveedor MCS superior, que la contesta retornando **TokenReleaseConfirm**.

**Cuadro 11-46/T.125 – MCSPDU TokenReleaseRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior

**TokenReleaseRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

Si el testigo es tomado por el solicitante, deberá quedar libre. Si es inhibido, el solicitante deberá suprimirse en el conjunto de inhibidores; si este conjunto se vacía, el testigo deberá quedar libre. Si el testigo está en curso de dejar de ser poseído por el solicitante, pasará al estado de cedido al receptor deseado, que es un estado intermedio distinto, mientras está pendiente la recepción de **TokenGiveResponse**. En otro caso, el estado del testigo no cambia.

#### 11.47 TokenReleaseConfirm

Se genera **TokenReleaseConfirm** en el proveedor MCS superior al recibirse **TokenReleaseRequest**. Cuando se devuelve al proveedor solicitante, genera una confirmación MCS-LIBERACIÓN-TESTIGO.

**Cuadro 11-47/T.125 – MCSPDU TokenReleaseConfirm**

Contenido	Fuente	Sumidero
Resultado	Proveedor superior	Confirmación
Iniciador	Proveedor superior	Encaminamiento MCSPDU
Identificador de testigo	Proveedor superior	Confirmación
Estado del testigo	Proveedor superior	Proveedores intermedios

El resultado será de *éxito* si el solicitante tomó o inhibió el testigo, o estaba en trámite de cederlo. El otro resultado posible es *testigo no poseído*.

Los proveedores que reciben **TokenReleaseConfirm** actualizarán el testigo en su base de información para que concuerde con el estado retornado.

Esta MCSPDU se encamina de la misma forma que **TokenGrabConfirm**.

#### 11.48 TokenTestRequest

**TokenTestRequest** se genera por una petición MCS-PRUEBA-TESTIGO. Si es válida, sube al proveedor MCS superior, que la contesta retornando **TokenTestConfirm**.

**TokenTestRequest** contiene el identificador del usuario iniciador, que se validará como se ha explicado para **ChannelJoinRequest**.

**Cuadro 11-48/T.125 – MCSPDU TokenTestRequest**

Contenido	Fuente	Sumidero
Iniciador	Proveedor solicitante	Proveedor superior
Identificador de testigo	Petición	Proveedor superior

#### 11.49 TokenTestConfirm

Se genera **TokenTestConfirm** en el proveedor MCS superior al recibirse **TokenTestRequest**. Cuando se devuelve al proveedor solicitante, genera una confirmación MCS-PRUEBA-TESTIGO.

Los proveedores que reciben **TokenTestConfirm** actualizarán el testigo en su base de información para que concuerde con el estado retornado.

Esta MCSPDU se encamina de la misma forma que **TokenGrabConfirm**.

**Cuadro 11-49/T.125 – MCSPDU TokenTestConfirm**

Contenido	Fuente	Sumidero
Iniciador	Proveedor superior	MCSPDU
Identificador de testigo	Proveedor superior	Confirmación
Estado del testigo	Proveedor superior	Confirmación

### 11.50 CapabilitiesNotificationRequest

**CapabilitiesNotificationRequest** es una PDU que se transmite en sentido ascendente y que se utiliza para alertar a nodos jerárquicamente superiores sobre cambios que se desean hacer a la lista de capacidades, o bien, para alertar a la cumbre de un cambio en el estado de las conexiones de nodos V2. Puede también ser enviada por nodos intermedios que detectan la pérdida de una conexión descendente. Esta PDU puede también ser enviada por cualquier nodo que detecte el establecimiento o la pérdida de una conexión con un nodo V2.

**Cuadro 11-50/T.125 – MCSPDU CapabilitiesNotificationRequest**

Contenido	Fuente	Sumidero
Nodo V2 presente	Proveedor solicitante	Proveedor más alto
Lista de adición	Proveedor solicitante	Proveedor más alto
Lista de supresión	Proveedor solicitante	Proveedor más alto

El contenido de **CapabilitiesNotificationRequest** es el siguiente:

*Nodo V2 presente:* Es una bandera que se inicializa cuando un nodo V2 se conecta con un nodo V3. Cuando el nodo V2 se desconecta, la bandera se reinicializa.

*Lista de adición:* Es una lista de las capacidades (descritas en 11.50.1) que un nodo solicita que se añadan a la lista de capacidades de la cumbre. Es un campo opcional porque no todas las **CapabilitiesNotificationRequest** que se generan tienen por objetivo añadir capacidades a la lista.

*Lista de supresión:* Es una lista de las capacidades que un nodo solicita que se supriman de la lista de capacidades de la cumbre. Es un campo opcional porque no todas las **CapabilitiesNotificationRequest** que se generan tienen por objetivo suprimir capacidades de la lista.

#### 11.50.1 Petición de capacidades

El cuadro siguiente resume los contenidos de la petición de capacidad. Los campos AddList (lista de adición) y RemoveList (lista de supresión) de una PDU **CapabilitiesNotificationRequest** se componen de petición de capacidades.

Identificador de capacidad
Clase de capacidad
Indicador de participación

*Identificador de capacidad:* Es el identificador asignado a la capacidad. El identificador puede estar normalizado (es decir, documentado según la Recomendación T.125) o no estar normalizado.

*Clase de capacidad:* Indica si la capacidad tiene un valor asociado y el procesamiento asociado con dicho valor. Todas las capacidades son de clase MIN, MAX o NULA.

*Indicador de participación:* Este campo indica el grado de participación requerido por parte de otros nodos V3 de la cumbre. Los valores que puede tomar este campo son "parcial" o "global".

### 11.51 CapabilitiesNotificationIndication

**CapabilitiesNotificationIndication** es una PDU que se transmite en sentido descendente y que se utiliza para informar a los nodos sobre los cambios en las capacidades de la cumbre o sobre cambios en el estado de las conexiones de nodos V2. Esta PDU envía el proveedor de la cumbre o cualquier nodo intermedio que tenga que informar sobre un cambio a un nodo inferior.

**Cuadro 11-51/T.125 – MCSPDU CapabilitiesNotificationIndication**

Contenido	Fuente	Sumidero
Nodo V2 presente	Proveedor superior	Subordinado
Lista de adición	Proveedor superior	Subordinado
Lista de supresión	Proveedor superior	Subordinado

El contenido de **CapabilitiesNotificationIndication** es el siguiente:

*Nodo V2 presente:* Es una bandera que se inicializa cuando un nodo V2 se conecta con la cumbre V3. Cuando todos los nodos V2 se desconectan, la bandera se reinicializa.

*Lista de adición:* Es un lista de las capacidades (descritas en 11.51.1) que un nodo identifica que es necesario que se añadan a la lista de capacidades de la cumbre. Es un campo opcional porque no todas las **CapabilitiesNotificationIndication** que se generan tienen por objetivo añadir capacidades a la lista.

*Lista de supresión:* Es una lista de las capacidades que un nodo identifica que es necesario que se supriman de la lista de capacidades de la cumbre. Es un campo opcional porque no todas las **CapabilitiesNotificationIndication** que se generan tienen por objetivo suprimir capacidades de la lista.

#### 11.51.1 Indicación de capacidades

El cuadro siguiente resume los contenidos de la indicación de capacidades. Los campos lista de adición (AddList) y lista de supresión (RemoveList) de una PDU **CapabilitiesNotificationIndication** se componen de indicación de capacidades.

Identificador de capacidad
Clase de capacidad
Proveedor de cumbre soportado
Nodo intermedio soportado

*Identificador de capacidad:* Es la identificación asignada a la capacidad. El identificador puede estar normalizado (es decir, documentado según la Recomendación T.125) o no normalizado.

*Clase de capacidad:* Indica si la capacidad tiene un valor asociado y el procesamiento asociado con dicho valor. Todas las capacidades son de clase MIN, MAX o NULA.

*Proveedor de cumbre soporta:* Es una bandera que indica si el proveedor de cumbre V3 soporta esta capacidad.

*Nodo intermedio soportado.* Es una bandera que indica si los nodos intervinientes en el trayecto al proveedor de cumbre soportan esta capacidad. Si dichos nodos no existen, es decir en el caso de un nodo con conexión directa con el proveedor de cumbre, la bandera se inicializa.

## 12 Base de información de proveedor MCS

### 12.1 Replicación de la jerarquía

Un proveedor MCS, aunque puede atender múltiples dominios, sirve a cada uno independientemente. Para cada uno de ellos mantiene bases de información con lógicas separadas en las que registra el estado de los recursos (canales y testigos) que se están utilizando. La descripción que sigue presupone el contexto de un solo dominio.

Los recursos MCS que es necesario manejar en un dominio son los identificadores de canal y los identificadores de testigo. Los identificadores de usuario son un subconjunto de los identificadores de canal. Parámetros del dominio limitan el número de identificadores de cada categoría que pueden usarse simultáneamente. Esto permite a un proveedor calcular la cantidad de memoria necesaria para la base de información en el caso más desfavorable de un dominio utilizado en su totalidad.

En la jerarquía de un dominio, los identificadores en uso en un proveedor MCS cualquiera dado se estabilizan tendiendo a formar un subconjunto de los que se están utilizando en su superior inmediato. La información sobre un identificador se registra en un lugar donde puede utilizarse para soportar servicios MCS en que interviene ese identificador. Un registro más extenso de la información entrañaría costos adicionales en el tráfico de las MCSPDU, para mantener actualizada la información. Puesto que la información registrada en un proveedor está de acuerdo con la registrada en proveedores más altos, dentro del límite del tiempo de propagación de las MCSPDU, puede decirse que la base de información del proveedor está parcialmente replicada a través de la jerarquía del dominio.

Al establecerse la primera conexión MCS de un dominio, los parámetros del dominio quedan fijados y no podrán modificarse. Un proveedor que no tiene la capacidad necesaria para trabajar con el número máximo de identificadores especificados en cada categoría puede negociar para incorporarse a un dominio sobre una base falsa. Puede especular con el hecho de que, dada su baja posición en una jerarquía, no se le exigirá retener más que una fracción de la base de información. Tal proveedor podría no soportar anexiones y subordinados con la gama completa de servicios MCS que éstos esperan. No obstante, mientras su capacidad no haya sido efectivamente rebasada, dicho proveedor podrá actuar como un miembro del dominio igual a los demás. Esta estrategia pudiera convenir a nodos terminales con pretensiones limitadas.

Los identificadores se ponen en uso por primera vez en el proveedor MCS superior. En los proveedores subordinados se ponen en uso mediante un flujo descendente selectivo de MCSPDU. La mayor parte de ellos se retiran del uso de la misma manera de arriba a abajo. Existen necesariamente intervalos en los que un proveedor subordinado registra como en uso un identificador que no está registrado como tal en sus superiores, porque la MCSPDU que suprime el identificador está todavía en tránsito. Esta situación, sin embargo, no perdura. Las MCSPDU de control se reciben y procesan en el mismo orden en que se transmiten. Las consecuencias del procesamiento de una MCSPDU, incluida la creación o supresión de identificadores de canal y de testigo, comienzan a surtir efecto antes de que la atención se concentre en el siguiente suceso de entrada.

Una excepción a lo expuesto en el párrafo anterior es la supresión de identificadores de canales estáticos y asignados. Estos identificadores de canal, aunque hayan sido puestos en uso por un flujo descendente de **ChannelJoinConfirm**, se suprimen en orden inverso, de abajo a arriba. Específicamente, se suprimen cuando una acumulación de peticiones MCS-ABANDONO-CANAL

procedentes de anexiones y MCSPDU **ChannelLeaveRequest** procedentes de proveedores subordinados, se combinan para abandonar un canal al que no ha habido incorporación. Esas transiciones dan motivo a la transmisión de **ChannelLeaveRequest** más hacia arriba. Así, en estos dos casos, los identificadores de canal registrados como en uso son un subconjunto estricto de los que se encuentran en uso en el proveedor más alto. Este es un corolario accidental de las optimizaciones que tienen por finalidad acelerar la gestión de canal como paso previo a la transferencia de datos.

Los identificadores de canal se ponen en uso mediante **MergeChannelsConfirm**, **AttachUserConfirm**, **ChannelJoinConfirm**, **ChannelConveneConfirm**, y **ChannelAdmitIndication**; se suprimen mediante **MergeChannelsConfirm**, **PurgeChannelsIndication**, **DetachUserIndication**, **ChannelLeaveRequest**, **ChannelDisbandIndication** y **ChannelExpelIndication**. Los identificadores de testigo se ponen en uso mediante **MergeTokensConfirm**, **TokenGrabConfirm**, **TokenInhibitConfirm**, y **TokenGiveIndication**; se suprimen mediante **MergeTokensConfirm**, **PurgeTokensIndication**, **TokenReleaseConfirm**, **TokenGiveResponse**, y **TokenGiveConfirm**. Cuando se pone en uso un identificador en un proveedor dado, la MCSPDU que la causó se puede reenviar a cero, uno, varios, o todos los proveedores subordinados. La utilización de un identificador puede aumentar o reducirse gradualmente cuando, por ejemplo, usuarios individuales se admiten a un canal privado, o se excluyen de éste. Cuando se suprime un identificador en un proveedor dado, la MCSPDU que efectúa esto se reenvía a todos los subordinados que pueden estar todavía registrando el identificador como en uso.

El uso de un identificador está vinculado, en último término, a acciones que sobre un canal o testigo ejecuta un usuario anexionado al dominio (aunque puede haber algún retardo, como se ha explicado, al comunicar cambios mediante la transmisión de MCSPDU). Los identificadores registrados establemente como en uso en un proveedor MCS dado son los que algún usuario emplea activamente en el subárbol del proveedor. Por eso constituyen un subconjunto de los registrados establemente en cualquier proveedor más alto.

La supresión de un identificador de usuario tiene como consecuencia natural la supresión de los identificadores de canal y de testigo de los cuales es el único usuario en un subárbol.

En las subcláusulas que siguen se especifican los criterios seguidos para considerar en uso los identificadores de canal y los identificadores de testigo.

## 12.2 Información de canal

Cada una de las cuatro clases de canal tiene sus propios criterios para determinar si ha de considerarse que una anexión dada está usando un identificador de canal y, en consecuencia, si ha de estar representado en la base de información del proveedor:

- a) Un identificador de canal estático (gama 1..1000) está en uso si el usuario se ha incorporado al canal con una confirmación MCS-INCORPORACIÓN-CANAL de éxito y no lo ha abandonado por una petición o indicación MCS-ABANDONO-CANAL.
- b) Un canal de identificador de usuario está en uso si se ha asignado al usuario por una confirmación MCS-ANEXIÓN-USUARIO de éxito y el usuario no se ha desanexionado por una petición o indicación MCS-DESANEXIÓN-USUARIO.
- c) Un identificador de canal privado está en uso si el usuario ha creado el canal con una confirmación MCS-FORMACIÓN-CANAL de éxito o ha sido admitido al mismo con una indicación MCS-ADMISIÓN-CANAL y no ha sido expulsado por una indicación MCS-EXCLUSIÓN-CANAL y el canal no ha sido disuelto por una petición o indicación MCS-DISOLUCIÓN-CANAL.

- d) Un identificador de canal asignado está en uso si el usuario se ha incorporado al canal con una confirmación MCS-INCORPORACIÓN-CANAL de éxito y no lo ha abandonado por una petición o indicación MCS-ABANDONO-CANAL.

Para un identificador de canal en uso se registrará la siguiente información:

- a) La clase de canal que representa (estático, de identificador de usuario, privado, o asignado).
- b) Por cuáles anexiones MCS y por cuáles conexiones MCS a proveedores subordinados se ha efectuado la incorporación al canal.
- c) Si se trata de un canal de identificador de usuario, el sentido de transmisión correspondiente al mismo, es decir, o bien la anexión MCS local a que se ha asignado el identificador de usuario, o la conexión MCS descendente a un proveedor subordinado en cuyo subárbol reside el usuario.
- d) Si se trata de un identificador de canal privado, el identificador de usuario del gestor que lo convocó (esté o no el propio gestor en el subárbol del proveedor) y el conjunto de todos los identificadores de usuario en el subárbol del proveedor que ha sido admitido en el canal.

La información registrada sobre los identificadores de canal se emplea como se ha explicado en la cláusula 10 para validar las MCSPDU de petición y para encaminar las MCSPDU de indicación y de confirmación.

### 12.3 Información de testigo

Las transiciones de estado de un identificador de testigo se muestran en la figura 12-1. A tales efectos se aplican las abreviaturas siguientes:

DUin – DetachUserIndication

TGcf – TokenGrabConfirm

TIcf – TokenInhibitConfirm

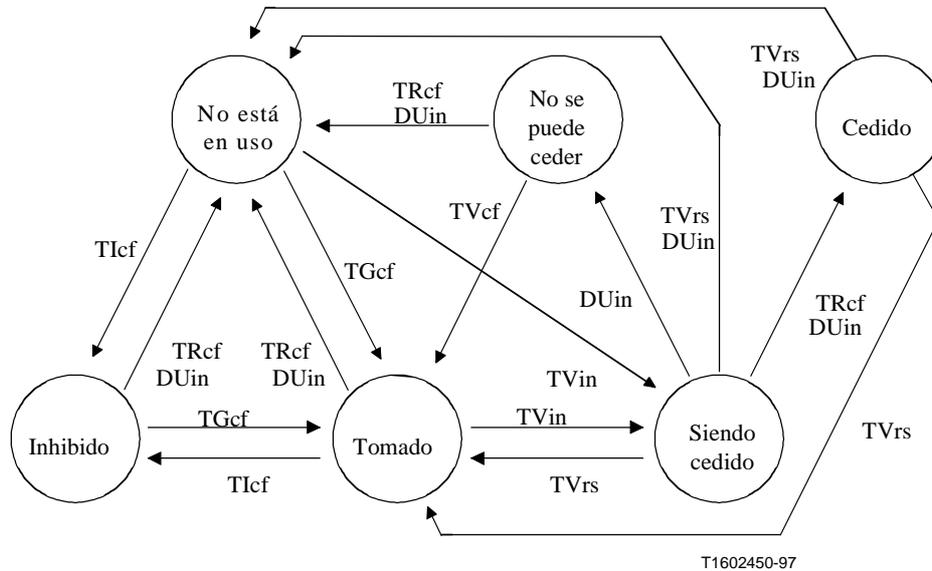
TRcf – TokenReleaseConfirm

TVin – TokenGiveIndication

TVrs – TokenGiveResponse

TVcf – TokenGiveConfirm.

Un identificador de un testigo puede estar *tomado* por un solo usuario o *inhibido* por uno o más usuarios. La acción de **TokenGiveIndication** convierte el estado a *cediéndose* a lo largo de la rama de una jerarquía de dominio que va del proveedor MCS superior al receptor deseado. Este estado se degrada a *no cedido* si el receptor se desanexiona antes de que el proveedor responda con **TokenGiveResponse**. En cambio, pasa a *cedido* si el cedente libera explícitamente el testigo o se desanexiona. Durante la cesión de un testigo, la rama de una jerarquía de dominio que parte del cedente intersecta la rama que conduce al receptor al menos en el proveedor MCS superior. El estado del testigo cambia de *tomado* a *cediéndose*, y posiblemente más adelante a *no cedido* o *cedido*, solamente a lo largo de esta intersección.



**Figura 12-1/T.125 – Transiciones de estado de un identificador de testigo**

El usuario de un testigo y el testigo están en una relación de tomador, inhibidor, receptor, o en la de tomador y receptor (cuando el usuario se cede el testigo a sí mismo):

- El usuario es un tomador si ha obtenido un testigo con una confirmación MCS-TOMA-TESTIGO de éxito y no lo ha liberado mediante una petición MCS-LIBERACIÓN-TESTIGO o una confirmación MCS-CESIÓN-TESTIGO de éxito, ni lo ha convertido con una confirmación MCS-INHIBICIÓN-TESTIGO de éxito o si ha aceptado un testigo ofrecido con una respuesta MCS-CESIÓN-TESTIGO de éxito.
- El usuario es un inhibidor si ha obtenido un testigo con una confirmación MCS-INHIBICIÓN-TESTIGO de éxito y no lo ha liberado con una petición MCS-LIBERACIÓN-TESTIGO ni lo ha convertido con una confirmación MCS TOMA TESTIGO de éxito.
- El usuario es un receptor si se le ha ofrecido un testigo por una indicación MCS-CESIÓN-TESTIGO y no lo ha liberado con una respuesta MCS-CESIÓN-TESTIGO de fracaso.

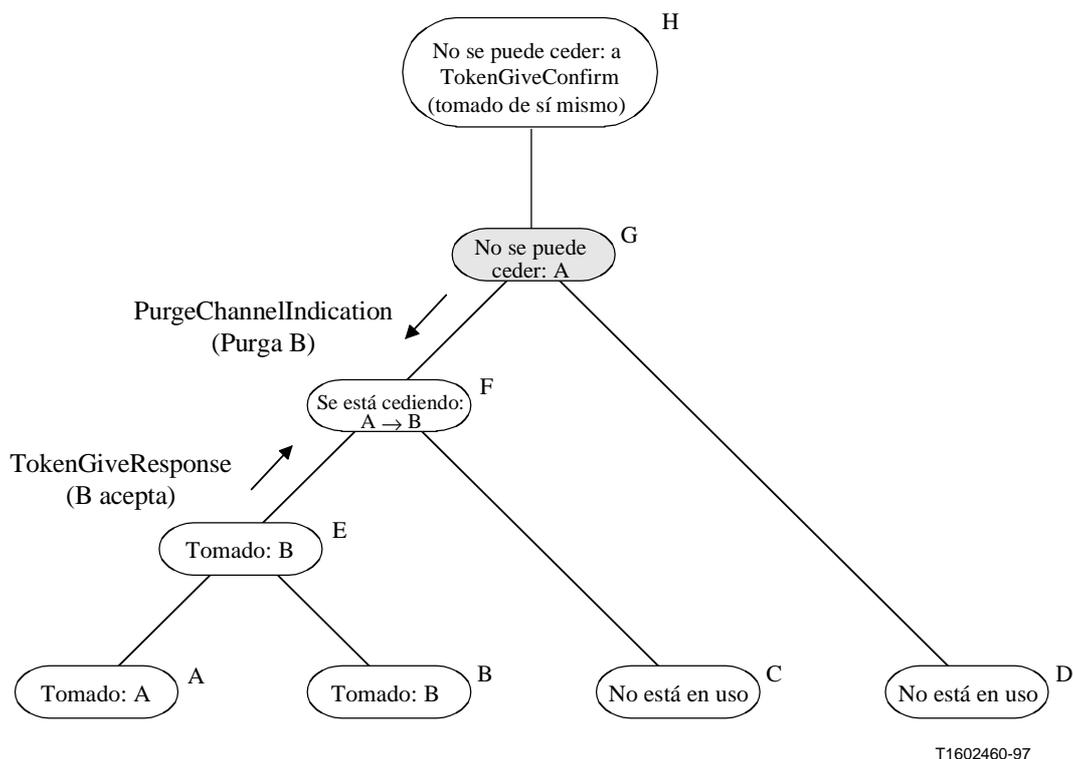
La siguiente información se registrará para un identificador de testigo en uso:

- El estado del identificador de testigo en el proveedor MCS (no necesariamente idéntico a su estado en el proveedor superior).
- Si el estado es tomado o no cedido, el identificador de usuario del tomador en el subárbol del proveedor.
- Si el estado es cediéndose, el identificador de usuario del receptor en el subárbol del proveedor, encuéntrese o no el tomador en el subárbol del proveedor.
- Si el estado es cediéndose o cedido, el identificador de usuario del receptor en el subárbol del proveedor.
- Si el estado es inhibido, el conjunto de todos los identificadores de usuario en el subárbol del proveedor que ha inhibido el testigo.

La información registrada para identificadores de testigo en uso se emplea como se ha explicado en la cláusula 10 para validar MCSPDU de respuesta y para encaminar MCSPDU de indicación.

El estado de un identificador de testigo en un proveedor subordinado no tiene que ser idéntico a su estado en el proveedor MCS superior. Esto se debe al hecho de que, por lo general, un cedente de testigo no procesa **TokenGiveIndication** ni **TokenGiveResponse**, y de que, por lo general, un

receptor no procesa la **TokenReleaseConfirm** de un cedente. La figura 11-2 muestra los estados que pueden presentarse en una interacción compleja de testigos.



**Figura 12-2/T.125 – Estados de testigo que pueden presentarse en una interacción compleja**

La figura destaca esencialmente el caso de un identificador de testigo en la base de información de los proveedores de A a H. Un antecedente plausible es que el usuario A, afectado al proveedor A, cedió el testigo al usuario B, afectado al proveedor B. Sin embargo, antes de que B pudiera responder, el proveedor G conectó el dominio a un nuevo proveedor superior H y comenzó una fusión. El nuevo dominio, al no poder aceptar al usuario B por existir un conflicto en los identificadores de canal, inició su purga mediante **PurgeChannelsIndication**. Esta MCSPDU se muestra habiendo recorrido parte de su trayectoria, a través de los proveedores G y D. El proveedor G, en consecuencia, ha ajustado el estado de su testigo pasándolo de *cediéndose* a *incedible*, y ha pasado (por fusión) el testigo, en este estado, al nuevo dominio. El nuevo proveedor superior H, al que se ha presentado un testigo en ese estado, ha introducido en la cola de transmisión una **TokenGiveConfirm** de fracaso, para retornar el testigo al usuario A. En el momento mostrado, el testigo ha sido también finalmente aceptado por el usuario B, y una **TokenGiveResponse** está subiendo por la rama del receptor en la jerarquía del dominio, cambiando el estado del testigo en cada proveedor de *cediéndose* a *tomado* por B. Convergen ahora en el proveedor F dos MCSPDU, y la primera de ellas que llegue determinará su ulterior estado allí: o bien volverá a ser *tomado* por A, o pasará transitoriamente por el estado *tomado* por B y, después que el usuario B se haya desanexionado, será de nuevo *no en uso*. Tanto en uno como en el otro caso, el estado del testigo se estabilizará en *tomado* por A cuando el nuevo proveedor superior transmita la **TokenGiveConfirm** que tiene pendiente.

## **13 Elementos de procedimiento**

### **13.1 Secuenciación de las MCSPDU**

Las MCSPDU de control se mantienen en secuencia entre cualquier par de proveedores MCS porque se transmiten por una TC única, la TC inicial de una conexión MCS. Los proveedores MCS procesarán las MCSPDU recibidas y transmitirán cualesquiera MCSPDU de salida resultantes en el mismo orden en que las recibieron. Esto se aplica tanto a las MCSPDU que son simplemente reenviadas en sentido ascendente o descendente a través de la jerarquía del dominio, como a las MCSPDU que son transformadas, por ejemplo, las peticiones y las respuestas se transforman respectivamente en indicaciones y confirmaciones. La secuenciación de las MCSPDU se mantendrá dentro de un proveedor MCS si es necesario para introducir en cola las salidas con miras a su ulterior transmisión, como consecuencia de la retropresión de control de flujo a lo largo de una TC.

Las MCSPDU de diferentes prioridades no tienen que mantenerse en secuencia. Por el contrario, la ventaja de las prioridades relativas sólo se pone de manifiesto cuando los datos de prioridad más alta son adelantados y sobrepasan a los de prioridad más baja. Ello significa que los datos deben transmitirse por TC separadas e introducirse en cola separadamente dentro de cada proveedor MCS. Si el número de prioridades de datos implementadas en un dominio es menor que el máximo, hay menos TC disponibles, pero los proveedores que optan por esto pueden mantener aun así, internamente, colas separadas. Con esto se obtienen algunas de las ventajas de la prioridad relativa, pero no todas.

Un proveedor MCS mantendrá la secuencia de las MCSPDU de control transmitidas con una prioridad dada. Esta es una limitación más estricta que la impuesta por la Recomendación T.122, que sólo garantiza la secuenciación de unidades de datos de servicio transmitidas con una prioridad dada a través del mismo canal de destino.

Las MCSPDU de control y las de datos de más alta prioridad se transmiten a través de la misma TC inicial y serán tratadas con la misma atención por un proveedor MCS. Los datos de prioridades más bajas pueden rezagarse. Las indicaciones de control que sobrepasan a otras unidades pueden llegar antes que datos con prioridad más baja transmitidos primero.

### **13.2 Control de flujo de entrada**

Los objetivos de un proveedor MCS a veces son contradictorios, por ejemplo: mantener los datos en rápido movimiento a través de un dominio pese a los bloqueos transitorios de las transmisiones hacia algunos receptores, dar a los transmisores un acceso equitativo a la anchura de banda disponible, y evitar que cualquiera de los participantes se quede demasiado rezagado con respecto a otros proveedores pares que están recibiendo los mismos datos multidifundidos. MCS es un servicio fiable que preserva la integridad de los datos de usuario. Dado que un proveedor tiene una capacidad limitada para almacenar las MCSPDU que no pueden transmitirse inmediatamente, tiene que disponer de un medio defensivo que le permita algunas veces rechazar ulteriores entradas. Si bien los detalles de la interfaz con el servicio de transporte es un asunto de índole local, el efecto, en un orden abstracto, deberá ser que las TSDU entrantes se mantengan en tuberías TC, intactas y en secuencia, para que se reciban posteriormente, cuando se haya dejado sin efecto el control de flujo. Cuando las tuberías TC se llenan, los proveedores MCS distantes pueden verse impedidos, como consecuencia de la retropresión, de transmitir ulteriores MCSPDU y tener que recurrir a un medio defensivo similar.

