



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**T.125**

(02/98)

SÉRIE T: TERMINAUX DES SERVICES TÉLÉMATIQUES

---

**Spécification du protocole du service de  
communication multipoint**

Recommandation UIT-T T.125

(Antérieurement Recommandation du CCITT)

---

RECOMMANDATIONS UIT-T DE LA SÉRIE T  
**TERMINAUX DES SERVICES TÉLÉMATIQUES**



*Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.*

## RECOMMANDATION UIT-T T.125

### SPECIFICATION DU PROTOCOLE DU SERVICE DE COMMUNICATION MULTIPPOINT

#### Résumé

La présente Recommandation définit un protocole exploitant toute la hiérarchie d'un domaine de communication multipoint. Elle spécifie le format des messages de protocole et les procédures qui régissent l'échange de tels messages sur un ensemble de connexions de transport. L'objet de ce protocole est de mettre en œuvre le service de communication multipoint qui est défini dans la Recommandation T.122.

L'Annexe A définit un protocole permettant d'utiliser des services de réseau multidiffusion dans une conférence T.120. Le processus repose sur l'acheminement des données dans le cadre du protocole T.125. Une fois assumée la responsabilité de l'acheminement, il aura recours à la multidiffusion dans la mesure du possible. La présente annexe traite les réseaux sous-optimaux par le contrôle des chemins de multidiffusion avant toute dépendance à l'égard de ceux-ci. Si le principe de la multidiffusion ne s'applique pas à un certain moment de la conférence, celui de la multidiffusion unique est utilisé à la place. La présente annexe est indépendante des réseaux auxquels elle s'applique. Toutes les questions spécifiques à un réseau donné sont traitées par les piles de protocole inférieures à ce protocole.

Les modifications suivantes ont été incorporées à la présente version de la Recommandation:

- le protocole MAP a été incorporé en tant qu'Annexe A;
- la version 3 du protocole MCS est spécifiée;
- des unités MCSPDU de *paramètre étendu* ont été ajoutées;
- des unités MCSPDU de *notification de capacités* ont été ajoutées;
- des champs ont été ajoutés aux unités MCSPDU Transmission de données et Transmission uniforme de données afin de permettre les modifications de la Recommandation T.122;
- des modifications mineures ont été apportées au format du document.

#### Source

La Recommandation UIT-T T.125, révisée par la Commission d'études 16 de l'UIT-T (1997-2000), a été approuvée le 6 février 1998 selon la procédure définie dans la Résolution n° 1 de la CMNT.

## AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

## NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

## DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 1998

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

## TABLE DES MATIÈRES

	<b>Page</b>
1	Domaine d'application ..... 1
2	Références normatives ..... 1
3	Définitions ..... 2
4	Abréviations ..... 4
5	Aperçu général du protocole MCS ..... 5
5.1	Modèle de couche MCS ..... 5
5.2	Services assurés par la couche MCS ..... 5
5.2.1	Services de couche MAP ..... 6
5.3	Services assurés par la couche Transport ..... 6
5.4	Fonctions de la couche MCS ..... 6
5.4.1	Gestion de domaine ..... 6
5.4.2	Gestion de canal ..... 9
5.4.3	Transfert de données ..... 9
5.4.4	Gestion de jeton ..... 9
5.4.5	Notification de capacités ..... 9
5.5	Traitement hiérarchique ..... 10
5.6	Paramètres de domaine ..... 11
5.7	Paramètres étendus ..... 12
6	Utilisation du service de transport ..... 13
6.1	Modèle du service de transport ..... 13
6.2	Utilisation de connexions multiples ..... 14
6.3	Libération d'une connexion de transport ..... 15
7	Structure des unités MCSPDU de la version 2 ..... 15
8	Structure des unités MCSPDU de la version 3 ..... 25
9	Codage des unités MCSPDU ..... 38
10	Acheminement des unités MCSPDU ..... 39
10.1	Unités MCSPDU de type connect et à paramètres étendus ..... 39
10.2	Unités MCSPDU de domaine ..... 41
11	Signification des unités MCSPDU ..... 44
11.1	Connect-Initial ..... 44
11.2	Connect-Response ..... 46
11.3	Connect-Additional ..... 46

	<b>Page</b>
11.4 Connect-Result.....	47
11.5 Extended-Parameters-Propose .....	47
11.6 Extended-Parameters-Accept.....	48
11.7 PlumbDomainIndication .....	48
11.8 ErectDomainRequest .....	49
11.9 MergeChannelsRequest .....	50
11.10 MergeChannelsConfirm.....	51
11.11 PurgeChannelsIndication .....	52
11.12 MergeTokensRequest .....	52
11.13 MergeTokensConfirm.....	54
11.14 PurgeTokensIndication .....	54
11.15 DisconnectProviderUltimatum .....	55
11.16 RejectMCSPDUUltimatum .....	55
11.17 AttachUserRequest .....	55
11.18 AttachUserConfirm.....	56
11.19 DetachUserRequest.....	56
11.20 DetachUserIndication.....	57
11.21 ChannelJoinRequest.....	58
11.22 ChannelJoinConfirm.....	59
11.23 ChannelLeaveRequest.....	60
11.24 ChannelConveneRequest .....	60
11.25 ChannelConveneConfirm .....	61
11.26 ChannelDisbandRequest.....	61
11.27 ChannelDisbandIndication.....	62
11.28 ChannelAdmitRequest .....	62
11.29 ChannelAdmitIndication.....	62
11.30 ChannelExpelRequest.....	63
11.31 ChannelExpelIndication.....	63
11.32 SendDataRequest .....	64
11.33 SendDataIndication.....	65
11.34 UniformSendDataRequest .....	66
11.35 UniformSendDataIndication .....	67
11.36 TokenGrabRequest .....	68
11.37 TokenGrabConfirm.....	69
11.38 TokenInhibitRequest.....	69

	<b>Page</b>
11.39 TokenInhibitConfirm .....	70
11.40 TokenGiveRequest.....	70
11.41 TokenGiveIndication .....	70
11.42 TokenGiveResponse .....	71
11.43 TokenGiveConfirm.....	72
11.44 TokenPleaseRequest .....	72
11.45 TokenPleaseIndication .....	72
11.46 TokenReleaseRequest.....	73
11.47 TokenReleaseConfirm .....	73
11.48 TokenTestRequest .....	74
11.49 TokenTestConfirm.....	74
11.50 CapabilitiesNotificationRequest .....	74
11.50.1 Capacité de demande .....	75
11.51 CapabilitiesNotificationIndication .....	75
11.51.1 Capacité d'indication.....	76
12 Base de données de fournisseur de service MCS.....	76
12.1 Copie hiérarchique .....	76
12.2 Informations relatives aux canaux .....	78
12.3 Informations relatives aux jetons .....	79
13 Eléments de procédure.....	81
13.1 Ordonnancement des unités MCSPDU.....	81
13.2 Commande du flux d'entrée .....	82
13.3 Application du débit.....	83
13.4 Configuration de domaine.....	84
13.5 Fusion de domaines .....	85
13.6 Déconnexion de domaine.....	88
13.7 Attribution des identificateurs de canal .....	89
13.8 Etat des jetons .....	89
13.9 Notification de capacités.....	90
13.9.1 Identificateurs de capacité .....	90
13.9.2 Sommets de la version 3.....	90
13.9.3 Règles de sélection de capacité.....	91
13.9.4 Etablissement du fournisseur de sommet V3 .....	92
13.9.5 Extension du sommet V3.....	92
13.9.6 Traitement des unités PDU à un nœud feuille.....	93
13.9.7 Unités PDU de traitement à un nœud intermédiaire.....	93

	<b>Page</b>
13.10 Arbitrage des versions de protocole.....	97
13.11 Interfonctionnement des versions de protocole .....	98
Annexe A – Protocole d'adaptation de multidiffusion.....	99
A.1 Domaine d'application .....	99
A.2 Références normatives .....	99
A.3 Définitions .....	100
A.4 Abréviations.....	101
A.5 Aperçu général .....	102
A.5.1 Utilisation du principe de la multidiffusion.....	103
A.5.2 Ilots multidiffusion .....	103
A.5.3 Fournisseurs de groupe de multidiffusion .....	105
A.5.4 Ordonnancement de données.....	106
A.5.5 Usage des principes de la multidiffusion et de la monodiffusion non fiable.	106
A.6 Utilisation du protocole MAP.....	107
A.6.1 Gestion de domaine .....	107
A.6.2 Gestion de canal.....	108
A.6.3 Transfert de données.....	108
A.6.4 Gestion des jetons.....	111
A.7 Utilisation des protocoles de transport.....	111
A.7.1 Protocoles de transport pour la monodiffusion.....	111
A.7.2 Protocoles de transport multidiffusion .....	113
A.7.3 Information locale.....	115
A.8 Spécification de protocole.....	116
A.8.1 Etablissement de la connexion de monodiffusion fiable initiale.....	117
A.8.2 Arbitrage de protocoles de transport.....	117
A.8.3 Attribution et distribution de groupes multidiffusion.....	121
A.8.4 Gestion du trafic de données multidiffusion (réglage automatique).....	122
A.8.5 Abandon de la multidiffusion .....	124
A.8.6 Suppression de groupes multidiffusion .....	125
A.9 Description des unités MAPPDU .....	126
A.9.1 MAPConnectRequest .....	126
A.9.2 MAPConnectConfirm.....	127
A.9.3 MAPDisconnectRequest.....	128
A.9.4 MAPDisconnectConfirm .....	128
A.9.5 MAPArbitrateProtocolsRequest .....	128
A.9.6 MAPArbitrateProtocolsConfirm .....	131
A.9.7 MAPData .....	133

	<b>Page</b>
A.9.8 MAPAddGroupRequest.....	136
A.9.9 MAPRemoveGroupRequest .....	138
A.9.10 MAPDisableUnicastRequest .....	138
A.9.11 MAPEnableUnicastRequest .....	138
A.9.12 MAPEnableUnicastConfirm.....	139
A.9.13 MAPDisableMulticastRequest .....	140
A.9.14 MAPDisableMulticastConfirm.....	140
A.9.15 MAPEnableMulticastRequest .....	141
A.9.16 MAPSequenceNumber .....	141
A.10 Définition de la notation ASN.1 des unités MAPPDU.....	142



## Recommandation T.125

### SPECIFICATION DU PROTOCOLE DU SERVICE DE COMMUNICATION MULTIPOINT

(révisée en 1998)

#### 1 Domaine d'application

La présente Recommandation spécifie:

- a) les procédures d'un unique protocole de transfert de données et d'informations de commande entre un fournisseur de service de communication multipoint (service MCS) et un fournisseur de service MCS homologue;
- b) la structure et le codage des unités de données du protocole MCS qui sont utilisés pour le transfert des données et des informations de commande.

Les procédures sont définies en termes:

- a) d'interactions entre fournisseurs de service MCS homologues par échange d'unités de données de protocole MCS;
- b) d'interactions entre un fournisseur de service MCS et des utilisateurs de ce service par échange de primitives MCS;
- c) d'interactions entre un fournisseur de service MCS et un fournisseur de service de transport par échange de primitives du service de transport.

Ces procédures sont applicables aux instances de communication multipoint entre des systèmes compatibles avec le service MCS et appelés à être interconnectés dans un environnement de systèmes ouverts.

#### 2 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui de ce fait en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

- Recommandation UIT-T T.122 (1998), *Service de communication multipoint – Définition du service.*
- Recommandation UIT-T T.123 (1996), *Piles protocolaires de données propres au réseau pour conférences multimédias.*
- Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèles de référence de base – le modèle de référence de base.*
- Recommandation UIT-T X.214 (1995) | ISO/CEI 8072:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition du service de transport.*
- Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*

- Recommandation UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*

### 3 Définitions

NOTE – Ces définitions font appel aux abréviations définies au paragraphe 4.

La présente Recommandation est fondée sur les concepts développés dans la Recommandation X.200. Elle fait appel aux termes suivants, qui y sont définis:

- a) commande de flux;
- b) réassemblage;
- c) recombinaison;
- d) segmentation;
- e) séquençement;
- f) éclatement;
- g) syntaxe de transfert;
- h) connexion de transport;
- i) identificateur d'extrémité de connexion de transport;
- j) service de transport;
- k) point d'accès au service de transport;
- l) adresse de point d'accès au service de transport;
- m) unité de données du service de transport.

La présente Recommandation est également fondée sur les concepts développés dans la Recommandation T.122. Elle fait appel aux termes suivants, qui y sont définis:

- a) point MCSAP de contrôle;
- b) rattachement au point MCS;
- c) canal MCS;
- d) connexion MCS;
- e) domaine MCS;
- f) sélecteur de domaine MCS;
- g) canal privé MCS;
- h) gestionnaire de canal privé MCS;
- i) fournisseur de service MCS;
- j) point d'accès au service MCS;
- k) utilisateur du service MCS;
- l) identificateur d'utilisateur du service MCS;
- m) fournisseur supérieur du service MCS.

Pour les besoins de la présente Recommandation, les définitions suivantes s'appliquent.

- 3.1 unité de données du service MCS:** partie des données d'utilisateur du service MCS dont l'identité est protégée pendant le transfert de l'émetteur aux récepteurs. Plus précisément, il s'agit du contenu d'une demande MCS-SEND-DATA ou d'une demande MCS-UNIFORM-SEND-DATA.
- 3.2 unité de données d'interface MCS:** unité d'information transférée via un point MCSAP entre un utilisateur du service MCS et un fournisseur de service MCS au cours d'une même interaction. Chaque unité de données d'interface MCS contient des informations de commande d'interface. Elle peut également contenir tout ou partie d'une unité de données du service MCS.
- 3.3 unité de données de protocole MCS:** unité d'information échangée dans le cadre du protocole MCS, composée des informations de commande transférées entre des fournisseurs de service MCS pour coordonner leur opération commune et, le cas échéant, de données transférées pour le compte d'utilisateurs du service MCS auxquels ces fournisseurs sont en train de fournir ce service.
- 3.4 priorité de transfert de données MCS:** priorité en cause d'un des quatre niveaux suivants: absolue, élevée, moyenne, basse. La valeur est communiquée sans changement de l'émetteur aux récepteurs. Selon la valeur d'un paramètre de domaine MCS indiquant le nombre de priorités de transfert de données distinctes qui sont mises en œuvre, deux priorités inférieures ou plus peuvent recevoir la même qualité de service.
- 3.5 unité MCSPDU valide:** unité MCSPDU dont la structure et le codage sont conformes à la présente Recommandation.
- 3.6 unité MCSPDU non valide:** unité MCSPDU qui n'est pas valide.
- 3.7 erreur de protocole:** utilisation d'une unité MCSPDU d'une manière non compatible avec les procédures de la présente Recommandation.
- 3.8 unité MCSPDU de type Connect:** unité MCSPDU d'un des types suivants: **Connect-Initial, Connect-Response, Connect-Additional, Connect-Result.**
- 3.9 unité MCSPDU à paramètres étendus:** unité MCSPDU d'un des types suivants: **Extended-Parameter-Propose** ou **Extended-Parameter-Accept.**
- 3.10 unité MCSPDU de domaine:** toute unité MCSPDU qui n'est pas une unité MCSPDU de type Connect ou à paramètres étendus.
- 3.11 unité MCSPDU de données:** unité MCSPDU d'un des types suivants: **SendDataRequest, SendDataIndication, UniformSendDataRequest, UniformSendDataIndication.**
- 3.12 unité MCSPDU de contrôle:** toute unité MCSPDU de domaine qui n'est pas une unité MCSPDU de données.
- 3.13 connexion TC initiale:** première connexion de transport d'une connexion de service MCS, utilisée pour échanger des unités MCSPDU de contrôle et de données ayant une priorité absolue.
- 3.14 connexion TC supplémentaire:** connexion de transport subséquente, appartenant à une connexion de service MCS, utilisée pour échanger des unités MCSPDU de données ayant une priorité inférieure.
- 3.15 sous-arbre de fournisseur de service MCS:** dans le cadre d'un domaine MCS, arborescence constituée par le fournisseur de service MCS lui-même et ses rattachements au service MCS, plus tous les fournisseurs de service MCS qui lui sont hiérarchiquement subordonnés, avec leurs propres rattachements au service MCS.

**3.16 hauteur d'un fournisseur de service MCS:** dans le cadre d'un domaine MCS, position la plus élevée de tous les fournisseurs de service MCS hiérarchiquement subordonnés. Un fournisseur de service MCS sans subordonnés a la hauteur zéro.

**3.17 sommet de version 3 ou V3:** fournisseurs de service MCS d'une hiérarchie de domaines qui utilisent un protocole MCS de la version 3 ou d'une version supérieure et:

- qui sont des fournisseurs supérieurs;
- qui disposent d'un trajet de connexion ascendante vers un fournisseur supérieur de version 3, qui passe seulement par d'autres fournisseurs de version 3.

**3.18 nœud feuille:** nœud comportant une connexion ascendante et aucune connexion descendante.

**3.19 nœud intermédiaire:** nœud comportant une connexion ascendante et au moins une connexion descendante.

**3.20 nœud supérieur:** tout nœud comportant une connexion ascendante est relié à son nœud supérieur.

**3.21 nœud intervenant:** par rapport à un nœud spécifique, tout nœud sur le trajet entre le nœud spécifique et le fournisseur de sommet est un nœud intervenant.

**3.22 capacité:** comportement de niveau supérieur du système, exigeant vraisemblablement la coopération entre deux nœuds ou plus.

**3.23 identificateur de capacité:** identificateur assigné à une capacité. Lorsqu'il est normalisé, cet identificateur est documenté dans la Recommandation T.125.

**3.24 indicateur de participation:** caractéristique d'une capacité, qui indique le niveau de participation attendu des nœuds V3 d'un sommet V3 en vue de l'exercice de la capacité. La valeur de l'indicateur de participation est soit partielle (deux nœuds ou plus doivent participer) soit globale (tous les nœuds doivent participer).

**3.25 capacité partielle:** capacité à indicateur de participation "partielle".

**3.26 capacité globale:** capacité à indicateur de participation "globale".

**3.27 liste de capacités de sommet:** liste des capacités disponibles au sommet V3.

**3.28 fournisseur de sommet V3:** tout fournisseur supérieur qui est un nœud V3.

## 4 Abréviations

La présente Recommandation utilise les abréviations suivantes.

MAP	protocole d'adaptation multidiffusion ( <i>multicast adaptation protocol</i> ) décrit dans l'Annexe A
MAPSAP	point d'accès de service MAP ( <i>MAP service access point</i> )
MC	connexion MAP ( <i>MAP connection</i> )
MCS	service de communication multipoint ( <i>multipoint communication service</i> )
MCSAP	point d'accès au service MCS ( <i>MCS service access point</i> )
MCSPDU	unité de données protocolaires du service MCS ( <i>MCS protocol data unit</i> )
PDU	unité de données protocolaires ( <i>protocol data unit</i> )
TC	connexion de transport ( <i>transport connection</i> )
TS	service de transport ( <i>transport service</i> )

TSAP point d'accès au service de transport (*transport service access point*)  
 TSDU unité de données du service de transport (*transport service data unit*)  
 V2 version 2  
 V3 version 3

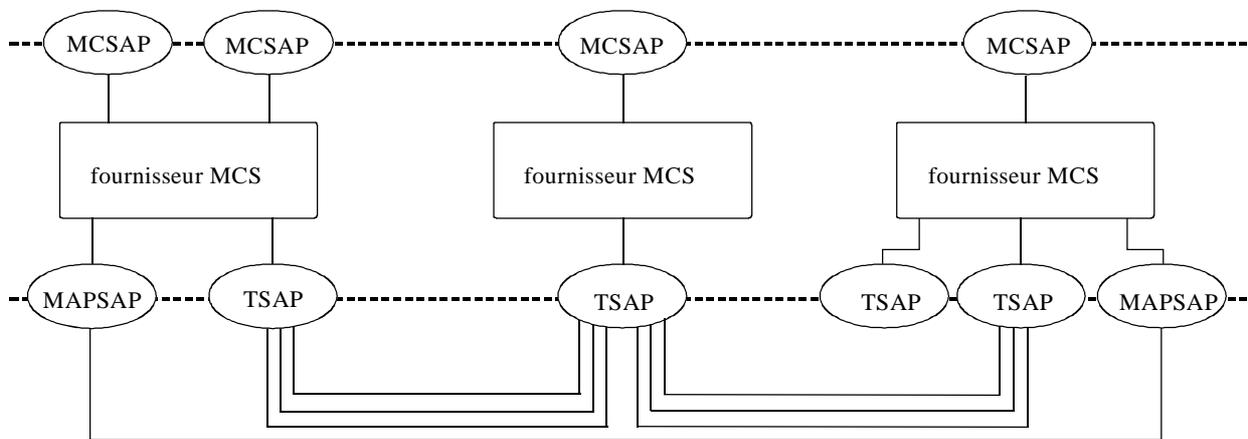
## 5 Aperçu général du protocole MCS

### 5.1 Modèle de couche MCS

Un fournisseur de service MCS communique avec les utilisateurs de ce service au moyen de primitives MCS définies dans la Recommandation T.122, passant par un point d'accès MCSAP. Ces primitives peuvent être la cause ou le résultat d'échanges d'unités MCSPDU entre fournisseurs MCS homologues utilisant une connexion MCS. Elles peuvent aussi être la cause ou le résultat d'actions prises dans le domaine d'un même fournisseur de service MCS. Les échanges d'unités MCSPDU ont lieu entre fournisseurs MCS qui couvrent le même domaine MCS.

Un fournisseur de service MCS peut avoir plusieurs homologues, à chacun desquels il est relié directement par une connexion MCS particulière ou indirectement par l'intermédiaire d'un fournisseur MCS homologue. Une connexion MCS comprend soit une connexion MAP, soit une ou plusieurs connexions de transport. Les connexions de type MAP fonctionneront de la façon décrite dans l'Annexe A. Pour les connexions de type transport, le nombre de connexions dépendra du nombre de priorités de transfert de données mises en jeu dans un domaine MCS. Les échanges de protocole s'effectuent au moyen des services de la couche Transport en passant par une paire de points TSAP.

Ce modèle de couche MCS est illustré à la Figure 5-1.



T1602400-97

**Figure 5-1/T.125 – Modèle de la couche MCS**

### 5.2 Services assurés par la couche MCS

Le protocole MCS prend en charge les services définis dans la Recommandation T.122. Les informations sont transférées à destination et en provenance d'un utilisateur MCS au moyen des primitives MCS énumérées dans le Tableau 5-1.

### 5.2.1 Services de couche MAP

Lorsqu'on utilise des connexions MAP, le service MCS fonctionne de la façon décrite à l'Annexe A.

### 5.3 Services assurés par la couche Transport

Pour les connexions de transport, le protocole MCS suppose l'utilisation d'un sous-ensemble du service de transport en mode connexion qui est défini dans la Recommandation X.214. Les informations sont transférées en direction et en provenance d'un fournisseur de service de transport (TS) au moyen des primitives énumérées au Tableau 5-2.

### 5.4 Fonctions de la couche MCS

Le Tableau 5-1 énumère les unités fonctionnelles du service MCS et les unités MCSPDU associées à chaque primitive MCS. Les unités MCSPDU sont définies au paragraphe 7. La relation entre primitives et unités MCSPDU peut être aussi simple qu'un rapport de cause à effet bilatéral. Par exemple, la demande MCS-ATTACH-USER produit les unités **AttachUserRequest**, alors que l'unité **AttachUserConfirm** produit la confirmation MCS-ATTACH-USER. D'autres cas peuvent être plus compliqués. L'exécution de la demande MCS-CONNECT-PROVIDER exige par exemple l'échange d'unités MCSPDU additionnelles, à titre d'effet secondaire de la primitive à quatre phases. L'une quelconque des cinq unités MCSPDU associées peut déclencher une indication MCS-CHANNEL-USER, et l'une quelconque des quatre unités MCSPDU associées peut déclencher une indication MCS-CHANNEL-EXPEL.

#### 5.4.1 Gestion de domaine

La couche de service MCS conserve l'intégrité des connexions MCS qui forment un domaine MCS. Une connexion MCS est orientée, c'est-à-dire qu'une de ses extrémités est hiérarchiquement supérieure à l'autre. Chaque domaine est supervisé par un seul fournisseur de service MCS.

L'établissement d'une connexion MCS consiste à fusionner deux domaines. La couche de service MCS fait en sorte qu'il ne reste qu'un seul fournisseur supérieur. Elle résout tout conflit d'identité unique ou de propriété exclusive pouvant se présenter.

La déconnexion d'une connexion MCS subdivise un domaine en deux parties. Celle qui contient le fournisseur supérieur se maintient. La partie inférieure se supprime automatiquement.

La couche MCS identifie de manière univoque les utilisateurs qui sont rattachés à un domaine donné. Les utilisateurs peuvent se percevoir les uns les autres par l'intermédiaire de leurs interactions au moyen de primitives MCS. La couche MCS notifie à tous les utilisateurs d'un domaine le moment où l'un d'entre eux se détache. La couche MCS récupère toutes les ressources éventuellement détenues par l'utilisateur détaché.

**Tableau 5-1/T.125 – Primitives du service MCS**

Unité fonctionnelle	Primitives	Unités MCSPDU associées
Gestion de domaine	demande MCS-CONNECT-PROVIDER indication MCS-CONNECT-PROVIDER réponse MCS-CONNECT-PROVIDER confirmation MCS-CONNECT-PROVIDER (effets secondaires)	<b>Connect-Initial</b> <b>Connect-Initial</b> <b>Connect-Response</b> <b>Connect-Response</b> <b>Extended-Parameters-Propose</b> <b>Extended-Parameters-Accept</b> <b>Connect-Additional</b> <b>Connect-Result</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>
	demande MCS-DISCONNECT-PROVIDER indication MCS-DISCONNECT-PROVIDER	<b>DisconnectProviderUltimatum</b> <b>DisconnectProviderUltimatum</b> <b>RejectMCSPDUUltimatum</b>
	demande MCS-ATTACH-USER confirmation MCS-ATTACH-USER	<b>AttachUserRequest</b> <b>AttachUserConfirm</b>
	demande MCS-DETACH-USER indication MCS-DETACH-USER	<b>DetachUserRequest</b> <b>DetachUserIndication</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b> <b>MergeTokensConfirm</b> <b>PurgeTokensIndication</b>
Gestion de canal	demande MCS-CHANNEL-JOIN confirmation MCS-CHANNEL-JOIN	<b>ChannelJoinRequest</b> <b>ChannelJoinConfirm</b>
	demande MCS-CHANNEL-LEAVE indication MCS-CHANNEL-LEAVE	<b>ChannelLeaveRequest</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>
	demande MCS-CHANNEL-CONVENE confirmation MCS-CHANNEL-CONVENE	<b>ChannelConveneRequest</b> <b>ChannelConveneConfirm</b>
	demande MCS-CHANNEL-DISBAND indication MCS-CHANNEL-DISBAND	<b>ChannelDisbandRequest</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>
	demande MCS-CHANNEL-ADMIT indication MCS-CHANNEL-ADMIT	<b>ChannelAdmitRequest</b> <b>ChannelAdmitIndication</b>
	demande MCS-CHANNEL-EXPEL indication MCS-CHANNEL-EXPEL	<b>ChannelExpelRequest</b> <b>ChannelExpelIndication</b> <b>ChannelDisbandIndication</b> <b>MergeChannelsConfirm</b> <b>PurgeChannelsIndication</b>

**Tableau 5-1/T.125 – Primitives du service MCS (fin)**

<b>Unité fonctionnelle</b>	<b>Primitives</b>	<b>Unités MCSPDU associées</b>
Transfert de données	demande MCS-SEND-DATA indication MCS-SEND-DATA	<b>SendDataRequest</b> <b>SendDataIndication</b>
	demande MCS-UNIFORM-SEND-DATA indication MCS-UNIFORM-SEND-DATA	<b>UniformSendDataRequest</b> <b>UniformSendDataIndication</b>
Gestion de jeton	demande MCS-TOKEN-GRAB confirmation MCS-TOKEN-GRAB	<b>TokenGrabRequest</b> <b>TokenGrabConfirm</b>
	demande MCS-TOKEN-INHIBIT confirmation MCS-TOKEN-INHIBIT	<b>TokenInhibitRequest</b> <b>TokenInhibitConfirm</b>
	demande MCS-TOKEN-GIVE indication MCS-TOKEN-GIVE réponse MCS-TOKEN-GIVE confirmation MCS-TOKEN-GIVE	<b>TokenGiveRequest</b> <b>TokenGiveIndication</b> <b>TokenGiveResponse</b> <b>TokenGiveConfirm</b>
	demande MCS-TOKEN-PLEASE indication MCS-TOKEN-PLEASE	<b>TokenPleaseRequest</b> <b>TokenPleaseIndication</b>
	demande MCS-TOKEN-RELEASE confirmation MCS-TOKEN-RELEASE	<b>TokenReleaseRequest</b> <b>TokenReleaseConfirm</b>
	demande MCS-TOKEN-TEST confirmation MCS-TOKEN-TEST	<b>TokenTestRequest</b> <b>TokenTestConfirm</b>
	Notification de capacités	

**Tableau 5-2/T.125 – Primitives du service de transport**

<b>Primitives</b>	<b>Utilisation</b>	<b>Paramètres</b>	<b>Utilisation</b>
demande T-CONNECT indication T-CONNECT	X X	Adresse appelée Adresse appelante Option de données exprès Qualité de service Données d'utilisateur TS	X X – X –
réponse T-CONNECT confirmation T-CONNECT	X X	Adresse répondante Option de données exprès Qualité de service Données d'utilisateur TS	– – X –
demande T-DATA indication T-DATA	X X	Données d'utilisateur TS	X
demande T-EXPEDITED-DATA indication T-EXPEDITED-DATA	– –	Données d'utilisateur TS	–
demande T-DISCONNECT	X	Données d'utilisateur TS	–
indication T-DISCONNECT	X	Cause Données d'utilisateur TS	– –

X Indique que le protocole MCS part du principe que cette caractéristique est toujours disponible.