El control de flujo no se señala explícitamente en el protocolo MCS. Esta es una función de capas inferiores cuya duplicación sería inútil. Por tanto, es difícil detectar, a través de un medio de transmisión constituido por una TC participante, si un proveedor distante está oponiéndose a aceptar

más entradas. Sin embargo, con el fin de satisfacer lo mejor posible objetivos contradictorios, se recomienda aplicar la política de control de flujo que se describe a continuación.

Un proveedor MCS puede asignar a cada TC entrante una cuota fija de memorias tampón que podrá llenar con MCSPDU antes de que se aplique retropresión. Cada memoria tampón se procesa como se especifica en este protocolo, después de lo cual se asigna para la salida de cero o más TC salientes. La salida puede producirse inmediatamente o demorarse porque una tubería de transporte esté llena. Mientras una memoria tampón no haya descargado sobre la última TC, se cargará contra la cuota de entrada de la TC por la que llegó dicha entrada. Una vez que haya descargado sobre todas las TC requeridas, será reciclada como un incremento de esa cuota. Cuando se ha agotado una cuota, se detienen las ulteriores entradas a través de la TC correspondiente. Las cuotas de entrada podrán fijarse atendiendo a que una TC forme parte de una conexión MCS ascendente o descendente, y a la prioridad de datos que representa.

Mediante el empleo de memorias tampón se puede mitigar los efectos de las diferencias entre las velocidades de los transmisores y los receptores. La fijación de cuotas a las entradas puede impedir que cualquiera de las TC monopolice los recursos. Puede también determinar hasta qué punto dos receptores de los mismos datos multidifundidos podrán estar fuera de paso. Sin embargo, el esquema recomendado no es lo suficientemente inteligente para prever todos los patrones de utilización y pudiera a veces reducir la velocidad de transferencia de datos a través de un dominio en situaciones en que existan alternativas aceptables. El establecimiento de mejores políticas de control de flujo (aplicables localmente y que no requieran la comunicación adicional de MCSPDU) podría ser un medio de conseguir una diferenciación del producto.

### 13.3 Cumplimentación del caudal

A diferencia del control de flujo de entrada, el protocolo MCS da cierto apoyo explícito a la cumplimentación del caudal. En primer lugar, la velocidad a que se cumplimenta el caudal es un parámetro de dominio que se negocia mediante MCS-CONEXIÓN-PROVEEDOR. En segundo lugar, el intervalo de tiempo durante el cual se supervisa el caudal antes de ejecutar una acción adversa lo comunica cada proveedor MCS a su superior mediante **ErectDomainRequest**. Para describir el intervalo de cumplimentación del caudal es necesario que los proveedores ajusten su comportamiento a ciertos principios comunes. De todas formas, la cumplimentación del caudal sigue siendo una técnica heurística con margen para la intervención.

La cumplimentación de una velocidad mínima de entrada en cada receptor es una opción de que disponen las aplicaciones de controlador que establecen conexiones MCS de un dominio. Esta opción se selecciona mediante el parámetro de dominio para el caudal cumplimentado, que se expresa en octetos por segundo. Si bien parece evidente que no se debe permitir a un participante funcionar a una velocidad arbitrariamente baja y obstaculizar así la transferencia de datos entre los otros, el tratar de cumplimentar un caudal de una manera demasiado estricta entraña un peligro.

Los complejos patrones de utilización de múltiples transmisores constituyen un problema. Las clases de retropresión contra las que reacciona una política de cumplimentación del caudal pueden no ser el resultado de un solo receptor anómalamente lento. En primer lugar, las MCSPDU transmitidas en sentido descendente deben competir, para ser atendidas, con las transmitidas en sentido ascendente, pero son reflejadas en retorno, sobre todo **SendDataIndication**. El flujo descendente experimentado por un proveedor par puede por tanto alcanzar sólo una fracción de la anchura de banda nominal y puede variar dinámicamente con el número de las otras conexiones y anexiones al proveedor par. En segundo lugar, cabe esperar que sólo las MCSPDU de máxima prioridad fluyan continuamente por una conexión MCS. Las de prioridades más bajas pueden, de hecho, quedar bloqueadas por un proveedor par durante largos periodos de tiempo debido a la intensidad del tráfico recibido de otras fuentes con prioridades más altas. Por último, puede haber una gran variación en el caudal

instantáneo si éste se mide por el tiempo que tiene que esperar una MCSPDU bloqueada para ser aceptada y pasar a una conexión MCS. Esto puede depender, entre otras cosas, del número de MCSPDU procedentes de otras fuentes que son introducidas en cola en el proveedor par, y en qué orden esto se efectúa.

No obstante, la cumplimentación del caudal es una opción valiosa en situaciones prácticas en las que se sabe que los patrones de transferencia de datos son más uniformes. Debe interpretarse como que requiere la aplicación de una salida mínima de MCSPDU a cada anexión MCS directa y a cada conexión MCS descendente durante un intervalo de tiempo que será especificado por el proveedor MCS cumplimentante. Cada salida de MCSPDU, tanto de control como de datos, deberá contar, a los efectos del caudal, como si fuera del tamaño máximo permitido por los parámetros del dominio. La salida de una MCSPDU aplicada a una anexión MCS deberá entrañar la entrega de la indicación o confirmación de primitiva conexas. La salida aplicada a una conexión MCS descendente entrañará la ausencia de retropresión en la interfaz del servicio de transporte y la aceptación para pasar a la correspondiente tubería TC.

La salida hacia una anexión dada o hacia una conexión descendente se deberá supervisar mientras exista una o más MCSPDU en la cola, independientemente de la prioridad de los datos. Cuando las colas se vacían, termina la supervisión, sin que se ejecuten acciones de cumplimentación del caudal. Al mismo tiempo, no se registrará ningún mérito por buena conducta para compensar futuros rezagos. La supervisión se reanudará tan pronto como la retropresión impida la obtención de cualquier MCSPDU a la salida, y, en lugar de ello, haya que introducirla en una cola. Mientras una o más MCSPDU permanezcan en la cola, el número de unidades efectivamente obtenidas a la salida se contará en un intervalo fijo de tiempo.

Cada proveedor MCS elegirá un intervalo de cumplimentación del caudal. El intervalo será lo suficientemente largo para que por lo menos una MCSPDU de tamaño máximo pueda obtenerse a la salida en el caudal mínimo. Un proveedor MCS informará a su superior sobre el intervalo elegido y sus cambios ulteriores, transmitiendo **ErectDomainRequest** hacia arriba. Un proveedor actuará contra la anexión MCS o la conexión MCS descendente infractora una vez terminado cualquier intervalo durante el cual el caudal supervisado no se ajuste a su comportamiento esperado. El proveedor desafectará al usuario o desanexionará la conexión.

Los proveedores más altos pueden fijar su intervalo de cumplimentación del caudal de modo que sea más largo que el de cualquier subordinado, más cierto margen de tiempo de reacción. Esto tiene por finalidad incitar a que las acciones de cumplimentación del caudal se ejecuten en primer lugar en el proveedor más bajo cuya posición le permita detectar el problema. Cuando, después del intervalo de algún proveedor más bajo, se elimina un infractor, los proveedores más altos deben tener tiempo suficiente para detectar el restablecimiento del caudal a los niveles adecuados. Si actúan con demasiada rapidez, pudieran penalizar a un subárbol que fuera mayor de lo necesario, y perjudicar a otros que no sean infractores en el dominio.

Las aplicaciones de controlador que establecen parámetros del dominio deberán ser conservadoras en sus pretensiones, y prever que el caudal pueda ocasionalmente sufrir bajas durante periodos de intensos estímulos de aplicación. Si su preocupación es simplemente dar protección contra receptores que dejan de aceptar todo, por completo, pueden fijar el caudal mínimo muy bajo. Sabiendo el tamaño máximo de las MCSPDU y el caudal cumplimentado, los controladores pueden calcular el intervalo mínimo que debe transcurrir antes de que se detecte y resuelva un bloqueo.

#### **13.4 Configuración de los dominios**

La Recomendación T.122 no proporciona ningún mecanismo para configurar el conjunto de los dominios soportados por un proveedor MCS. Deberá considerarse que esto es un asunto local cuya normalización queda en estudio. Este protocolo presupone que un proveedor MCS reconocerá

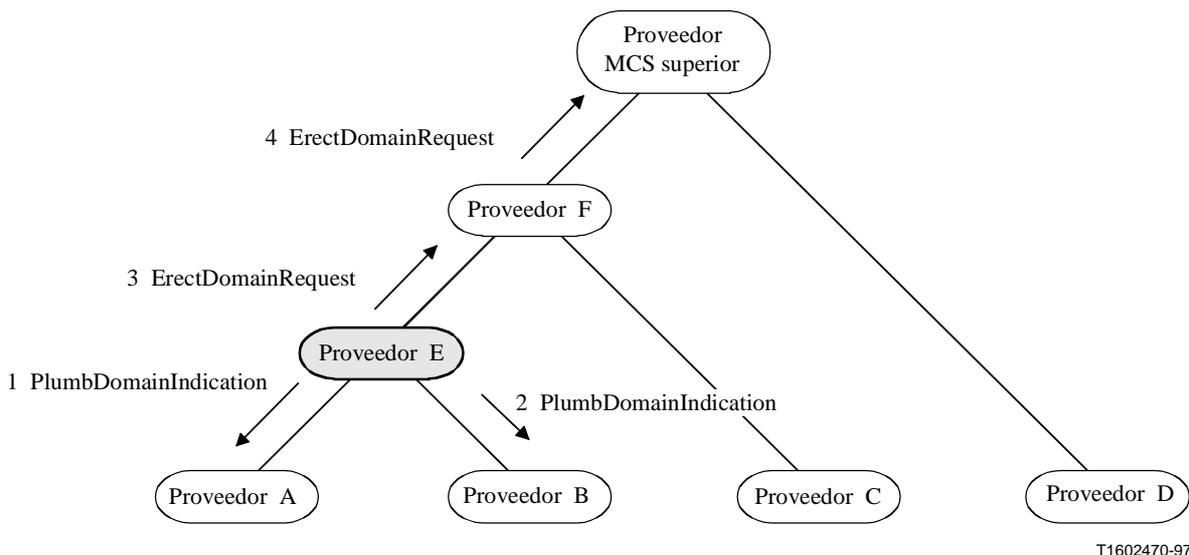
algunos selectores de dominio como válidos y otros como inválidos. Prevé que los selectores de dominio se comuniquen como parte del establecimiento de una conexión MCS.

Un proveedor MCS participa implícitamente en la negociación de parámetros de dominio. Sea el lado llamante o el llamado, restringe la gama de valores de parámetro admitidos de acuerdo con los límites para los que está configurado el dominio. Un proveedor MCS congelará la negociabilidad de los parámetros de dominio cuando cualquier usuario se haya anexionado a un dominio o cuando se haya establecido la primera conexión MCS.

### 13.5 Fusión de dominios

Se fusionan dominios como consecuencia de MCS-CONEXIÓN-PROVEEDOR. Si es fácil conseguir que uno u otro dominio esté vacío en este punto, la fusión no presentará gran complicación. En el caso más general, sin embargo, debe preverse la actualización de la base de información del proveedor superior restante de modo que contenga la base de información del antiguo proveedor superior, y la solución de cualquier conflicto que surja. En la cláusula 10 se da una información más detallada al respecto.

Para facilitar la comprensión, en el grupo de figuras 13-1 a 13-4 se presenta un ejemplo de fusión de dominios. En este ejemplo, el proveedor E representa un antiguo proveedor superior que se ha incorporado a un nuevo dominio mediante la conexión MCS (dibujada en trazo más grueso) con un proveedor intermedio F. Es intrascendente que la conexión MCS haya sido iniciada por el proveedor E o el proveedor F.



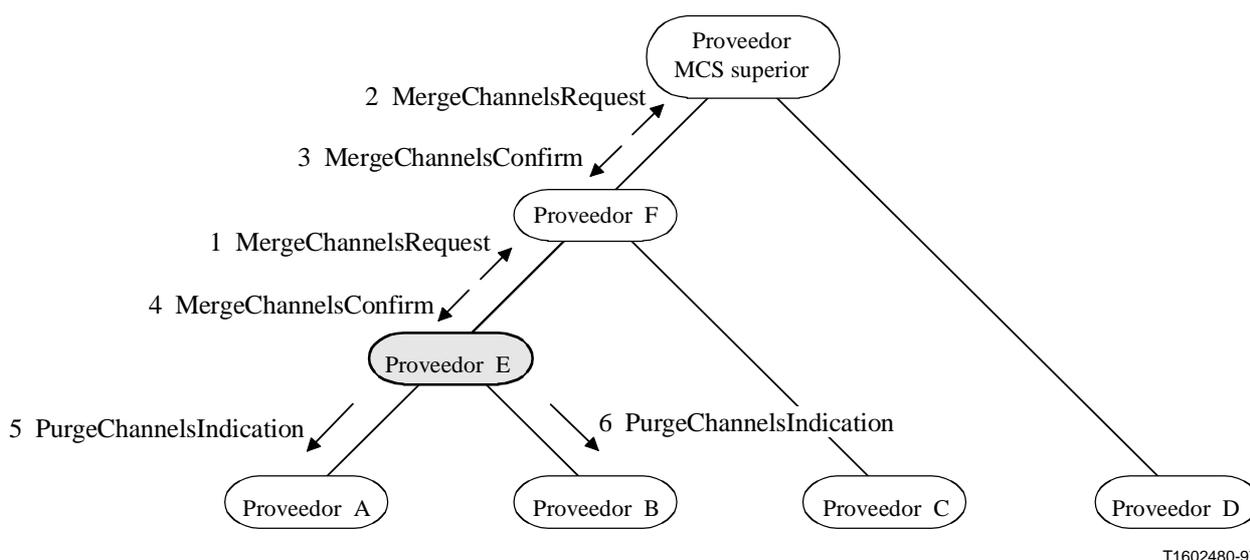
**Figura 13-1/T.125 – Primera etapa de la fusión de dominios – Establecimiento de la jerarquía**

El proveedor E, al observar que se encuentra en el extremo inferior de una conexión ascendente, asume la responsabilidad de la ejecución de la fusión. Como no es seguro efectuar una actividad cuando la base de información está activa, el proveedor E detiene la aceptación de entradas de su subárbol. Toda MCSPDU que se encuentre ya en tránsito procedente de los proveedores A y B o las nuevas generadas antes de completarse la fusión serán salvaguardadas y procesadas posteriormente. En cambio, no se impide el flujo descendente de MCSPDU, hayan sido generadas por el proveedor E o reenviadas desde más arriba.

Hasta que la fusión se completa, las únicas MCSPDU de confirmación que el proveedor E recibirá son **MergeChannelsConfirm** y **MergeTokensConfirm**, pues no se permiten peticiones de usuario ascendentes. Estas confirmarán o purgarán los identificadores de canal y de testigo ya en uso. Las MCSPDU de indicación **PurgeChannelsIndication**, **PurgeTokensIndication**, **DetachUserIndication** y **ChannelDisbandIndication** del dominio superior pueden llegar también, suprimiendo identificadores de canal y de testigo del dominio más bajo cuya fusión ha sido confirmada individualmente. Los identificadores no confirmados del dominio más bajo están protegidos contra la supresión porque significan algo diferente con respecto al mismo identificador del dominio superior. **PlumbDomainIndication** debe ser obedecida para cumplir el límite de altura del dominio. Las indicaciones restantes no tienen que ser aplicadas por el proveedor E mientras se está efectuando la fusión. En particular, no deben fluir transferencias de datos entre los dominios separados anteriormente hasta que la fusión se completa; no pueden fluir hasta que hayan sido clasificados al menos los canales que las transportan. Puede no ser conveniente aplicar dos indicaciones que pueden poner en uso nuevos identificadores. Para mantener la coherencia de la base de información, si el proveedor E rechaza **ChannelAdmitIndication** o **TokenGiveIndication**, reaccionará como se especifica en la cláusula 10.

Las primeras acciones del proveedor E son enviar **PlumbDomainIndication** hacia abajo, cerciorarse de que la última conexión MCS no haya creado un ciclo que invalidaría el principio de que cada jerarquía de dominio tenga exactamente un proveedor superior, y enviar **ErectDomainRequest** hacia arriba para comunicar su altura y su intervalo de cumplimentación de canal. El proveedor F, al ascender de ese modo de la altura 2 a la 3, pasa de largo **ErectDomainRequest** al proveedor superior, que alcanza entonces la altura 4.

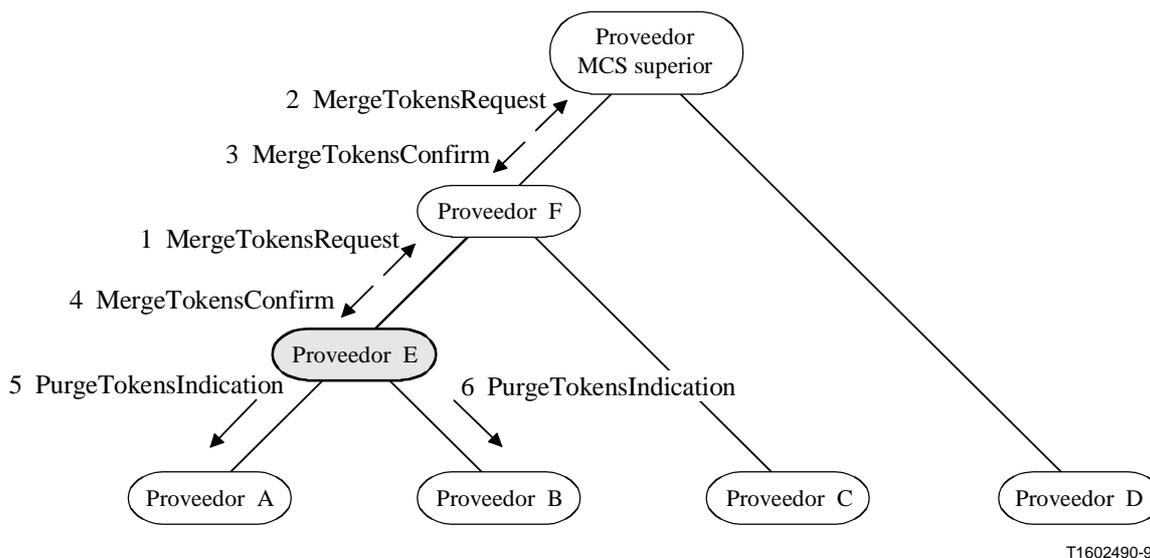
En la segunda etapa de la fusión de dominios, el proveedor E envía hacia arriba cuantas instancias de **MergeChannelsRequest** sean necesarias para contener los identificadores de usuario que existen en su base de información. Los identificadores de usuario que no estén en conflicto con el dominio superior se confirman y las restantes se purgan en el mismo número de instancias de **MergeChannelsConfirm**. En base a **MergeChannelsConfirm**, el proveedor E genera **PurgeChannelsIndication** para informar sobre las purgas que se hayan efectuado a todo lo largo de su subárbol. Esta etapa termina cuando todos los identificadores de usuario hayan sido explícitamente confirmados o purgados.



T1602480-97

**Figura 13-2/T.125 – Segunda etapa de la fusión de dominios – Fusión de los canales de identificador de usuario**

La tercera etapa es similar a la segunda; se diferencian en que ésta se refiere a los identificadores de testigo en lugar de los de canal, y se utiliza un conjunto paralelo de MCSPDU. Si los identificadores de usuario no se hubiesen fusionado previamente, porciones de una ulterior **MergeTokensRequest** podrían rechazarse como inválidas y los identificadores de testigo influidos serían innecesariamente purgados. En el caso de un testigo inhibido, es posible que el conjunto completo de usuarios inhibidores no quepan en una sola MCSPDU. El proveedor E espera por la confirmación del primer subconjunto que envía hacia arriba antes de volver a enviar el testigo inhibido con los identificadores de usuario restantes. De esta forma se evita que el primer conjunto se rechace porque se esté usando demasiados identificadores de testigo y que, no obstante, se acepten después los restantes, lo que adulteraría la base de información. Esta etapa termina cuando todos los identificadores de testigo han sido explícitamente confirmados o purgados.



**Figura 13-3/T.125 – Tercera etapa de la fusión de dominios –  
Fusión de los identificadores de testigo**

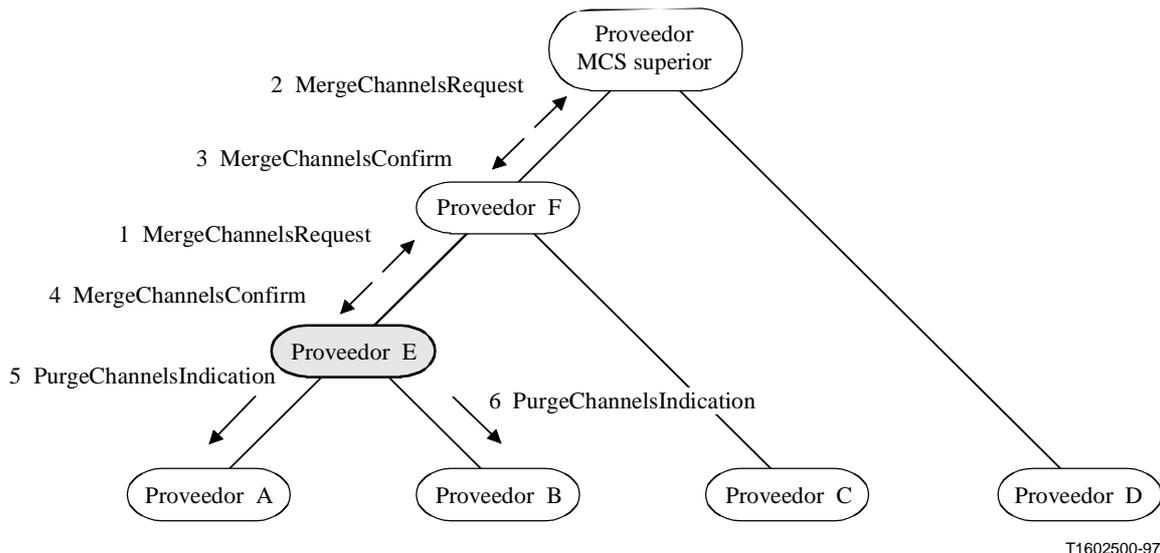
En la cuarta etapa se utilizan las mismas MCSPDU que en la segunda, pero contienen diferentes identificadores de canal. Una vez tratados los canales de identificador de usuario, los identificadores restantes son los de canales estáticos, privados, y asignados. Si los identificadores de usuario no se hubiesen fusionado previamente, porciones de una ulterior **MergeChannelsRequest** podrían rechazarse como inválidos y los identificadores de canal influidos serían innecesariamente purgados. En el caso de un canal privado, es posible que el conjunto completo de usuarios admitidos por el gestor del canal no quepan en una sola MCSPDU. El proveedor E espera por la confirmación del primer subconjunto que envía hacia arriba antes de volver a enviar el canal privado con los identificadores de usuario restantes. De esta forma se evita que el primer conjunto se rechace porque se esté usando demasiados identificadores de canal y que, no obstante, se acepten después los restantes, lo que adulteraría la base de información. Esta etapa termina cuando todos los identificadores de testigo han sido explícitamente confirmados o purgados.

NOTA – Cuando los identificadores de testigo han sido previamente fusionados, su posesión exclusiva es un medio más eficaz para resolver los conflictos de flujos de datos. En otro caso, mientras se confirman los identificadores de canal, y antes de que el conflicto haya sido revelado por una purga de testigos, pudiera haber fugas de datos entre los dominios.

### 13.6 Desconexión de dominio

Cuando se desconecta una conexión MCS ascendente, un proveedor MCS eliminará su árbol del dominio desanexionando todas sus anexiones MCS directas y desconectando todas sus otras conexiones MCS. El proveedor influido no puede, en general, establecer un dominio residual en su propio subárbol, porque no dispone de un registro de las MCSPDU de petición que se enviaron hacia arriba y para las cuales nunca recibirá la correspondiente MCSPDU de confirmación.

Cuando se desconecta una conexión MCS descendente, un proveedor MCS generará MCSPDU **DetachUserRequest** para todos los usuarios que residen en esa porción de su subárbol, expresando como motivo *dominio desconectado*.



T1602500-97

**Figura 13-4/T.125 – Cuarta etapa de la fusión de dominios – Fusión de los restantes identificadores de canal**

### 13.7 Atribución de identificadores de canal

Los identificadores de canal en la gama de 1001 en adelante se atribuyen dinámicamente en el proveedor MCS superior durante el procesamiento de peticiones de primitivas MCS-ANEXIÓN-USUARIO, MCS-INCORPORACIÓN-CANAL cero, y MCS-FORMACIÓN-CANAL. No se exige que los valores atribuidos se ajusten a un determinado patrón. Más bien es conveniente que los valores estén dispersados al azar en toda la gama permitida. Esto hace más probable que dos dominios que operan independientemente durante un periodo considerable de tiempo puedan fusionarse posteriormente sin conflictos en sus respectivas atribuciones de identificadores de canal. Con esto se obtiene también protección contra un reciclado demasiado rápido de los identificadores dentro de un dominio, pues se liberan de un uso y se reatribuyen después para un uso diferente. Las aplicaciones que emplean MCS deben tener tiempo para adaptarse a la desaparición de un identificador de canal o de usuario, antes de que el identificador reaparezca.

En los casos en que se requiere una fusión "inconsutil" de dominios activos, se puede evitar conflictos seleccionando identificadores de canal de una subgama única de cada proveedor. Las subgamas pueden ser creadas dividiendo identificadores de canal atribuidos dinámicamente 1001,..65535 en varias bandas. Además, dentro de una subgama, se pueden asignar secuencialmente identificadores en un sentido, desde arriba o desde abajo. Los proveedores que emplean asignación de subgama obedecerán aún a todos los aspectos de este protocolo, incluido el

procedimiento para la fusión de dominios. De este modo, acomodarán proveedores pares que pueden carecer de subgama únicas.

NOTA – Corresponde a la realización local decidir el modo en que un proveedor elige una subgama a partir de la cual efectuar atribuciones. Las conferencias organizadas previamente pueden ser agrupadas por un sistema de gestión de red.

Los identificadores de testigo, a diferencia de los de canal, no son atribuidos, por lo que la línea de demarcación representada por el valor 1001, no tiene ningún significado para ellos. Un identificador de testigo libre con un valor dado está estáticamente disponible para ser tomado o inhibido en cualquier momento, a condición de que se respete el límite impuesto por el dominio al número total de identificadores que se estén utilizando a la vez.

### 13.8 Estado del testigo

El estado del testigo se define formalmente en la cláusula 7. Se retorna como un componente de las MCSPDU de confirmación de testigo y se utiliza para actualizar el registro para un identificador de testigo en la base de información de los proveedores subordinados. El estado del testigo no tiene que comunicarse directamente mediante una primitiva de confirmación al usuario iniciador, sino que puede comunicarse indirectamente mediante un valor de resultado.

Cuando más de un valor de estado del testigo describen la relación de un usuario dado con un identificador de testigo especificado, el orden de preferencia debe ser el siguiente: *Recibiente de sí mismo* debe comunicarse primero, si cabe, para recordar al usuario que se debe responder a una indicación MCS-CESIÓN-TESTIGO. El siguiente valor preferido es *cediéndose así mismo* para recordar al usuario que se está actuando en una operación no completada, y el siguiente es *tomado por sí mismo o inhibido por sí mismo*. Los últimos en el orden de preferencia son los restantes valores de estado, que reflejan el estado actual del testigo atendiendo exclusivamente a su uso por otros participantes.

Un proveedor MCS se basa en la preferencia especificada con respecto a los valores de estado de testigo para actualizar correctamente el estado del testigo en su base de información.

### 13.9 Notificación de capacidades

Mediante los mecanismos de extensibilidad del protocolo V3 pueden implementarse capacidades en el MCS añadiendo componentes a las PDU descritas en esta Recomendación o incluso añadiendo nuevas PDU sin que se requiera una nueva versión del protocolo. Un nodo puede actuar sobre los parámetros de la PDU que entiende, retransmitiendo intactos los parámetros que no reconoce.

Para poder utilizar un capacidad, los nodos del dominio deben cumplir determinadas condiciones. Dichas condiciones o reglas pueden incluir aspectos tales como el grado de participación de otros nodos o la presencia o ausencia de nodos de versión 2. Así, por ejemplo, una cierta capacidad puede requerir la participación de todos los nodos del dominio. Por otro lado, para que sea de utilidad, otra capacidad puede exigir la participación de dos o más nodos. Dicho de otra forma, a cada capacidad se asocian un conjunto de reglas, debiendo cada nodo determinar si utiliza dicha capacidad en base a dichas reglas y al estado del dominio.

Las PDU **CapabilitiesNotificationRequest** y **CapabilitiesNotificationIndication** proporcionan la forma de facilitar la notificación de capacidades en una cumbre V3. La PDU **CapabilitiesNotificationRequest** transporta capacidades que algunos nodos desean que se añadan o se supriman de la lista de capacidades de la cumbre. Además, esta PDU se utiliza también para notificar a otros nodos cambios de estado de las conexiones V2. La PDU **CapabilitiesNotificationIndication** se utiliza para cambiar la lista de capacidades de la cumbre y para informar del estado de las conexiones V2. En el resto de esta subcláusula la PDU

**CapabilitiesNotificationRequest** se denominará "**petición**" y la PDU **CapabilitiesNotificationIndication** de denominará "**indicación**".

### **13.9.1 Identificadores de capacidad**

Una capacidad es un comportamiento de alto nivel del sistema, que normalmente requiere la cooperación entre dos o más nodos. Conforme se desarrollan las capacidades, se asigna un identificador (ID) a cada una de ellas. Este ID puede estar normalizado y documentado en esta Recomendación. Por lo tanto, un ID de capacidad designa a un conjunto de parámetros (no necesariamente de la misma PDU), probablemente un conjunto de valores para dichos parámetros, así como el procesamiento que dichas PDU sufren en los nodos participantes.