– Indique que le protocole MCS ne fait pas appel à cette caractéristique.

### **5.4.2 Gestion de canal**

La couche MCS enregistre les parties d'un domaine MCS qui contiennent un ou plusieurs utilisateurs ayant adhéré à un canal donné, de manière à pouvoir optimiser le transfert de données vers les destinations souhaitant les recevoir.

La couche MCS traite les identificateurs d'utilisateur comme des canaux pour membre unique, auxquels seuls des utilisateurs désignés sont autorisés à adhérer. Sur demande, la couche MCS peut créer des canaux privés auxquels seuls des utilisateurs habilités auront le droit d'accéder. Elle peut également attribuer des canaux publics, auxquels aucun autre utilisateur n'a encore adhéré.

### **5.4.3 Transfert de données**

La couche du service MCS entretient un flux séquentiel de données à destination des utilisateurs qui ont adhéré à un canal. Un canal devient, en pratique, une liste de distribution multidestinataire avec une gamme comprise entre zéro destination et une diffusion générale.

Par défaut, la couche du service MCS aiguille les données vers chaque récepteur en passant par le plus court chemin offert par les connexions MCS. Sur option, elle achemine des unités de données spécifiées du service MCS en passant par le fournisseur MCS supérieur, garantissant ainsi leur réception uniforme par tous les récepteurs, émetteur éventuellement compris.

La couche du service MCS reconnaît une ou plusieurs priorités de transfert de données et leur applique un traitement préférentiel. Pour les données totalement fiables, la couche MCS autorise des longueurs quelconques d'unités de données du service MCS, par des opérations de segmentation. Pour les données non fiables, la taille des unités de données du service MCS ne peut toutefois pas excéder la taille maximale de l'unité MCSPDU moins l'information supplémentaire d'en-tête de l'unité PDU, déterminée par les paramètres de domaine.

La couche du service MCS règle le débit global des données à l'intérieur d'un domaine. L'incapacité d'un récepteur d'accepter des données à leur débit de présentation provoque une contre-pression qui elle-même provoque le blocage des émetteurs. Un utilisateur peut être détaché contre sa volonté s'il ne parvient pas à conserver un débit de réception minimal.

La couche du service MCS permet les transferts de données à la fois totalement fiables (livraison garantie) et non fiables (livraison non garantie). Pour les transferts de données totalement fiables, la couche du service MCS garantit une réception exempte d'erreurs dans les données émises, du moment que les utilisateurs d'origine et de destination restent rattachés au canal. Pour les transferts de données non fiables, le service MCS assure la réception ordonnée des données transmises, bien que des intervalles puissent exister dans le flux de données. Par ailleurs, à la fois pour les transferts de données totalement fiables et non fiables, les données à priorité de niveau plus élevé doivent toutefois prendre le pas, et une surabondance de telles données peut retarder indéfiniment l'acheminement de données à priorité plus faible.

### **5.4.4 Gestion de jeton**

La couche du service MCS met en œuvre des opérations de jeton au niveau du fournisseur MCS supérieur, assurant ainsi la cohérence et l'exclusion.

### **5.4.5 Notification de capacités**

La couche du service MCS offre un mécanisme d'échange d'information concernant les capacités. Grâce à ce mécanisme, les nœuds peuvent déterminer l'état du domaine par rapport à une certaine capacité. Un nœud peut ainsi prendre une décision quant à l'exercice de cette capacité.

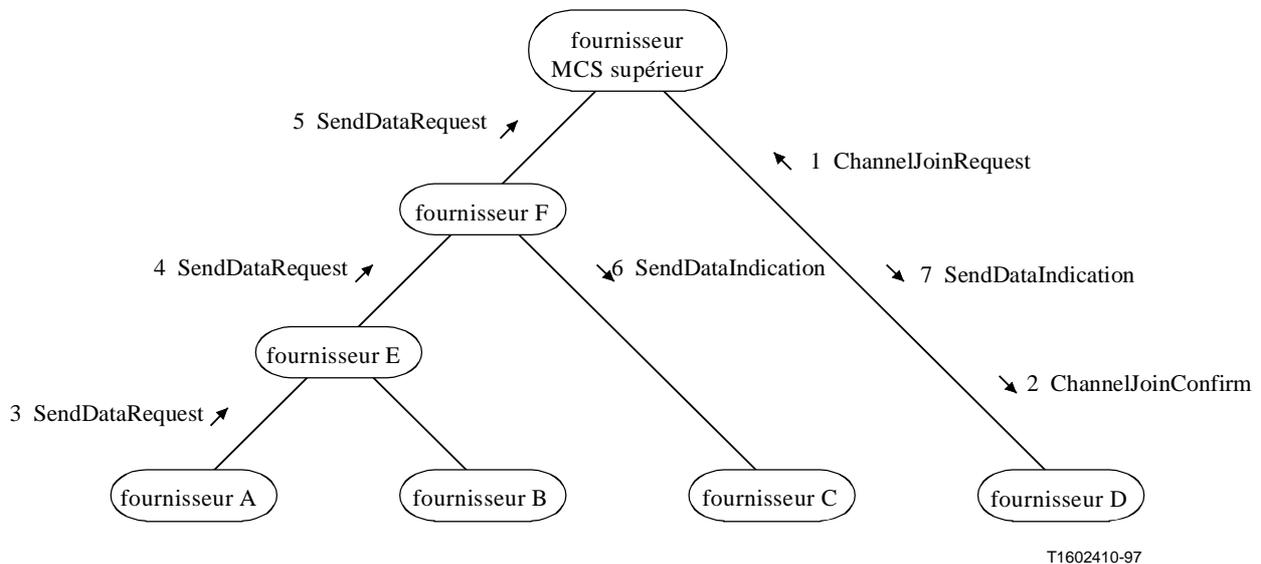
## 5.5 Traitement hiérarchique

Le traitement hiérarchique dans un domaine MCS est illustré par la Figure 5-2.

Les nœuds illustrés sur la figure représentent des fournisseurs de service MCS. Les flèches sont assorties de noms d'unités MCSPDU. Cet exemple porte sur une certaine période de temps après l'établissement du domaine au moyen de connexions entre fournisseurs MCS, lorsque l'utilisation du transfert de données commence à se développer. A l'étape 1, le fournisseur D demande, pour le compte d'un utilisateur, à adhérer à un canal sur lequel des données seront réparties. A l'étape 2, l'acceptation de la demande est confirmée. A l'étape 3, un utilisateur rattaché au fournisseur A envoie des données et l'unité **SendDataRequest** correspondante commence à remonter vers le sommet de la hiérarchie. En admettant que seuls des utilisateurs rattachés aux fournisseurs A, C et D ont adhéré au canal sur lequel les données sont actuellement envoyées, l'unité MCSPDU de demande est répercutée en aval aux étapes 6 et 7 sous la forme **SendDataIndication**. Le fournisseur E, percevant qu'aucun autre subordonné n'a besoin de recevoir ces données, expédie simplement l'unité **SendDataRequest** en amont (étape 4). Le fournisseur F expédie l'unité **SendDataRequest** en amont à l'étape 5 mais la répercute également en aval à l'étape 6, sachant que le fournisseur C a fait part de son intérêt pour ce canal.

Les fournisseurs de service MCS ne sont pas particulièrement concernés par leur hauteur dans la hiérarchie, sauf pour ce qui est de leur rôle dans le maintien d'une limite globale de hauteur du domaine et du fait que, d'une manière générale, ils sont ou non le fournisseur supérieur. Celui-ci ne possède pas de connexion ascendante. Tous les autres fournisseurs en ont exactement une seule.

Un fournisseur de service MCS enregistre des informations sur les canaux et jetons utilisés dans son sous-arbre de domaine MCS.



**Figure 5-2/T.125 – Traitement hiérarchique dans un domaine MCS**

Un fournisseur de service MCS enregistre les canaux auxquels des utilisateurs ont adhéré à l'intérieur du sous-arbre. Pour chacun de ces canaux, il enregistre également l'origine de l'adhésion, c'est-à-dire les rattachements et les connexions MCS descendantes dont ils proviennent. Il enregistre les identificateurs d'utilisateur qui sont attribués dans le sous-arbre et leur origine. Il enregistre les canaux privés qui possèdent soit un gestionnaire ou un utilisateur admis dans le sous-arbre. Il enregistre les identificateurs d'utilisateur correspondants.

Un fournisseur de service MCS enregistre les jetons qui sont saisis ou inhibés par des utilisateurs dans le sous-arbre et il enregistre les identificateurs d'utilisateur correspondants.

Un fournisseur de service MCS examine les demandes issues de son sous-arbre pour vérifier que l'identificateur de l'utilisateur initiateur est légitimement attribué au rattachement ou à la connexion MCS descendante d'origine. Cela crée des anneaux de protection autour du fournisseur MCS supérieur et limite le nombre d'interruptions qu'un participant malveillant peut provoquer dans un domaine par ailleurs coopératif.

En termes généraux, le fonctionnement de la couche de service MCS peut, à plusieurs exceptions près, être décrit comme suit:

- a) une primitive de demande de service MCS, invoquée à un point de rattachement MCS, produit une unité MCSPDU au niveau du fournisseur MCS correspondant et la fait remonter vers le fournisseur MCS supérieur, où elle sera suivie d'effet si toutes les informations sur le domaine MCS sont présentes;
- b) une unité MCSPDU de confirmation peut être produite au niveau du fournisseur MCS supérieur afin de retourner des résultats vers le rattachement demandeur. Les fournisseurs MCS qui la transmettent mettent à jour leurs enregistrements en fonction des effets de l'opération sur leur sous-arbre. Une confirmation est acheminée vers l'utilisateur initiateur identifié après consultation des enregistrements locaux à chaque bond descendant successif;
- c) une unité MCSPDU d'indication au lieu de confirmation peut être produite pour informer d'autres rattachements de l'action entreprise. Ces indications peuvent être copiées et descendre vers plusieurs connexions aboutissant à des utilisateurs touchés. Les fournisseurs de service MCS peuvent également mettre à jour leurs enregistrements en fonction de l'effet de l'opération, dans le cadre du traitement d'une indication.

La description ci-dessus donne un aperçu général destiné à créer un cadre théorique. Les paragraphes suivants spécifient en plus amples détails les informations enregistrées chez un fournisseur de service MCS et la façon dont des unités MCSPDU spécifiques sont traitées.

Les exceptions sont en particulier les suivantes : certaines demandes, comme **ChannelJoinRequest** et **ChannelLeaveRequest**, peuvent ne pas remonter jusqu'au fournisseur supérieur et certaines indications, comme **SendDataIndication**, peuvent être produites à un niveau inférieur au fournisseur supérieur. Une seule unité MCSPDU, **TokenGiveResponse**, appartient à la catégorie des réponses. Et certaines unités MCSPDU peuvent être produites par un fournisseur de service MCS pour la poursuite d'un traitement, contrairement à des unités telles que **ChannelLeaveRequest** faisant suite à **DetachUserIndication**.

## 5.6 Paramètres de domaine

Les fournisseurs de service MCS qui couvrent un seul domaine MCS attribuent les ressources et exécutent les procédures conformément aux paramètres ci-après. Les valeurs de ces paramètres sont identiques dans un domaine donné, exception faite de la version du protocole.

- a) nombre maximal de canaux MCS pouvant être utilisés simultanément. Ce paramètre comprend les canaux auxquels un utilisateur quelconque adhère, les identificateurs d'utilisateur qui ont été attribués et les canaux privés qui ont été créés;
- b) nombre maximal d'identificateurs d'utilisateur MCS pouvant être attribués simultanément. Ce paramètre indique une limite inférieure à l'intérieur de la contrainte représentée par le paramètre précédent;
- c) nombre maximal d'identificateurs de jeton pouvant être saisis ou inhibés simultanément;

- d) nombre de priorités de transfert de données mises en œuvre. Ce nombre est égal à celui des connexions de transport (TC) d'une connexion MCS. Un utilisateur du service MCS peut toujours émettre et recevoir des données avec des priorités extérieures à la limite. Mais toutes ces priorités peuvent être traitées de la même manière que la plus basse priorité mise en œuvre;
- e) débit renforcé. Bien que la commande de flux globale impose une limite au transfert de données à l'intérieur d'un domaine en fonction de la vitesse du récepteur le plus lent, les récepteurs ne doivent jamais être autorisés à adopter une vitesse arbitrairement basse. Sinon, un des participants à une conférence peut bloquer tous les autres. Ce paramètre invite les fournisseurs de service MCS à appliquer un débit de réception minimal à chaque rattachement MCS et sur chaque connexion MCS descendante. Les transgresseurs courent le risque d'être détachés ou déconnectés, selon le cas, contre leur volonté;
- f) hauteur maximale. Ce paramètre limite la hauteur de tous les fournisseurs MCS, en particulier le fournisseur supérieur;
- g) longueur maximale des unités MCSPDU de domaine. La commande de flux globale est fondée sur la mise en mémoire tampon d'unités MCSPDU de domaine (mais non des unités MCSPDU de type Connect), chez un fournisseur de service MCS. Par souci de simplicité, on suppose que les mémoires tampons ont une capacité fixe. Un fournisseur de service MCS ne doit pas émettre d'unités MCSPDU plus longues. Cela limite la quantité d'informations qui peuvent être assemblées pour former une même unité MCSPDU de commande et donne une indication du point où un flux continu de données d'utilisateur devrait être segmenté pour former des unités MCSPDU de données;
- h) version de protocole. Ce paramètre peut prendre l'une des trois valeurs qui définissent des codages différents pour les unités MCSPDU de domaine. Dans la version trois du protocole de service MCS, cette valeur est établie connexion par connexion. Des fournisseurs de service MCS de version deux peuvent ainsi exister en aval dans une conférence permettant la version trois.

NOTE – Une certaine instance d'un fournisseur de service MCS peut fonctionner avec des contraintes de ressources locales qui sont également paramétrées. Il peut s'agir de la capacité de mémoire disponible pour mettre en tampon les unités MCSPDU en attente de transport, du nombre maximal de rattachements MCS et du nombre maximal de connexions MCS vers d'autres fournisseurs. De tels paramètres relèvent de décisions locales et ne sont pas communiqués dans un domaine MCS.

## 5.7 Paramètres étendus

Les fournisseurs de service MCS qui sont membres du sommet V3 d'un domaine MCS unique attribueront des ressources et exécuteront des procédures conformément aux paramètres ci-dessous.

- a) les données non fiables doivent pouvoir s'utiliser. Si des données non fiables sont utilisées dans le sommet V3, cette valeur booléenne sera au niveau vrai. La valeur de ce paramètre est identique dans l'ensemble du sommet V3 d'un domaine;
- b) identificateur de référence de domaine. Il ne s'agit pas d'une valeur négociée, mais elle sera générée par un fournisseur de service MCS pour identifier de façon unique un domaine spécifique. Bien que la même valeur puisse s'utiliser afin d'identifier différents domaines pour des fournisseurs différents, elle doit représenter de façon unique un domaine spécifique pour le fournisseur qui a généré la valeur.