### **13.9.2 Cumbres de la versión 3**

No es habitual que toda la jerarquía de un dominio esté exclusivamente formada por nodos V3. Según lo establecido en esta Recomendación en el sentido de que la versión del protocolo se negocie para cada conexión, es posible que un dominio conste de una combinación de nodos V2 y V3. En esta Recomendación (revisada) se estipula asimismo que un nodo con capacidades V3 sólo debe ser fuente y sumidero de PDU V2 si su conexión ascendente negocia con su nodo superior la utilización del protocolo V2. Dichos nodos son equivalentes a nodos V2 y como tales se consideran en el resto de la subcláusula 13.9.

Una cumbre V3 es un grupo de nodos V3 conectados de forma contigua y situados en la parte superior de la jerarquía del dominio. Para que exista una cumbre V3 el proveedor superior del dominio debe ser un nodo V3. Los nodos que componen la cumbre pueden intercambiar información de las capacidades, pero no pueden intercambiar dicha información con ninguno de los nodos V2 del dominio. Ello puede impedir la utilización de una determinada capacidad. En la cumbre V3, al proveedor superior del dominio se le conoce como el "proveedor de cumbre V3".

### **13.9.3 Reglas para la selección de una capacidad**

Existen varios factores que influyen en que se haga uso de una determinada capacidad en una cumbre V3. Estos factores constituyen normalmente un índice de la participación que otros nodos tienen en dicha capacidad. Supóngase una determinada capacidad que exija que todos los nodos de un dominio realicen simultáneamente una determinada tarea. Se podría intercambiar una PDU entre todos los nodos que, en un instante dado, deban realizar dicha tarea. Para que ello ocurra, deben participar, por definición, todos los nodos del dominio. Ello no sería posible si en el dominio hubiera un nodo V2 o un nodo V3 que no tuviera implementada esta capacidad. Por el contrario, algunas capacidades no requieren la participación de todo el dominio o de toda la cumbre.

#### **13.9.3.1 Presencia de nodos V2 en el dominio**

Cualquier nodo V3 de una cumbre conoce si dispone de una conexión descendente con un nodo V2. Mediante una serie de cambios propuestos en el protocolo V3, es posible que el nodo V3 informe a todos los nodos V3 de la cumbre que un nodo V2 dispone de una conexión con la misma. Algunas capacidades pueden resultar inútiles en un dominio en el que haya nodos V2. Otras capacidades pueden no verse afectadas en absoluto. Un nodo puede decidir si hace uso de una determinada capacidad cuando conoce si un nodo V2 está conectado o no está conectado a la cumbre.

#### **13.9.3.2 Participación del nodo V3**

Para utilizar algunas capacidades, éstas requieren de una cierta participación por parte de los nodos V3 de la cumbre. Cada capacidad se puede categorizar en función de que requiera una participación "global" o "parcial". La participación global significa que cada nodo de la cumbre debe participar para ejercer la capacidad en cuestión. La participación parcial requiere la participación de

dos o más nodos. Este requisito de participación se conoce como "indicador de participación". Las capacidades con un indicador de participación parcial se conocen como "capacidades parciales". Las capacidades con un indicador de participación global se conocen como "capacidades globales". El conocimiento del indicador de participación de una capacidad determinada permite al proveedor de la cumbre decidir si dicha capacidad debe de estar disponible en la cumbre.

### **13.9.3.3 Nodos V3 partícipes**

Debido a las marcas de extensión de ASN.1, un nodo V3 puede recibir una PDU con componentes que no reconoce. Sin embargo, el nodo reconoce las PDU a las que están asociadas dichas componentes y retransmite intactas en la PDU las componentes que no reconoce.

Además de ampliar las PDU existentes, el protocolo V3 permite añadir nuevas PDU. El problema que esto conlleva es que los nodos que no reconocen las nuevas PDU no saben como encaminarlas. En esta situación, sería beneficioso que un nodo que generara dicha PDU conociese si los nodos partícipes del trayecto hacia la cumbre V3 soportan dicha PDU. Si así es, el nodo emisor puede asumir que cada nodo a lo largo del trayecto conoce como tratar la PDU. Los cambios propuestos al protocolo V3 contienen una facilidad según la cual, para cada capacidad, todos los nodos son informados sobre si los nodos partícipes soportan dicha capacidad.

### **13.9.3.4 Clase de capacidad**

Una vez que se ha decidido que una determinada capacidad debe estar disponible en una cumbre V3, debe existir una facilidad para, si es necesario, asociar un valor con dicha capacidad. Esto se realiza mediante la clase de capacidad. Cada capacidad se clasifica como de clase MIN, MAX o NULA. Para las capacidades de clase MIN, el proveedor de la cumbre V3 es responsable de determinar el valor mínimo que debe utilizar una capacidad, seleccionando el valor mínimo requerido por cada uno de los nodos participantes. Asimismo, para capacidades de clase MAX, el proveedor de la cumbre V3 es responsable de determinar el valor máximo que debe utilizarse. Una capacidad de clase NULA no tiene un valor asociado a la misma.

Una vez que una capacidad se normaliza, es de esperar que la clase de capacidad asociada también se normalice. Cada vez que un nodo inferior pide una determinada capacidad, el proveedor de cumbre debe considerar que sólo existe una sola clase de capacidad asociada con dicha capacidad. Si el proveedor de la cumbre recibe una petición de una capacidad con clase de capacidad distinta a la de una petición original, deberá suponer que se trata de una condición de error e ignorar la petición.

### **13.9.4 Establecimiento del proveedor de cumbre V3**

Cada cumbre V3 se establece inicialmente por un solo nodo V3. El nodo V3 se convierte en el "proveedor superior" de un dominio de un solo nodo. El proveedor de cumbre sólo conoce sus propias capacidades.

Cada una de las capacidades del proveedor de cumbre exige participación parcial o global en dicha cumbre de un solo nodo. La lista de capacidades de la cumbre no incluye ninguna de las capacidades parciales debido a que éstas requieren, por definición, la participación de dos o más nodos. La cuestión radica en si las capacidades que requieren participación global deben incluirse en la lista. Puede argumentarse que "todos los nodos" de la cumbre participan y, por lo tanto, la lista debe incluir todas las capacidades globales del nodo. Si alguna de dichas capacidades puede ser llevada a cabo, debe obviamente de tratarse de una capacidad que requiera la participación de un solo nodo. Dichas capacidades no constituyen el objeto de esta Recomendación. Este protocolo es irrelevante para cualquier capacidad que pueda realizarse mediante un solo nodo. Por todo ello, la lista de capacidades de la cumbre se considera vacía si la cumbre consta de un solo nodo V3.

### 13.9.5 Ampliación de la cumbre V3

Una vez que se ha establecido el proveedor de la cumbre V3, ésta se amplía conforme otros nodos V3 se conectan al proveedor de la cumbre. Cada nuevo nodo hoja genera una **petición** que se envía al proveedor de la cumbre. Dicha PDU contiene las capacidades y los aspectos de dichas capacidades que el nodo hoja desea que se ubiquen en la lista de capacidades de la cumbre.

A partir de dicha PDU el proveedor de la cumbre puede hacer dos cosas. En primer lugar, puede formular cual es la información de estado de las capacidades solicitadas por la nueva conexión. En segundo lugar, y en base a información de estado de toda la cumbre, puede hacer un juicio cabal sobre que capacidades deben incluirse en la lista de capacidades de la cumbre. Se genera y transmite hacia abajo entonces una **indicación** para informar al nodo hoja del contenido de la lista de la cumbre, así como del estado de los nodos V2 conectados a la cumbre.

De igual forma, la cumbre V3 puede seguir creciendo. Conforme nuevos nodos hoja se conectan a nodos intermedios, se genera una **petición** que se envía en sentido ascendente. A partir de la información de esta PDU, el nodo intermedio actualiza su información de estado. En base a la información de estado de su subárbol y al contenido actual de la lista de capacidades de la cumbre, el nodo intermedio tomará alguna acción, probablemente transmitiendo una **petición** en sentido ascendente o una **indicación** en sentido descendente. Eventualmente, una **petición** puede llegar hasta el proveedor de cumbre. Si éste determina que debe modificarse la lista de capacidades de la cumbre, genera y transmite hacia abajo una **indicación** para informar a otros nodos de la cumbre.

### 13.9.6 Procesamiento de las PDU en un nodo hoja

Una vez que un nodo hoja V3 establece una conexión con un nodo V3 superior, debe informar al nodo superior de las capacidades que soporta. Para cada una de las capacidades solicitadas, el nodo hoja especifica la clase e identificador de la capacidad. Si está normalizada, el identificador especificado es aquél que se asigna a la capacidad en esta Recomendación. Si la clase pertenece a la categoría MIN o MAX, el nodo hoja debe también especificar un valor asociado. Si la clase es NULA, no se especifica ningún valor.

Además, el nodo hoja debe especificar el indicador de participación para cada capacidad. En el caso de capacidad parcial, el nodo hoja especifica el valor 1, indicando así al nodo superior que sólo el nodo hoja ha solicitado la capacidad por debajo de esta nueva conexión.

Estas capacidades se empaquetan en el componente AddList (lista de adición) de una sola **petición**. La bandera V2NodePresent se reinicializa en la **petición**. Debido a que se trata de un nodo hoja que se ha incorporado recientemente, no tiene conectados nodos V2. Igualmente, el componente RemoveList (lista de supresión) estará vacío. Cuando se empaqueta, esta PDU se transmite al nodo superior.

El nodo hoja recién conectado desconoce el contenido de la lista de capacidades de la cumbre. El nodo espera ser informado del contenido de dicha lista mediante una **indicación** procedente del nodo superior. El contenido de la lista figurará en el componente AddList de la **indicación**. RemoveList debe estar vacío. La bandera V2NodePresent informa al nodo hoja si existe actualmente un nodo V2 conectado a cualquiera de los nodos de la cumbre. El nodo hoja es informado a partir de entonces de cualquier modificación a la lista de la cumbre mediante una **indicación** procedente del nodo superior. Si cambiase el estado de las conexiones a los nodos V2, el nodo hoja también es informado de ello mediante una **indicación**.

El nodo hoja es informado si el proveedor de la cumbre soporta cada una de las capacidades que figuran en el componente AddList de la **indicación** recibida y si los nodos partícipes existentes entre el nodo hoja y el proveedor de la cumbre soportan dicha capacidad. Con esta información, el nodo hoja puede decidir si hace uso de una determinada capacidad.

En tanto que un nodo permanece como nodo hoja, no necesitará enviar de nuevo una **petición**. Cuando otro nodo (V2 o V3) se conecte desde abajo con dicho nodo, éste se convierte en un nodo intermedio. En 13.9.7 se describe el procesamiento de las PDU en dicho nodo intermedio.

### 13.9.7 Procesamiento de las PDU en un nodo intermedio

Cuando un nodo intermedio recibe una **petición** de una conexión descendente, actualiza en primer lugar la información de estado de la conexión asociada. El nodo se ocupa entonces de las capacidades incluidas en los componentes AddList y RemoveList, así como la bandera V2NodePresent. Aunque una única **petición** puede transportar varias capacidades que requieran actuación, éstas se procesan de forma independiente. En consecuencia, el nodo intermedio puede tener que transmitir varios elementos de información a nodos adyacentes. El nodo combina dichas acciones de forma que, como consecuencia del procesamiento de la **petición** original, una conexión no reciba más de una PDU.

#### 13.9.7.1 Mantenimiento de la información de estado

Los nodos intermedios mantienen información de estado para cada una de la conexiones descendentes. Tras la recepción de una **petición** desde un subnodo, el nodo intermedio actualiza la información de estado asociada con dicha conexión. En esta información se incluyen dos listas. La primera lista consta de todas las capacidades parciales solicitadas por nodos situados por debajo de dicha conexión. Cada una de dichas capacidades se etiqueta con el valor 1 ó 2. El valor 1 significa que esta capacidad ha sido solicitada por un único nodo ubicado a un nivel inferior a la conexión. Un valor de 2 significa que dos o más nodos han solicitado dicha capacidad.

La segunda lista consta de un conjunto de capacidades globales. En contraposición con las capacidades parciales, las capacidades globales figuran en la lista si han sido solicitadas por todos los nodos ubicados por debajo de la conexión. Para cualquier capacidad que se encuentre en una u otra lista, el nodo intermedio hará igualmente un seguimiento de la clase de la capacidad y de cualquier valor asociado con esta clase.

Finalmente, se dispone de una bandera para cada una de las conexiones descendentes. Dicha bandera indica si un nodo V2 se conecta a la cumbre por medio de dicha conexión.

El algoritmo para la actualización de la información de estado es sencillo. Si el nodo intermedio recibe una petición para añadir una capacidad que no está actualmente en ninguna de las dos listas, la capacidad se añade a la lista adecuada dependiendo de cual sea el indicador de participación. Si se recibe una petición para eliminar una capacidad que actualmente está en una de las dos listas, se suprime la capacidad de dicha lista.

Si el nodo intermedio recibe una petición para añadir una capacidad que actualmente está en una de las dos listas, el procesamiento depende de la clase y del indicador de participación de la capacidad. En caso de que sea clase MAX o MIN, el valor de la clase en la información de estado se actualiza de forma distinta que en la **petición**. En el caso de un indicador de participación parcial, el valor asociado en la información de estado se actualiza en caso de que sea diferente.

Finalmente, si el nodo intermedio recibe una **petición** que tenga la bandera V2NodePresent, el valor de dicha bandera se compara con la bandera de la información de estado. Si es diferente, se actualiza la información de estado.

#### 13.9.7.2 Procesamiento de la PDU CapabilitiesNotificationRequest

Un nodo intermedio puede transmitir una **petición** bajo alguna de las condiciones siguientes:

- 1) El nodo recibe una **petición** procedente de una conexión descendente
- 2) El nodo detecta la pérdida de una conexión V3.

3) El nodo detecta una conexión/desconexión directa con un nodo V2.

Las subcláusulas siguientes describen como un nodo intermedio maneja dichas situaciones.

#### 13.9.7.2.1 Capacidades globales en la lista de adición

Cuando un nodo intermedio recibe una **petición** solicitando la adición de una capacidad global, el nodo examina su copia local de la lista de capacidades de la cumbre para determinar si la capacidad solicitada figura en la misma. La siguiente acción que realiza el nodo depende de si la capacidad está en la lista.

Si la capacidad no está en la lista, el nodo intermedio debe determinar si ésta es la primera **petición** que se ha recibido de esa conexión. Si así es, significa que es una **petición** de un nodo hoja recién conectado y no se requiere ninguna acción ulterior. En otro caso, puede ser necesario transmitir una **petición** solicitando que se añada la capacidad, dependiendo de la información de estado de cualquiera de las conexiones descendentes. La **petición** se transmite si el nodo intermedio determina, a partir de su información de estado, que cada nodo de su subárbol soporta dicha capacidad. Esta acción puede eventualmente dar lugar a que la capacidad se añada a la lista de la cumbre.

Si la capacidad solicitada figura en la lista, implica que todos los nodos de la cumbre soportan dicha capacidad. La necesidad de un procesamiento ulterior depende de la clase de la capacidad. Si ésta es de clase MIN, el nodo intermedio debe determinar si el valor asociado es el menor de los valores solicitados por cada nodo del subárbol para dicha capacidad. Igualmente, si la capacidad es de clase MAX, el nodo intermedio debe determinar si el valor asociado es el mayor de los valores solicitados por los otros nodos. Si se cumple cualquiera de las dos condiciones anteriores, el nodo intermedio transmite una **petición** ascendente solicitando que se añada la capacidad con su valor asociado. Ello sirve para informar al nodo superior de que existe un valor menor MIN o mayor MAX en el sentido descendente de su subárbol para la capacidad en cuestión. Si la capacidad es de clase NULA, no se requiere ninguna **petición**. Por último, el nodo intermedio genera una **indicación** para notificar al nodo inferior que añada la capacidad (y su valor asociado, si existiera) a su copia local de la lista de capacidades de la cumbre.

Cuando un nodo intermedio recibe una **petición** de un nodo hoja recién conectado, puede tener que tomar acciones debido a capacidades globales que NO han sido solicitadas por el nodo hoja. La primera petición de un nodo hoja contiene una lista de todas las capacidades globales que dicho nodo soporta. Si el nodo intermedio tiene capacidades globales en su propia copia de la lista de la cumbre que no son solicitadas por el nodo hoja, significa que el nodo hoja no soporta dichas capacidades. Por definición, dichas capacidades pasan a estar entonces indisponibles para la cumbre y deben ser eliminadas de la lista de la cumbre. En ese momento el nodo intermedio genera una **petición** y coloca dicha capacidad en la lista de supresión. Dicha PDU se transmite en sentido ascendente informando al nodo superior de que la capacidad ya no es soportada en este subárbol. Si el nodo intermedio tiene otras conexiones descendentes, genera una **indicación** que se transmite en sentido descendente. Dicha **indicación** contiene la capacidad en la lista de supresión. Por último, el nodo intermedio suprime la capacidad de su propia lista de capacidades de la cumbre.

#### 13.9.7.2.2 Capacidades globales en la lista de supresión

Si un nodo intermedio recibe una **petición** en la que se solicita que se suprima una capacidad global, significa que en algún lugar en sentido descendente de la conexión un nodo que no soporta dicha capacidad se ha incorporado a la cumbre. Dicha capacidad queda indisponible para la cumbre. El procesamiento que se requiere es inmediato. El nodo genera una **petición** solicitando que se suprima la capacidad, la cual se transmite en sentido ascendente. Si existieran otras conexiones descendentes, el nodo intermedio genera una **indicación** solicitando que se suprima la capacidad que es transmitida en todas las conexiones descendentes (excepto en la conexión que originó la **petición**). El nodo suprime entonces la capacidad de su copia local de la lista de capacidades de la cumbre.

### 13.9.7.2.3 Capacidades parciales en la lista de adición

Si un nodo intermedio recibe una **petición** solicitando que se añada una capacidad parcial, el nodo debe determinar si se trata de la primera, segunda o superior petición relativa a dicha capacidad. Ello se conoce en base a la información de estado del nodo intermedio.

Si se trata de la primera petición, el nodo intermedio genera una **petición** solicitando que se añada la capacidad. El valor suministrado en el indicador de participación parcial es 1, indicando con ello que sólo se ha realizado una petición de esta capacidad en el subárbol de dicho nodo intermedio. Si se trata de la segunda petición, el nodo intermedio genera una **petición** solicitando que se añada dicha capacidad y se suministra el valor 2 en el indicador de participación parcial. Si el nodo intermedio determina que esta capacidad se ha solicitado al menos dos veces, no se requiere ninguna **petición**.

A continuación, el nodo intermedio debe examinar su copia local de la lista de la cumbre para comprobar si en ella se encuentra dicha capacidad. Si así es, el nodo genera una **indicación** que incluye la capacidad en el componente lista de adición (AddList) y la transmite hacia la conexión solicitante.

El nodo intermedio debe asimismo tener en cuenta cual es la clase de la capacidad. Este aspecto del procesamiento es idéntico al explicado anteriormente para capacidades globales. Si una capacidad aparece con el mínimo valor MIN del subárbol o el máximo valor MAX del mismo, el nodo superior debe ser informado de ello mediante un **petición**.

### 13.9.7.2.4 Capacidades parciales en la lista de supresión

Cuando un nodo intermedio recibe una **petición** en la que se pide que se suprima una determinada capacidad parcial, el nodo debe determinar si la supresión podría afectar al número de peticiones relativas a dicha capacidad en el subárbol. Si así es, el nodo intermedio debe generar una **petición** para dicha capacidad, solicitando que la capacidad se añada con un indicador de participación parcial de valor 1 o que la capacidad se elimine definitivamente, en función de cual sea el cambio de estado.

Si el nodo intermedio determina el efecto directo sobre la lista de la cumbre, el nodo puede ajustar su propia copia local de la lista y generar **indicaciones** para notificarlo a las restantes conexiones descendentes. Si el nodo no puede determinar dicho efecto, cualquier ajuste provendrá de un nodo superior que sí pueda determinarlo.

### 13.9.7.2.5 Procesamiento de la bandera de presencia de nodo V2

En la práctica, y aunque no se requiera, un nodo intermedio recibirá una **petición** con una bandera V2NodePresent o una lista de capacidades (la lista de adición y/o la lista de supresión), pero normalmente no ambas.

Un nodo intermedio mantiene para cada una de sus conexiones descendentes información sobre si hay algún nodo V2 conectado a la cumbre a un nivel inferior. Cuando un nodo intermedio es consciente por vez primera que un nodo V2 se ha conectado a un nivel inferior, incluida la posibilidad de que la conexión sea directa con dicho nodo, genera una **petición** que sólo contiene la bandera V2NodePresent inicializada. Esta **petición** se envía entonces en sentido ascendente. Asimismo, se envía un **indicación** hacia todas las conexiones que hay en sentido descendente salvo hacia aquélla que generó la **petición** original. De esta forma cada nodo de la cumbre conoce la existencia del nodo V2. Este hecho es tenido en cuenta por cada nodo a la hora de decidir si hace uso de una determinada capacidad.

De igual forma, un nodo intermedio puede recibir una **petición** informándole de que todas las conexiones V2 anteriores se han separado del subárbol. Mediante cada una de dichas **peticiones**, el nodo intermedio puede determinar si existe algún nodo V2 que permanece conectado a su subárbol. Si hubiera alguno, el nodo intermedio no toma ninguna acción. Si se desconectan todos los

nodos V2, el nodo intermedio genera una **petición** con la bandera V2NodePresent reinicializada que se envía en sentido ascendente.

Si todos los nodos V2 conectados a la cumbre se desconectan, el proveedor de la cumbre lo detecta comprobando que ninguna de las conexiones descendentes tiene un nodo V2 en su subárbol, y que no existen nodos V2 directamente conectados a él. Cuando esto ocurre, el proveedor de la cumbre genera una **indicación** con la bandera V2NodePresent reinicializada y la transmite a toda la cumbre.

#### 13.9.7.2.6 Procesamiento de desconexiones del nodo V3

Cuando un nodo de la cumbre detecta que se ha deshecho una conexión descendente, se descarta toda la información de estado para dicha conexión. La información de estado de las restantes conexiones descendentes, si existen, puede afectar al contenido de la lista de capacidades de la cumbre.

Si todos los nodos de la cumbre soportan las capacidades globales, excepto el nodo o los nodos situados justamente debajo de la conexión perdida, dichas capacidades pueden agregarse a la lista de la cumbre. Si el número de peticiones sobre una capacidad parcial de la cumbre es inferior a dos como consecuencia de nodos segregados por debajo de la conexión perdida, la capacidad en cuestión debe desaparecer de la cumbre. El proceso para adaptarse a la conexión perdida comienza en el nodo que detecta que la conexión se ha roto. El nodo maneja esta situación actuando como si se hubiera recibido una **petición** en la que se solicitara la supresión de todas las capacidades de la información de estado para dicha conexión. La conexión la procesa tal como se describe en 13.9.7.2.

#### 13.9.7.3 Procesamiento de la PDU CapabilitiesNotificationIndication

De vez en cuando, un nodo intermedio recibe una **indicación** de su conexión ascendente. La indicación instruye al nodo local para que añada o suprima capacidades a la lista de la cumbre. Estas modificaciones se hacen sobre la lista "local" de la cumbre, y la **indicación** se envía a todas las conexiones descendentes. Si se trata de la primera **indicación** enviada en una conexión, el nodo intermedio transmite toda la lista de la cumbre mediante el componente lista de adición de la **indicación**. Si un nodo intermedio no soporta una capacidad, debe reinicializar la bandera que informa del soporte de la misma por el nodo intermedio.

#### 13.9.7.4 Procesamiento de las PDU en el proveedor de la cumbre

La forma en la que el proveedor de la cumbre procesa las PDU es casi idéntica a como lo hace el nodo intermedio. Al recibir una **petición** de una conexión descendente, el proveedor de la cumbre actualiza la información de estado de la conexión asociada igual que lo hace un nodo intermedio, tal como se describe en 13.9.7.1. El proveedor de la cumbre trata las capacidades globales y las capacidades parciales de forma idéntica a como lo hacen los nodos intermedios, con la excepción obvia de que no se envía ninguna **petición** a un nodo superior. Junto a cada petición para añadir o suprimir una capacidad, el proveedor de cumbre decide si debe modificarse la lista de la cumbre. Si así es, el proveedor de la cumbre genera una **indicación** que contiene las capacidades adecuadas que deben añadirse o suprimirse y transmite esta PDU hacia abajo a través de las conexiones pertinentes.

El protocolo se ha diseñado de forma que cada nodo que establezca que debe modificarse la lista de capacidades de la cumbre debe iniciar dicho cambio notificándolo a sus conexiones descendentes y a su nodo superior. De esta forma, la lista de la cumbre será actualizada en toda la cumbre.

Si un nodo no tiene información suficiente como para determinar si puede hacerse un cambio en la lista de la cumbre, envía una **petición** al nodo superior siguiente, que puede estar capacitado para tomar tal decisión. Todas las decisiones sobre esta materia pueden ser, en última instancia, resueltas por el proveedor de la cumbre.

### 13.10 Arbitraje sobre la versión del protocolo

Esta Recomendación define tres versiones de protocolo (véase más información en la cláusula 9). Sin embargo, la Recomendación T.120 declara obsoleta la versión 1. Un nodo puede rechazar soportar la versión 1 requiriendo la versión 2 (o una superior) durante el establecimiento de la conexión (si un nodo distante *sólo* soporta la versión 1, la conexión puede ser rechazada). En el caso de nodos que deseen soportar la versión 1 para proporcionar retrocompatibilidad, se recomienda que no se permita la mezcla de otras versiones de protocolo en el mismo dominio (según las reglas de la versión de 1994 de la presente Recomendación).

La Recomendación T.120 debe ser (o ha sido) modificada para permitir las versiones 2 ó 3. El parámetro número de la versión de protocolo de los parámetros del dominio se utilizan para negociar la capacidad de ser origen o sumidero PDU de la versión 3. En relación con las PDU, un nodo que implemente la versión 3 debe ser capaz de transmitir y recibir las PDU de versión 2.

El parámetro número de la versión de protocolo se negocia independientemente para cada conexión (con la limitación abajo descrita). Cada nodo de versión 3 debe mantener información relativa a las versiones de protocolo seleccionadas por sus conexiones ascendentes y descendentes. Todo nodo que no sea nodo hoja y que implemente la versión 3 del protocolo debe facilitar la traducción entre las distintas versiones del protocolo (esto se analiza en detalle en la sección 13.11).

Un nodo de versión 3 que haya negociado la utilización de la versión 2 en su dirección ascendente *sólo* negociará la utilización de la versión 2 en sus conexiones descendentes. Esto tiene como consecuencia que se concentra la utilización de la versión 3 en la parte superior de la jerarquía del dominio (esta zona de utilización de la versión 3 se denomina "cumbre V3").

Un nodo de versión 3 que haya negociado la utilización de dicha versión en su conexión ascendente negociará la utilización de la misma en sus conexiones descendentes (aunque la versión 2 será aceptada cuando ello sea preciso).

### 13.11 Interfuncionamiento de la versión de protocolo

La traducción de la versión 2 a la versión 3 se realiza copiando los parámetros de las PDU de versión 2 en PDU equivalentes de versión 3.

La traducción de la versión 3 a la versión 2 se realiza copiando los parámetros de las PDU de versión 3 en PDU equivalentes de versión 2. Los parámetros de la versión 3 que no se correspondan con parámetros válidos equivalentes de la versión 2 se descartan. También se descartan las PDU de versión 3 sin equivalentes de versión 2.

Cuando un nodo recibe una PDU de versión 3 que no reconoce, ésta se descarta (ello no constituye un error).

Cuando un nodo recibe una PDU de versión 3 que *sí* reconoce, la PDU se procesa normalmente. Si la PDU contiene parámetros que no son reconocidos por el receptor, éstos se ignoran. Si la PDU debe ser retransmitida hacia otro nodo utilizando la versión 3 del protocolo, todos los parámetros se retransmiten intactos (incluidos aquéllos que no habían sido reconocidos localmente en dicho nodo).

Consideraciones sobre la utilización de parámetros no normalizados en conferencias multiprotocolo:

NOTA 1 – Antes de utilizar parámetros normalizados, el implementador debe asegurarse que el Interfuncionamiento entre los nodos de versión 2 y versión 3 es satisfactorio. No es posible enviar a un nodo de versión 2 mediante una PDU de versión 2 información que incluya una PDU de versión 3 en la forma de parámetro no normalizado, o como parámetro normalizado más allá el marcador de extensión. Por lo tanto, los parámetros no normalizados sólo son adecuados para ser utilizados en extensiones cuya ausencia no afecte al funcionamiento básico en nodos V2 o V3 (funcionamiento libre de fracaso). Para garantizar un

interfuncionamiento exitoso deben de analizarse en detalle los procedimientos a utilizar por una MCU o por otros nodos partícipes.

NOTA 2 – Cuando se utilicen parámetros no normalizados, debe tenerse en cuenta que dicha utilización debe ser una solución transitoria hasta que se normalice una funcionalidad equivalente.

NOTA 3 – Esta Recomendación describe un mecanismo para el intercambio de información entre nodos a fin de determinar cuales son las capacidades no normalizadas disponibles en una cumbre V3. Deben de considerarse los aspectos de escalabilidad, siendo probable que en conferencias grandes disminuya la probabilidad de que existan capacidades no normalizadas comunes.