## 6 Utilisation du service de transport

### 6.1 Modèle du service de transport

La présente description paraphrase les parties applicables de la Recommandation X.214, étant supposé qu'il n'est jamais fait appel à des données exprès.

Le service de transport offre les possibilités suivantes à un utilisateur TS:

- a) le moyen d'établir une connexion de transport avec un autre utilisateur du service de transport afin d'échanger des unités TSDU. Plusieurs connexions de transport peuvent exister entre un même couple d'utilisateurs du service de transport;
- b) la possibilité de demander, de négocier et de faire agréer par le fournisseur du service de transport, pour chaque connexion de transport au moment de son établissement, une certaine qualité de service spécifiée par des paramètres représentant des caractéristiques comme le débit, le temps de transit, le taux d'erreur résiduel et la priorité;
- c) le moyen de transférer des unités TSDU sur une connexion de transport. Les unités TSDU, qui comprennent un nombre entier d'octets, sont transférées en transparence, en ce sens que les limites et le contenu des unités TSDU sont préservés sans changement par le fournisseur TS;
- d) le moyen, pour l'utilisateur destinataire du service de transport, de commander la vitesse à laquelle l'utilisateur expéditeur peut transmettre les données;
- e) la libération inconditionnelle et donc éventuellement destructive d'une connexion de transport.

Le fonctionnement d'une connexion de transport est modélisé sous forme abstraite par une paire de files d'attente reliant deux points TSAP. Chaque file correspond à un sens de transmission du flux d'informations. Chaque connexion de transport est modélisée par une paire distincte de files d'attente.

Le modèle des files d'attente sert à exprimer la fonction de commande de flux. Une file d'attente a une capacité limitée, mais celle-ci n'est pas forcément fixe ni déterminable. L'introduction et l'extraction d'objets d'unités de type Connect, d'unités TSDU et d'unités de type disconnect résultent des interactions au niveau des deux points TSAP. La capacité d'un utilisateur du service de transport (TS) à ajouter des objets à une file d'attente dépend du comportement de l'utilisateur TS qui retire des objets de cette file ainsi que de l'état de celle-ci. Les seuls objets qui peuvent être placés dans une file d'attente par le fournisseur du service TS sont des objets de type disconnect. Des objets sont ajoutés à une file d'attente sous le contrôle du fournisseur TS. Les objets sont normalement retirés de la file d'attente sous réserve de commande par l'utilisateur TS récepteur. Les objets sont normalement retirés dans l'ordre où ils ont été introduits. La seule exception à leur retrait normal est le fait qu'un objet peut être supprimé par le fournisseur TS si, et seulement si, l'objet suivant est du type disconnect.

Un mécanisme d'identification d'extrémité de connexion de transport doit être prévu au niveau local si l'utilisateur TS et le fournisseur TS ont besoin de distinguer entre elles plusieurs connexions de transport au niveau d'un même point d'accès au service de transport (TSAP). Toutes les primitives doivent alors utiliser ce mécanisme d'identification pour identifier la connexion de transport à laquelle elles s'appliquent. Cette identification implicite n'est pas représentée sous la forme d'un paramètre des primitives du service TS et ne doit pas être confondue avec les paramètres d'adresse des primitives de type T-CONNECT.

## 6.2 Utilisation de connexions multiples

Une connexion MCS se compose d'une ou de plusieurs connexions de transport (TC) entre la même paire de fournisseurs de service MCS. La première connexion TC établie est dite TC initiale; les connexions TC établies ultérieurement sont dites TC additionnelles. Toutes les connexions TC qui appartiennent à une même connexion MCS sont établies par le même fournisseur de service MCS, en réaction à une demande MCS-CONNECT-PROVIDER. Cette demande contiendra les paramètres d'adresse, c'est-à-dire les adresses de point TSAP appelant et appelé. Ces paramètres seront utilisés sans modification dans les demandes T-CONNECT qui en résulteront.

Le nombre de connexions TC par connexion MCS est uniforme dans un même domaine MCS. Ce paramètre de domaine est égal au nombre de niveaux de priorité de transfert de données mis en œuvre. Des connexions TC distinctes sont requises parce que chacune véhicule une commande de flux. Des blocages affectant des données de priorité inférieure ne doivent normalement pas provoquer de contre-pression sur des données de priorité supérieure. Pour pouvoir être mises en œuvre totalement, les données de priorité inférieure et de priorité supérieure doivent être acheminées sur des connexions TC différentes.

La qualité de service demandée pour une connexion TC peut varier selon la priorité de transfert de données pour laquelle cette connexion a été établie. Ces objectifs de qualité de service peuvent ne pas être uniformes à l'intérieur d'un domaine MCS.

Les aspects de qualité de service à prendre en compte sont le débit maximal ou moyen et le temps de transit. Des données à priorité élevée peuvent favoriser un temps de transit court pour une réponse en temps réel mais ne peuvent pas exiger un débit utile élevé. Des données à priorité basse, par ailleurs, peuvent favoriser un débit utile élevé pour des transferts de masse mais peuvent ne pas exiger un temps de transit court.

La priorité des connexions de transport constitue un autre aspect de la qualité de service, bien que celui-ci ne recouvre pas exactement le concept de priorité de transfert des données du service MCS. Il s'agit plutôt de l'ordre relatif dans lequel les connexions TC verront leur qualité de service se dégrader, le cas échéant. Une priorité TC élevée peut être requise en même temps que d'autres caractéristiques comme un faible temps de transit, afin de garantir que la priorité MCS des données recevra le traitement préférentiel qu'elle mérite.

Une unité MCSPDU de type connect n'apparaît qu'en tant que première unité TSDU acheminée dans un sens ou dans l'autre d'une connexion TC. Les unités de type **Connect-Initial** et **Connect-Response** traversent la connexion de transport initiale d'une connexion MCS. Les unités de type **Connect-Additional** et **Connect-Result** traversent des connexions de transport additionnelles, si elles existent.

Un fournisseur de service MCS appelant, qui émet des demandes T-CONNECT, détermine par ses propres actions les connexions TC qui feront partie de la même connexion MCS ainsi que les priorités de transfert de données que ces connexions achemineront.

Un fournisseur de service MCS appelé, qui reçoit des indications T-CONNECT, doit en général accepter une connexion TC et lire sa première unité TSDU avant de prendre connaissance de sa signification. Une unité de type **Connect-Initial** annonce une connexion TC entrante qui formera le début d'une nouvelle connexion MCS. Une unité de type **Connect-Additional** annonce une connexion TC qui fera partie d'une connexion MCS en cours.

L'unité **Connect-Additional** contient une valeur qui est attribuée par le fournisseur MCS appelé et qui est transmise au fournisseur MCS appelant dans une unité **Connect-Response** acheminée sur la connexion TC initiale qui désignera la connexion TC additionnelle comme appartenant à la même connexion MCS. Une unité **Connect-Additional** indique explicitement la priorité des données qui est représentée par la connexion de transport.

Les unités MCSPDU de type connect sont échangées immédiatement après l'établissement de la connexion TC. Une fois qu'une connexion MCS a été confirmée, elle est intégrée dans le domaine MCS hiérarchique. C'est alors que la connexion MCS achemine les unités MCSPDU du domaine.

A l'exception des unités MCSPDU de données, les unités MCSPDU de domaine traversent la connexion TC initiale d'une connexion MCS. Les unités MCSPDU de données traversent la connexion TC qui correspond à leur niveau de priorité de données. Si la priorité spécifiée est supérieure au nombre mis en œuvre dans un domaine MCS, l'unité MCSPDU de données traversera la connexion TC ayant la plus basse priorité mise en œuvre.

Si une seule priorité est mise en œuvre dans un domaine MCS, chacune des connexions MCS de ce domaine comportera une seule connexion de transport, il ne sera pas fait appel aux unités **Connect-Additional** ou **Connect-Result** et toutes les unités MCSPDU seront acheminées en séquence entre les fournisseurs.

### 6.3 Libération d'une connexion de transport

Un fournisseur de service de transport (TS) augmente la fiabilité de ses connexions de bout en bout en exécutant un protocole suffisant pour compenser toute faiblesse éventuelle se manifestant dans les services des couches Réseau sous-jacentes. Un fournisseur de service MCS ne duplique pas cette fonction. En cas de panne de transport, il ne tente pas d'effectuer un autre processus de rétablissement automatique.

Les erreurs irrémédiables sont annoncées par une indication T-DISCONNECT. Si l'une quelconque des connexions de transport faisant partie d'une connexion MCS est coupée, les autres sont immédiatement coupées aussi. Une indication MCS-DISCONNECT-PROVIDER est alors émise avec la cause engendrée par le fournisseur, sauf si cette déconnexion a été demandée par l'utilisateur.

Par ailleurs, une demande MCS-DISCONNECT-PROVIDER doit normalement apparaître sous forme d'indication à l'autre extrémité avec la cause "demandé par l'utilisateur". Malgré les indications données dans la Recommandation X.214, le protocole de transport de la classe la plus simple n'autorise pas le passage de données d'utilisateur dans la demande T-DISCONNECT. Le code de cause de déconnexion est transféré dans une unité MCSPDU spéciale, qui obligera le fournisseur MCS qui la recevra à couper la connexion MCS ayant acheminé ce code.

## 7 Structure des unités MCSPDU de la version 2

La structure des unités MCSPDU de la version 2 est spécifiée au moyen de la notation ASN.1 de la Recommandation X.208. Les paragraphes 9 et 10 décriront plus en détail l'utilisation et la signification de ces unités MCSPDU.

**MCS-PROTOCOL DEFINITIONS ::=**

**BEGIN**

-- Partie 1: types MCS fondamentaux

**ChannelId ::= INTEGER (0..65535)** -- domaine de 16 bits

**StaticChannelId ::= ChannelId (1..1000)** -- pour les canaux connus en permanence

**DynamicChannelId ::= ChannelId (1001..65535)** -- canaux créés et supprimés

**UserId ::= DynamicChannelId**  
-- canal créé par utilisateur de rattachement  
-- canal supprimé par utilisateur de détachement

```

PrivateChannelId ::= DynamicChannelId           -- canal créé par constitution de canal
                                                    -- canal supprimé par dissolution de canal

AssignedChannelId ::= DynamicChannelId         -- canal créé par adhésion à un canal zéro
                                                    -- canal supprimé par sortie du dernier canal

TokenId ::= INTEGER (1..65535)                -- tous ces entiers sont connus en permanence

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),
    -- limite sur les identificateurs de canal utilisés,
    -- canal statique + ident. utilisateur + privé + attribué
    maxUserIds           INTEGER (0..MAX),
    -- sous-limite sur les seuls canaux d'identification d'utilisateur
    maxTokenIds          INTEGER (0..MAX),
    -- limite sur les identificateurs de jeton utilisés
    -- jeton saisi + inhibé + donnant + incessible + donné
    numPriorities        INTEGER (0..MAX),
    -- nombre de connexions TC dans une connexion MCS
    minThroughput        INTEGER (0..MAX),
    -- nombre réel d'octets par seconde
    maxHeight            INTEGER (0..MAX),
    -- limite quant à la hauteur d'un fournisseur
    maxMCSPDUsize        INTEGER (0..MAX),
    -- limite quant au nombre d'octets des unités MCSPDU du domaine
    protocolVersion      INTEGER (0..MAX)
}

-- Partie 2: fournisseur de connexions

Connect-Initial ::= [APPLICATION 101] IMPLICIT SEQUENCE
{
    callingDomainSelector OCTET STRING,
    calledDomainSelector OCTET STRING,

```

```

    upwardFlag                BOOLEAN,
                               -- prend la valeur TRUE si le fournisseur appelé est plus élevé
    targetParameters          DomainParameters,
    minimumParameters         DomainParameters,
    maximumParameters         DomainParameters,
    userData                  OCTET STRING
}

```

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

```

{
    result                    Result,
    calledConnectId          INTEGER (0..MAX),
                               -- cet entier est attribué par le fournisseur appelé afin d'identifier
                               -- les connexions de transport additionnelles appartenant à la
                               -- même connexion MCS
    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
}

```

-- Partie 3: fusion de domaines

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

```

{
    heightLimit              INTEGER (0..MAX)
                               -- restriction relative au récepteur d'unités MCSPDU
}

```

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

```

{
    subHeight                INTEGER (0..MAX),
                               -- hauteur dans le domaine de l'émetteur d'unités MCSPDU
    subInterval              INTEGER (0..MAX)
                               -- son intervalle d'application de débit, en millisecondes
}

```

**ChannelAttributes ::= CHOICE**

```

{
    static                   [0] IMPLICIT SEQUENCE
    {
        channelId            StaticChannelId
                               -- la valeur d'adhésion est implicitement TRUE
    },
    userId                   [1] IMPLICIT SEQUENCE
    {
        joined                BOOLEAN,
                               -- valeur TRUE si l'utilisateur a adhéré à son canal
                               -- d'identification d'utilisateur
        userId                UserId
    },
}

```

```

private
{
    joined
        BOOLEAN,
        -- valeur TRUE si l'adhésion à l'identificateur de canal
        -- est effectuée à un niveau inférieur
    channelId
        PrivateChannelId,
    manager
        UserId,
    admitted
        SET OF UserId
        -- cet ensemble peut couvrir plusieurs unités MergeChannelsRequest
},

assigned
{
    channelId
        AssignedChannelId
        -- la valeur d'adhésion est implicitement à 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 de canaux d'identification d'utilisateur
    purgeChannelIds
        SET OF ChannelId
        -- purge d'autres canaux
}

TokenAttributes ::= CHOICE
{
    grabbed
        [0] IMPLICIT SEQUENCE
        {
            tokenId
                TokenId,
            grabber
                UserId
        },
    inhibited
        [1] IMPLICIT SEQUENCE
        {
            tokenId
                TokenId,
            inhibitors
                SET OF UserId
                -- cet ensemble peut couvrir plusieurs demandes MergeTokensRequest
        },
    giving
        [2] IMPLICIT SEQUENCE
        {
            tokenId
                TokenId,
            grabber
                UserId,
            recipient
                UserId
        },
}

```

```

ungivable                                [3] IMPLICIT SEQUENCE
{
    tokenId                               TokenId,
    grabber                               UserId
    -- le destinataire s'est détaché entre-temps
},

given                                    [4] IMPLICIT SEQUENCE
{
    tokenId                               TokenId,
    recipient                             UserId
    -- le saisisseur a été libéré ou détaché
}
}

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
}
-- Partie 4: fournisseur du service de déconnexion

DisconnectProviderUltimatum ::= [APPLICATION 8] IMPLICIT SEQUENCE
{
    reason                                 Reason
}

RejectMCSPDUUltimatum ::= [APPLICATION 9] IMPLICIT SEQUENCE
{
    diagnostic                             Diagnostic,
    initialOctets                         OCTET STRING
}
-- Partie 5: rattachement/détachement d'utilisateur

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
}

```

-- *Partie 6: gestion de canal*

```

ChannelJoinRequest ::= [APPLICATION 14] IMPLICIT SEQUENCE
{
    initiator            UserId,
    channelId           ChannelId
                        -- peut prendre la valeur zéro
}

```

```

ChannelJoinConfirm ::= [APPLICATION 15] IMPLICIT SEQUENCE
{
    result              Result,
    initiator          UserId,
    requested         ChannelId,
                        -- peut prendre la valeur zéro
    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
}

```

-- Partie 7: transfert de données

```

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
}

```

-- Partie 8: gestion de jeton

**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
}
```

-- *Partie 9: codes d'état*

**Reason ::= ENUMERATED** -- *codes contenus dans des unités 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** -- *codes contenus dans des unités de type Connect, réponse, confirmation*

```
{
    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** -- *codes contenus dans des unités 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)
}

```

-- *Partie 10: répertoire des unités MCSPDU*

**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,
}

```

<b>channelAdmitRequest</b>	<b>ChannelAdmitRequest,</b>
<b>channelAdmitIndication</b>	<b>ChannelAdmitIndication,</b>
<b>channelExpelRequest</b>	<b>ChannelExpelRequest,</b>
<b>channelExpelIndication</b>	<b>ChannelExpelIndication,</b>
<b>sendDataRequest</b>	<b>SendDataRequest,</b>
<b>sendDataIndication</b>	<b>SendDataIndication,</b>
<b>uniformSendDataRequest</b>	<b>UniformSendDataRequest,</b>
<b>uniformSendDataIndication</b>	<b>UniformSendDataIndication,</b>
<b>tokenGrabRequest</b>	<b>TokenGrabRequest,</b>
<b>tokenGrabConfirm</b>	<b>TokenGrabConfirm,</b>
<b>tokenInhibitRequest</b>	<b>TokenInhibitRequest,</b>
<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>

}

END

## 8 Structure des unités MCSPDU de la version 3

La structure des unités MCSPDU de la version 3 est spécifiée au moyen de la notation ASN.1 de la Recommandation X.208. Les paragraphes 9 et 10 décriront plus en détail l'utilisation et la signification de ces unités MCSPDU.

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

**BEGIN**

*-- Partie 1: types fondamentaux de service MCS*

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

*-- Les quatre premiers octets doivent comprendre  
-- le code de pays et le code de fabricant, assignés  
-- selon les spécifications de l'Annexe A/H.221  
-- pour NS-cap et NS-comm*

**Key ::= CHOICE**

*-- Identificateur d'objet standard ou non standard*

{		
	<b>object</b>	<b>OBJECT IDENTIFIER,</b>
	<b>h221NonStandard</b>	<b>H221NonStandardIdentifier</b>
}		

**NonStandardParameter ::= SEQUENCE**

{		
	<b>key</b>	<b>Key,</b>
	<b>data</b>	<b>OCTET STRING</b>
}		

**ChannelId ::= INTEGER (0..65535)**

*-- domaine de 16 bits*

**StaticChannelId ::= ChannelId (1..1000)**

*-- pour les canaux connus en permanence*

```

DynamicChannelId ::= ChannelId (1001..65535)           -- canaux créés et supprimés

UserId ::= DynamicChannelId                          -- canal créé par utilisateur de rattachement
PrivateChannelId ::= DynamicChannelId                -- canal supprimé par utilisateur de détachement
                                                    -- canal créé par constitution de canal
                                                    -- canal supprimé par dissolution de canal

AssignedChannelId ::= DynamicChannelId               -- canal créé par adhésion à un canal zéro
                                                    -- canal supprimé par sortie du dernier canal

TokenId ::= INTEGER (1..65535)                      -- tous ces entiers sont connus en permanence

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))

-- Partie 2: paramètre étendu

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,
    ...
}

```

-- Partie 3: fusion de domaines

**PlumbDomainIndication ::= SEQUENCE**

```

{
    heightLimit              INTEGER (0..MAX),
                            -- restriction relative au récepteur d'unités MCSPDU
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**ErectDomainRequest ::= SEQUENCE**

```

{
    subHeight                INTEGER (0..MAX),
                            -- hauteur dans le domaine de l'émetteur d'unités MCSPDU
    subInterval              INTEGER (0..MAX),
                            -- son intervalle d'application de débit, en millisecondes
    nonStandard              SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

```

**ChannelAttributes ::= CHOICE**

```

{
    static                   SEQUENCE
    {
        channelId            StaticChannelId,
                            -- la valeur d'adhésion est implicitement TRUE
        nonStandard          SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },

    userId                   SEQUENCE
    {
        joined               BOOLEAN,
                            -- valeur TRUE si l'utilisateur a adhéré à son canal
                            -- d'identification d'utilisateur
        userId               UserId,
        nonStandard          SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },

    private                  SEQUENCE
    {
        joined               BOOLEAN,
                            -- valeur TRUE si l'adhésion à l'identificateur de canal
                            -- est effectuée à un niveau inférieur
        channelId            PrivateChannelId,
        manager              UserId,
        admitted             SET OF UserId,
                            -- cet ensemble peut couvrir plusieurs unités MergeChannelsRequest
        nonStandard          SEQUENCE OF NonStandardParameter OPTIONAL,
        ...
    },

    assigned                 SEQUENCE
    {
        channelId            AssignedChannelId,

```

```

        nonStandard          -- la valeur d'adhésion est implicitement à TRUE
                               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 de canaux d'identification d'utilisateur
    purgeChannelIds          SET OF ChannelId,
                               -- purge d'autres canaux
    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,
                               -- cet ensemble peut couvrir plusieurs unités 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,
                                           -- le destinataire s'est détaché entre-temps
    nonStandard                           SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
},

given                                    SEQUENCE
{
    tokenId                               TokenId,
    recipient                             UserId,
                                           -- le saisisseur a été libéré ou détaché
    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,
    ...
}

-- Partie 4: fournisseur du service de déconnexion

DisconnectProviderUltimatum ::= SEQUENCE
{
    reason                                 Reason,
    nonStandard                           SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

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

```

-- Partie 5: rattachement/détachement d'utilisateur

```
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,  
    ...  
}
```

-- Partie 6: gestion de canal

```
ChannelJoinRequest ::= SEQUENCE  
{  
    initiator          UserId,  
    channelId         ChannelId, -- peut prendre la valeur zéro  
    nonStandard       SEQUENCE OF NonStandardParameter OPTIONAL,  
    ...  
}
```

```
ChannelJoinConfirm ::= SEQUENCE  
{  
    result Result,  
    initiator          UserId,  
    requested         ChannelId, -- peut prendre la valeur zéro  
    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,
        ...
    }

```

-- Partie 7: transfert de données

**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,
    ...
}

```

-- Partie 8: gestion de jeton

**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,
    ...
}
-- Partie 9: notification de capacités

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,
    ...
}

```

-- Partie 10: codes d'état

```

Reason ::= CHOICE      -- codes contenus dans des unités 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      -- codes contenus dans des unités de type 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 -- codes contenus dans des unités 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,
    ...

```

```

}

```

-- Partie 11: répertoire des unités MCSPDU

**NonStandardPDU ::= SEQUENCE**

```
{  
    data                NonStandardParameter,  
    ...  
}
```

**ExtendedParameterMCSPDU ::= CHOICE**