NOTA 4 – Debe tenerse en cuenta cual es el coste del procesamiento entre PDU V2 y PDU V3. Además del hecho de que una MCU tiene que generar las PDU de versión 2 eliminando determinados parámetros de las PDU equivalentes de versión 3, así como generar las PDU de versión 3 generando parámetros a partir de PDU equivalentes de versión 2, el coste de incluir parámetros no normalizados es relativamente elevado. La codificación mínima es del orden de 8 octetos, excluyendo el tamaño de los datos incluidos en los parámetros. Ello es equivalente a la cabecera MCS de parámetros normalizados en una PDU de datos y supone una pérdida adicional de la anchura de banda disponible. Tras la recepción de un parámetro no normalizado, un nodo de versión 3 debe analizar cual es su clave e identificar el código del fabricante para determinar su significado. Esto constituye una pérdida de potencia de procesamiento que debe minimizarse siempre que sea posible. En esta subcláusula se describen los procedimientos que deben seguir los nodos que no entiendan parámetros no normalizados.

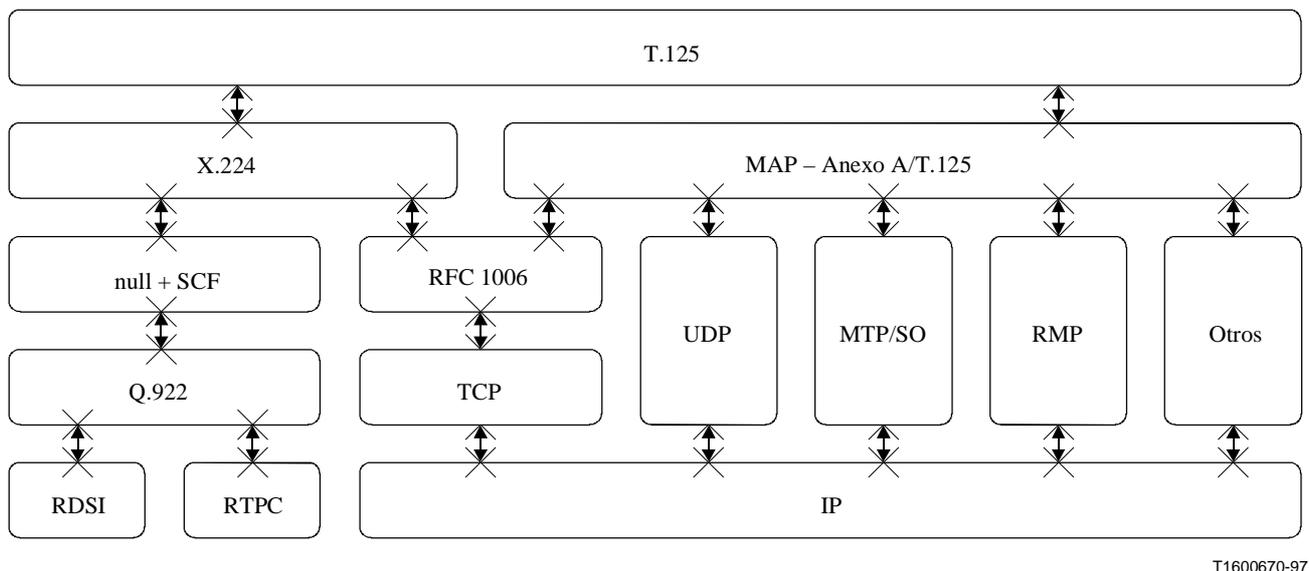
## ANEXO A

### Protocolo de adaptación multidifusión

#### A.1 Campo de aplicación

Este anexo define un protocolo que permite la utilización de los servicios de multidifusión en una conferencia T.120. El protocolo es independiente del tipo de red en la que se utiliza. Los aspectos específicos de una red en particular tales como el formato de las direcciones y las asignación de la dirección de grupo multidifusión, quedan fuera del campo de aplicación de este anexo. Este protocolo también permite la transferencia de datos no fiables.

La figura A.1-1 ilustra como el protocolo de adaptación de multidifusión (MAP, *multicast adaptation protocol*) se adapta al modelo T.120. El MAP se ocupa de la traducción de los servicios T.122 en servicio de multidifusión. Por debajo de MAP puede haber una considerable variedad de protocolos de datos, tales como TCP, UDP, MTP/SO y RMP (MTP/SO y RMP son ejemplos de multidifusión fiable en redes IP).



**Figura A.1-1/T.125 – Modelo T.120 que muestra la relación entre T.120 y las demás capas**

## A.2 Referencias normativas

Las siguientes Recomendaciones del UIT-T y otras Normas Internacionales contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones del presente anexo. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y Normas son objeto de revisiones por lo que se preconiza que los usuarios de este anexo investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y Normas citadas a continuación. Los miembros de la CEI y de la ISO mantienen registros de las Normas Internacionales actualmente vigentes. La Oficina de Normalización de las Telecomunicaciones de la UIT mantiene una lista de las Recomendaciones UIT-T actualmente vigentes.

- Recomendación T.35 del CCITT (1991), *Procedimiento para la asignación de códigos definidos por el CCITT para facilidades no normalizadas.*
- Recomendación UIT-T T.120 (1996), *Protocolo de datos para conferencia multimedia.*
- Recomendación UIT-T T.122 (1998), *Servicio de comunicación multipunto – Definición de los servicios.*
- Recomendación UIT-T T.123 (1996), *Pilas de protocolos de datos específicos de la red para conferencias multimedios.*
- Recomendación UIT-T T X.214 (1995) | ISO/CEI 8072:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Definición del servicio de transporte.*
- Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno – Especificación de la notación básica.*
- Recomendación UIT-T X.691 (1997) / ISO/CEI 8825-2:1998, *Tecnología de la información – Reglas de codificación de notación de sintaxis abstracta uno – Especificación de las reglas de codificación paquetizada.*

### A.3 Definiciones

En esta Recomendación se definen los términos siguientes.

**A.3.1 flujo de datos:** Secuencia de datos transmitida desde un emisor a todos los receptores interesados. Los flujos de datos se identifican inequívocamente mediante los cinco campos siguientes: ID del emisor; ID del canal; nivel de fiabilidad; prioridad; tipos de datos. Cada flujo de datos tiene un contador de secuencia independiente y puede ser encaminado de forma distinta a cualquier otro flujo de datos.

**A.3.2 tipo de datos:** Todas las PDU de datos MCS son no uniformes (ordenadas según la fuente), uniformes (ordenadas de forma global) o datos en representación (datos no uniformes que se envían en multidifusión a través del proveedor de grupo de multidifusión en nombre del originador).

**A.3.3 conexión de protocolo de adaptación multidifusión:** Conexión MCS entre dos nodos a través de MAP. Una conexión MAP está, a su vez, compuesta de una o más conexiones de transporte. No es necesario que nodos adyacentes dispongan de protocolos de multidifusión comunes a fin de establecer una conexión MAP.

**A.3.4 conexión de servicio de comunicaciones multipunto:** Conexión entre dos proveedores MCS adyacentes en la jerarquía del dominio. Puede establecerse por medio de MAP o directamente a través de una de las pilas de protocolos definidas en la Recomendación T.123.

**A.3.5 metacanal:** Canal MCS con un nivel dado de fiabilidad en un margen de prioridad dado. MAP asigna grupos de multidifusión a metacanales.

**A.3.6 multidifusión:** Hace referencia a la capacidad para enviar un paquete de datos a más de un receptor en una sola operación.

**A.3.7 grupo multidifusión:** Grupo de receptores que han expresado su interés en recibir determinados datos en multidifusión. El grupo se identifica inequívocamente mediante la dirección de grupo de multidifusión.

**A.3.8 dirección de grupo multidifusión:** Dirección de red que identifica inequívocamente un grupo multidifusión. Los datos que se envían a esta dirección se entrega, en multidifusión, a todos los nodos que son alcanzables y que se han incorporado al grupo multidifusión.

**A.3.9 proveedor de grupo multidifusión:** Nodo que puede tener o no una conexión MAP ascendente, pero que no goza de protocolos multidifusión comunes con el nodo superior. El proveedor de grupo multidifusión es responsable de la asociación de metacanales con grupos multidifusión. Una vez que se realiza dicha asociación, los datos que se transmiten al metacanal pueden entregarse en multidifusión.

**A.3.10 isla multidifusión:** Grupo de conexiones MAP contiguas en un dominio MCS que ha acordado la utilización de un protocolo de transporte multidifusión para un metacanal determinado.

**A.3.11 emisor multidifusión:** Nodo de una isla multidifusión que transmite un determinado flujo de datos en multidifusión. Para datos no uniformes, se trata del nodo que constituye el punto de entrada. Para datos uniformes y en representación, se trata del proveedor de grupo multidifusión.

**A.3.12 punto de entrada:** Nodo de una isla multidifusión en el que un determinado flujo de datos entra en la isla.

**A.3.13 nivel de fiabilidad:** Hace referencia al nivel con el que se garantiza la transferencia de datos. MAP soporta dos niveles de fiabilidad: totalmente fiable (entrega garantizada) y no fiable (no garantizada).

**A.3.14 número de secuencia:** Es un número acumulativo que se asocia con cada flujo de datos. Se utiliza para el reensamblado de datos no fiables y para facilitar la transición entre la transferencia uniforme de datos unidifusión y multidifusión.

**A.3.15 conexión de transporte:** Conexión unidifusión a través de una de las pilas de protocolo inferiores al MAP.

**A.3.16 protocolo de transporte:** Protocolos que utiliza el MAP para entregar los datos. Existen cuatro tipos de protocolos de transporte: unidifusión fiable; multidifusión fiable; unidifusión no fiable; multidifusión no fiable. Los nodos del dominio arbitran (deciden) cuales son los protocolos de transporte que deberán utilizar.

#### **A.4 Abreviaturas**

En este anexo se utilizan las siguientes siglas.

ATM	Modo de transferencia asíncrona ( <i>asynchronous transfer mode</i> )
IP	Protocolo internet ( <i>internet protocol</i> )
MAP	Protocolo de adaptación multidifusión ( <i>multicast adaptation protocol</i> )
MAPPDU	Protocolo de adaptación multidifusión – Unidad de datos de protocolo ( <i>multicast adaptation protocol – protocol data unit</i> )
MAPSAP	Protocolo de adaptación multidifusión – Punto de acceso al servicio ( <i>multicast adaptation protocol – service access point</i> )
MCS	Servicio de comunicaciones multipunto ( <i>multipoint communication service</i> )
MTP/SO	Protocolo de transporte multidifusión/autoorganizable ( <i>multicast transport protocol/self-organizing</i> )
PDU	Unidad de datos de protocolo ( <i>protocol data unit</i> )
RDSI	Red digital de servicios integrados
RMP	Protocolo multidifusión fiable ( <i>reliable multicast protocol</i> )
RTPC	Red telefónica pública conmutada
SCF	Función de sincronización y convergencia ( <i>synchronization and convergence function</i> )
TCP	Protocolo de control de transmisión ( <i>transmission control protocol</i> )
UDP	Protocolo de datagrama de usuario ( <i>user datagram protocol</i> )

#### **A.5 Visión de conjunto**

Cuando se utiliza protocolos de transporte exclusivamente de unidifusión, T.125 crea conexiones punto a punto para "construir" una conferencia multipunto. Los datos fluyen a través de dichas conexiones, siguiendo la jerarquía del dominio, hasta que alcanzan a todos los receptores deseados. En redes que pueden realizar una auténtica multidifusión, ello supone un considerable desperdicio de recursos ya que los mismos datos a menudo se transmiten varias veces sobre el mismo dispositivo físico. Este anexo describe un protocolo que permite la utilización de la transferencia de datos multidifusión en una conferencia T.120. Este protocolo se denomina protocolo de adaptación multidifusión (MAP, *multicast adaptation protocol*).

MAP es completamente independiente de la red subyacente. Está aislado de la misma por un protocolo de transporte que cumple los requisitos del servicio definidos en 5.2. Algunos de los ejemplos de este anexo hacen referencia a protocolos que se implementan sobre el protocolo internet

(IP, *internet protocol*), pero MAP puede también utilizarse en otros tipos de redes multidifusión, tales como ATM multidifusión o la retransmisión de tramas multidifusión.

Para la gestión de la multidifusión, MAP debe conocer algo más que las conexiones unidifusión. MAP precisa información sobre dominios y canales. Esta información le permite agrupar adecuadamente conexiones unidifusión y utilizar la multidifusión de la forma más eficaz posible. La gestión de dominios y canales forma parte del MCS, pero dicha información se comparte con MAP si es necesario.

A alto nivel, el problema de utilizar la multidifusión en la T.120 puede subdividirse en tres partes:

- 1) *Asignación de grupos multidifusión:* Antes de utilizar la multidifusión, alguien debe asignar las direcciones de grupo multidifusión que no se estén utilizando por nadie más en toda la red. El MAP determina que nodo asigna la dirección. Sin embargo, la forma en que se realiza dicha asignación queda fuera del campo de aplicación del MAP (ello es objeto de trabajo por parte de otros grupos de normalización).
- 2) *Distribución de grupos multidifusión:* Una vez que se asigna una dirección de grupo multidifusión, es necesario distribuirla a aquellos nodos que necesitan utilizarla. MAP maneja la distribución de direcciones de grupo multidifusión.
- 3) *Gestión de transferencia de datos multidifusión:* No es suficiente enviar la dirección de un grupo multidifusión e indicar a todas las partes que comiencen a utilizarla. Existen muchos motivos por los que la multidifusión puede no funcionar de forma uniforme en todo el dominio. Es por lo tanto necesario "probar" la red subyacente para comprobar quien puede recibir los datos multidifusión (y de quien). El MAP realiza esta prueba y utiliza la multidifusión siempre que puede. Si un determinado nodo no puede recibir datos en multidifusión, los datos se envían a dicho nodo en unidifusión.

El MAP maneja asimismo la selección dinámica de los protocolos de transporte unidifusión y multidifusión. Dichos protocolos son seleccionados de forma automática por dos nodos cualesquiera que se comunican mediante MAP. Una vez que se han seleccionado los protocolos, MAP realiza la entrega eficiente de los datos en el dominio. MAP constituye una capa separada por encima de dichos protocolos de transporte.

### **A.5.1 La utilización de la multidifusión**

El MAP soporta la transmisión de datos con dos niveles de fiabilidad, lo cual afecta directamente a la utilización de la multidifusión. Los distintos niveles de fiabilidad utilizarán normalmente distintos protocolos de transporte y, por lo tanto, diferentes grupos multidifusión. El MAP puede asimismo utilizar diferentes grupos multidifusión para distintas gamas de prioridad (para permitir que distintas prioridades tengan un control de flujo independiente). En este anexo se utiliza el término *metacanal* para hacer referencia a un canal MCS con un nivel de fiabilidad dado y ubicado en un nivel de prioridad específico.

Para permitir la utilización de la multidifusión, MAP asigna un grupo multidifusión a cada metacanal de cada isla multidifusión (las islas multidifusión son descritas extensamente en A.5.2). La única excepción a ello la constituyen los canales de ID de usuario, lo cual viola el concepto de multidifusión).

MAP puede gozar de una elevada flexibilidad en la forma en que realiza las asignaciones de grupo. Por ejemplo, podría asignar el mismo grupo multidifusión a cada metacanal (para minimizar el número de grupos multidifusión). O podría asignar un grupo multidifusión separado a cada metacanal (para reflejar el hecho de que cada metacanal puede tener un conjunto distinto de receptores de interés). Estas asignaciones de grupo se controlan mediante un conjunto de políticas que se pueden adaptar a lo largo del tiempo, teniendo en cuenta la evolución tecnológica.

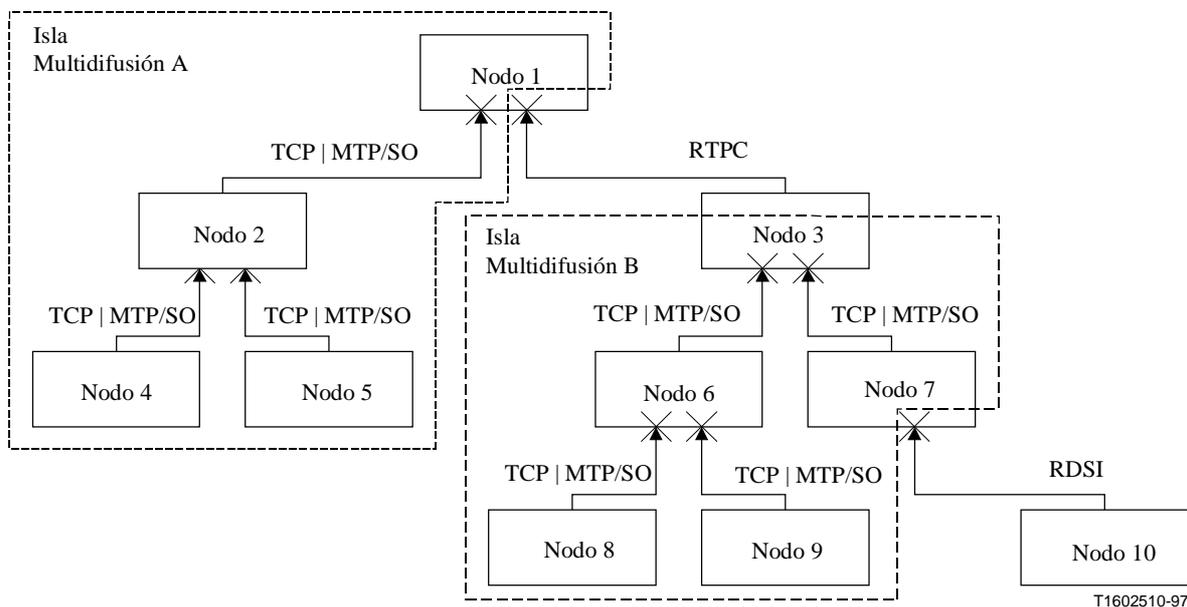
Cuando se transmiten los datos, MAP utiliza el canal, el nivel de fiabilidad y la prioridad para seleccionar el grupo multidifusión adecuado. Ello permite que los datos se entreguen directamente a los receptores deseados sin que se propaguen a través de la jerarquía del dominio (cuando funciona la multidifusión).

### A.5.2 Islas multidifusión

La multidifusión no siempre puede ser utilizada para la entrega de datos a cada uno de los nodos de un dominio. Algunos nodos pueden estar conectados al dominio a través de redes que no pueden realizar la multidifusión (tales como la RTPC o la RDSI). Es asimismo posible que dos grupos de nodos que pueden trabajar en multidifusión estén separados en la jerarquía del dominio por una conexión que no pueda utilizarla. Por lo tanto, es preciso reconocer la existencia de "islas multidifusión" en un dominio.

Un isla multidifusión es un grupo de conexiones MAP contiguas en un dominio MCS que han acordado utilizar un protocolo de transporte particular en un metacanal dado.

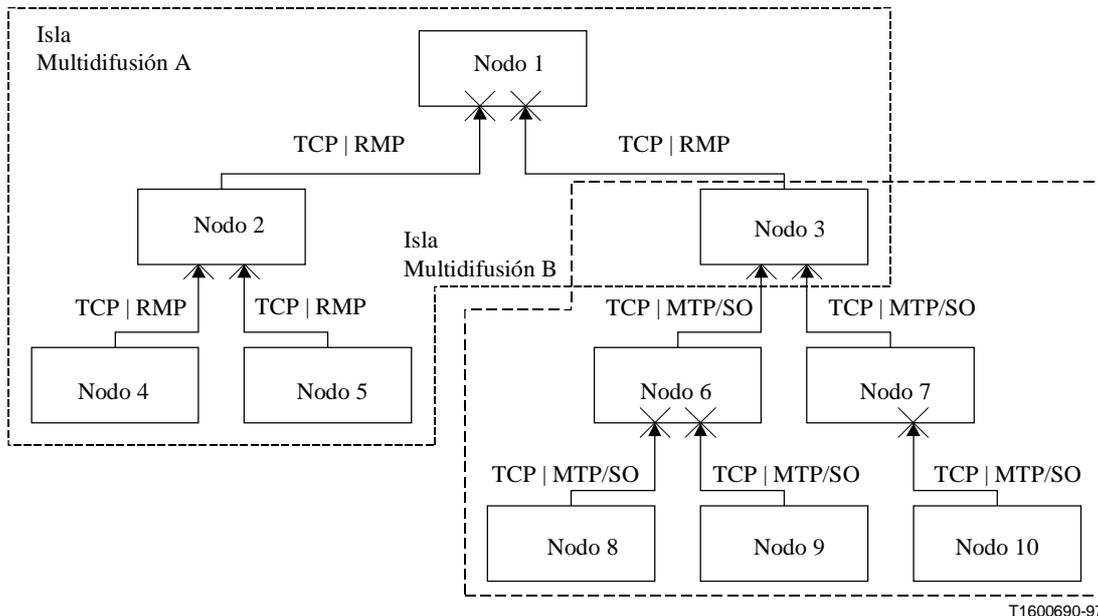
La figura A.5-1 describe una conferencia de 10 nodos. Cada una de las conexiones que constituyen la conferencia se etiqueta con el protocolo o los protocolos de transporte que los nodos adyacentes han acordado utilizar para la transferencia de datos. "RTPC" y RDSI" hacen referencia a las pilas de protocolo definidas en la Recomendación T.123. TCP | MTP/SO" indica una conexión MAP que utiliza TCP para el transporte unidifusión y MTP/SO para el transporte multidifusión (para un metacanal dado). Los nodos 1, 2, 4 y 5 están en la isla multidifusión A, mientras que los nodos 3, 6, 7, 8 y 9 están en la isla multidifusión B. Estas islas se encuentran separadas porque la conexión entre los nodos 1 y 3 es la RTPC (esta conexión no puede utilizarse para transportar información de control).



**Figura A.5-1/T.125 – Islas multidifusión separadas por conexiones que no permiten multidifusión**

Las islas multidifusión pueden también ser divididas como consecuencia de la selección de distintos protocolos de transporte multidifusión para un metacanal dado. La figura A.5-2 representa una conferencia con la misma jerarquía de dominio que la figura A.5-1, pero con una selección de protocolo diferente. Los nodos 1, 2, 3, 4 y 5 están en la isla multidifusión A y han acordado

utilizar RMP para el transporte multidifusión . Los nodos 3, 6, 7, 8 y 9 se encuentran en la isla multidifusión B y han acordado utilizar MTP/SO para el transporte multidifusión. Nótese que el nodo 3 se encuentra en las dos islas multidifusión, actuando como puente entre ellas.



**Figura A.5-2/T.125 – Islas multidifusión separadas por la selección del protocolo multidifusión**

La selección del protocolo se realiza de forma diferente para los protocolos unidifusión y multidifusión. Los protocolos unidifusión se seleccionan *para cada conexión MCS*. Los protocolos unidifusión seleccionados se utilizan para transferir datos entre nodos adyacentes en la jerarquía de dominio MCS. Es perfectamente válido que distintas conexiones utilicen distintos protocolos multidifusión. Los protocolos multidifusión se seleccionan *para cada metacanal*. Cuando se establece la correspondencia entre un metacanal y la multidifusión, cada nodo MAP selecciona los protocolos multidifusión que deben utilizar los nodos ubicados debajo del mismo. Este proceso de selección define los límites de la isla multidifusión *para dicho metacanal*. Es posible que los límites de las islas multidifusión varíen en función de los metacanales.

Como regla general, cada nodo debe seleccionar los protocolos multidifusión que den lugar al menor número de islas multidifusión y que éstas sean lo más grandes posible. Cuando sea posible, cada nodo selecciona el mismo protocolo multidifusión que para él ha seleccionado el nodo superior. En la figura A.5-2, los nodos 6 y 7 deben de haber indicado que no soportan RMP ya que de no haber sido así, el nodo 3 lo habría seleccionado. Esta división en las islas multidifusión es el resultado de que se soporten distintos protocolos de transporte multidifusión.

Este anexo se refiere a los protocolos unidifusión y multidifusión como *protocolos de transporte*. Los nodos que son adyacentes en la jerarquía de dominio MCS realizan un arbitraje para acordar el conjunto de protocolos de transporte que pueden ser utilizados en la conferencia. Durante dicho arbitraje, los nodos adyacentes deben seleccionar un único protocolo unidifusión fiable (y, opcionalmente, un único protocolo unidifusión no fiable). Asimismo deben acordar un conjunto de cero o más protocolos multidifusión que pueden ser utilizados en cada nivel de fiabilidad. Cuando se realiza la correspondencia entre metacanal y multidifusión, un nodo sólo selecciona protocolos multidifusión cuya utilización haya sido acordada durante el proceso de arbitraje por alguno de los nodos que se encuentran más abajo.

### A.5.3 Proveedores de grupo multidifusión

En cada isla multidifusión existe un nodo que es responsable de la asignación de direcciones de grupo multidifusión. Este nodo se conoce como proveedor de grupo multidifusión.

El proveedor de grupo multidifusión es el nodo que tiene conexión MAP ascendente o bien no la tiene pero que no tiene protocolos multidifusión comunes con el nodo superior. El proveedor superior del dominio puede también ser el proveedor de grupo multidifusión para una isla multidifusión, pero ello no es necesario. En las figuras A.5-1 y A.5-2, el nodo 1 es el proveedor de grupo multidifusión para la isla multidifusión A, mientras que el nodo 3 es el proveedor de grupo multidifusión para la isla multidifusión B.

### A.5.4 Ordenamiento de los datos

MCS permite dos tipos de ordenamiento de datos: no uniforme (ordenado según la fuente) y uniforme (ordenado globalmente).

Los datos no uniformes se envían normalmente al grupo multidifusión a través del "punto de entrada" inicial de la isla multidifusión. El punto de entrada es simplemente el nodo por el que acceden los datos a la isla multidifusión. No es el nodo al que se anexa la aplicación emisora en caso de que dicho nodo se encuentre fuera de la isla multidifusión. En la figura A.5-1, si el nodo 10 transmite datos, el nodo 7 se constituye en el punto de entrada para la isla multidifusión B, mientras que el nodo 1 es el punto de entrada para la isla multidifusión A.

Opcionalmente un nodo punto de entrada puede optar por utilizar el proveedor de grupo multidifusión como un representante de multidifusión. Cuando se utiliza esta opción, los datos no uniformes se envían en primer lugar en sentido ascendente en unidifusión, desde donde el proveedor de grupo multidifusión los envía al grupo multidifusión. Esto se describe en más detalle en A.8.4.

A fin de preservar la ordenación global, los datos uniformes deben ser encaminados en sentido ascendente hasta el proveedor superior en unidifusión. Este flujo de datos tiene un punto de entrada "ascendente", pero no se trata de un concepto de interés y no se considerará en lo sucesivo. Los datos se envían al grupo multidifusión a través del punto de entrada "descendente" en su camino de vuelta (para los datos uniformes, se hace referencia al punto de entrada descendente de forma simplificada como punto de entrada). En la figura A.5-1, si el nodo 10 transmite datos uniformes, se tratará en primera instancia de una unidifusión ascendente hacia el proveedor superior. En su camino hacia abajo, el nodo 1 es el punto de entrada para la isla multidifusión A, mientras que el nodo 3 es el punto de entrada para la isla multidifusión B. Para datos uniformes, el punto de entrada será *siempre* el proveedor de grupo multidifusión de cada isla multidifusión.

### A.5.5 Utilización de unidifusión no fiable y de multidifusión

Todos los datos de control se envían sobre conexiones unidifusión fiables. **MAPData** es la única MAPPDU que se transmite en multidifusión o en unidifusión no fiable. El cuadro A.5-1 resume la forma en la que MAP utiliza cada uno de los cuatro tipos de protocolo.

**Cuadro A.5-1/T.125 – Utilización de los tipos de protocolo de transporte**

	<b>Unidifusión</b>	<b>Multidifusión</b>
Fiable	Cualquier MAPDU  <b>MAPData</b> puede contener cualquier MCSPDU	Sólo <b>MAPData</b>  <b>MAPData</b> sólo puede contener una de las MCSPDU siguientes:  <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataIndication</b>
No fiable	Sólo <b>MAPData</b>  <b>MAPData</b> sólo puede contener una de las MCSPDU siguientes:  <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataRequest</b> <b>MCSUniformSendDataIndication</b>	Sólo <b>MAPData</b>  <b>MAPData</b> sólo puede contener una de las MCSPDU siguientes:  <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataIndication</b>

## **A.6 Utilización del MAP**

En esta subcláusula se describe como el MCS gestiona las conexiones y la transferencia de datos cuando utiliza el protocolo de adaptación de multidifusión (MAP). No se pretende aquí describir exhaustivamente las interacciones entre MCS y MAP. Más bien, en esta subcláusula se describe como el MCS se comporta de forma diferente a cuando utiliza la pila de protocolo exclusivamente unidifusión (tal como se define en la Recomendación T.123).

A fin de describir completamente como el MCS se relaciona con el MAP, es de utilidad descomponer T.125 en cuatro unidades funcionales: gestión del dominio, gestión del canal, transferencia de datos y gestión del testigo. Cada una de ellas se describe brevemente a continuación.

### **A.6.1 Gestión del dominio**

Cuando se establece una conexión con un proveedor MCS remoto utilizando una pila de protocolo exclusivamente unidifusión, el MCS crea una conexión de transporte (TC, *transport connection*) para cada prioridad de datos (el número de prioridades forma parte de los parámetros del dominio). Las MCSPDU de control se envían siempre a la TC de mayor prioridad. Las MCSPDU de datos pueden enviarse en todas las TC (la que se utilice en particular depende de la prioridad).