```
{  
    extendedParameterPropose    ExtendedParameterPropose,  
    extendedParameterAcceptExtendedParameterAccept,  
    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,  
    tokenInhibitConfirm           TokenInhibitConfirm,  
    tokenGiveRequest              TokenGiveRequest,  
    tokenGiveIndication            TokenGiveIndication,  
    tokenGiveResponse              TokenGiveResponse,  
    tokenGiveConfirm              TokenGiveConfirm,  
    tokenPleaseRequest              TokenPleaseRequest,  
    tokenPleaseIndication            TokenPleaseIndication,  
    tokenReleaseRequest            TokenReleaseRequest,  
    tokenReleaseConfirm            TokenReleaseConfirm,  
    tokenTestRequest              TokenTestRequest,  
}
```

**tokenTestConfirm**  
**nonStandard**

**TokenTestConfirm,**  
**NonStandardPDU,**

...

}

**END**

## **9 Codage des unités MCSPDU**

Chaque unité MCSPDU est transportée sous forme d'unité TSDU sur une connexion de transport (TC) appartenant à une connexion du service MCS. Les unités MCSPDU de type connect sont de longueur illimitée. Les unités MCSPDU de domaine sont limitées en longueur par un paramètre du domaine MCS.

Un codage normalisé des valeurs de données ASN.1 est utilisé pour transférer des unités MCSPDU entre des fournisseurs de service MCS homologues. Les règles de codage sont sélectionnées dans le cadre d'une négociation de version de protocole. Cette négociation implique l'échange d'une unité MCSPDU de type **Connect-Initial** et d'une unité MCSPDU de type **Connect-Response** sur la connexion de transport initiale. Trois versions du présent protocole sont définies:

- *version 1* – Toutes les unités MCSPDU sont décrites au paragraphe 7 et utilisent les règles de codage de base de la Recommandation X.690;
- *version 2* – Toutes les unités MCSPDU sont décrites au paragraphe 7. Cette version utilise les règles de codage de base pour les unités MCSPDU de type connect et les règles de codage en métalangage condensé (PER, *packed encoding rules*) de la Recommandation X.691 pour toutes les unités MCSPDU de domaine suivantes. Plus précisément, la variante ALIGNED de l'élément BASIC-PER doit être appliquée aux unités de type ASN.1 **DomainMCSPDU**. La chaîne binaire ainsi produite doit être acheminée sous forme d'un nombre entier d'octets. Le bit initial de cette chaîne doit coïncider avec le bit de poids le plus fort du premier octet;
- *version 3* – Toutes les unités MCSPDU de type connect sont décrites au paragraphe 7 et utilisent les règles de codage de base pour assurer la compatibilité amont avec la version 2. Les unités MCSPDU à paramètres étendus et de domaine sont décrites au paragraphe 8 et utilisent les règles de codage en métalangage condensé spécifiées pour la version 2 ci-dessus. Les nœuds validés pour la version 3 doivent pouvoir traduire un protocole de version 3 en protocole de version 2 pour fins de transmission par une connexion vers un nœud qui a spécifié un protocole de version 2 dans les paramètres de domaine. Cette traduction est décrite au 13.11.

NOTE 1 – Les règles de codage en métalangage condensé produisent des en-têtes plus compacts pour les unités MCSPDU.

NOTE 2 – Les règles de codage de base (BER) et condensées (PER) sont auto-délimitatrices, en ce sens qu'elles contiennent assez de renseignements pour localiser la fin de chaque unité MCSPDU codée. On pourrait en tirer la conclusion que cela rend inutile l'utilisation d'unités TSDU et que ce protocole pourrait être mis en œuvre au moyen de services de transport non normalisés, acheminant des trains d'octets sans tenir compte des frontières entre unités TSDU. Une telle conception prête cependant plus le flanc à des erreurs de mise en œuvre car si jamais les frontières entre unités MCSPDU étaient perdues, le rétablissement serait très difficile.

## 10 Acheminement des unités MCSPDU

### 10.1 Unités MCSPDU de type connect et à paramètres étendus

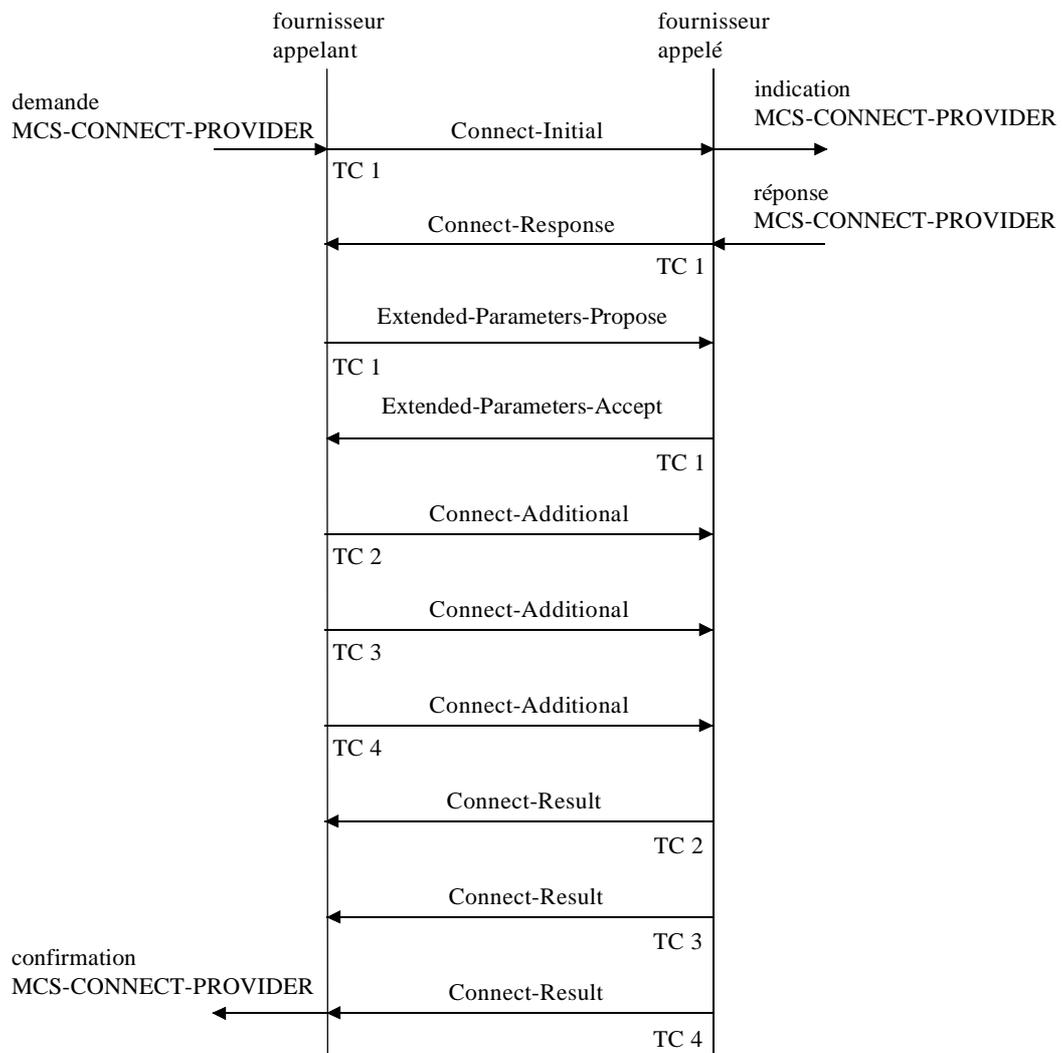
Les Figures 10-1 et 10-2 spécifient l'échange d'unités MCSPDU de type connect et à paramètres étendus (le nombre et l'ordre relatif des connexions de transport supplémentaires peut varier).

Dès qu'il reçoit une unité de type **Connect-Response**, le fournisseur MCS appelant déterminera si la nouvelle connexion se situe entre deux membres du sommet V3 en examinant le numéro de version négocié. Si tel est le cas, il amorcera l'échange des unités MCSPDU à **paramètres étendus**. Si l'un ou l'autre des nœuds ne fait pas partie du sommet V3, le fournisseur appelant n'émettra pas d'unité **Extended-Parameters-Propose**, mais passera aux unités MCSPDU de type **Connect-Additional**.

Après avoir reçu une unité de type **Connect-Response**, le fournisseur MCS appelant prend connaissance de la valeur négociée du nombre de priorités de transfert de données mises en œuvre dans le domaine. A titre d'exemple, on a représenté des unités MCSPDU de type connect sur des connexions de transport additionnelles suivant strictement la séquence 2, 3, 4 au niveau du fournisseur MCS appelé. En réalité, on ne peut pas établir de connexions de transport dans le même ordre que celui de leur demande: une unité de type **Connect-Additional** peut arriver dans un ordre différent de celui où elle a été envoyée, ce qui entraînera le retour d'une unité **Connect-Result** dans un ordre également différent. Ou encore, cette dernière unité peut, même si elle a été envoyée en séquence, être réordonnée lors de son transit. Un fournisseur MCS appelant n'a pas besoin d'attendre que toutes les connexions de transport additionnelles soient établies avant d'envoyer la première unité **Connect-Additional**. Le fournisseur de service MCS appelé n'a pas besoin d'attendre l'arrivée d'une série complète d'unités MCSPDU **Connect-Additional** pour retourner la première unité **Connect-Result**. La réception d'une série complète de résultats favorables au niveau du fournisseur MCS appelant, dans un ordre ou dans un autre, engendre une confirmation MCS-CONNECT-PROVIDER contenant une valeur favorable.

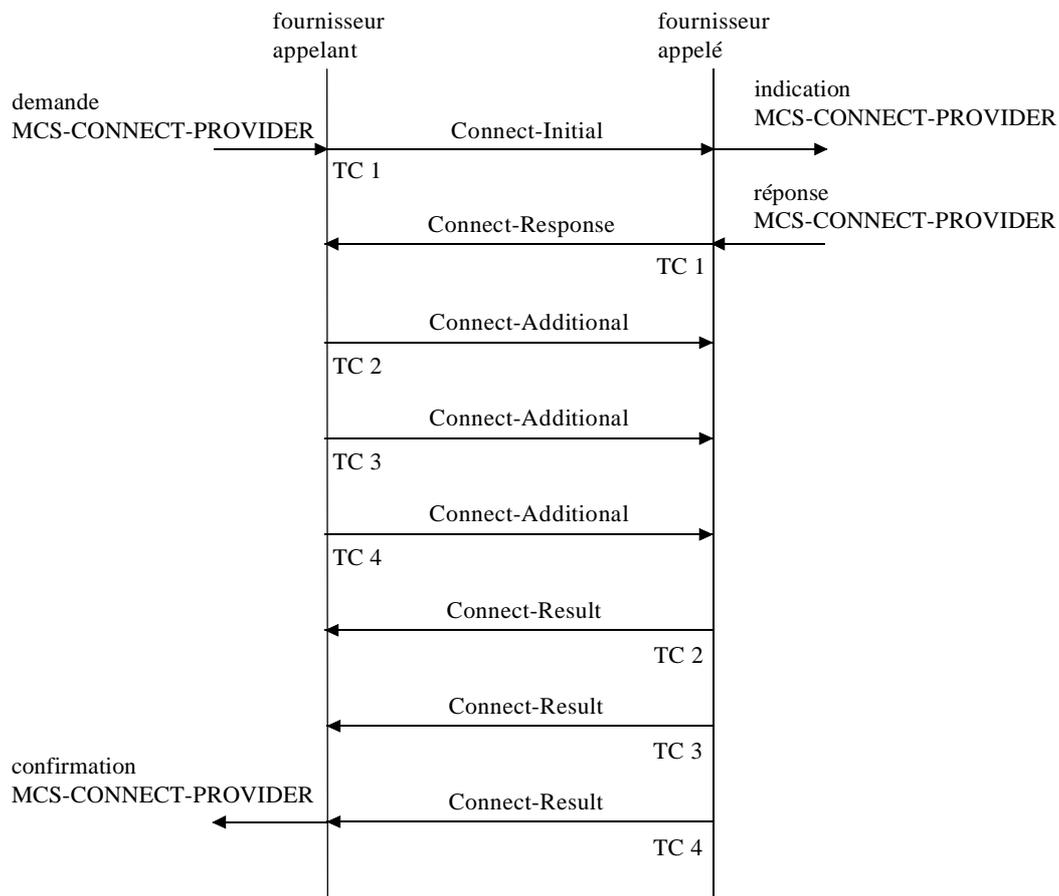
Une unité de type **Connect-Response** ou **Connect-Result** contenant une valeur défavorable, ou une indication T-DISCONNECT, reçue à un point intermédiaire quelconque, provoquera la déconnexion de toutes les connexions de transport appartenant jusque là à la connexion MCS tout en engendrant une confirmation MCS-CONNECT-PROVIDER contenant une valeur défavorable.

Une demande MCS-CONNECT-PROVIDER spécifiera lequel des deux fournisseurs de service MCS a le niveau hiérarchique le plus élevé. Cette relation hiérarchique détermine l'acheminement ultérieur des unités MCSPDU de domaine, et la distinction entre fournisseur de service MCS appelant et fournisseur de service MCS appelé ne sera donc plus applicable. Par exemple, l'étape suivante consistera, pour la couche du service MCS, à fusionner les ressources des deux domaines jusque là indépendants. Des unités de type **MergeChannelsRequest** et **MergeTokensRequest** sont produites par le fournisseur de service MCS inférieur et transmises sur la nouvelle connexion MCS à destination du fournisseur MCS de niveau supérieur. Le sens dans lequel ces unités MCSPDU sont envoyées peut être d'appelant à appelé ou d'appelé à appelant, selon la façon dont le fanion de montée a été positionné.



T1602430-97

**Figure 10-1/T.125 – Flux de messages des unités MCSPDU de type connect et à paramètres étendus entre nœuds membres du sommet V3**



T1602440-97

**Figure 10-2/T.125 – Flux de messages des unités MCSPDU de type connect entre nœuds qui ne sont pas tous deux membres du sommet V3**

## 10.2 Unités MCSPDU de domaine

Le Tableau 10-1 spécifie l'acheminement des unités MCSPDU de domaine.

Si un fournisseur MCS produit ou envoie une unité MCSPDU de la catégorie *demande*, cette unité transitera en amont sur l'unique connexion MCS. Les demandes **ErectDomainRequest**, **ChannelJoinRequest**, **ChannelLeaveRequest** et **CapabilitiesNotificationRequest** peuvent être absorbées au niveau d'un quelconque fournisseur MCS intermédiaire. Les autres demandes remontent vers le fournisseur MCS supérieur pour y être traitées, à moins que le contenu d'une de ces demandes ne soit jugé non valide, auquel cas il sera possible de ne pas tenir compte de l'unité MCSPDU correspondante, sans confirmation.

**Tableau 10-1/T.125 – Acheminement des unités MCSPDU de domaine**

Catégorie	Unités MCSPDU	TC	Sens
Demande	<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>ChannelAdmitRequest</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	Ascendant
	<b>SendDataRequest</b> <b>UniformSendDataRequest</b>	A	
Indication	<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	Descendant
	<b>SendDataIndication</b> <b>UniformSendDataIndication</b>	A	

Si un fournisseur de service MCS produit ou envoie une unité MCSPDU de la catégorie *indication*, des copies de cette unité, de contenu éventuellement modifié, descendront sur zéro ou un autre nombre de connexions MCS, conformément aux règles suivantes:

- a) une unité **PlumbDomainIndication** est envoyée sur toutes les connexions MCS descendantes. La hauteur limite qu'elle contient est diminuée de 1. Un fournisseur MCS recevant cette unité MCSPDU avec une hauteur limite égale à zéro doit se déconnecter;

- b) une unité **PurgeChannelsIndication** est envoyée sur toutes les connexions MCS descendantes. L'ensemble des identificateurs d'utilisateur envoyé n'est pas modifié, de manière que tous les utilisateurs détachés soient annoncés à ceux qui restent attachés. L'ensemble des autres identificateurs de canal expédiés peut être limité à ceux qui sont en cours d'utilisation dans un sous-arbre. Chez un ancien fournisseur supérieur qui est encore en cours de fusion avec un domaine supérieur, aussi bien les identificateurs d'utilisateur que les identificateurs d'autres canaux sont limités à ceux dont l'acceptation dans le domaine supérieur a été confirmée;
- c) une unité **PurgeTokensIndication** est envoyée sur toutes les connexions MCS descendantes. L'ensemble des identificateurs de jeton expédiés peut être limité à ceux qui sont en cours d'utilisation dans un sous-arbre. Chez un ancien fournisseur supérieur qui est encore en cours de fusion avec un domaine supérieur, les identificateurs d'utilisateur sont limités à ceux dont l'acceptation dans le domaine supérieur a été confirmée;
- d) une unité **DetachUserIndication** est envoyée sur toute les connexions MCS descendantes. L'ensemble d'identificateurs d'utilisateur expédié reste inchangé, de sorte que tous les utilisateurs détachés seront annoncés à ceux qui restent. Chez un ancien fournisseur supérieur qui est encore en cours de fusion avec un domaine supérieur, les identificateurs d'utilisateur sont limités à ceux dont l'acceptation dans le domaine supérieur a été confirmée;
- e) une unité **ChannelDisbandIndication** est envoyée sur toutes les connexions MCS descendantes qui contiennent dans leur sous-arbre le gestionnaire du canal privé ou tout utilisateur admis;
- f) des unités **ChannelAdmitIndication** et **ChannelExpelIndication** sont envoyées sur toutes les connexions MCS descendantes qui contiennent dans leur sous-arbre un ou plusieurs des utilisateurs touchés. L'ensemble des identificateurs d'utilisateur expédié peut être limité à ceux qui résident dans un sous-arbre;
- g) une unité **TokenGiveIndication** est envoyée sur une seule connexion MCS descendante qui contient le destinataire désigné dans son sous-arbre;
- h) une unité **TokenPleaseIndication** est envoyée sur toutes les connexions MCS descendantes qui contiennent dans leur sous-arbre un utilisateur qui a saisi, inhibé ou va recevoir le jeton;
- i) une unité **CapabilitiesNotificationIndication** est envoyée sur toutes les connexions MCS descendantes du sommet V3. Chaque nœud qui reçoit cette unité MCSPDU modifie sa copie locale de la liste de capacités du sommet V3 selon le contenu de la liste d'ajout et de la liste de suppression;
- j) des unités **SendDataIndication** et **UniformSendDataIndication** sont envoyées sur toutes les connexions MCS descendantes qui adhèrent au canal spécifié, sauf qu'une unité **SendDataIndication**, lorsqu'elle est produite, n'est pas réexpédiée sur la connexion par laquelle l'unité **SendDataRequest** est arrivée.

Les indications **PurgeChannelsIndication**, **PurgeTokensIndication**, **DetachUserIndication**, **ChannelAdmitIndication** et **ChannelExpelIndication** n'ont pas besoin d'être envoyées si les ensembles d'identificateurs qu'elles contiennent sont vides.

Si un fournisseur de service MCS produit ou envoie une unité MCSPDU de la catégorie *réponse*, celle-ci transitera sur l'unique connexion MCS ascendante. Elle remontera vers le fournisseur MCS supérieur pour y être traitée, à moins que son contenu ne soit jugé non valide.

Si un fournisseur de service MCS produit ou envoie une unité MCSPDU de la catégorie *confirmation*, celle-ci transitera sur une seule connexion MCS descendante, conformément aux règles suivantes:

- a) une unité **MergeChannelsConfirm** refait, dans le sens opposé, le chemin parcouru par la plus récente demande **MergeChannelsRequest** qui n'a pas encore fait l'objet d'une confirmation en réponse. Cela implique que chaque fournisseur MCS maintienne une file d'attente des demandes en instance, du type premier entré-premier sorti;
- b) une unité **MergeTokensConfirm** refait, dans le sens opposé, le chemin parcouru par la plus récente demande **MergeTokensRequest** qui n'a pas encore fait l'objet d'une confirmation en réponse. Cela implique que chaque fournisseur de service MCS maintienne une file d'attente des demandes en instance, du type premier entré-premier sorti;
- c) une unité **AttachUserConfirm** refait, dans le sens opposé, le chemin parcouru par une demande **AttachUserRequest** antérieure – peu importe laquelle si plusieurs sont en instance – n'ayant pas encore fait l'objet d'une confirmation en réponse. Pour la bonne règle, il convient cependant que chaque fournisseur de service MCS maintienne une file d'attente de ces unités, du type premier entré-premier sorti. Lorsqu'il expédie une confirmation **AttachUserConfirm**, un fournisseur de service MCS doit enregistrer à quel sous-arbre l'identificateur d'utilisateur contenu dans cette unité va être ainsi assigné;
- d) d'autres unités MCSPDU de la catégorie des confirmations contiennent un identificateur d'utilisateur initiateur qui avait déjà été attribué dans le cadre de l'action de confirmation par l'unité **AttachUserConfirm** qui vient d'être exposée. Ces unités MCSPDU sont envoyées sur la connexion MCS descendante qui aboutit au sous-arbre auquel l'identificateur d'utilisateur a été attribué. Poursuivant leur chemin, les unités finissent par revenir au fournisseur qui couvre le rattachement MCS demandeur.

Les confirmations sont produites au cours du traitement, comme les demandes. Toutes ces unités, sauf **ChannelJoinConfirm**, sont produites par le fournisseur MCS supérieur.

Si un fournisseur de service MCS produit une unité MCSPDU du type *ultimatum*, cette unité transitera sur une seule connexion MCS, ascendante ou descendante. L'unité **DisconnectProviderUltimatum** donne au fournisseur de service MCS récepteur l'ordre de couper la connexion MCS qui l'achemine. L'unité **RejectMCSPDUUltimatum** rejette une unité MCSPDU erronée avec un code de diagnostic. Elle invite également le fournisseur qui l'a transmise à se déconnecter. Les unités d'ultimatum ne sont pas expédiées.