Cuando se establece una conexión con un proveedor MCS remoto utilizando un MAP, el MCS solo crea una conexión MAP sencilla (MC, *MAP connection*), con independencia del número de prioridades de transferencia de datos. Las MCSPDU de control se envían sobre dicha conexión MAP. Las MCSPDU de datos no se envían sobre ninguna MC, sino que se transfieren al MAP en su totalidad (esto se trata en la subcláusula sobre transferencia de datos).

Las MCSPDU de conexión realizan el mismo papel de inicialización de una nueva conexión MCS y de arbitraje sobre cuales deben ser los parámetros del dominio. El MCS envía **Connect-Initial** como la primera MCSPDU de una nueva MC. **Connect-Response** es la primera MCSPDU que envía el MCS llamado como respuesta en una nueva MC. Debido a que sólo hay una MC por cada conexión MCS, no son necesarios **Connect-Additional** ni **Connect-Result**.

Para que el MAP realice adecuadamente sus funciones de encaminamiento, el MCS debe compartir información sobre cada una de las conexiones que se crean. Ello debe incluir: a qué dominio está ligada la conexión, si se trata de una conexión ascendente o descendente y cual es la versión de protocolo que se utiliza en la conexión. Además, cuando el MCS vincula la primera MC con un dominio, debe informar al MAP del número de niveles de prioridad y de fiabilidad que se utilizan en el dominio.

Todas las MCSPDU del dominio que se ocupan de la gestión de éste se envían en una MC exactamente de la misma forma en que se envían en una TC cuando se utiliza una pila de protocolo unidifusión.

### **A.6.2 Gestión del canal**

Todas las MCSPDU del dominio que se ocupan de la gestión de canal se envían en una MC exactamente de la misma forma en que se envían en una TC cuando se utiliza una pila de protocolo unidifusión.

Para que el MAP pueda realizar multidifusión de datos en nombre del MCS, debe conocer los canales (de forma que conozca donde enviar los datos). El MCS debe compartir con el MAP la información siguiente:

- 1) Para cada canal activo, el MAP debe conocer el dominio con el que están asociado el canal y el tipo de canal de que se trata.
- 2) Para cada canal activo, el MAP debe conocer las MC que conducen a los usuarios que se han incorporado.
- 3) Para cada canal activo, el MAP debe conocer si el MCS necesita recibir los datos. El MCS necesita los datos si los puede redireccionar hacia una aplicación de usuario anexada localmente o a otra pila de protocolo.

Los canales estáticos se consideran activos si existe al menos una aplicación de usuario incorporada al subárbol del nodo local. Los datos dinámicos se consideran activos tan pronto como se asignan.

### **A.6.3 Transferencia de datos**

La característica diferencial más significativa de la utilización del MAP radica en el tratamiento de la transferencia de datos. Cuando se utiliza una pila de protocolo exclusivamente unidifusión, el MCS asume toda la responsabilidad del envío de datos a las TC que conducen a los usuarios deseados. Cuando se utiliza MAP, gran parte de esta responsabilidad se delega en ella.

El MCS normalmente envía una MCSPDU de datos al MAP sólo una vez, sin tener en cuenta el número de nodos tengan que recibirla. La única excepción ocurre cuando el MCS ha decidido utilizar una versión de protocolo diferente en algunas de las MC que están vinculadas con el dominio. El MCS debe enviar la MCSPDU de datos adecuadamente codificada para cada versión de protocolo activo. El MAP comprueba que todo el que necesita los datos los recibe.

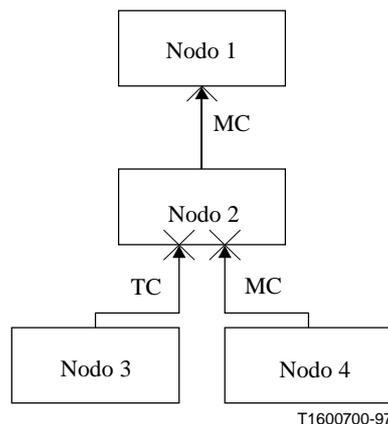
Cuando el MCS recibe datos del MAP, no es necesario que los vuelva a enviar de nuevo al mismo (salvo que se estén utilizando múltiples versiones en cuyo caso el MCS debe realizar la traducción de la versión).

Cuando se envían datos al MAP, el MCS identifica inequívocamente el conjunto de receptores deseados especificando un identificador de dominio, una versión de protocolo MCS y un identificador de flujo de datos. Ello constituye información suficiente dado que el MAP ya conoce cuales son las MC que están vinculadas al dominio así como la versión de protocolo que utiliza cada una de ellas. El MAP puede utilizar el identificador del flujo de datos para encaminar adecuadamente los datos en unidifusión o multidifusión. Nótese que el identificador del flujo de datos también comunica la prioridad relativa de los datos al MAP. Esta prioridad estará comprendida entre 0 y  $N-1$ ,

siendo 0 la máxima prioridad ( $N$  es el número de prioridades que el MCS ha indicado que se pueden utilizar en el dominio). No se requiere traducción de prioridad entre el MCS y el MAP.

Considérese la figura A.6-1 en la que se muestra una conferencia de cuatro nodos. Dos de las tres conexiones que forman la conferencia son MC. La tercera conexión, que vincula el nodo 3 al nodo 2 es una conexión de transporte a través de la pila de protocolo y de RTPC que se define en la Recomendación T.123. Es de utilidad examinar como el MCS del nodo 2 reacciona a los datos emitidos desde cada uno de los cuatro nodos. En este ejemplo se supone que se envían datos no uniformes en un canal en el que existen usuarios incorporados a cada nodo. Nótese que este ejemplo sólo analiza el tratamiento de los datos en el nodo 2.

- 1) Cuando los datos se originan en el nodo 1, se reciben en el nodo 2 mediante MAP. Se pasan al MCS que, a su vez, envía los datos a la conexión de transporte que conduce el nodo 3. MCS NO envía los datos de vuelta al MAP.
- 2) Cuando los datos se originan en el nodo 2, el MCS envía los datos a la conexión de transporte que conduce al nodo 3. También envía los datos (una sola vez) al MAP.
- 3) Cuando los datos se originan en el nodo 3, se pasan al MCS del nodo 2 mediante la pila de protocolo RTPC. El MCS envía los datos (una sola vez) al MAP.
- 4) Cuando los datos se originan en el nodo 4, se reciben mediante MAP en el nodo 2. Se pasan al MCS que, a su vez, envía los datos a la conexión de transporte que conduce el nodo 3. MCS NO envía los datos de vuelta al MAP.



**Figura A.6-1/T.125 – Ejemplo de conferencia de cuatro vías**

Tal como ilustra este ejemplo, el MCS encarga al MAP el encaminamiento de todas las conexiones MAP que están agrupadas lógicamente (por dominio y por versión de protocolo). Esta delegación es necesaria debido a que sólo el MAP conoce cuándo realmente se utiliza la multidifusión y si ésta está en marcha o no. La abstracción que está presente en el MCS es que la multidifusión *siempre* está funcionando.

### **A.6.3.1 Diferenciación de las versiones de protocolo**

Tal como se ha mencionado antes, es necesario diferenciar las PDU de datos basadas en distintas versiones de protocolo. Ello se debe a que el formato de las PDU es diferente. Si la mitad de las conexiones MC de un dominio utilizan una versión de protocolo mientras que la otra mitad utiliza otra diferente, el MCS debe realizar la traducción de cualquier dato que pase de un grupo a otro. Este requisito da lugar al único caso en que el MCS puede necesitar redireccionar los datos de vuelta a la misma pila de protocolo MAP de la que proceden (después de la traducción).

### A.6.3.2 Tratamiento de datos uniformes

Debido al requisito de ordenación global, es necesario tratar los datos uniformes de forma algo diferente que los datos no uniformes. MAP puede determinar si una PDU es uniforme o no lo es examinando el identificador del flujo de datos.

Cuando MAP recibe PDU de datos uniformes del MCS, verifica si existe una MC ascendente para el dominio y la versión de protocolo especificada. Si así es, MAP envía hacia arriba la PDU en unidifusión (ya que debe de ser una petición). Si no existe MC ascendente para el dominio y la versión de protocolo especificada, MAP envía la PDU hacia abajo a todos los receptores incorporados (en multidifusión, si ello es posible).

Cuando MAP recibe una PDU de datos uniformes de una MC descendente, verifica si existe una MC ascendente para el mismo dominio y versión de protocolo. Si así es, MAP envía hacia arriba la PDU en unidifusión (la PDU NO se envía al MCS). Si no existe MC ascendente para el mismo dominio y versión de protocolo, MAP envía la PDU al MCS.

Cuando MAP recibe una PDU de datos uniforme en multidifusión o de la MC ascendente, la PDU se envía al MCS (si ello es necesario) e igualmente se envía en sentido descendente a aquellos nodos que no reciben datos en multidifusión.

### A.6.3.3 Petición versus indicación

Cuando se envían datos no uniformes mediante MAP, los datos pueden llegar a nodos ubicados por encima Y por debajo del nodo actual. Por este motivo, es necesario especificar si el MCS debe identificar dichas PDU de datos como peticiones o como indicaciones.

Si la conexión ascendente es una MC que utiliza la misma versión de protocolo, la PDU enviada al MAP debe de ser una petición. En otro caso, debe ser una indicación. En otras palabras, si un dato SOLO fluye en sentido descendente es una indicación, mientras que si un dato fluye en sentido ascendente es una petición (incluso si también fluye en sentido descendente).

Ello da lugar a una situación en la que es posible que un nodo reciba una **Send-Data-Request** de un nodo que se encuentra jerárquicamente por encima de él. El MCS necesita reconocer que ésta no es una situación de error cuando los datos provienen del MAP, manejando la PDU como si estuviera correctamente orientada.

Nótese que esta situación no se da si se trata de datos uniformes ya que las peticiones se envían en primer lugar en sentido ascendente en unidifusión. Los datos uniformes sólo se transmiten en multidifusión desde la parte superior, garantizándose así que todos los receptores reciben una indicación desde un nodo que se encuentra jerárquicamente por encima de ellos.

### A.6.3.4 Datos en representación

Cuando se envían datos no uniformes, MAP puede hacer la multidifusión de los datos en el punto de entrada o puede decidir utilizar la multidifusión en representación. La forma en que el punto de entrada toma esta decisión es un asunto que queda fuera del campo de aplicación de este anexo.

La multidifusión por medio de un representante significa sencillamente que se hace unidifusión de los datos en sentido ascendente hasta que alcanzan al proveedor de grupo multidifusión, el cual los envía en sentido descendente en multidifusión. Es algo semejante al tratamiento de datos uniformes, excepto en que los datos no tienen que alcanzar al proveedor superior antes de comenzar su camino descendente (sólo tienen que ir hasta el proveedor de grupo multidifusión). Esta técnica puede utilizarse para disminuir la información de estado del flujo de datos en conferencias muy grandes. También puede utilizarse para permitir que se haga multidifusión de los datos cuando el punto de entrada no puede transmitir en multidifusión.

La utilización de datos en representación es transparente para el MCS, para el cual sólo aparecen como datos no uniformes.

#### **A.6.4 Gestión del testigo**

Todas las MCSPDU del dominio que se ocupan de la gestión del testigo se envían en una MC de la misma forma en que se envían a través de una conexión de transporte cuando se utiliza una pila de protocolo unidifusión.

### **A.7 Utilización de protocolos de transporte**

Esta subcláusula describe los servicios que el MAP espera recibir de los protocolos de transporte.

#### **A.7.1 Protocolos de transporte unidifusión**

Los servicios que se requieren de los protocolos unidifusión son un subconjunto de los definidos en la Recomendación X.214. El MAP no utiliza exactamente los mismos parámetros descritos en dicha Recomendación ni hace uso de los datos enviados. En esta subcláusula se presentan los servicios que *utiliza* el MAP.

Durante el establecimiento de la conexión, el MAP crea una o más conexiones unidifusión con el nodo adyacente. Estas conexiones se utilizan para transportar datos de control y datos de usuario. Si se realiza más de una conexión, éstas pueden ser de distintas prioridades, permitiendo que el MAP indique al protocolo de transporte la importancia relativa de una MAPPDU dada.

En el caso de protocolos unidifusión fiables, el MAP asume que el servicio está libre de errores y ordenado. En el caso de protocolos unidifusión no fiables no pueden hacerse ninguna de las dos hipótesis.

El MAP también asume que todos los protocolos unidifusión fiables pueden segmentar las MAPPDU que sean demasiado grandes para ser transmitidas como una sola unidad. Dichas MAPPDU deben ser reensambladas en el receptor antes de ser enviadas al MAP. Para prevenir la necesidad de manejar MAPPDU arbitrariamente grandes, se recomienda utilizar el límite de 16 384 octetos en el MAP. El MAP debe garantizar que ninguna MAPPDU excede dicho tamaño.

En el caso de unidifusión fiable, el MAP asume que se trata de un servicio "orientado a la conexión". Ello implica que ambos extremos de la conexión son conscientes uno del otro y que se utiliza algún protocolo en el proceso de establecimiento de la conexión. Ello implica que ambos nodos mantienen información de estado de la conexión entre ellos, pudiendo detectar de forma automática la pérdida de paquetes y otros problemas de red que se produzcan. Cuando se crea una conexión unidifusión fiable, MAP aporta su propio protocolo de transmisión (una vez que el protocolo de transporte lo autoriza).

En el caso de unidifusión no fiable, el MAP asume un servicio "no orientado a la conexión". Ello significa que MAP no hace ningún supuesto sobre cual es el servicio que se proporciona. No asume que los datos vayan a estar ordenados, ni que se van a detectar las pérdidas de paquetes ni ningún otro problema de red que se produzca.

En el cuadro A.7-1 se resumen las primitivas de unidifusión que utiliza el MAP.

**Cuadro A.7-1/T.127 – Definición del servicio del protocolo de transporte unidifusión**

<b>Primitiva</b>	<b>Parámetros</b>	<b>Fiable</b>	<b>No fiable</b>
Petición T-Conexión	Dirección llamada, dirección llamante de prioridad	X	
Indicación T-Conexión		X	
Respuesta T-Conexión		X	
Confrmación T-Conexión		X	
Petición T-Datos	Datos de usuario	X	
Indicación T-Datos	Datos de usuario	X	
Petición T-Unidad-Datos	Dirección de destino, prioridad, dirección del origen de datos de usuario, datos de usuario		X
Indicación T-Unidad-Datos			X
Petición T-Desconexión	Motivo	X	
Indicación T-Desconexión		X	

### **A.7.1.1 T-Conexión**

Las primitivas *T-Conexión* se utilizan para crear conexiones unidifusión fiables con nodos remotos.

El nodo llamante invoca una *petición T-Conexión* para iniciar la creación de una nueva conexión. Si dicho nodo es accesible se entrega una *indicación T-Conexión* a la MAP del mismo. El MAP normalmente acepta la conexión invocando una *respuesta T-Conexión*. Ello da lugar a una *confirmación T-Conexión* en el nodo llamante. Una vez que se completa este proceso, la conexión está lista y MAP continua con su parte del establecimiento de la conexión.

#### **Definición de los parámetros**

*Dirección llamada:* Dirección de red del nodo al que se llama.

*Dirección llamante:* Dirección de red del nodo que realiza la llamada.

*Prioridad:* Este campo se utiliza para especificar la prioridad de esta conexión (en relación con otras conexiones). El rango permitido es de 0 hasta 15, siendo 0 la prioridad más elevada.

### **A.7.1.2 T-Datos**

Las primitivas *T-Datos* se utilizan para la transferencia de MAPPDU al nodo remoto en unidifusión fiable.

Cuando el MAP desea enviar una MAPPDU en unidifusión fiable, invoca una *petición T-Datos*. Ello da lugar a un *indicación T-Datos* en el nodo distante.

#### **Definición de parámetro**

*Datos de usuario:* Es una cadena de octetos de longitud variable. Se utiliza para transportar las MAPPDU.

### **A.7.1.3 T-Unidad-Datos**

Las primitivas *T-Unidad-Datos* se utilizan para transferir las MAPPDU al nodo remoto en unidifusión fiable.

Cuando el MAP desea enviar una MAPPDU en unidifusión no fiable, invoca una *petición T-Unidad-Datos*. Esta da lugar a cero o más indicaciones *T-Unidad-Datos* en el nodo remoto. El orden de las peticiones puede no estar en consonancia con el orden de las indicaciones. Además,

puede no existir una correspondencia biunívoca entre peticiones e indicaciones (algunas peticiones pueden no dar lugar a indicación alguna, mientras que otras peticiones pueden dar lugar a varias indicaciones).

### **Definición de parámetros**

*Dirección de destino:* Es la dirección de red del nodo que debe recibir los datos.

*Dirección fuente:* Es la dirección de red del nodo que origina la llamada.

*Prioridad:* Este campo se utiliza para especificar la prioridad de esta MAPPDU (en relación con todas las demás MAPPDU). El rango permitido es de 0 hasta 15, siendo 0 la prioridad más elevada.

*Datos de usuario:* Es una cadena de octetos de longitud variable. Se utiliza para transportar las MAPPDU.

#### **A.7.1.4 T-Desconexión**

Las primitivas *T-Desconexión* se utilizan para terminar conexiones unidifusión fiables existentes.

Un lado invocará normalmente una *peticion T-Desconexión*, que rompe la conexión y provoca que el otro extremo reciba una *indicación T-Desconexión*. Sin embargo, si la conexión cae debido a problemas de la red, ambos extremo reciben una *indicación T-Desconexión*.

### **Definición de parámetro**

*Motivo:* Contiene el motivo por el que se cae la conexión. Ello se deberá a uno de los motivos siguientes: iniciada por el usuario, iniciada por el proveedor o fallo no especificado.

#### **A.7.2 Protocolos de transporte multidifusión**

Esta subcláusula describe los servicios que requiere el MAP para la gestión de grupo de multidifusión. Ello incluye la incorporación a un grupo multidifusión antes del flujo de datos, la transmisión y recepción de datos en multidifusión y el abandono de un grupo cuando ya no se necesitan los servicios multidifusión.

Todos los flujos de datos de una isla multidifusión son inicialmente unidifusión. En respuesta al flujo de datos, el MAP asigna y distribuye direcciones de grupo multidifusión que deben utilizarse en la optimización del tráfico de la red. En cada nodo de la isla, el MAP decide si es necesario que dicho nodo se incorpore a un grupo asignado. Si así es, recurre a los protocolos multidifusión para realizar dicho servicio.

Cuando el MAP decide unirse a un grupo multidifusión, puede especificar uno de los tres modos siguientes: sólo transmisión, solo recepción o transmisión/recepción. Algunos protocolos multidifusión pueden optimizar su comportamiento mediante el conocimiento de cual es la combinación requerida, de tal forma que se les hace llegar dicha información durante la operación de incorporación al grupo.

Para la multidifusión fiable, el MAP asume que el servicio está libre de errores, probablemente después de una pérdida indeterminada de datos iniciales en cada transmisor. Este supuesto no se hace para la multidifusión no fiable.

MAP nunca asume que los datos multidifusión van a estar ordenados. Sin embargo, en el caso de multidifusión fiable, el MAP asume que se llenarán todos los espacios vacíos dejados por datos en un orden incorrecto. Si un nuevo nodo comienza la transmisión hacia un grupo multidifusión, el protocolo multidifusión fiable *no* comienza a enviar los datos al MAP hasta que no se haya estabilizado la recepción de lo que procede de dicho nodo (es decir, se llenan todos los huecos vacíos en recepción). Esto significa que en cada receptor el flujo de datos en recepción comienza después de una período de pérdidas iniciales distinto.

No es preciso que los protocolos multidifusión soporten la segmentación. Ello se debe a que **MAPData** es la única MAPPDU que se envía en multidifusión, siendo segmentada al tamaño adecuado por el MAP antes de ser enviada al protocolo multidifusión.

En el cuadro A.7-2 se resumen las primitivas multidifusión utilizadas por el MAP.

**Cuadro A.7-2/T.125 – Definición del servicio del protocolo de transporte multidifusión**

<b>Primitiva</b>	<b>Parámetros</b>
Petición MT-Incorporación Confirmación MT-Incorporación	Dirección de grupo de multidifusión, prioridad, banderas de transmisión / recepción (tx/rx)
Petición MT-Datos Indicación MT-Datos	Datos de usuario Datos de usuario
Petición MT-Abandono Indicación MT-Abandono	Motivo

### **A.7.2.1 MT-Incorporación**

Las primitivas *MT-Incorporación* se utilizan para la incorporación a un grupo multidifusión.

Existen banderas en la *petición MT-Incorporación* que permiten al MAP especificar si desea transmitir, recibir o ambas cosas. El estado de dichas banderas puede cambiar mediante la emisión de *peticiones MT-Incorporación* adicionales sin necesidad de invocar *petición MT-Abandono*.

*Confirmación MT-Incorporación* informa al MAP si la *petición MT-Incorporación* pendiente ha tenido éxito.

#### **Definición de parámetro**

*Dirección de grupo multidifusión:* Es la dirección del grupo multidifusión al que se realiza la incorporación. Un valor NULO indica que el protocolo de transporte debe crear un nuevo grupo y devolver la dirección al MAP para su distribución.

*Prioridad:* Este campo se utiliza para especificar la prioridad de este grupo (en relación con todos los grupos restantes). El rango permitido es de 0 hasta 15, siendo 0 la prioridad más elevada.

*Banderas TX/RX:* Estas banderas permiten al MAP indicar al protocolo multidifusión cual de los tres modos requiere: sólo transmisión, sólo recepción o transmisión/recepción.

### **A.7.2.2 MT-Datos**

Las primitivas *MT-Datos* se utilizan para la transferencia de las MAPPDU a todos los nodos de un grupo en multidifusión.

Estas primitivas trabajan de la misma forma para protocolos fiables y para protocolos no fiables. Cuando MAP desea enviar una MAPPDU en multidifusión, invoca una *petición MT-Datos*. Ello da lugar a una *indicación MT-Datos* en todos los nodos que deben recibir los datos. El orden de las peticiones puede no coincidir con el orden de las indicaciones. En el caso de multidifusión no fiable, puede no existir una correspondencia biunívoca entre peticiones e indicaciones (algunas peticiones pueden no dar lugar a indicación alguna, mientras que otras pueden provocar múltiples indicaciones).

#### **Definición de parámetro**

*Datos de usuario:* Cadena de octetos de longitud variable. Se utiliza para transportar las MAPPDU.

### A.7.2.3 MT-Abandono

Las primitivas *MT-Abandono* se utilizan para abandonar un grupo multidifusión. La *indicación MT-Abandono* se utiliza para informar al MAP que su pertenencia al grupo ha finalizado de forma prematura.

Cuando MAP decide que no necesita transmitir o recibir datos en un grupo multidifusión dado, invoca una *petición MT-Abandono*. Este protocolo multidifusión debe detener inmediatamente el envío de datos desde el grupo al MAP. Esta primitiva no tiene confirmación.

La *indicación MT-Abandono* sólo tiene lugar si el protocolo multidifusión necesita indicar al MAP que ya no está incorporado al grupo. El abandono de otros nodos del grupo *no* provoca esta indicación.

#### Definición de parámetro

*Motivo*: Contiene el motivo por el que se abandonó el grupo multidifusión. Puede ser: iniciado por el proveedor o fallo no especificado.

### A.7.3 Información local

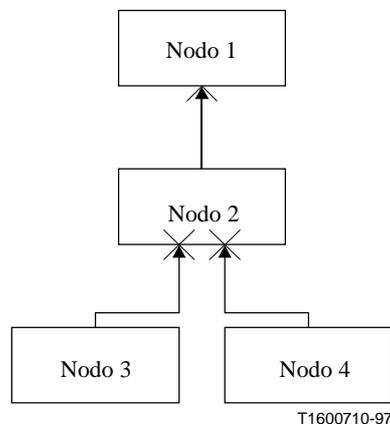
Para que el MAP pueda manejar simultáneamente varios protocolos de transporte sin un conocimiento previo de cuales son sus capacidades, es necesario que exista información local de la configuración. Cada protocolo de transporte se describe de forma genérica ante el MAP. El MAP utiliza esta información durante el proceso de arbitraje del protocolo de transporte, la cual incluye los parámetros siguientes:

- *Identificador del protocolo de transporte*: Se utiliza para identificar inequívocamente el protocolo de transporte durante el proceso de arbitraje del protocolo.
- *Tipo de protocolo de transporte*: Cada protocolo de transporte debe identificarse como perteneciente a uno de los tipos siguientes: unidifusión fiable, unidifusión no fiable, multidifusión fiable o multidifusión no fiable.
- *Peso de preferencia*: Peso de referencia que se utiliza para determinar cual es el protocolo de transporte preferido entre varios.
- *Número máximo de conexiones*: Sólo es relevante para protocolos unidifusión. Es el máximo número de conexiones que el protocolo unidifusión recomienda *por cada conexión MAP*.
- *Tamaño máximo de la carga útil*: Es el tamaño máximo de la carga útil que puede manejar el protocolo. Nótese que para la unidifusión fiable, el protocolo de transporte *debe* poder manejar hasta 16 384 octetos (con segmentación si ello es necesario), de forma que sólo se trata de un tamaño máximo recomendado.
- *Datos de configuración*: Cuando el MAP está listo para realizar el arbitraje de protocolos con un nodo remoto, cada protocolo candidato le envía un bloque de datos de configuración (que puede ser NULO). Esta información se transfiere al nodo remoto durante el arbitraje de protocolo, y allí es transferido al protocolo equivalente de dicho nodo (si existe). En el nodo llamado el MAP obtiene los datos de configuración que deben ser devueltos desde su pila local a fin de completar el proceso de arbitraje de protocolo. Estos datos de configuración deben de ser recibidos de forma separada por cada conexión MAP que se arbitra.

La información contenida en el bloque de datos de configuración se define de forma separada para cada protocolo de transporte. El formato de estos bloques puede ser normalizado (junto con los perfiles del protocolo de transporte) o de carácter propietario. Puede utilizarse, por ejemplo, para comunicar al llamante un número de puerto IP dinámico.

## A.8 Especificación de protocolo

Esta subcláusula contiene una descripción paso a paso de como trabaja el MAP, incluyendo los diagramas de tiempo de varias PDU. La figura A.8-1 ilustra la jerarquía del dominio que se ha utilizado para generar estos diagramas.



**Figura A.8-1/T.125 – Dominio al que se hace referencia en los ejemplos subsiguientes**

### A.8.1 Establecimiento de la conexión unidifusión fiable inicial

Durante el procesamiento de una petición de conexión MCS, lo primero que debe hacerse es crear una conexión unidifusión fiable. La conexión se realiza utilizando el protocolo especificado por el MCS o el protocolo por defecto de la red (si no se especifica protocolo alguno). Esta conexión se utiliza para transmitir todas las PDU de control (tanto MCS como MAP). La primera MAPPDU que el nodo llamante envía a través de la nueva conexión es **MAPConnectRequest**. Esta MAPPDU contiene los campos siguientes:

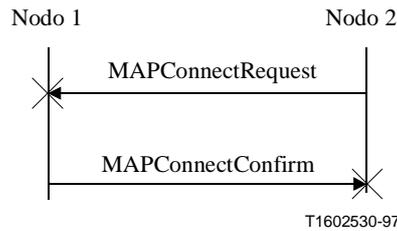
- Un número de versión de protocolo que identifica la versión del MAP que el nodo llamante desea utilizar.
- Un punto de acceso al servicio MAP (MAPSAP, *MAP service access point*), que identifica quien debe aceptar la llamada en el nodo llamado (lo especifica el MCS como parte de la dirección de red).
- Un identificador (ID) de referencia del dominio que puede vincular esta conexión con un dominio en particular (para la conexión inicial este campo se pone a cero).
- Valores de prioridad baja y alta que indican la gama de prioridades que deben enviarse en esta conexión (para la conexión inicial ambos campos se ponen a cero).

Si el nodo llamado desea aceptar la llamada, debe transmitir un **MAPConnectConfirm** al nodo llamante. Esta MAPPDU contiene los campos siguientes:

- Un número de versión de protocolo que identifica la versión de MAP que *será* utilizada (debe ser un valor menor o igual que el especificado por el llamante).
- El MAPSAP del nodo llamado que acepta la conexión entrante.

Si el nodo llamado no desea aceptar la llamada entrante, debe transmitir una **MAPDisconnectRequest** al nodo llamante. Esta MAPPDU contiene el motivo del rechazo. También contiene un campo que indica que no se requiere confirmación alguna, de forma que la conexión se deshace inmediatamente.

La figura A.8-2 muestra el diagrama de tiempos de un establecimiento de llamada con éxito.



**Figura A.8-2/T.125 – Temporización de las MAPPDU de conexión**

### A.8.2 Arbitraje de protocolos de transporte

Una vez que se ha establecido la conexión unidifusión fiable, el MCS realiza el establecimiento de su conexión. Cuando el MCS ha finalizado el establecimiento de su conexión, pasa la siguiente información al MAP: el dominio de conexión al que está vinculado, la versión de protocolo que el MCS utilizará en la conexión y cuantas prioridades y niveles de fiabilidad se utilizarán en el dominio (todo ello se proporciona al MAP cuando se fijan por vez primera los parámetros del dominio). Una vez que el MAP dispone de dicha información, puede realizar el proceso de arbitraje de protocolos de transporte.

El nodo llamante envía una **MAPArbitrateProtocolsRequest** que contiene una lista de los protocolos de transporte que está dispuesto a utilizar (unidifusión y multidifusión). Como mínimo, esta lista debe incluir el protocolo unidifusión fiable utilizado para crear la conexión actual. Es opcional soportar otros protocolos. Después de comparar esta lista con sus propias capacidades, el nodo llamado responde con un **MAPArbitrateProtocolsConfirm** que indica los protocolos que deben utilizarse.

Si no se seleccionan protocolos no fiables, los datos no fiables se envían utilizando protocolos fiables (en el caso unidifusión los datos no fiables se envían a través de las mismas conexiones que los datos fiables). Si el MCS ha indicado que en el dominio no debe haber datos no fiables, no es necesario incluir protocolos no fiables en cualquiera de las MAPPDU de arbitraje. Si no se seleccionan protocolos multidifusión, todos los datos se envían mediante unidifusión (entre estos dos nodos).

La confirmación incluye siempre exactamente un protocolo unidifusión fiable. Opcionalmente incluirá un protocolo unidifusión no fiable. En el caso de multidifusión, incluirá todos los protocolos que ambos nodos han acordado que se pueden utilizar (puede haber más de uno para cada nivel de fiabilidad).

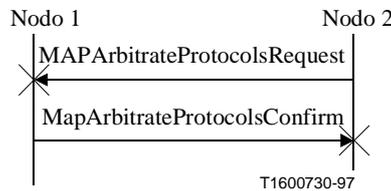
En el caso de protocolo unidifusión fiable, los nodos arbitran también el número de conexiones de transporte que constituirán esta conexión MAP. Si existe más de una conexión de transporte, los datos unidifusión se dividen entre ellos de acuerdo con su prioridad. Las distintas conexiones MAP pueden estar compuestas por un número distinto de conexiones de transporte, incluso en la misma isla multidifusión. Las subcláusulas siguientes describe en detalle el proceso de creación de conexiones de transporte adicionales.

Las MAPPDU de arbitraje también contienen un identificador de referencia del dominio. Este valor es utilizado más tarde por el emisor para indicar a qué dominio está asociado una MAPPDU, en caso de que el dominio no pueda ser implícitamente inferido. De forma específica, este valor se utiliza cuando se deben crear conexiones unidifusión fiables adicionales y cuando se transmiten datos no fiables.

Cualquier nodo puede provocar que se produzca un nuevo arbitraje de los protocolos de transporte con un nodo vecino simplemente emitiendo una **MAPArbitrateProtocolsRequest**. Para garantizar que no se pierden conexiones, es necesario que el protocolo unidifusión fiable anteriormente seleccionado esté disponible en el menú del protocolo. Si el resultado del nuevo arbitraje es un cambio de los protocolos unidifusión, la transición debe hacerse de forma inmediata (este proceso se describe en detalle en la última subcláusula). Los cambios en el conjunto de protocolos multidifusión no tienen un efecto directo en los grupos multidifusión existentes. Estos cambios sólo se reflejan en ulteriores asignaciones de grupos (este proceso se describe en detalle en la subcláusula sobre asignación y distribución de grupos multidifusión, véase A.8.3).

Entre nodos adyacentes puede darse una "colisión de arbitraje". Esto ocurre cuando ambos nodos emiten una **MAPArbitrateProtocolRequest** al mismo tiempo. Esta situación se detecta en ambos nodos porque ambos reciben una petición de arbitraje cuando estaban esperando una confirmación. Cuando esto ocurre, se ignora la petición procedente del nodo que se encuentra a un nivel inferior (y éste detiene la espera de confirmación). La petición del nodo superior se procesa normalmente.

La figura A.8-3 muestra el proceso de arbitraje cuando el nodo 2 llama al nodo 1.



**Figura A.8-3/T.125 – Temporización de las MAPPDU de arbitraje**

### A.8.2.1 Creación de conexiones unidifusión fiables adicionales

Tal como se ha mencionado en las secciones anteriores, los nodos también realizan arbitraje sobre cual debe ser el número de conexiones unidifusión fiables que deben constituir una conexión MAP. Si deciden utilizar más de una, es necesario crear las conexiones adicionales. En esta subcláusula se utiliza el término *conexión* para hacer referencia a una conexión unidifusión fiable.

El proceso lo inicia en nodo llamante, que hace una llamada al protocolo unidifusión fiable seleccionado a fin de crear el número adecuado de conexiones adicionales. Conforme cada conexión queda lista para ser utilizada, el nodo llamante emite una **MAPConnectRequest** a través de ella. Esta MAPPDU contiene los campos siguientes:

- El número de versión de protocolo es el mismo que se ha seleccionado para la conexión inicial.
- El MAPSAP es el mismo que se ha utilizado para crear la conexión inicial.
- El identificador de referencia del dominio es el que se ha enviado en la **MAPArbitrateProtocolsRequest** en la conexión inicial.
- Los valores de prioridad baja y alta indican la gama de prioridades que serán enviadas a través de esta conexión (tal como se especifica en el MCS, las prioridades requeridas normalmente se distribuyen uniformemente entre las conexiones).

El nodo llamado acepta cada una de las conexiones adicionales emitiendo una **MAPConnectConfirm** sobre cada una de ellas. Esta MAPPDU contiene una copia exacta de los primeros dos campos de la petición anterior.

El rechazo de cualquiera de las conexiones adicionales constituye un error del protocolo. Si uno de los dos nodos no puede crear el número adicional de conexiones acordado, *todas* las conexiones correspondientes a esta conexión MAP deben clausurarse, separando al nodo inferior del resto del dominio.

Es importante que cuando finalice el proceso, se establezca una correspondencia entre todas las prioridades requeridas y una conexión. Conforme se establece cada conexión adicional, se establece una correspondencia entre ésta y una gama de prioridades. Todas las prioridades requeridas que no gozan de una correspondencia explícita con una conexión adicional son manejadas por la conexión inicial. Dichas prioridades *deben* mantenerse contiguas con la de prioridad cero (la cual se ha hecho corresponder con la conexión inicial).

#### A.8.2.2 Cambio de las conexiones unidifusión fiables

Una vez que ha tenido lugar el rearbitraje, puede ser necesario que los dos nodos cambien de forma dinámica a un conjunto diferente de conexiones unidifusión fiables. Ello sucede si se produce una de las condiciones siguientes: los nodos han decidido conmutar a un protocolo unidifusión fiable distinto, los nodos han decidido utilizar un número distinto de conexiones de transporte, los nodos han decidido cambiar el tamaño máximo de la carga útil de las conexiones unidifusión fiables (esto no puede ocurrir mientras aún haya grupos multidifusión activos) o los nodos han intercambiado nuevos datos de configuración para el protocolo unidifusión fiable. Nótese que la transición dinámica no puede tener lugar hasta que no se haya completado el conjunto previo de conexiones unidifusión fiables.

Para migrar de forma dinámica a un conjunto diferente de conexiones, es necesario realizar los pasos siguientes:

- 1) El nodo iniciador del arbitraje debe crear el número acordado de nuevas conexiones con el nodo remoto utilizando el protocolo seleccionado. Este nodo se denomina en los sucesivos "nodo llamante".
- 2) El nodo llamante emite una **MAPConnectRequest** sobre cada una de las nuevas conexiones conforme éstas quedan disponibles. Esta MAPPDU contiene los campos siguientes:
  - El número de la versión de protocolo es el mismo seleccionado en las conexiones anteriores.
  - El MAPSAP es el mismo que se ha utilizado para crear las conexiones anteriores.
  - El identificador de referencia del dominio es el que se ha enviado en la **MAPArbitrateProtocolsRequest** de la conexión inicial anterior.
  - Los valores de prioridad baja y alta indican la gama de prioridades que serán enviadas a través de esta conexión (tal como se especifica en el MCS, las prioridades requeridas normalmente se distribuyen uniformemente entre las conexiones).
- 3) Cuando recibe la petición, el nodo llamado responde enviando una **MAPConnectConfirm** en cada nueva conexión. El número de versión de protocolo es el mismo que el de la petición.
- 4) Cada vez que se recibe una nueva confirmación, el nodo llamante debe conocer cuales son las conexiones que están listas para ser utilizadas. Comparando la gama de prioridades de las conexiones anteriores con la gama de prioridades de las nuevas conexiones que han sido confirmadas, el nodo llamante puede determinar cuando debe suprimirse una de las conexiones anteriores (porque todas sus prioridades quedan cubiertas por nuevas conexiones). Cuando esto ocurre, el nodo llamante emite una **MAPDisconnectRequest** hacia la conexión anterior. En este momento, el nodo llamante comienza el envío de todo el

tráfico unidifusión fiable de la gama de prioridades de la conexión anterior pero sobre la nueva o nuevas conexiones pertinentes.

- 5) Al recibir una petición de desconexión, el nodo llamado emite una **MAPDisconnectConfirm** en la conexión previa. El nodo llamado inicia entonces el envío de todo el tráfico unidifusión fiable de la gama de prioridad de la conexión previa pero sobre la nueva o nuevas conexiones pertinentes. El nodo llamado debe conocer en todo momento cuales son las conexiones previas que han sido desconectadas. Comparando la gama de prioridades de las nuevas conexiones con las de las conexiones anteriores, el nodo llamado determina cuando es seguro comenzar el procesamiento de los datos recibidos en una nueva conexión (esto ocurre cuando se han desconectado todas las conexiones previas que transportaban datos en la gama de prioridad de la nueva conexión).
- 6) Cuando el nodo llamante recibe una confirmación de desconexión, asume que todos los datos pendientes de la conexión anterior han sido eliminados y la conexión suprimida. Comparando la gama de prioridades de las nuevas conexiones con las de las conexiones anteriores, el nodo llamante determina si es seguro iniciar el procesamiento de los datos recibidos en una nueva conexión (esto ocurre cuando se han desconectado todas las conexiones previas que transportaban datos en la gama de prioridad de la nueva conexión).

### A.8.3 Asignación y distribución de grupos multidifusión

Una vez que se ha realizado el arbitraje de un conjunto de protocolos de transporte, el MAP puede comenzar la asignación y distribución de direcciones de grupos multidifusión. Para minimizar el número de grupos multidifusión activos, el MAP no debe asignar una dirección de un grupo multidifusión a un metacanal hasta que se detecte tráfico de datos que puede utilizar dicho grupo. No se producirá pérdida de datos ya que todos ellos se transmiten inicialmente en unidifusión.

Cuando el MAP detecta un flujo de datos para el que no hay grupo multidifusión, puede asignarse una dirección de grupo multidifusión. Este proceso comienza en cada nodo del dominio que no tiene conexiones MAP ascendentes o que tiene una conexión MAP ascendente pero carece de protocolos multidifusión comunes con el nodo superior. Dichos nodos se convierten en proveedores de grupo multidifusión. Estos examinan la lista de protocolos multidifusión soportados por los nodos que se encuentran por debajo de ellos e intentan elegir el protocolo o los protocolos que generan el menor número de islas multidifusión. Idealmente, es posible elegir un único protocolo que sea soportado por *todos* los nodos ubicados más abajo en la jerarquía (permitiendo que todos los nodos inferiores permanezcan en la misma isla multidifusión).

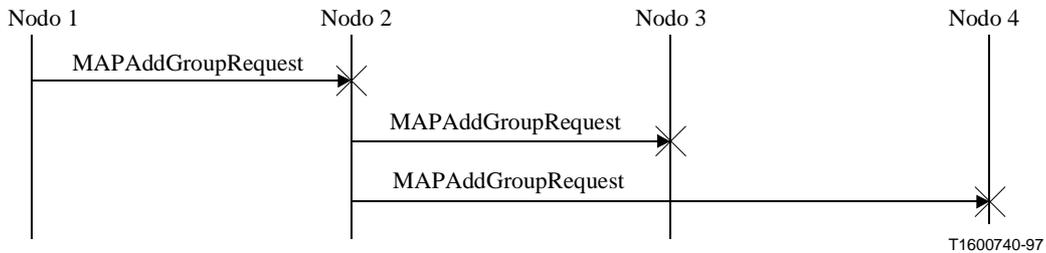
Una vez que se selecciona un protocolo multidifusión para cada nodo inferior, el MAP obtiene una dirección de grupo multidifusión para cada uno de los distintos protocolos que se van a utilizar. Dependiendo de las políticas locales, la dirección de grupo que se va a asignar puede estar siendo utilizada por otro metacanal, o puede ser de nueva asignación. Queda fuera del ámbito de este protocolo el determinar como se realiza esta asignación. Una vez que ha decidido cual es la dirección de un grupo multidifusión, se disemina hacia abajo utilizando **MAPAddGroupRequest**.

**MAPAddGroupRequest** indica al receptor cual es el canal, el nivel de fiabilidad y la gama de prioridad que corresponden a la asignación. También informa al receptor del protocolo multidifusión y de la dirección de grupo que debe utilizar. Tal como se describe en la subcláusula anterior, el hecho de seleccionar protocolos multidifusión para los nodos inferiores da lugar a la definición de límites de islas multidifusión. Todos los nodos que se les asigna el mismo protocolo multidifusión, y direcciones de grupo están, por definición, en la misma isla multidifusión.

Cuando un nodo recibe una **MAPAddGroupRequest**, añade dicho grupo a su lista de grupos multidifusión activos. Si dispone de conexiones MAP descendentes, debe seleccionar un protocolo multidifusión para cada uno de los nodos inferiores. Si ello es posible, debe utilizar el mismo

protocolo que ha sido seleccionado para él por el nodo superior. Si ello no es posible para algunos de los nodos inferiores, debe seleccionar aquellos protocolos que den lugar al menor número posible de islas multidifusión. Una vez que se realiza dicha selección, el nodo obtiene una dirección de grupo multidifusión para cada uno de los protocolos a utilizar y disemina la información hacia abajo utilizando **MAPAddGroupRequest**.

La figura A.8-4 ilustra el proceso de diseminación de la información de grupo multidifusión en sentido descendente.



**Figura A.8-4/T.125 – Temporización del procedimiento de adición de grupo**

#### A.8.4 Gestión del tráfico de datos multidifusión (autoajuste)

Una vez que se ha distribuido una dirección de grupo multidifusión, los nodos de la isla multidifusión pueden comenzar a utilizarla. Sin embargo, los nodos no asumen que la multidifusión comienza a funcionar de forma inmediata. Existen numerosos factores que pueden provocar que fracase la multidifusión. En primer lugar, no todas las redes soportan la multidifusión. Ello puede provocar que los datos multidifusión no alcancen a uno o más nodos de la isla multidifusión. Incluso si las redes soportan la multidifusión, otros factores tales como el tiempo de vida (TTL, *time-to-live*) pueden dar lugar a que los datos multidifusión no alcancen su destino.

Por estos y otros motivos, el MAP utiliza una estrategia autoajutable que le garantiza depender de la multidifusión sólo cuando ésta funciona. La estrategia consiste en enviar todos los datos tanto en unidifusión como en multidifusión, inhabilitando la unidifusión para aquellos nodos que reciben correctamente los datos en multidifusión.

Los datos unidifusión en una isla multidifusión se controlan de forma independiente para cada "flujo de datos". Un flujo de datos se identifica de forma inequívoca por los siguientes cinco campos: identificador del emisor, identificador del canal, nivel de fiabilidad, prioridad y tipo de datos. Por defecto, todos los datos se envían en unidifusión a todos los nodos de una isla multidifusión que deben recibirlos. Un único nodo de la isla multidifusión (el emisor multidifusión) envía asimismo los datos al grupo multidifusión. Cuando un nodo detecta que recibe un determinado flujo de datos en multidifusión, envía una **MAPDisableUnicastRequest** a su nodo vecino en la dirección del emisor multidifusión. El nodo vecino detiene entonces el envío del flujo de datos en unidifusión.

La identidad del emisor multidifusión para un determinado flujo de datos viene dado por el tipo de datos, tal como se indica a continuación:

- Para datos no uniformes, el emisor multidifusión es sencillamente el nodo punto de entrada original del flujo de datos.
- Para datos uniformes, todos los datos se envían hacia arriba al proveedor superior antes de ser candidatos para la multidifusión. El proveedor del grupo multidifusión envía los datos al grupo multidifusión de vuelta hacia abajo (el emisor multidifusión *siempre* es el proveedor de grupo multidifusión). Cuando se envía una **MAPDisableUnicastRequest** para datos

uniformes, siempre se hace en sentido ascendente. Además, el identificador del emisor debe ignorarse cuando se identifiquen inequívocamente los flujos de datos uniformes.

- Para datos en representación, todos los datos se envían hacia arriba al proveedor de grupo multidifusión. Éste envía los datos en sentido ascendente en unidifusión (si es necesario), al tiempo que los envía de vuelta hacia abajo (en unidifusión y en multidifusión). Tal como ocurre con los datos uniformes, el emisor multidifusión es *siempre* el proveedor de grupo multidifusión. Cuando se envía una **MAPDisableUnicastRequest** para datos en representación, siempre se hace en sentido ascendente. Además, el identificador del emisor debe ignorarse cuando se identifiquen inequívocamente los flujos de datos en representación.

Los tipos de datos en representación proporcionan una forma alternativa de enviar datos no uniformes. Los datos en representación **no pueden** utilizarse para transmitir datos uniformes. La utilización de datos en representación reduce la cantidad de información de estado del flujo de datos, ya que todos los emisores de datos en representación para un determinado identificador de canal/nivel de fiabilidad/prioridad, se fusionan en un único flujo de datos. Los datos en representación también son útiles en redes que no permiten que los nodos generen información multidifusión. El conocimiento de cuando se deben utilizar datos en representación es un asunto propio de la implementación local. No obstante, cuando un nodo comienza a enviar datos en representación para un determinado identificador de canal/nivel de fiabilidad/prioridad, no puede ulteriormente decidir utilizar el tipo de datos no uniforme (lo contrario es también cierto). La decisión de utilizar datos en representación es estrictamente local y cada nodo puede manejar los emisores locales de forma diferente.

Existe otra consideración importante a la hora de utilizar datos en representación. Cada nodo del trayecto entre el originador y el proveedor de grupo multidifusión verá los datos dos veces (una vez cuando van en sentido ascendente y otra en sentido descendente). Estos nodos sólo deben enviar los datos al MCS una sola vez. No importa si ello se realiza cuando el flujo de datos asciende o cuando vuelve descendiendo (pero debe de hacerse de forma consistente). Además, el nodo originador *nunca* debe devolver los datos al anexo del usuario que los ha enviado.

Antes de que un nodo informe a su vecino que debe inhabilitar la unidifusión para un flujo de datos, debe asegurarse que la transición a multidifusión puede realizarse sin pérdida de datos. Para garantizarlo, debe recibir al menos un paquete de datos en unidifusión, de forma que conozca cual debe ser el siguiente número de secuencia (nótese que los números de secuencia son exclusivos para cada flujo de datos y que los asigna el originador del flujo de datos). Antes o después de recibir el paquete unidifusión, el nodo debe recibir un paquete de datos en multidifusión con un número de secuencia que sea menor o igual que el siguiente número de secuencia esperado. Ello indica un solapamiento en los dos flujos de datos, permitiendo que se inhabilite la unidifusión.

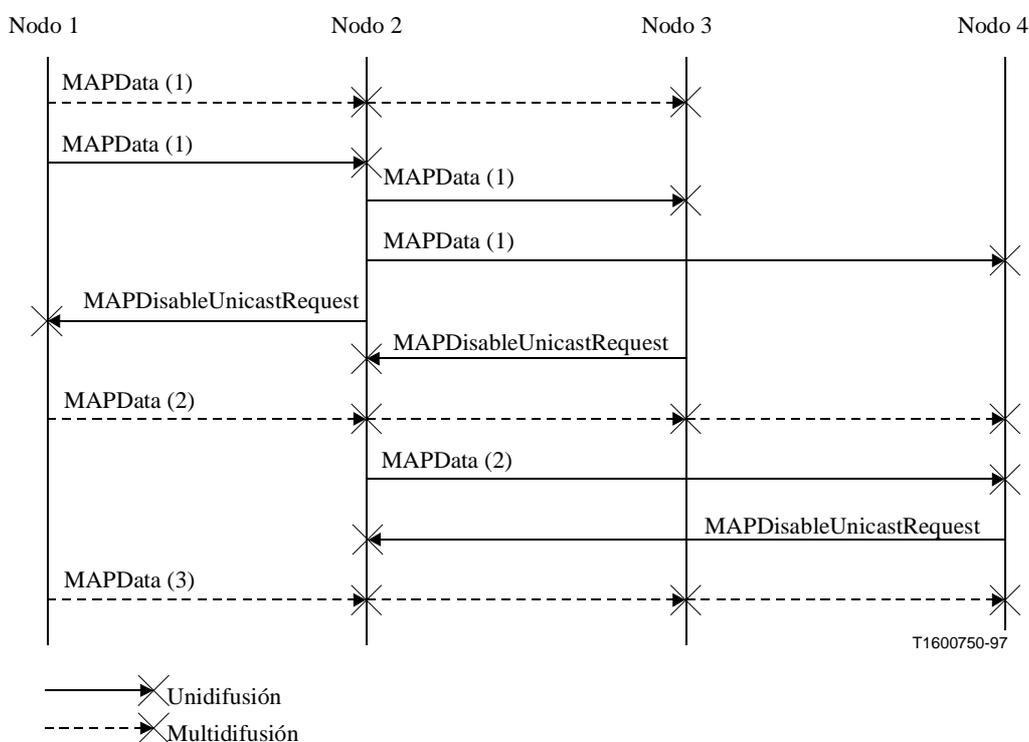
Cuando los datos se transmiten en un nuevo flujo de datos, se envían en unidifusión Y multidifusión a toda la isla. Progresivamente, y conforme va siendo posible, la unidifusión se va inhabilitando dando lugar a un tráfico de red óptimo.

La figura A.8-5 muestra un ejemplo de esta estrategia de autoajuste (véase la jerarquía del dominio en la figura A.8-1). En este ejemplo se hace un seguimiento de la transmisión de tres paquetes de datos de un flujo de datos anterior desconocido:

- 1) Una aplicación de usuario anexa al nodo 1 transmite un paquete de datos. El nodo 1 transmite los datos al grupo multidifusión. También transmite los datos al nodo 2 en unidifusión ya que el nodo 2 no ha inhabilitado aún la unidifusión para este flujo de datos. El nodo 4 no se ha incorporado aún al grupo, por lo que el paquete de datos multidifusión sólo llega a los nodos 2 y 3. Los nodos 2 y 3 retienen los datos ya que no conocen cual es el número de secuencia válido. Cuando el nodo 2 recibe los datos en unidifusión, redirecciona los mismo a los nodos 3 y 4 (en unidifusión). Los nodos 2 y 3 reconocen entonces que el paquete multidifusión es

idéntico al paquete unidifusión, por lo que emiten una **MAPDisableUnicastRequest** en la dirección del emisor.

- 2) La aplicación de usuario en el nodo 1 transmite otro paquete de datos en el mismo flujo de datos. Al igual que antes, el nodo 1 transmite los datos al grupo multidifusión. En esta ocasión el nodo 1 *no* transmite el paquete al nodo 2 mediante unidifusión, ya que el nodo 2 ha inhabilitado la unidifusión para este flujo de datos. Esta vez, la transmisión multidifusión llega a los tres nodos inferiores (incluyendo el nodo 4). El nodo 2 transmite los datos en unidifusión al nodo 4, pero no al nodo 3 (ya que éste ha inhabilitado la unidifusión para este flujo de datos). El nodo 4 conoce entonces que está sincronizado con el flujo de datos multidifusión y, por lo tanto, envía una **MAPDisableUnicastRequest** en sentido ascendente.
- 3) La aplicación de usuario en el nodo 1 transmite un tercer paquete de datos en el mismo flujo de datos. Al igual que antes, el nodo 1 transmite los datos al grupo multidifusión. Sin embargo, en esta ocasión, ninguno hará unidifusión de los datos ya que todos los nodos de la isla han inhabilitado la unidifusión para este flujo de datos.



**Figura A.8-5/T.125 – Ejemplo de temporización del procedimiento de inhabilitación de unidifusión para un flujo de datos**

### A.8.5 Abandono de la multidifusión

La subcláusula anterior se ha centrado en lo que ocurre cuando se trabaja en modo multidifusión. En esta subcláusula se describe lo que puede hacer un nodo para optimizar la transmisión cuando el modo multidifusión no funciona.

Cuando un nodo comienza la transmisión de datos de un flujo de datos no uniforme, lo hace tanto en modo unidifusión como multidifusión. Si los nodos adyacentes reciben los datos multidifusión, se inhabilita el modo unidifusión tal como se ha descrito en la subcláusula anterior. Si los nodos adyacentes no reciben los datos en multidifusión, no se producirá la inhabilitación del modo unidifusión. Ello da lugar a que el nodo emisor transmita cada PDU más veces de las necesarias (una

por cada nodo adyacente y una en multidifusión). Este problema se agudiza en un nodo hoja, que puede acabar utilizando un ancho de banda dos veces el necesario, lo cual puede ser inaceptable (especialmente si el nodo se anexa utilizando una conexión de baja velocidad). Desafortunadamente, el nodo no puede simplemente dejar de transmitir datos en multidifusión. Es posible que un nodo de la isla (no adyacente) reciba datos multidifusión, pudiendo haber inhabilitado ya el modo unidifusión.

Cada PDU **MAPData** contiene una bandera denominada bandera de envío unidifusión que puede utilizar un nodo para solventar este problema. Esta bandera fuerza a todos los nodos intermedios de la isla multidifusión a enviar de nuevo el flujo de datos en unidifusión a todos los receptores (incluso aunque previamente hayan inhabilitado la unidifusión). Cuando un nodo decide detener el envío de un determinado flujo de datos en multidifusión, sólo tiene que enviar todas las PDU **MAPData** posteriores (para dicho flujo de datos) exclusivamente en unidifusión, con la bandera de envío unidifusión inicializada. Ello permite al nodo detener con seguridad la transmisión multidifusión.

#### **A.8.6 Supresión de los grupos multidifusión**

Cuando un proveedor de grupo multidifusión detecta que ya no se necesita un grupo multidifusión, éste puede ser eliminado de la isla. En el caso de datos fiables, es muy importante que el grupo se suprima con la garantía de que no se van a perder datos.

Para que así sea, el proveedor de grupo multidifusión inhabilita en primer lugar la recepción multidifusión para el grupo en cuestión. Ello se consigue enviando una **MAPDisableMulticastRequest** en sentido descendente a todos los nodos de la isla (cada nodo envía automáticamente esta petición a todos los nodos que se encuentran por debajo de él). Esta MAPPDU hace que cada nodo comience el proceso de habilitar de nuevo la unidifusión para todos los flujos de datos del grupo para los que anteriormente se suprimió la unidifusión. Esto se realiza enviando una **MAPEnableUnicastRequest** en la dirección del emisor para cada flujo de datos afectado. Esta MAPPDU también hace que el grupo se marque como inhabilitado, con lo cual se prohíbe cualquier ulterior inhabilitación de la unidifusión para el grupo. Nótese que esto NO causa que los nodos detengan la transmisión de datos al grupo multidifusión. Además, los nodos deben permanecer incorporados al grupo multidifusión como receptores, al menos hasta que la unidifusión haya sido habilitada con éxito para todos los flujos de datos afectados.

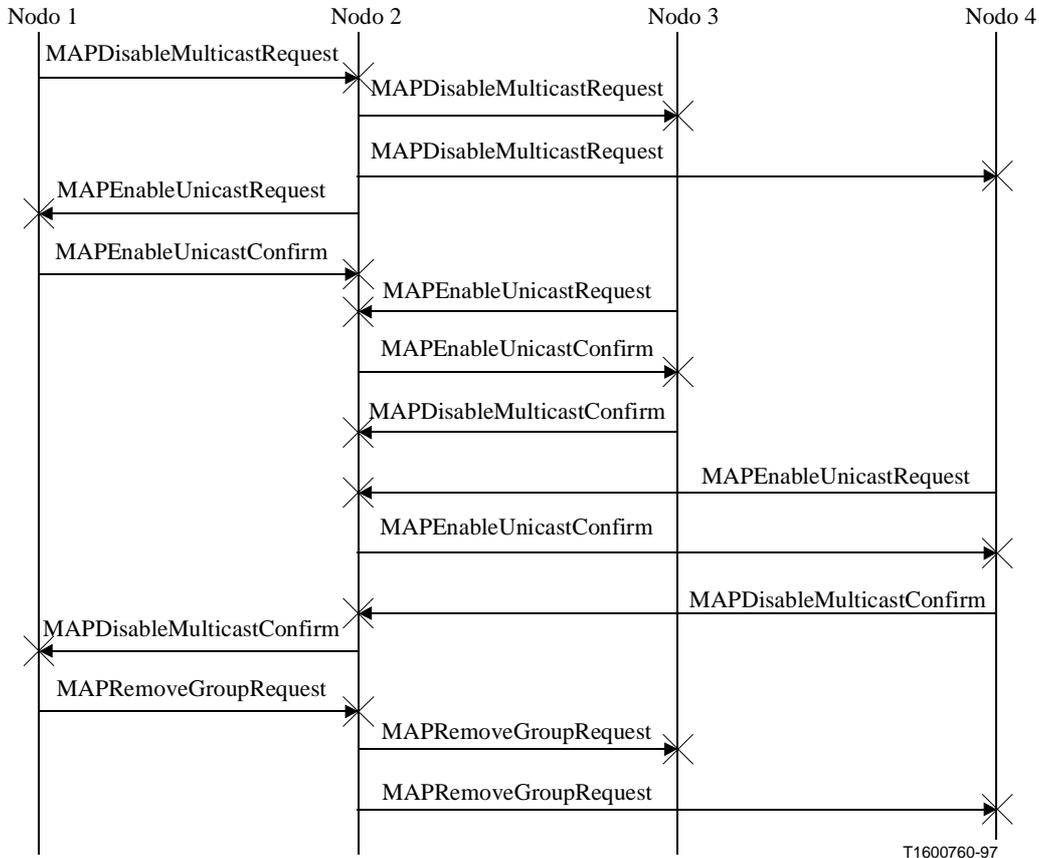
Cuando la **MAPDisableMulticastRequest** alcanza los nodos inferiores de la isla multidifusión, éstos responden con **MAPDisableMulticastConfirm**. Antes de hacerlo, deben garantizar que todos los flujos de datos del grupo para el que la unidifusión fue inhabilitada anteriormente son habilitados de nuevo. Para que ello sea así, cada nodo debe haber recibido un **MAPEnableUnicastConfirm** para cada petición pendiente. Además, cada nodo debe examinar el número de secuencia en la confirmación para asegurarse que se han recibido todos los datos multidifusión en circulación (si no, debe esperar a que éstos lleguen). Una vez que esto ocurre, se envía **MAPDisableMulticastConfirm** al nodo inmediatamente superior.