## 11 Signification des unités MCSPDU

Les Tableaux 11-1 à 11-51 reprennent le contenu de chacune des unités MCSPDU qui ont été définies au paragraphe 8.

### 11.1 Connect-Initial

L'unité de type **Connect-Initial** est produite par une demande MCS-CONNECT-PROVIDER. Elle est envoyée en tant que première unité TSDU sur la connexion de transport initiale d'une nouvelle connexion MCS. Elle produit, au niveau du récepteur, une indication MCS-CONNECT-PROVIDER.

**Tableau 11-1/T.125 – Unité MCSPDU de type Connect-Initial**

Contenu	Source	Collecteur
Sélecteur de domaine appelant	Demande	Indication
Sélecteur de domaine appelé	Demande	Indication
Fanion ascendant	Demande	Indication
Paramètres de domaine cibles	Demande	Indication
Paramètres de domaine minimaux	Demande	Indication
Paramètres de domaine maximaux	Demande	Indication
Données d'utilisateur	Demande	Indication

L'adresse de transport appelante et l'adresse de transport appelée sont des paramètres additionnels de la demande et de l'indication MCS-CONNECT-PROVIDER qui deviennent des paramètres de la primitive T-CONNECT et qui ne sont pas explicitement insérés dans une unité MCSPDU. La même paire d'adresses de transport doit être utilisée pour demander toutes les connexions de transport appartenant à la même connexion MCS.

La qualité du service de transport est un paramètre additionnel de la demande – mais non de l'indication – MCS-CONNECT-PROVIDER. La qualité de service peut varier d'une connexion de transport à une autre et la qualité disponible n'est révélée que lors de l'établissement de chaque connexion de transport. Comme le nombre de connexions de transport additionnelles n'est pas connu avant que la primitive MCS-CONNECT-PROVIDER ait négocié le paramètre de domaine relatif au nombre de priorités de données mises en œuvre, cette primitive ne peut pas en même temps négocier complètement la qualité du service de transport de ces connexions. Un fournisseur de service MCS appelé doit donc accepter automatiquement les connexions de transport entrantes au niveau de qualité de service offert, du moment que ce niveau est conforme à toute valeur minimale spécifiée à titre d'option par le fournisseur de service MCS appelant et indiquée par chaque primitive T-CONNECT.

L'interprétation des valeurs du sélecteur de domaine relève d'une décision locale pour chaque fournisseur de service MCS. Ces valeurs sont des chaînes d'octets qui ont les caractéristiques d'une adresse. Les valeurs acceptables peuvent être déterminées au cours du processus de configuration d'un fournisseur de service MCS. Un même domaine peut être sélectionné par plusieurs valeurs. Un sélecteur de domaine non spécifié est représenté par une chaîne d'octets de longueur nulle. Ce conflit peut être résolu, conformément à des accords locaux, en donnant à cette chaîne une valeur explicite quelconque.

Le fanion ascendant spécifie le sens d'une nouvelle connexion MCS; il prend la valeur *true* si le fournisseur appelé est à considérer comme ayant un niveau supérieur à celui du fournisseur appelant et la valeur *false* dans le cas contraire. Un fournisseur de service MCS joue un rôle dans la hiérarchie d'un domaine selon le sens des connexions MCS auxquelles il participe. Aucun fournisseur ne doit autoriser deux connexions vers des fournisseurs de niveau supérieur. Un fournisseur sans connexion vers un fournisseur de niveau supérieur doit remplir le rôle de fournisseur MCS supérieur.

Les paramètres de domaine cibles de l'unité **Connect-Initial** doivent chacun avoir une valeur comprise entre les valeurs maximale et minimale spécifiées. Un fournisseur de service MCS doit réviser les paramètres de domaine demandés pour tenir compte des limites de sa mise en œuvre ou pour imposer des valeurs déjà convenues par les membres existants d'un domaine. Le fournisseur peut augmenter les valeurs minimales et diminuer les valeurs maximales. Il ne doit modifier les valeurs cibles que pour les faire cadrer avec l'intervalle. Il y a lieu qu'un fournisseur de service MCS soit disposé à donner suite à toute réponse satisfaisant à la gamme de valeurs qu'il propose.

Les données d'utilisateur forment une chaîne d'octets arbitraire, qui peut avoir une longueur nulle.

Un fournisseur de service MCS accepte automatiquement chaque connexion de transport, selon ses limites de capacité. Les données d'utilisateur contenues dans la primitive T-CONNECT ne sont pas utilisées. La première unité TSDU reçue lors du transfert de données, qui sera une unité MCSPDU de type **Connect-Initial** ou **Connect-Additional**, détermine la nature de la connexion de transport. Si le contenu n'est pas acceptable, un fournisseur MCS appelé peut couper immédiatement la connexion de transport. La réaction préférée consiste à retourner une unité **Connect-Response** ou **Connect-Result**, selon le cas, expliquant pourquoi la connexion MCS n'a pu être établie. Le fournisseur MCS appelant doit ensuite se déconnecter.

## 11.2 Connect-Response

L'unité **Connect-Response** est produite par une réponse MCS-CONNECT-PROVIDER. C'est la première unité TSDU qui est renvoyée dans le sens inverse sur la connexion de transport initiale d'une nouvelle connexion MCS. Cette unité signale au fournisseur de service MCS appelant l'acceptation d'une connexion MCS. Ce fournisseur procède ensuite à l'établissement de toutes les autres connexions de transport requises.

**Tableau 11-2/T.125 – Unité MCSPDU de type Connect-Response**

Contenu	Source	Collecteur
Résultat	Réponse	Confirmation
Paramètres de domaine	Réponse	Confirmation
Identificateur de connexion chez l'appelé	Fournisseur appelé	Fournisseur appelant
Données d'utilisateur	Réponse	Confirmation

Si le résultat est favorable, cette unité MCSPDU fixe les paramètres de domaine effectifs, dont le nombre de priorités de transfert de données MCS mises en œuvre, égal au nombre de connexions de transport appartenant à une connexion de couche MCS. Si ce nombre est supérieur à un, des connexions de transport additionnelles doivent être créées et associées à la connexion MCS par l'échange d'unités MCSPDU de type **Connect-Additional** et **Connect-Result**.

L'identificateur de connexion pour l'appelé sert d'outil pour associer des connexions de transport additionnelles entrantes à la connexion de transport initiale, au niveau du fournisseur de service MCS appelé. Sa valeur est choisie à cette seule fin. Il doit identifier uniquement une connexion MCS en cours au niveau du fournisseur appelé. La signification de cet identificateur ne dure pas au-delà de l'exécution de la primitive MCS-CONNECT-PROVIDER.

La plupart des paramètres de la confirmation MCS-CONNECT-PROVIDER sont acheminés dans l'unité **Connect-Response**. Si le résultat est défavorable ou si aucune connexion de transport additionnelle n'est requise, la confirmation est émise immédiatement. Sinon, elle est retardée jusqu'au moment où les résultats sont connus pour l'affectation de connexions de transport additionnelles à la connexion MCS.

## 11.3 Connect-Additional

Si le fournisseur de service MCS appelant détermine que les deux fournisseurs font partie du sommet V3, l'unité **Connect-Additional** est produite après la réception de l'unité **Extended-Parameters-Accept**, sinon après la réception de l'unité **Connect-Response**. Elle est envoyée en tant que première unité TSDU sur une connexion de transport additionnelle d'une nouvelle connexion MCS.

Le paramètre priorité des données prend les valeurs *haute*, *moyenne*, *basse*, en séquence, jusqu'au nombre de connexions de transport additionnelles requises.

**Tableau 11-3/T.125 – Unité MCSPDU de type Connect-Additional**

Contenu	Source	Collecteur
Identificateur de connexion chez l'appelé	Fournisseur appelant	Fournisseur appelé
Priorité des données	Fournisseur appelant	Fournisseur appelé

#### 11.4 Connect-Result

L'unité **Connect-Result** est produite après réception d'une unité **Connect-Additional**. Il s'agit de la première unité TSDU renvoyée dans le sens inverse sur une connexion de transport additionnelle d'une nouvelle connexion MCS.

**Tableau 11-4/T.125 – Unité MCSPDU de type Connect-Result**

Contenu	Source	Collecteur
Résultat	Fournisseur appelé	Confirmation

Si un résultat quelconque est défavorable, la confirmation MCS-CONNECT-PROVIDER doit être immédiatement produite. Toutes les connexions de transport associées à la connexion MCS doivent être coupées et il ne doit être tenu compte d'aucune des unités MCSPDU acheminées.

Dans le cas contraire, on attendra des résultats favorables pour chaque connexion de transport additionnelle. Ces résultats pourront être retournés hors séquence. Lorsqu'ils auront tous été collectés, une confirmation MCS-CONNECT-PROVIDER sera émise.

Après la confirmation MCS-CONNECT-PROVIDER, des unités MCSPDU de domaine pourront transiter par une connexion MCS. Chaque connexion MCS appartient à un seul domaine. Dans les configurations où un fournisseur de service MCS couvre plusieurs domaines, la connexion MCS qui achemine les unités MCSPDU détermine le domaine auquel celles-ci sont applicables. Dans le reste du présent sous-paragraphe, la description de chaque unité MCSPDU est placée dans le contexte d'un seul domaine.

#### 11.5 Extended-Parameters-Propose

Après la réception de l'unité **Connect-Response**, si le fournisseur de service MCS appelant détermine que les deux fournisseurs font partie du sommet V3, l'unité de type **Extended-Parameters-Propose** est produite. Elle est envoyée sur la connexion de transport initiale d'une nouvelle connexion MCS.

**Tableau 11-5/T.125 – Unité MCSPDU de type Extended-Parameters-Propose**

Contenu	Source	Collecteur
Paramètres étendus cibles	Fournisseur appelant	Fournisseur appelé
Paramètres étendus minimaux	Fournisseur appelant	Fournisseur appelé
Paramètres étendus maximaux	Fournisseur appelant	Fournisseur appelé

Pour les paramètres étendus comportant une gamme de valeurs possibles, les paramètres cibles de l'unité **Extended-Parameters-Propose** doivent tous se situer entre les valeurs minimales et maximales spécifiées. Un fournisseur de service MCS doit réviser les paramètres étendus demandés pour refléter les limites de son implémentation ou pour imposer des valeurs faisant déjà l'objet d'une entente entre les membres existants d'un domaine. Il peut augmenter les valeurs minimales et diminuer les valeurs maximales. Il ne doit changer les valeurs cibles qu'afin de les maintenir à l'intérieur de l'intervalle. Un fournisseur de service MCS doit être prêt à donner suite à tout objectif accepté situé à l'intérieur de la plage des valeurs qu'il propose. Si les paramètres étendus sont inacceptables, un fournisseur de service MCS appelé coupera immédiatement la connexion de transport.

Pour la valeur *Données non fiables traitées*, si le fournisseur appelant souhaite permettre la négociation de la valeur, la valeur minimale sera False, la valeur maximale sera True et la valeur cible indiquera la préférence de l'appelant. Si le soutien de données non fiables n'est pas négociable, les valeurs cibles, minimales et maximales seront établies selon les préférences des appelants.

La valeur *Identificateur de référence de domaine* sera identique pour les valeurs cibles, minimales et maximales et identifiera ce domaine de façon unique pour le fournisseur appelant. Le fournisseur appelé peut utiliser cette valeur pour indiquer ce domaine au fournisseur appelant.

### 11.6 Extended-Parameters-Accept

L'unité **Extended-Parameters-Accept** est produite après la réception de l'unité **Extended-Parameters-Propose**. Elle véhicule l'acceptation des valeurs de paramètre étendu au fournisseur de service MCS appelant, qui entreprend ensuite d'établir toutes les connexions de transport additionnelles requises.

**Tableau 11-6/T.125 – Unité MCSPDU de type Extended-Parameters-Accept**

Contenu	Source	Collecteur
Paramètres étendus	Fournisseur appelé	Fournisseur appelant

La valeur *Identificateur de référence de domaine* sera identique pour les valeurs cibles, minimales et maximales et identifiera ce domaine de façon unique pour le fournisseur appelé. Le fournisseur appelant peut utiliser cette valeur pour indiquer ce domaine au fournisseur appelé.

### 11.7 PlumbDomainIndication

L'unité **PlumbDomainIndication** est produite dès que la primitive MCS-CONNECT-PROVIDER a été suivie d'effet. Cette unité raccorde par "plombage" la hiérarchie des fournisseurs de service MCS au-dessous d'une nouvelle connexion MCS pour faire en sorte qu'aucune boucle ne soit créée. L'unité **PlumbDomainIndication** est également produite par le fournisseur de service MCS supérieur pour mettre en application la hauteur maximale d'un domaine.

**Tableau 11-7/T.125 – Unité MCSPDU de type PlumbDomainIndication**

Contenu	Source	Collecteur
Limite de hauteur	Fournisseur supérieur antérieur ou direct	Subordonnés

L'unité **PlumbDomainIndication** est produite à l'extrémité inférieure d'une nouvelle connexion MCS, par le fournisseur qui a cessé d'être le fournisseur supérieur d'un domaine. Son contenu est initialisé conformément au paramètre de domaine relatif à la hauteur maximale du domaine. L'unité **PlumbDomainIndication** est transmise sur toutes les connexions MCS descendantes.

Le cas échéant, l'unité **PlumbDomainIndication** est produite de la même façon au niveau du fournisseur de service MCS supérieur.

Chaque fois qu'une unité **PlumbDomainIndication** est reçue, la hauteur limite qu'elle contient est contrôlée. Si elle est supérieure à zéro, cette limite est diminuée de 1 et l'unité **PlumbDomainIndication** est envoyée sur toutes les connexions MCS descendantes. Une valeur égale à zéro signifie par ailleurs que le récepteur est trop éloigné du fournisseur supérieur. Il doit réagir en coupant la connexion MCS ascendante, ce qui supprime un sous-arbre entier et contribue à reconstituer la hauteur du domaine.

En présence d'une boucle, la hauteur limite contenue dans l'unité **PlumbDomainIndication** doit diminuer jusqu'à atteindre zéro. Le fournisseur qui détecte la boucle doit l'interrompre, quitte à retrancher du domaine tous les fournisseurs de la boucle ainsi que leurs subordonnés.

NOTE – Un fournisseur de service MCS, n'ayant qu'une connaissance purement locale des connexions MCS, ne peut pas empêcher la formation de boucles. Il peut veiller à ce qu'il n'y ait, à tout moment, qu'une seule connexion ascendante. Mais il ne peut pas garantir que cette connexion ascendante ne se reboucle pas avec un certain fournisseur situé plus bas dans la hiérarchie. Lorsqu'une boucle se forme, la cause immédiate est la création d'une connexion ascendante erronée à partir du fournisseur de service MCS supérieur. Les applications de régulation, qui spécifient les connexions MCS à créer, doivent faire en sorte que de telles erreurs ne se produisent pas.

### **11.8 ErectDomainRequest**

L'unité **ErectDomainRequest** est produite dès que la primitive MCS-CONNECT-PROVIDER a été suivie d'effet. Elle communique dans le sens ascendant des modifications de hauteur de fournisseurs ainsi que les intervalles d'application de débit correspondants. L'unité **ErectDomainRequest** est produite par un fournisseur de service MCS chaque fois que la hauteur ou l'intervalle de celui-ci change.