Los nodos intermedios de la isla deben esperar hasta que han recibido las PDU **MAPDisableMulticastConfirm** de todos los nodos inferiores a ellos de la isla. También deben esperar hasta que todos los flujos de datos unidifusión previamente inhabilitados son habilitados de nuevo, tal como se ha descrito anteriormente. Pueden entonces enviar un **MAPDisableMulticastConfirm** al nodo superior.

Una vez que el proveedor del grupo multidifusión ha recibido las PDU **MAPDisableMulticastConfirm** procedentes de todos los nodos inferiores y se ha asegurado que sus flujos de datos unidifusión se han habilitado de nuevo, se puede afirmar que el grupo ha dejado de ser utilizado. En este momento el proveedor de grupo multidifusión puede enviar una **MAPRemoveGroupRequest** hacia abajo. Esta MAPPDU se disemina a todos los nodos de la isla,

haciendo que cada nodo abandone el grupo (si es que está incorporado al mismo) y los elimina de sus listas de grupos activos.

La figura A.8-6 ilustra el proceso de supresión de un grupo del dominio de la figura A.8-1. Se trata de una continuación del ejemplo de la figura A.8-5, en el que la unidifusión se ha inhabilitado para un único emisor anexo al nodo 1.



**Figura A.8-6/T.125 – Temporización del procedimiento de supresión de grupo**

## A.9 Descripción de las MAPPDU

En esta subcláusula se describen las PDU que utiliza el MAP. Cada MAPPDU comienza con una breve descripción sobre el objeto de su utilización, sigue un cuadro que muestra los campos que se codifican en la MAPPDU y, finalmente, se presenta una descripción de cada campo de la MAPPDU.

La subcláusula siguiente contiene la definición ASN.1 de cada MAPPDU.

### A.9.1 MAPConnectRequest

Cuando se crea una nueva conexión entre dos nodos para un dominio dado, el primer paso consiste en abrir una conexión unidifusión fiable. **MAPConnectRequest** es la primera MAPPDU enviada a través de dicha conexión. El nodo llamante envía esta MAPPDU para indicar al nodo llamado la versión del protocolo que desea utilizar, sólo es un valor recomendado. Es responsabilidad del nodo llamado tomar la decisión final y comunicársela al nodo llamante mediante una **MAPConnectConfirm**.

Esta MAPPDU incluye un campo de punto de acceso al servicio MAP (MAPSAP) que permite al nodo llamante especificar quién debe recibir la llamada en el nodo llamado.

Esta MAPPDU contiene también un identificador de referencia que permite al nodo llamante asociar esta conexión con un dominio en particular. Se utiliza cuando el arbitraje del protocolo de transporte da lugar a la creación de conexiones adicionales o a la transición a un conjunto distinto de conexiones.

El cuadro siguiente resumen el contenido de **MAPConnectRequest**.

Número de versión
MAPSAP de la conexión
Identificador de referencia del dominio
Rango de prioridad

*Número de versión:* Este campo indica la versión del protocolo que el nodo llamante desea utilizar. Consta de un número de versión mayor y menor. El valor de este campo para el protocolo definido en este anexo es 1.0 (versión mayor 1, versión menor 0). Nótese que se trata de un valor recomendado. El nodo llamado determinará la versión del protocolo que se debe utilizar.

*MAPSAP de conexión:* Este campo informa al receptor de cual es el punto de acceso al servicio del MAP que maneja la conexión entrante. El valor cero (0) indica que el llamante desea acceder al MAPSAP por defecto. Todos los demás valores se asignan y obtienen fuera de banda respecto a este protocolo.

*Identificador de referencia del dominio:* Este campo permite al nodo llamante asociar esta conexión con un dominio en particular del nodo llamado. Cuando se crea la conexión inicial con un nodo remoto, este campo debe tomar el valor cero (0). Cuando se crean conexiones adicionales, o transiciones a un nuevo conjunto de conexiones, este campo debe tomar el valor que se envió en la **MAPArbitrateProtocolsRequest** de la conexión inicial. Este valor, junto con la dirección llamante, permite conocer al nodo llamado cual es el dominio al que la conexión entrante debe vincularse.

*Gama de prioridad:* Es la gama de prioridad de los datos que se transportan en esta conexión. La conexión inicial con un nodo remoto siempre transporta *exclusivamente* prioridades cero (recuérdese que cero es la prioridad más alta).

### A.9.2 MAPConnectConfirm

Esta MAPPDU se utiliza para aceptar una nueva conexión con otro nodo. Informa al nodo llamante de la versión de protocolo que se debe utilizar en la conexión. Asimismo informa al nodo llamante de cual es el MAPSAP remoto que aceptó la llamada.

El cuadro siguiente resume el contenido de **MAPConnectConfirm**.

Número de versión
MAPSAP de la conexión

*Número de versión:* Este campo indica la versión del protocolo que será utilizado en esta conexión. Consta de un número de versión mayor y menor. El valor de este campo para el protocolo definido en este anexo es 1.0 (versión mayor 1, versión menor 0). Si el nodo llamante no puede manejar (o no desea manejar) la versión especificada por el nodo llamado en esta MAPPDU, debe desconectar.

*MAPSAP de conexión:* Es el MAPSAP en el nodo llamado que ha aceptado la llamada. Si el MAPSAP especificado en la **MAPConnectRequest** es distinto de cero, este campo debería tomar el mismo valor. Si el MAPSAP especificado en la **MAPConnectRequest** es cero, este campo toma el valor válido del MAPSAP que ha aceptado la llamada (que también puede ser cero).

### A.9.3 MAPDisconnectRequest

Esta MAPPDU se utiliza para deshacer una conexión de transporte existente entre dos nodos o para rechazar una conexión entrante si el nodo llamado no desea participar en la conferencia. Una vez que se manda esta MAPPDU, la conexión no puede utilizarse para la transmisión de ningún otro dato.

Si esta MAPPDU se envía como parte de la transición a un nuevo conjunto de conexiones unidifusión fiables, el emisor debe inicializar la bandera de confirmación requerida. Ello informa al receptor de que debe responder enviando una **MAPDisconnectConfirm** en la misma conexión.

Si la bandera de confirmación requerida se inicializa, el nodo emisor debe esperar la **MAPDisconnectConfirm** antes de deshacer la conexión. Esto se hace para garantizar que se eliminan todos los datos la conexión antes de deshacerla.

El cuadro siguiente resume el contenido de **MAPDisconnectRequest**.

Motivo
Confirmación requerida

*Motivo:* Este campo indica al receptor porqué se ha deshecho o se ha rechazado la conexión.

*Confirmación requerida:* Es una bandera booleana que indica si el receptor debe responder con una **MAPDisconnectConfirm**. La bandera se inicializa si la desconexión ocurre como parte de una transición a un conjunto diferente de conexiones unidifusión fiables. La espera de la confirmación permite a ambas partes conocer que todas las MAPPDU transmitidas en esta conexión han sido procesadas.

### A.9.4 MAPDisconnectConfirm

Esta MAPPDU se envía en respuesta a una **MAPDisconnectRequest** en la que se ha inicializado la bandera de confirmación requerida. Enviando esta confirmación, ambas partes se aseguran que todos los datos han desaparecido de la conexión antes de que ésta se haya deshecho y comienzan a utilizar una nueva.

Al recibir esta MAPPDU, el nodo puede cerrar con seguridad la conexión sobre la cual ésta ha llegado. Dependiendo del estado de las conexiones que la han sustituido, el nodo puede comenzar a procesar los datos en una o varias de dichas conexiones.

Esta MAPPDU no tiene campos asociados a la misma.

### A.9.5 MAPArbitrateProtocolsRequest

Una vez que se ha establecido una conexión para un determinado dominio entre dos nodos, puede utilizarse esta MAPPDU para arbitrar cuales deben ser los protocolos de transporte que han de utilizarse para la distribución de los datos. Esta MAPPDU *debe* ser enviada por el nodo llamante cuando se realiza la conexión inicial. También puede ser enviada por cualquier otro nodo si se desea redefinir los protocolos de transporte a utilizar.

Esta MAPPDU informa al receptor sobre cuales son los protocolos de transporte soportados por el emisor. Se incluyen los protocolos unidifusión y multidifusión, para los dos niveles de fiabilidad. Nótese que es *obligatorio* que el protocolo unidifusión fiable empleado en la conexión actual se represente en esta MAPPDU (ello permite al receptor rechazar una transición unidifusión fiable simplemente seleccionando el mismo protocolo que está siendo utilizado).

El emisor puede asignar un indicador de preferencia a cada uno de los protocolos de transporte que se listan en esta MAPPDU. Para los protocolos multidifusión, existe un campo que puede ser utilizado por un nodo inferior para indicar cuantos nodos de su subárbol soportan el protocolo. El

nodo superior puede utilizar esta información para seleccionar con mayor precisión protocolos multidifusión para un metacanal dado.

Esta MAPPDU también se utiliza para comunicar un identificador de referencia del dominio al receptor. Este valor, conjuntamente con la dirección de red del emisor, puede ser ulteriormente utilizado para dar a conocer al receptor de forma inequívoca un dominio en particular. Ello es necesario en aquellos casos en los que el dominio no esté implícito. Este valor se utiliza específicamente cuando se crean conexiones unidifusión fiables adicionales, o cuando se envían PDU de datos no fiables.

Si los dos nodos adyacentes intentan iniciar un nuevo proceso de arbitraje simultáneamente, se ignora la petición del nodo inferior. Esta situación se detecta en ambos nodos porque ambos reciben una petición de arbitraje cuando esperan una confirmación. El nodo superior responde rechazando la petición (y quedando a la espera de confirmación) y el nodo inferior responde procesando la petición (y abandonando la espera de confirmación). Si el nodo inferior aún desea realizar un nuevo arbitraje, puede emitir otra petición (después de enviar la confirmación).

El cuadro siguiente resume el contenido de **MAPArbitrateProtocolsRequest**.

Identificador de referencia del dominio
Bandera indicadora de protocolos posteriores
Menú del protocolo de transporte

*Identificador de referencia del dominio:* Es un valor que el emisor asigna localmente para hacer referencia a un dominio en particular. Una vez que el proceso de arbitraje se ha completado, el emisor puede utilizar este valor para indicar el dominio al que está asociada una PDU **MAPConnectRequest**. También se utiliza para indicar el dominio al que está asociada una PDU **MAPData** no fiable.

*Bandera indicadora de protocolos posteriores:* Cuando se inicializa, esta bandera indica que el número de protocolos de transporte es demasiado alto como para caber en una sola MAPPDU, por lo que el receptor debe esperar ulteriores PDU **MAPArbitrateProtocolsRequest** antes de procesarlos. La bandera de la última MAPPDU de la secuencia debe de estar reinicializada.

*Menú del protocolo de transporte:* Este campo contiene un conjunto de entradas de peticiones de protocolo de transporte. Es la lista de los protocolos de transporte que el emisor puede utilizar en esta conferencia. Debe ser la lista completa de los protocolos soportados por el emisor (los protocolos no fiables son opcionales). Es obligatorio incluir en esta lista los protocolos unidifusión fiables utilizados para transportar esta MAPPDU. Existe un campo en cada entrada que permite al llamante especificar los protocolos que prefiere utilizar. Para un tipo de protocolo dado, el emisor debe también listar las entradas en orden de preferencia. El receptor selecciona los protocolos que deben utilizarse en la conexión. En el caso de protocolos unidifusión, el receptor intenta seleccionar los protocolos preferidos si ello es posible. Para protocolos multidifusión, el receptor debe seleccionar todos los protocolos comunes al emisor y al receptor. Puede ocurrir que no existan protocolos multidifusión comunes para un nivel de fiabilidad dado (todos los datos son unidifusión para dicho nivel de fiabilidad). También puede ocurrir que no hayan protocolos unidifusión no fiables comunes (los datos no fiables se envían a través de conexiones fiables).

### A.9.5.1 Entrada de petición de protocolo de transporte

El menú sólo es un conjunto de entradas de petición de protocolo de transporte, cuyo formato se define en el cuadro siguiente.

Identificador del protocolo de transporte
Tipo de protocolo de transporte
Dirección de red
Bandera de carga útil máxima fijada
Tamaño máximo de carga útil
Peso de preferencia (opcional)
Cuenta de nodo (opcional)
Número de conexiones (opcional)
Datos de configuración (opcional)

*Identificador del protocolo de transporte:* Este campo identifica inequívocamente el protocolo de transporte. Los valores de este campo deben estar normalizados aunque también es posible especificar valores propietarios.

*Tipo de protocolo de transporte:* Este campo toma uno de los cuatro valores siguientes: unidifusión fiable, unidifusión no fiable, multidifusión fiable o multidifusión no fiable.

*Dirección de red:* Es la dirección de acceso al originador utilizando este protocolo.

*Bandera de carga útil máxima fijada:* Es una bandera booleana que indica si el tamaño máximo de la carga útil es negociable para esta entrada de protocolo de transporte. Para más información véase la descripción del tamaño máximo de la carga útil.

*Tamaño máximo de la carga útil:* Este campo especifica el tamaño máximo de la carga útil que soporta este protocolo. Si la bandera de carga útil máxima fijada está reinicializada, este es el valor máximo que soporta la implementación local (que puede ser negociado a la baja). Si la bandera de carga útil máxima fijada está inicializada, este es el valor exacto que debe utilizarse para aceptar esta entrada de protocolo de transporte. MAP utiliza los valores negociados para determinar el tamaño máximo de una trama de datos (las tramas de datos se describen con más detalle en la sección relativa a **MAPData**). El tamaño máximo de la carga útil más pequeño que soporta el MAP es 128.

*Peso de referencia:* Este campo opcional se utiliza para indicar al receptor cuales son los protocolos preferidos. Un valor alto indica un protocolo preferido con respecto a otro con un valor más bajo. Si este campo no está incluido, se asume un peso de preferencia cero. Este campo no se utiliza en protocolos multidifusión cuando la MAPPDU se envía en sentido descendente.

*Cuenta de nodo:* Este campo opcional sólo está presente en protocolos multidifusión (fiables y no fiables) y sólo cuando la MAPPDU se envía en sentido ascendente. Informa sobre cuantos nodos al mismo o inferior nivel que el emisor soportan esta entrada de protocolo multidifusión. Este número puede ser utilizado por el nodo superior para tomar las decisiones más apropiadas sobre el protocolo multidifusión a utilizar cuando se establece la correspondencia entre una nuevo metacanal y la multidifusión. Para mantener la precisión de este número, debe tener lugar un rearbitraje siempre que un nuevo nodo con MAP se incorpore o abandone la conferencia. Este arbitraje solo tiene lugar en la rama del árbol del dominio que conduce a donde el nodo se ha incorporado o ha abandonado. A fin de reducir el tráfico de red y una vez que la conferencia ha alcanzado un determinado tamaño, el rearbitraje no debe tener lugar *cada vez* que un nodo se incorpore o abandone la conferencia. Este campo puede también utilizarse para evitar la utilización de multidifusión cuando la vayan a utilizar menos de  $N$  nodos ( $N=3?$ ).

*Número de conexiones:* Este campo opcional sólo está presente en el caso de protocolo unidifusión fiable. Indica el número de conexiones de transporte que el emisor desea crear para esta conexión MAP (si se selecciona este protocolo). Si el receptor no selecciona este protocolo, debe elegir un número menor o igual al número especificado por el emisor.

*Datos de configuración:* Este campo opcional es una cadena de octetos de longitud variable que es utilizada por las pilas de protocolo para pasar información sobre la configuración. Este campo no está presente si las pilas del protocolo no desean intercambiar datos de configuración o si se trata de un rearbitraje y las pilas de transporte quieren mantener el valor previamente acordado. Nótese que si este campo *está* presente durante un rearbitraje y el protocolo se selecciona como protocolo unidifusión, es necesaria una transición a un nuevo conjunto de conexiones unidifusión (ya que los datos de configuración pueden haber cambiado). El formato de este campo es propio de cada protocolo de transporte. Para protocolos de transporte normalizados, este campo debe formar parte de la norma. Para protocolos de transporte propietarios, puede ser cualquiera que el fabricante considere adecuado. Este campo puede utilizarse para comunicar capacidades y puede ser modificado en la **MAPArbitrateProtocolsConfirm** para arbitrar dichas capacidades. Este campo no debe ser de un tamaño tal que haga que la MAPPDU sea de un tamaño superior a los 16 384 octetos.

#### **A.9.6 MAPArbitrateProtocolsConfirm**

Esta MAPPDU se utiliza para dar por terminado el arbitraje de los protocolos de transporte para una conexión dada. Después de recibir una **MAPArbitrateProtocolsRequest** el receptor debe responder emitiendo esta MAPDU.

La MAPPDU de petición contiene una lista de todos los protocolos de transporte soportados por el emisor (incluyendo las preferencias). El receptor debe seleccionar de dicha lista los protocolos que puede utilizar en esta conexión. Ello incluye *exactamente* un protocolo unidifusión fiable (y, opcionalmente, un protocolo unidifusión no fiable). Debe de tenerse en cuenta si el emisor especificó preferencias para los protocolos unidifusión. El receptor debe y también determinar todo el conjunto de protocolos multidifusión comunes a los dos nodos. El envío de dicho conjunto común en la confirmación contribuye a garantizar que durante la asignación de grupo se seleccionan los mejores protocolos multidifusión.

Esta MAPPDU también se utiliza para comunicar un identificador de referencia del dominio al receptor. Este valor, junto con la dirección de red del emisor, puede ser utilizado ulteriormente para identificar inequívocamente un dominio particular al receptor. Ello es necesario en aquellos casos en los que la referencia al dominio no está implícita. Específicamente, este valor se utiliza cuando se envían PDU de datos no fiables.

El cuadro siguiente resume el contenido de **MAPArbitrateProtocolsConfirm**.

Identificador de referencia del dominio
Bandera indicadora de protocolos posteriores
Menú del protocolo de transporte

*Identificador de referencia del dominio:* Es un valor que el emisor asigna localmente para hacer referencia a un dominio en particular. Una vez que el proceso de arbitraje se ha completado, el emisor puede utilizar este valor para indicar el dominio al que está asociada una PDU **MAPData** no fiable.

*Bandera indicadora de protocolos posteriores:* Cuando esta bandera se inicializa indica que el número de protocolos de transporte es demasiado alto como para caber en una sola MAPPDU, por lo que el receptor debe esperar ulteriores PDU **MAPArbitrateProtocolsConfirm** antes de procesarlos. La bandera de la última MAPPDU de la secuencia debe de estar reinicializada.

*Menú del protocolo de transporte:* Este campo contiene un conjunto de entradas de confirmación de protocolo de transporte. Es una lista de los protocolos de transporte que el nodo llamado ha decidido utilizar en esta conexión. Como mínimo el nodo llamado debe indicar un protocolo unidifusión fiable. Opcionalmente, puede indicar un protocolo unidifusión no fiable y varios protocolos para los niveles de fiabilidad multidifusión. Si no se incluye un protocolo unidifusión no fiable, se hará unidifusión de los datos no fiables sobre conexiones fiables. Si no se incluye un determinado nivel de fiabilidad multidifusión, se hará unidifusión entre dichos nodos de todos los datos con dicho nivel de fiabilidad (las islas multidifusión se dividen para dicho nivel de fiabilidad). Si los protocolos unidifusión arbitrados son distintos de los que ya están siendo utilizados, debe realizarse una transición de protocolo unidifusión.

#### **A.9.6.1 Entrada de confirmación del protocolo de transporte**

El menú consta de un conjunto de entradas de confirmación del protocolo de transporte, cuyo formato se define en el cuadro siguiente.

Identificador del protocolo de transporte
Tipo de protocolo de transporte
Dirección de red
Tamaño máximo de carga útil
Peso de preferencia (opcional)
Cuenta de nodo (opcional)
Número de conexiones (opcional)
Datos de configuración (opcional)

*Identificador del protocolo de transporte:* Este campo identifica inequívocamente el protocolo de transporte. Los valores de este campo deben estar normalizados aunque también es posible especificar valores propietarios.

*Tipo de protocolo de transporte:* Este campo toma uno de los cuatro valores siguientes: unidifusión fiable, unidifusión no fiable, multidifusión fiable o multidifusión no fiable.

*Dirección de red:* Es la dirección de acceso al originador utilizando este protocolo.

*Tamaño máximo de la carga útil:* Este campo especifica el tamaño máximo de la carga útil que utiliza este protocolo de transporte. Nótese que el tamaño máximo de la carga útil para un determinado protocolo multidifusión *debe* ser igual al tamaño máximo de carga útil para el protocolo unidifusión con el mismo nivel de fiabilidad. Ello se debe a que las tramas de referencia y el nivel de fiabilidad deben enviarse en unidifusión desde algunos puntos de la isla multidifusión. El tamaño máximo de la carga útil más pequeño que soporta MAP es 128.

*Peso de referencia:* Este campo opcional se utiliza para indicar al receptor cuales son los protocolos preferidos. Un valor alto indica un protocolo preferido con respecto a otro con un valor más bajo. Si este campo no está incluido, se asume un peso de preferencia cero. Este campo no se utiliza en protocolos multidifusión cuando la MAPPDU se envía en sentido ascendente.

*Cuenta de nodo:* Este campo opcional sólo está presente en protocolos multidifusión (fiables y no fiables) y sólo cuando la MAPPDU se envía en sentido ascendente. Informa sobre cuantos nodos al mismo nivel o nivel inferior al emisor soportan esta entrada de protocolo multidifusión. El nodo superior puede utilizar este número para tomar decisiones más apropiadas sobre el protocolo multidifusión a usar cuando se establece la correspondencia entre una nuevo metacanal y la

multidifusión. Para mantener la precisión de este número, debe de tener lugar un rearbitraje siempre que un nuevo nodo con MAP se incorpore o abandone la conferencia. Este rearbitraje sólo tendrá lugar en la rama del árbol del dominio que conduce al punto donde el nodo se unió o abandonó. A fin de reducir el tráfico de red y una vez que la conferencia ha alcanzado un determinado tamaño, no se debe producir un rearbitraje *cada vez* que un nodo se incorpore o abandone la conferencia. Este campo puede también utilizarse para evitar la utilización de multidifusión cuando la vayan a utilizar menos de N nodos ( $N=3?$ ).

*Número de conexiones:* Este campo opcional sólo está presente para protocolos unidifusión. Informa sobre el número de conexiones de transporte que se van a crear para esta conexión MAP (a este nivel de fiabilidad). Este número debe ser el menor de entre el valor solicitado y el valor soportado localmente.

*Datos de configuración:* Este campo opcional es una cadena de octetos de longitud variable que utilizado por las pilas de protocolo para pasar información sobre la configuración. Este campo no está presente si las pilas del protocolo no desean intercambiar datos de configuración o si se trata de un rearbitraje y las pilas de transporte quieren mantener el valor previamente acordado. Nótese que si este campo *está* presente durante un rearbitraje, y el protocolo se selecciona como protocolo unidifusión, es necesaria una transición a un nuevo conjunto de conexiones unidifusión (ya que los datos de configuración pueden haber cambiado). El formato de este campo es propio de cada protocolo de transporte. Para protocolos de transporte normalizados, este campo debe formar parte de la norma. Para pilas propietarias, puede ser cualquiera que el fabricante considere adecuado. Este campo puede utilizarse para comunicar capacidades y puede ser modificado para arbitrar dichas capacidades. Este campo no debe ser tan grande que haga que la MAPPDU sea de un tamaño superior a los 16 384 octetos.

### A.9.7 MAPData

Esta MAPPDU es pieza clave del protocolo. Se utiliza para transportar todos los datos de usuario a su eventual destino. A diferencia de la pila TCP/IP definida en la Recomendación T.123, el MAP permite la concatenación de múltiples MCSPDU en una sola PDU **MAPData** (para promover la utilización eficiente de la red). MAP también permite la segmentación de una sola MCSPDU en múltiples PDU **MAPData** (cuando la MCSPDU excede el tamaño máximo de la carga útil).

Cada componente de datos individual de una PDU **MAPData** se denomina *trama de datos*. En una PDU **MAPData** puede haber una o varias tramas de datos. El tamaño máximo de una trama de datos se determina durante el arbitraje del protocolo de transporte y vendrá dado por el tamaño máximo de la trama de datos que se ajusta a la carga útil máxima para un nivel de fiabilidad dado (teniendo en cuenta la cabecera). El tamaño máximo de la trama de datos para un nivel de fiabilidad puede ser distinto del que corresponda a otro nivel de fiabilidad.

Las MCSPDU se dividen en tramas de datos en el nodo que actúa como el punto de entrada original del flujo de datos. Si es necesario son reensambladas en todos los nodos receptores.

**MAPData** soporta la noción de compresión de la cabecera. Existen varios campos en la cabecera de **MAPData** que también se incluyen en la cabecera de la PDU de datos MCS. Si dichos campos pueden extraerse de la cabecera MCS de la primera trama de datos de la MAPPDU, pueden no estar incluidos en la cabecera de **MAPData** (se trata de campos opcionales). Si la primera trama de datos de la MAPPDU es una PDU de datos de MCS segmentada, el emisor decide si la cabecera MCS que está presente es suficiente como para extraer dichos datos. El receptor debe estar preparado para extraer dichos campos en caso de que éstos no estén explícitamente incluidos en la cabecera de **MAPData**.

El cuadro siguiente resume el contenido de **MAPData**.

Descriptor de datos (opcional)
Matriz de la trama de datos

*Descriptor de datos:* Este campo opcional sólo está presente cuando se envían PDU de datos de MCS. Cuando está presente, los campos de este descriptor describen los datos contenidos en esta MAPPDU. Si este campo no está presente, esta MAPPDU contiene las PDU de control de MCS, las cuales deben enviarse directamente al MCS en el nodo receptor.

*Matriz de la trama de datos:* Es una secuencia de entradas de tramas de datos (tal como se define en los apartados siguientes).

#### **A.9.7.1 Descriptor de datos**

Un descriptor de datos sólo figura en una PDU **MAPData** si ésta transporta PDU de datos de MCS. El cuadro siguiente resume el contenido del descriptor de datos.

Bandera de envío unidifusión
Número de secuencia de inicio
Identificador del flujo de datos (opcional)
Identificador de dominio de referencia (opcional)

*Bandera de envío unidifusión:* Esta bandera se utiliza para forzar al receptor a que envíe esta MAPPDU en unidifusión. Cuando se inicializa esta bandera, el receptor debe iniciar la transmisión de todos los datos de este flujo de datos en unidifusión (incluso a nodos vecinos que habían inhibido la unidifusión). La transmisión unidifusión obligatoria comienza con esta MAPPDU. Una vez que un emisor decide inicializar esta bandera para un flujo de datos en concreto, la bandera debe inicializarse para todas las PDU **MAPData** subsiguientes del mismo flujo de datos (para evitar situaciones de aceleración). Si esta bandera está reiniciada, la MAPPDU debe ser tratada normalmente.

*Número de secuencia de inicio:* Este campo contiene el número de secuencia único asignado a la primera trama de datos de esta MAPPDU. Las tramas de datos subsiguientes de esta MAPPDU incrementan este número de secuencia de inicio. El originador del flujo de datos asigna de forma independiente los números de secuencia de cada trama de datos.

*Identificador del flujo de datos:* Este campo opcional debe estar presente si la información no puede ser extraída de la primera trama de datos. Los campos que constituyen el identificador del flujo de datos son los que se requieren para encaminar correctamente los datos y para determinar a qué flujo de datos único están asociadas las tramas de datos (lo cual a su vez, identifica el contador de la secuencia).

*Identificador de referencia del dominio:* Este campo opcional sólo está presente en el caso de datos no fiables. Es el identificador de referencia del dominio que ha sido transmitido por el emisor durante el arbitraje del protocolo. Este campo puede estar presente en el caso de transmisiones multidifusión (motivado por las características de funcionamiento), pero en ese caso debe ignorarse. Este valor, combinado con la dirección de red del originador es suficiente para identificar inequívocamente cual es el dominio al que se asocian los datos.

#### **A.9.7.2 Identificador del flujo de datos**

El identificador del flujo de datos se utiliza para identificar el flujo de datos al que está asociado un conjunto de tramas de datos. Esta información debe de estar explícitamente incluida en caso de que

no pueda extraerse de la primera trama de datos. El cuadro siguiente resume el contenido del identificador del flujo de datos.

Identificador del emisor (opcional)
Identificador del metacanal (opcional)
Identificador del canal
Nivel de fiabilidad
Prioridad
Tipo de datos (opcional)

*Identificador del emisor:* Este campo opcional solo está presente en el caso de datos no uniformes, de datos uniformes o de datos en representación que circulan en sentido ascendente en unidifusión. Este campo puede también excluirse si la información se puede extraer de la cabecera MCS de la primera trama de datos. Informa sobre el identificador del usuario MCS de la aplicación de usuario que ha originado los datos asociados con este flujo de datos.

*Identificador del metacanal:* Este campo opcional puede ser excluido si la información puede extraerse de la cabecera MCS de la primera trama de datos. Identifica el metacanal al que se hace referencia y contiene tres subcampos: identificador de canal, nivel de fiabilidad y prioridad.

*Tipo de datos:* Este campo opcional puede ser excluido si la información puede extraerse de la cabecera MCS de la primera trama de datos. Informa del tipo de datos de que se trata, que puede ser: no uniforme, uniforme o en representación.

### A.9.7.3 Entrada de la trama de datos

Las PDU **MAPData** contienen una matriz de tramas de datos. El cuadro siguiente resume el contenido de la entrada de trama de datos.

Bandera del primer segmento
Bandera del último segmento
Datos de usuario

*Bandera del primer segmento:* Esta bandera indica si la trama de datos es el primer segmento de una MCSPDU. Si la bandera está inicializada, se trata de la primera trama de datos de una MCSPDU. Si esta bandera está reinicializada, no se trata de la primera trama de datos de una MCSPDU.

*Bandera del último segmento:* Esta bandera indica si la trama de datos es el último segmento de una MCSPDU. Si la bandera está inicializada, se trata de la última trama de datos de una MCSPDU. Si esta bandera está reinicializada, no se trata de la última trama de datos de una MCSPDU.

*Datos de usuario:* Datos de usuario de la MCSPDU.

### A.9.8 MAPAddGroupRequest

Esta MAPPDU se utiliza para diseminar asignaciones de grupo multidifusión en el sentido descendente de un dominio. Tiene lugar siempre que MAP detecta un flujo de datos para el que no hay asignación de grupo. Este proceso es originado por cada nodo de un dominio que no tiene ninguna conexión MAP ascendente o bien sí la tiene pero que carece de protocolos multidifusión comunes con el nodo superior. Cada nodo que recibe un **MAPAddGroupRequest** del nodo superior al mismo repite este proceso.

Cada nodo examina la lista de protocolos multidifusión que soportan los nodos situados por debajo del mismo (dichas lista se construyen durante el arbitraje del protocolo de transporte). En base a esta

información, cada nodo debe informar a sus nodos inferiores sobre cuales son los protocolos multidifusión y la dirección de grupo que hay que utilizar. Debe seleccionar el protocolo o los protocolos que den lugar al menor número de islas del mayor tamaño posible.

Cuando se selecciona un protocolo para cada uno de los nodos inferiores, deben tenerse en cuenta varios factores:

- 1) *¿Cuáles son las versiones de protocolos que se utilizan en los nodos inferiores?.* Durante el establecimiento de la conexión, el MCS y el MAP realizan un arbitraje sobre las versiones de protocolo con cada nodo adyacente. Cuando se seleccionan protocolos multidifusión para los nodos inferiores, debe recordarse que en la misma isla multidifusión *no puede* haber nodos que utilicen versiones distintas de protocolo (aunque soporten el mismo protocolo multidifusión ). Ello se debe a que los formatos de las PDU son distintos.
- 2) *¿Cuáles son los protocolos que soportan los nodos inferiores?.* Obviamente un nodo sólo puede seleccionar uno de los protocolos que los nodos inferiores han acordado utilizar. Debe intentar seleccionar los protocolos más comúnmente utilizados.
- 3) *¿Cuáles han sido los tamaños máximos de la carga útil que han sido arbitrados con los nodos inferiores?.* Durante el arbitraje de protocolo, cada nodo debe intentar seleccionar el mismo tamaño máximo de la carga útil que ya ha sido arbitrado con otras conexiones del mismo dominio. Si diversos nodos insisten en disponer de tamaños máximos de carga útil diferentes, no pueden estar en la misma isla multidifusión (todos los nodos de una isla deben acordar el mismo tamaño máximo de carga útil, para unidifusión y para multidifusión).
- 4) *¿Cuál es el protocolo que se ha asignado desde un nivel superior?.* Si un nodo está realizando esta selección en respuesta a una **MAPAddGroupRequest** procedente de arriba, debe intentar utilizar el mismo protocolo que le fue asignado (cuando ello es posible). Ello se debe a que el nodo superior tiene una mayor visibilidad y, por lo tanto, puede tomar decisiones más acertadas.
- 5) *¿Cuántos nodos de cada subárbol soportan cada uno de los protocolos?.* Durante el arbitraje del protocolo, los nodos inferiores incluyen una cuenta de nodo con cada uno de los protocolos multidifusión que ha acordado utilizar. Este número informa al nodo superior sobre cuantos nodos de cada uno de sus subárboles soportan dicho protocolo. Agregando dichos números para todas las conexiones descendentes, un nodo puede elegir el protocolo multidifusión que goza de mayor soporte. Ello proporciona cierta visibilidad para más de un nivel en la jerarquía del dominio.
- 6) *¿Cuáles son los protocolos preferidos?.* Durante el arbitraje de protocolo los nodos inferiores pueden asimismo comunicar sus preferencias de protocolo. Esta información puede utilizarse para finalizar una selección de protocolo si el resto de los factores no inducen claramente a una mejor elección.

Una vez que se ha seleccionado un protocolo multidifusión para cada uno de los nodos inferiores, la información se transmite a los mismos enviando un **MAPAddGroupRequest** a cada uno de ellos. Ello da lugar a que cada nodo receptor añada dicho grupo a su lista de grupos activos. Asimismo también da lugar a que cada nodo continúe diseminando la asignación de grupo en sentido descendente repitiendo los pasos anteriores para cada una de sus conexiones descendentes.

Debido a que **MAPAddGroupRequest** especifica la gama de prioridades que deben utilizar el grupo multidifusión, es posible utilizar distintos grupos para distintas prioridades (incluso para el mismo canal). Cuando se determina cual es el grupo multidifusión al que se deben enviar datos, MAP utiliza el identificador del canal, el nivel de fiabilidad y la prioridad. Si no se establece la correspondencia entre una determinada prioridad y la multidifusión, los datos con dicha prioridad se envían en unidifusión.

Nótese que no todos los nodos de una isla multidifusión necesitan incorporarse de forma inmediata a un grupo multidifusión. Un nodo se incorporará al grupo si se cumple cualquiera de las condiciones siguientes: si existen anexos del usuario local que necesitan recibir los datos, si existen conexiones a través de otras pilas de transporte que necesitan recibir los datos o si el nodo debe redireccionar los datos en unidifusión a cualquiera de sus vecinos. Algunos protocolos de transporte pueden requerir que un nodo se incorpore a un grupo antes de transmitirle datos (incluso si dicho nodo carece de interés alguno en recibir datos del grupo).

El cuadro siguiente resume el contenido de **MAPAddGroupRequest**.

Metacanal
Identificador del protocolo de transporte
Dirección del grupo multidifusión

*Metacanal:* Este campo informa sobre cual es el metacanal que se hace corresponder con un grupo multidifusión.

*Identificador del protocolo de transporte:* Este campo identifica inequívocamente el protocolo de transporte que debe utilizarse para este metacanal.

*Dirección del grupo multidifusión:* Es la dirección del grupo multidifusión a la que se enviarán los datos para el metacanal. El formato de este campo es idéntico al de una dirección de NSAP.

#### A.9.8.1 Metacanal

El cuadro siguiente define los campos que componen un metacanal. Este tipo se utiliza para identificar la abstracción a la que el MAP asigna grupos multidifusión.

Identificador del canal
Nivel de fiabilidad
Gama de prioridad

*Identificador del canal:* Este campo indica cual es el canal MCS asociado con el metacanal.

*Nivel de fiabilidad:* Este campo informa del nivel de fiabilidad asociado con el metacanal.

*Gama de prioridad:* Es la gama de prioridades asociadas con el metacanal (cero es la prioridad más elevada).

#### A.9.9 MAPRemoveGroupRequest

Esta MAPPDU se utiliza para suprimir grupos multidifusión de nodos inferiores de la isla multidifusión. Este proceso es iniciado por el proveedor de grupo multidifusión cuando es necesario suprimir la utilización de un grupo multidifusión. Cuando un nodo recibe una **MAPRemoveGroupRequest** debe enviar la petición en sentido descendente a todos los nodos ubicados por debajo del mismo en la isla multidifusión.

Típicamente, una **MAPRemoveGroupRequest** no se envía hacia abajo hasta que no haya cesado la utilización del grupo multidifusión. Es posible asegurar que un grupo multidifusión no está siendo utilizado activamente emitiendo una **MAPDisableMulticastRequest** hacia abajo y esperando una **MAPDisableMulticastConfirm** (de todos los nodos inferiores de la isla multidifusión). Una vez que esto se ha realizado, el proveedor de grupo multidifusión puede estar seguro que nadie está utilizando el grupo, de forma que éste se puede suprimir sin temor a perder datos.

Cuando un nodo recibe una **MAPRemoveGroupRequest** debe suprimir de forma inmediata el grupo y enviar la petición hacia abajo a todos los nodos inferiores de la isla.

El cuadro siguiente resume el contenido de **MAPRemoveGroupRequest**.

Metacanal
-----------

*Metacanal*: Este campo informa sobre cual es el metacanal del que se suprime el grupo.

#### **A.9.10 MAPDisableUnicastRequest**

Esta MAPPDU se utiliza para inhabilitar datos unidifusión procedentes de un nodo vecino para un flujo de datos específico.

Un nodo no debe transmitir esta MAPPDU hasta que no haya verificado que puede cambiar a una situación de recepción multidifusión sin pérdida de datos. Para ello debe haber recibido al menos una trama de datos en unidifusión a fin de conocer cual es el número siguiente de la secuencia. Además, debe recibir una trama de datos en multidifusión con un número de secuencia menor o igual que el número de secuencia esperado (de forma que sepa que está sincronizado). Solo entonces considera que está preparado para dar por terminado el flujo de datos unidifusión (para dicho flujo de datos).

El cuadro siguiente resume el contenido de **MAPDisableUnicastRequest**.

Identificador del flujo de datos
----------------------------------

*Identificador del flujo de datos*: Este campo contiene la información necesaria para determinar el flujo de datos para el cual se inhabilita la unidifusión. El identificador del flujo de datos se define completamente en la subcláusula sobre la PDU **MAPData**. El ID del emisor se omite para flujos de datos descendentes uniformes y para flujos de datos en representación, estando presente en cualquier otro caso. El ID del metacanal y el tipo de datos siempre estará presente.

#### **A.9.11 MAPEnableUnicastRequest**

Esta MAPPDU se utiliza para el que se rehabilita la unidifusión procedentes de un nodo vecino para un flujo de datos específico.

Esta MAPDU se envía si un nodo determina que no recibe más datos en unidifusión o que la tasa de errores de la transmisión multidifusión es tan alta que no es eficiente seguir utilizándola. Debe de haber una comunicación local entre MAP y el protocolo de transporte para determinar cuando es necesaria. Ello puede ocurrir como respuesta a las PDU **MAPSequenceNumber** que revelan que se ha detenido la recepción de un flujo de datos multidifusión.

El cuadro siguiente resume el contenido de **MAPEnableUnicastRequest**.

Identificador del flujo de datos
----------------------------------

*Identificador del flujo de datos*: Este campo contiene la información necesaria para determinar cual es el flujo de datos para el que se habilita la unidifusión. El identificador del flujo de datos se define completamente en la subcláusula sobre la PDU **MAPData**. El identificador del emisor se omite para flujos de datos descendentes uniformes y para flujos de datos en representación, estando presente en cualquier otro caso. El identificador del metacanal y el tipo de datos siempre estarán presentes.

#### **A.9.12 MAPEnableUnicastConfirm**

Esta MAPPDU confirma que la unidifusión se ha restablecido para un determinado flujo de datos. Cuando un nodo informa a un nodo vecino que vuelva a ponerse en modo unidifusión, puede ser necesario que dicho nodo reciba aún datos en multidifusión. Esto puede ocurrir si es demasiado tarde para que el nodo vecino envíe algunas tramas de datos en unidifusión debido a que ya

han sido enviadas. Esta MAPPDU permite al nodo vecino informar al originador de la **MAPEnableUnicastConfirm** cual es el número de la siguiente secuencia que va a ser enviada en unidifusión. De esta forma, el nodo puede seguir utilizando la multidifusión (si ello es posible) de forma que pueda atender todas las tramas de datos hasta aquélla. Esto es necesario para garantizar que no hay pérdida de datos durante la transición de multidifusión a unidifusión.

Tómese, por ejemplo, el caso de un nodo que ha recibido secuencias numeradas hasta la 20 para un determinado flujo de datos (pero con una tasa de errores muy alta). Puede optar por volver a unidifusión a fin de mejorar la tasa de errores. Sin embargo, el nodo vecino puede haber procesado secuencias de dicho flujo de datos numeradas hasta la 23. Si el nodo comienza inmediatamente a trabajar en unidifusión, perderá las secuencias 21 á 23. Esta confirmación informa al originador de la petición que la siguiente secuencia que enviará en unidifusión será la 24, de tal forma que todas las secuencias hasta la 23 deben ser recibidas en multidifusión antes de pasar a multidifusión.

El cuadro siguiente resume el contenido de **MAPEnableUnicastConfirm**.

Identificador del flujo de datos
Número de secuencia

*Identificador del flujo de datos:* Este campo contiene la información necesaria para determinar cual es el flujo de datos para el que se habilita la unidifusión. El identificador del flujo de datos se define completamente en la subcláusula sobre la PDU **MAPData**. El identificador del emisor se omite para flujos de datos descendentes uniformes y para flujos de datos en representación, estando presente en cualquier otro caso. El identificador del metacanal y el tipo de datos siempre estarán presentes.

*Número de secuencia:* Es el siguiente número de secuencia del flujo de datos que se enviará al nodo en unidifusión. Todo lo anterior a este número de secuencia debe recibirse en multidifusión antes de que el nodo pase a trabajar en unidifusión.

### A.9.13 **MAPDisableMulticastRequest**

Esta MAPPDU se transmite en sentido descendente en la isla multidifusión para inhabilitar la utilización de la multidifusión antes de la supresión del grupo.

Cuando un nodo recibe una **MAPDisableMulticastRequest**, debe iniciar el proceso para la rehabilitación de la unidifusión para cualquier flujo de datos para el que la unidifusión se inhabilitó anteriormente (para este grupo). Lo hace enviando una **MAPEnableUnicastRequest** hacia el emisor para cada flujo de datos afectado. El nodo debe asimismo marcar al grupo indicando que se encuentra en el estado inhabilitado, lo cual impide al nodo inhabilitar la unidifusión para cualquier flujo de datos del grupo. Sin embargo, NO debe dejar de transmitir hacia el grupo.

La **MAPDisableMulticastRequest** debe enviarse en sentido descendente de forma sucesiva a todos los nodos que se encuentran por debajo del receptor. Un nodo no puede responder con una **MAPDisableMulticastConfirm** hasta que no haya recibido confirmaciones similares de todos los nodos ubicados más abajo del mismo Y hasta que no haya rehabilitado la unidifusión para todos los flujos de datos para los que ésta fue previamente inhabilitada (para este grupo). No puede asumir que la unidifusión se rehabilita para un flujo de datos hasta que no haya recibido una **MAPEnableUnicastConfirm** para cada petición pendiente (y debe haber recibido todos los datos en multidifusión con un número de secuencia menor de uno en cada confirmación).

El cuadro siguiente resume el contenido de **MAPDisableMulticastRequest**.

Metacanal
-----------

*Metacanal:* Es el metacanal asociado con el grupo para el que se inhabilita la multidifusión. El tipo de metacanal se describe completamente en la subcláusula sobre **MAPRemoveGroupRequest**.

#### **A.9.14 MAPDisableMulticastConfirm**

Esta MAPPDU se utiliza para confirmar que todos los nodos ubicados por debajo del receptor han dejado de utilizar un grupo multidifusión.

Cuando un nodo recibe una **MAPDisableMulticastRequest** del nodo superior debe responder enviando una **MAPDisableMulticastConfirm** después de que se hayan cumplido dos condiciones: que haya recibido una PDU **MAPDisableMulticastConfirm** de todos los nodos situados debajo de él en la isla multidifusión (si hay alguno) y que esté seguro de que la unidifusión está habilitada para todos los flujos de datos del grupo.

La primera condición se cumple como resultado de enviar la **MAPDisableMulticastRequest** a los nodos inferiores. La segunda condición se cumple rehabilitando la unidifusión para todos los flujos de datos que fueron anteriormente inhabilitados (esto se hace utilizando la PDU **MAPEnableUnicastRequest**).

El cuadro siguiente resume el contenido de **MAPDisableMulticastConfirm**.

Metacanal
-----------

*Metacanal:* Es el metacanal asociado con el grupo para el que se inhabilita la multidifusión. El tipo de metacanal se describe completamente en la subcláusula sobre **MAPRemoveGroupRequest**.

#### **A.9.15 MAPEnableMulticastRequest**

Esta MAPPDU se utiliza para rehabilitar un grupo multidifusión que había sido previamente inhabilitado.

Cuando un nodo recibe una **MAPEnableMulticastRequest** para un grupo que está marcado como inhabilitado (pero que aún es válido) debe rehabilitar la utilización de dicho grupo. Ello significa que las PDU **MAPDisableUnicastRequest** pueden ser procesadas de nuevo a fin de clausurar el flujo de datos unidifusión. Un grupo rehabilitado adquiere el mismo estado que cuando fue creado.

Esta petición debe enviarse en sentido descendente sucesivamente a todos los nodos que se encuentran por debajo del receptor.

El cuadro siguiente resume el contenido de **MAPEnableMulticastRequest**.

Metacanal
-----------

*Metacanal:* Es el metacanal asociado con el grupo para el que se inhabilita la multidifusión. El tipo de metacanal se describe completamente en la subcláusula sobre **MAPRemoveGroupRequest**.

#### **A.9.16 MAPSequenceNumber**

Esta MAPPDU se utiliza para informar a todos los receptores que se encuentran por debajo sobre cuales son los números de secuencia que deben haber recibido hasta ese momento. Esta MAPPDU se disemina a partir del originador del flujo de datos en unidifusión y la repiten todos los nodos receptores hacia los nodos que se encuentran aún más distantes del emisor (hacia todas las conexiones en sentido ascendente y en sentido descendente que conducen a usuarios que están incorporados al canal especificado). De esta forma todos los nodos pueden ser informados de cuales son los números de secuencia que deben haberse recibido para dicho flujo de datos.

Esta MAPPDU puede utilizarse para determinar cuando un nodo deja de recibir datos en un grupo multidifusión. Esto puede ser de utilidad para protocolos no fiables en los que no exista la corrección de errores. También puede ser útil para protocolos de transporte que no detectan con claridad la pérdida del trayecto multidifusión desde un receptor a un transmisor. La utilización precisa de esta MAPPDU requiere estudios adicionales.

El cuadro siguiente resume el contenido de **MAPSequenceNumber**.

Identificador del flujo de datos
Número de secuencia

*Identificador del flujo de datos:* Este campo contiene la información necesaria para determinar cual es el flujo de datos asociado al número de secuencia. El identificador del flujo de datos se define completamente en la subcláusula sobre la PDU **MAPData**. El identificador del emisor se omite para flujos de datos descendentes uniformes y para flujos de datos en representación, estando presente en cualquier otro caso. El identificador del metacanal y el tipo de datos siempre estarán presentes.

*Número de secuencia:* Es el último número de secuencia que ha sido transmitido para el flujo de datos especificado.

#### A.10 Definición ASN.1 de las MAPPDU

Esta subcláusula contiene el formato real de las MAPPDU utilizando la notación ASN.1. El contenido de esta subcláusula también está disponible en forma de fichero de texto compilable y separado.

```

--*****
--*
--*      ASN.1 Definition for MAP PDUs
--*
--*****

```

**MAP-PROTOCOL DEFINITIONS AUTOMATIC TAGS ::=**

**BEGIN**

**H221NonStandardIdentifier ::= OCTET STRING (SIZE (4..255))**

**Key ::= CHOICE**

```

{
    object          OBJECT IDENTIFIER,
    h221NonStandard H221NonStandardIdentifier
}

```

**NonStandardParameter ::= SEQUENCE**

```

{
    key          Key,
    data        OCTET STRING
}

```

**NonStandardPDU ::= SEQUENCE**

```

{
    data        NonStandardParameter,
    ...
}

```

**VersionNumber ::= SEQUENCE**  
 {  
   **majorVersionNumber**                   **INTEGER (0 .. 65535),**  
   **minorVersionNumber**               **INTEGER (0 .. 65535),**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**Priority ::= INTEGER (0 .. 15)**

**PriorityRange ::= SEQUENCE**  
 {  
   **highPriority**                       **Priority,**  
   **lowPriority**                       **Priority,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPConnectRequestPDU ::= SEQUENCE**  
 {  
   **versionNumber**                   **VersionNumber,**  
   **connectionMAPSAP**               **INTEGER (0 .. 65535),**  
   **domainReferenceID**               **INTEGER (0 .. 65535),**  
   **priorityRange**                   **PriorityRange,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPConnectConfirmPDU ::= SEQUENCE**  
 {  
   **versionNumber**                   **VersionNumber,**  
   **connectionMAPSAP**               **INTEGER (0 .. 65535),**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**Reason ::= CHOICE**  
 {  
   **providerInitiated**               **NULL,**  
   **userRejected**                   **NULL,**  
   **userInitiated**                   **NULL,**  
   **invalidMAPSAP**                   **NULL,**  
   **invalidDomainReferenceID**       **NULL,**  
   **unicastTransition**               **NULL,**  
   **unspecifiedFailure**             **NULL,**  
   **nonStandardReason**             **NonStandardParameter,**  
   ...  
 }

**MAPDisconnectRequestPDU ::= SEQUENCE**  
 {  
   **reason**                           **Reason,**  
   **confirmRequired**               **BOOLEAN,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

```

MAPDisconnectConfirmPDU ::= SEQUENCE
{
    nonStandardParameters      SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

TransportProtocolID ::= CHOICE
{
    objectProtocolID           OBJECT IDENTIFIER,
    h221NonStandardProtocolID H221NonStandardIdentifier,
    snapProtocolID             OCTET STRING (SIZE (5)),
    nonStandardProtocolID     NonStandardParameter,
    ...
}

```

```

TransportProtocolType ::= CHOICE
{
    reliableUnicast            NULL,
    unreliableUnicast          NULL,
    reliableMulticast          NULL,
    unreliableMulticast        NULL,
    nonStandardProtocolType    NonStandardParameter,
    ...
}

```

```

NetworkAddress ::= SEQUENCE
{
    nsapAddress                OCTET STRING (SIZE (1..20)),
    nonStandardParameters      SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**PayloadSize ::= INTEGER (128 .. 65535)**

```

TransportProtocolRequestEntry ::= SEQUENCE
{
    transportProtocolID        TransportProtocolID,
    transportProtocolType      TransportProtocolType,
    networkAddress             NetworkAddress,
    maximumPayloadFixedFlag    BOOLEAN,
    maximumPayloadSize         PayloadSize,
    preferenceWeighting         INTEGER (0 .. 65535) OPTIONAL,
    nodeCount                  INTEGER (0 .. 65535) OPTIONAL,
    numberOfConnections         INTEGER (0 .. 65535) OPTIONAL,
    configurationData          OCTET STRING OPTIONAL,
    nonStandardParameters      SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

```

MAPArbitrateProtocolsRequestPDU ::= SEQUENCE
{
    domainReferenceID          INTEGER (0 .. 65535),
    moreToComeFlag             BOOLEAN,
    transportProtocolMenu      SEQUENCE OF TransportProtocolRequestEntry,
    nonStandardParameters      SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**TransportProtocolConfirmEntry ::= SEQUENCE**

```
{
    transportProtocolID          TransportProtocolID,
    transportProtocolType        TransportProtocolType,
    networkAddress               NetworkAddress,
    maximumPayloadSize           PayloadSize,
    preferenceWeighting          INTEGER (0 .. 65535) OPTIONAL,
    nodeCount                    INTEGER (0 .. 65535) OPTIONAL,
    numberOfConnections          INTEGER (0 .. 65535) OPTIONAL,
    configurationData            OCTET STRING OPTIONAL,
    nonStandardParameters        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}
```

**MAPArbitrateProtocolsConfirmPDU ::= SEQUENCE**

```
{
    domainReferenceID           INTEGER (0 .. 65535),
    moreToComeFlag              BOOLEAN,
    transportProtocolMenu        SEQUENCE OF TransportProtocolConfirmEntry,
    nonStandardParameters        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}
```

**SenderID ::= INTEGER (1001 .. 65535)**

**ChannelID ::= INTEGER (1 .. 65535)**

**ReliabilityLevel ::= CHOICE**

```
{
    reliable                     NULL,
    unreliable                    NULL,
    nonStandardReliabilityLevel  NonStandardParameter,
    ...
}
```

**DataType ::= CHOICE**

```
{
    nonuniform                   NULL,
    uniform                      NULL,
    proxy                       NULL,
    nonStandardDataType          NonStandardParameter,
    ...
}
```

**DataFlowIdentifier ::= SEQUENCE**

```
{
    senderID                     SenderID OPTIONAL,

    metachannelID SEQUENCE
    {
        channelID                ChannelID,
        reliabilityLevel          ReliabilityLevel,
        priority                  Priority,
        ...
    } OPTIONAL,

    dataType                     DataType OPTIONAL,
    nonStandardParameters        SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}
```

**SequenceNumber ::= INTEGER (0 .. 65535)**

**DataDescriptor ::= SEQUENCE**

```
{  
    unicastForwardFlag          BOOLEAN,  
    startingSequenceNumber      SequenceNumber,  
    dataFlowIdentifier          DataFlowIdentifier OPTIONAL,  
    domainReferenceID           INTEGER (0 .. 65535) OPTIONAL,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**DataFrameEntry ::= SEQUENCE**

```
{  
    firstSegmentFlag           BOOLEAN,  
    lastSegmentFlag           BOOLEAN,  
    userData                   OCTET STRING,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**MAPDataPDU ::= SEQUENCE**

```
{  
    dataDescriptor             DataDescriptor OPTIONAL,  
    dataFrameArray            SEQUENCE OF DataFrameEntry,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**Metachannel ::= SEQUENCE**

```
{  
    channelID                  ChannelID,  
    reliabilityLevel           ReliabilityLevel,  
    priorityRange              PriorityRange,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**MAPAddGroupRequestPDU ::= SEQUENCE**

```
{  
    metachannel                Metachannel,  
    transportProtocolID        TransportProtocolID,  
    multicastGroupAddress      NetworkAddress,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**MAPRemoveGroupRequestPDU ::= SEQUENCE**

```
{  
    metachannel                Metachannel,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**MAPDisableUnicastRequestPDU ::= SEQUENCE**

```
{  
    dataFlowIdentifier          DataFlowIdentifier,  
    nonStandardParameters       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

**MAPEnableUnicastRequestPDU ::= SEQUENCE**  
 {  
   **dataFlowIdentifier**                   **DataFlowIdentifier,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPEnableUnicastConfirmPDU ::= SEQUENCE**  
 {  
   **dataFlowIdentifier**                   **DataFlowIdentifier,**  
   **sequenceNumber**                   **SequenceNumber,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPDisableMulticastRequestPDU ::= SEQUENCE**  
 {  
   **metachannel**                       **Metachannel,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPDisableMulticastConfirmPDU ::= SEQUENCE**  
 {  
   **metachannel**                       **Metachannel,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPEnableMulticastRequestPDU ::= SEQUENCE**  
 {  
   **metachannel**                       **Metachannel,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPSequenceNumberPDU ::= SEQUENCE**  
 {  
   **dataFlowIdentifier**                   **DataFlowIdentifier,**  
   **sequenceNumber**                   **SequenceNumber,**  
   **nonStandardParameters**           **SEQUENCE OF NonStandardParameter OPTIONAL,**  
   ...  
 }

**MAPPDU ::= CHOICE**  
 {  
   **mapConnectRequest**                   **MAPConnectRequestPDU,**  
   **mapConnectConfirm**                   **MAPConnectConfirmPDU,**  
   **mapDisconnectRequest**               **MAPDisconnectRequestPDU,**  
   **mapDisconnectConfirm**               **MAPDisconnectConfirmPDU,**  
   **mapArbitrateProtocolsRequest**       **MAPArbitrateProtocolsRequestPDU,**  
   **mapArbitrateProtocolsConfirm**       **MAPArbitrateProtocolsConfirmPDU,**  
   **mapData**                               **MAPDataPDU,**  
   **mapAddGroupRequest**                   **MAPAddGroupRequestPDU,**  
   **mapRemoveGroupRequest**               **MAPRemoveGroupRequestPDU,**  
   **mapDisableUnicastRequest**           **MAPDisableUnicastRequestPDU,**  
   **mapEnableUnicastRequest**           **MAPEnableUnicastRequestPDU,**  
   **mapEnableUnicastConfirm**           **MAPEnableUnicastConfirmPDU,**  
   **mapDisableMulticastRequest**       **MAPDisableMulticastRequestPDU,**  
   **mapDisableMulticastConfirm**       **MAPDisableMulticastConfirmPDU,**  
 }

mapEnableMulticastRequest  
mapSequenceNumber  
nonStandardPDU

MAPEnableMulticastRequestPDU,  
MAPSequenceNumberPDU,  
NonStandardPDU,

...

}

END

```
--*****  
--*  
--*      End of ASN.1 Definition for MAP PDUs  
--*  
--*****
```



## SERIES DE RECOMENDACIONES DEL UIT-T

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