La hauteur d'un fournisseur de service MCS peut changer lorsqu'une connexion MCS est ajoutée ou supprimée ou lorsqu'un fournisseur subordonné signale une modification au moyen d'une unité **ErectDomainRequest**. Son intervalle de supervision pour mettre en application le débit minimal spécifié sous forme d'un paramètre de domaine peut se modifier par adaptation aux intervalles signalés par des subordonnés ou pour d'autres raisons. Si la valeur de hauteur ou d'intervalle change, le fournisseur de service MCS doit transmettre une unité de demande **ErectDomainRequest** à son supérieur immédiat.

**Tableau 11-8/T.125 – Unité MCSPDU de type ErectDomainRequest**

Contenu	Source	Collecteur
Hauteur dans le domaine	Subordonné	Fournisseur supérieur
Intervalle d'application de débit	Subordonné	Fournisseur supérieur

### 11.9 MergeChannelsRequest

L'unité **MergeChannelsRequest** est produite dès que la primitive MCS-CONNECT-PROVIDER est suivie d'effet. Elle communique dans le sens ascendant les attributs des canaux dépendant d'un précédent fournisseur supérieur de manière qu'ils puissent être intégrés dans le domaine fusionné.

**Tableau 11-9/T.125 – Unité MCSPDU de type MergeChannelsRequest**

Contenu	Source	Collecteur
Canaux de fusion	Précédent fournisseur supérieur	Fournisseur supérieur
Identificateurs de canal de purge	Fournisseurs intermédiaires	Fournisseur supérieur

On peut insérer dans l'unité **MergeChannelsRequest** les attributs de plusieurs canaux, dans les limites fixées par le domaine quant à la longueur des unités MCSPDU. Comme indiqué dans les définitions ASN.1 du paragraphe 7, chacun des quatre types de canaux utilisés (statiques, d'identificateurs d'utilisateur, privés, attribués) possède son propre ensemble d'attributs. Ceux-ci sont conservés dans la base de données du fournisseur de service MCS supérieur et sont partiellement copiés dans les sous-arbres où un canal est utilisé. Lorsque deux domaines sont fusionnés par l'action d'une primitive MCS-CONNECT-PROVIDER, les canaux qui sont en service dans le domaine inférieur doivent soit être intégrés dans la base de données du domaine supérieur ou être purgés du domaine inférieur. Cette décision relève du fournisseur supérieur du domaine fusionné.

Chaque canal doit être pris en compte individuellement. Si les limites de domaine concernant les canaux utilisés le permettent et si l'identificateur de canal n'a pas déjà été utilisé de manière conflictuelle, le domaine supérieur doit s'agrandir pour intégrer ce canal. L'utilisation d'un identificateur de canal statique ne provoque jamais de conflit. L'utilisation d'un identificateur de canal privé dans le domaine inférieur ne provoque pas de conflit si cet identificateur est également utilisé pour identifier un canal privé dans le domaine supérieur et s'il possède le même identificateur d'utilisateur que le gestionnaire. Toutes les autres combinaisons d'utilisations simultanées sont interdites, car elles imposent que l'identificateur de canal soit purgé du domaine inférieur.

Si un canal privé possède un ensemble de nombreux utilisateurs admis, ses attributs peuvent ne pas s'intégrer dans une même unité de demande **MergeChannelsRequest**: ces attributs doivent être envoyés vers le haut de la hiérarchie dans plusieurs unités MCSPDU. La deuxième demande de fusion du même canal privé, ainsi que les demandes subséquentes, doivent être tenues en attente de la réception d'une confirmation **MergeChannelsConfirm** en réponse à la première demande. Ce n'est qu'à ce moment que l'on sait si les limites de domaine ont permis que le canal soit utilisé dans le domaine supérieur. Si la première demande n'est pas suivie d'effet, elle ne doit pas être renouvelée avec un sous-ensemble restant d'utilisateurs admis.

Chaque unité **MergeChannelsRequest** suscite une réponse de confirmation de type **MergeChannelsConfirm** de la part du fournisseur de service MCS supérieur, dans la même séquence. Une unité **MergeChannelsConfirm** ne contient rien qui permette d'identifier expressément l'unité **MergeChannelsRequest** précédente. Les réponses ne doivent être acheminées que dans l'ordre où ces unités ont été reçues. Les fournisseurs MCS situés au-dessus du précédent supérieur doivent effectuer un enregistrement de chaque unité **MergeChannelsRequest** non suivie

de réponse et du point d'où elle provient, de manière que la confirmation **MergeChannelsConfirm** correspondante puisse être renvoyée via la même connexion MCS.

Les fournisseurs MCS intermédiaires doivent valider les identificateurs d'utilisateur annoncés dans les attributs de canal privé, pour faire en sorte qu'ils soient légitimement attribués au sous-arbre d'où l'unité **MergeChannelsRequest** provient. Les identificateurs d'utilisateur non valides doivent être supprimés. Si le gestionnaire d'un canal privé est supprimé, tous les attributs de ce canal doivent être retranchés de la demande de fusion et seul l'identificateur du canal doit être inséré dans l'ensemble à purger. A l'exception de cette validation des identificateurs d'utilisateur, les fournisseurs de service MCS intermédiaires ne doivent pas modifier le contenu d'une unité **MergeChannelsRequest**.

Un précédent fournisseur supérieur doit attendre un message individuel de confirmation que tous les identificateurs d'utilisateur et tous les identificateurs de jeton ont été intégrés dans un domaine fusionné ou ont été purgés avant que ce fournisseur commence à soumettre des attributs de canal statique, attribué ou privé en vue d'une fusion.

### 11.10 MergeChannelsConfirm

L'unité **MergeChannelsConfirm** répond à une précédente unité **MergeChannelsRequest**. Elle correspond au même ensemble d'identificateurs de canal et à un sous-ensemble de leurs attributs. Les attributs de canal non intégrés dans le domaine fusionné sont signalés comme étant des identificateurs de canal à purger.

Les identificateurs de canal acceptés sont repris avec les attributs introduits dans la base de données du fournisseur de service MCS supérieur. Les fournisseurs intermédiaires doivent mettre à jour leur base de données en conséquence.

**Tableau 11-10/T.125 – Unité MCSPDU de type MergeChannelsConfirm**

Contenu	Source	Collecteur
Canaux de fusion	Fournisseur supérieur	Fournisseurs
Identificateurs de canal de purge	Fournisseur supérieur	Supérieur précédent

Les canaux à purger du domaine inférieur ne sont énumérés que par leurs identificateurs. Si les mêmes identificateurs de canal sont utilisés dans le domaine supérieur, ils sont laissés tels quels. Les fournisseurs intermédiaires doivent expédier les identificateurs de canal purgés sans les traiter.

Les fournisseurs de service MCS doivent acheminer l'unité **MergeChannelsConfirm** vers la source de l'unité **MergeChannelsRequest** précédente, en appliquant le principe qu'il existe une seule réponse à chaque demande. La confirmation **MergeChannelsConfirm** retourne vers le précédent fournisseur supérieur qui avait émis la demande **MergeChannelsRequest**. Ensuite, il est possible de ne pas tenir compte des canaux fusionnés, car ils sont restés dans la base de données en attendant la réponse. Les identificateurs de canal purgés doivent être supprimés, car ils doivent faire l'objet d'une indication **PurgeChannelsIndication**.

Les fournisseurs de service MCS de niveau intermédiaire doivent confirmer que les identificateurs d'utilisateur annoncés dans des attributs de canal privé sont attribués au sous-arbre vers lequel l'unité **MergeChannelsConfirm** est acheminée. Si un gestionnaire de canal privé a été détaché et réaffecté ailleurs dans l'intervalle qui s'est écoulé depuis la validation de l'unité **MergeChannelsRequest** antérieure, un fournisseur de niveau intermédiaire doit émettre une demande **ChannelDisbandRequest** faisant du canal privé une victime de la fusion de domaines, puis transférer l'identificateur de ce canal dans l'ensemble purgé. Si des utilisateurs autorisés ont été réaffectés ailleurs, ce fournisseur doit les exclure du canal.

### 11.11 PurgeChannelsIndication

L'unité **PurgeChannelsIndication** est produite chez un précédent fournisseur supérieur dès réception d'une confirmation **MergeChannelsConfirm**. Elle est diffusée en aval et supprime l'utilisation d'identificateurs de canal donnés chez des fournisseurs subordonnés.

**Tableau 11-11/T.125 – Unité MCSPDU de type PurgeChannelsIndication**

Contenu	Source	Collecteur
Identificateurs de détachement d'utilisateur	Supérieur précédent	Subordonnés
Identificateurs de purge de canal	Supérieur précédent	Subordonnés

Selon l'utilisation en cours d'un identificateur de canal, l'effet de sa purge est l'envoi d'une indication MCS-DETACH-USER à tous les utilisateurs d'un canal d'identification d'utilisateur; d'une indication MCS-CHANNEL-LEAVE aux utilisateurs qui ont adhéré à un identificateur de canal statique ou attribué; d'une indication MCS-CHANNEL-DISBAND au gestionnaire et d'une indication MCS-CHANNEL-EXPEL aux utilisateurs admis d'un identificateur de canal privé.

Un précédent fournisseur supérieur, connaissant l'utilisation de tous les canaux dans son domaine inférieur, peut produire les indications appropriées à partir des seuls identificateurs de canal. Ses subordonnés peuvent cependant n'en avoir qu'une connaissance partielle. Ils doivent toujours être informés des identificateurs de canaux qui représentent des utilisateurs détachés, pour lesquels une indication est toujours émise, et qui représentent d'autres types de canaux, pour lesquels une indication n'est émise que si le canal en cause est en cours d'utilisation chez le fournisseur subordonné. L'unité **PurgeChannelsIndication** subdivisera donc les identificateurs de canal à purger selon ces deux catégories.

La purge d'un identificateur d'utilisateur par l'intermédiaire d'une unité **PurgeChannelsIndication** doit avoir les mêmes conséquences que leur suppression par l'intermédiaire d'une unité **DetachUserIndication**, sauf que celle-ci n'a pas besoin de produire, à titre d'effet secondaire, une unité **ChannelDisbandIndication** ou **TokenGiveConfirm**, puisque le récepteur de l'unité **PurgeChannels Indication** n'est plus le fournisseur supérieur.

NOTE – Un fournisseur peut recevoir, insérés dans une indication **PurgeChannelsIndication**, des identificateurs de canal statique et de canal attribué auxquels il n'a pas adhéré. Il peut également recevoir des identificateurs de canal privé pour lesquels ses rattachements ne sont ni des gestionnaires ni des utilisateurs admis. Les enregistrements d'utilisation contenus dans la base de données permettront à un fournisseur de supprimer les primitives d'indication correspondant à de tels identificateurs de canal.

### 11.12 MergeTokensRequest

L'unité **MergeTokensRequest** est produite dès que la primitive MCS-CONNECT-PROVIDER est suivie d'effet. Elle transmet en amont de la hiérarchie les attributs des jetons détenus par un précédent fournisseur supérieur, de manière que ces attributs puissent être intégrés dans le domaine fusionné.

**Tableau 11-12/T.125 – Unité MCSPDU de type MergeTokensRequest**

Contenu	Source	Collecteur
Jetons de fusion	Supérieur précédent	Fournisseur supérieur
Identificateurs de jeton de purge	Intermédiaires	Fournisseur supérieur

On peut insérer dans l'unité **MergeTokensRequest** les attributs de plusieurs jetons, dans les limites fixées par le domaine quant à la longueur des unités MCSPDU. Comme indiqué dans les définitions ASN.1 du paragraphe 7, chaque état de jeton utilisé (saisi, inhibé, donnant, inaccessible, donné) possède son propre ensemble d'attributs. Ceux-ci sont conservés dans la base de données du fournisseur de service MCS supérieur et sont partiellement copiés dans les sous-arbres où un jeton est utilisé. Lorsque deux domaines sont fusionnés par l'action d'une primitive MCS-CONNECT-PROVIDER, les jetons qui sont en service dans le domaine inférieur doivent soit être intégrés dans la base de données du domaine supérieur soit être purgés du domaine inférieur. Cette décision relève du fournisseur supérieur du domaine fusionné.

Chaque jeton doit être pris en compte individuellement. Si les limites de domaine concernant les jetons utilisés le permettent et si l'identificateur de jeton n'a pas déjà été utilisé de manière conflictuelle, le domaine supérieur doit s'agrandir pour intégrer ce jeton. L'inhibition d'un identificateur de jeton dans le domaine inférieur ne provoque pas de conflit s'il est également inhibé dans le domaine supérieur. Toutes les autres combinaisons pour usage simultané sont interdites, ce qui nécessite que l'identificateur de jeton soit purgé du domaine inférieur.

Si un jeton possède un grand ensemble d'utilisateurs qui l'inhibent, il se peut que ses attributs ne puissent pas s'insérer dans une seule unité **MergeTokensRequest** et qu'il soit nécessaire de les faire remonter dans plusieurs unités MCSPDU. Cependant, la deuxième demande et les demandes subséquentes concernant le même jeton inhibé doivent être tenues en attente de la réception d'une confirmation **MergeTokensConfirm** en réponse à la première demande. Ce n'est qu'à ce moment que l'on saura si les limites du domaine ont permis de mettre en œuvre ce jeton dans le domaine supérieur. Si la première demande n'est pas suivie d'effet, il ne faut pas la répéter avec un sous-ensemble restant d'inhibiteurs.

Chaque unité **MergeTokensRequest** suscite une réponse **MergeTokensConfirm** de la part du fournisseur de service MCS supérieur, dans la même séquence. Une unité **MergeTokensConfirm** ne contient rien qui permette d'identifier expressément l'unité **MergeTokensRequest** précédente. Les réponses ne doivent être acheminées que dans l'ordre où ces unités ont été reçues. Les fournisseurs de service MCS situés au-dessus du précédent supérieur doivent effectuer un enregistrement de chaque unité **MergeTokensRequest** non suivie de réponse et du point d'où elle provient, de manière que la confirmation **MergeTokensConfirm** correspondante puisse être renvoyée via la même connexion MCS.

Les fournisseurs de service MCS intermédiaires doivent valider les identificateurs d'utilisateur annoncés dans les attributs de jeton, pour faire en sorte qu'ils soient légitimement attribués au sous-arbre d'où l'unité **MergeTokensRequest** provient. Les identificateurs d'utilisateur non valides doivent être supprimés. Un jeton en cours de cession doit rester saisi si l'utilisateur saisisseur ou l'utilisateur destinataire est supprimé, mais non les deux. Si un jeton est libéré à la suite de cette suppression d'identificateurs d'utilisateur, tous les attributs de ce jeton doivent être retranchés de la demande de fusion et seul l'identificateur du jeton doit être inséré dans l'ensemble à purger. Un jeton inhibé doit rester inhibé dans l'unité **MergeTokensRequest** même si tous les inhibiteurs sont supprimés, ce qui laissera un ensemble vide dans les attributs, parce que certains inhibiteurs peuvent continuer à exister après la purge d'autres unités MCSPDU. A l'exception de cette validation des identificateurs d'utilisateur, les fournisseurs de service MCS intermédiaires ne doivent pas modifier le contenu d'une unité **MergeTokensRequest**.

Un précédent fournisseur supérieur doit attendre un message individuel de confirmation que tous les identificateurs d'utilisateur ont été intégrés dans un domaine fusionné ou ont été purgés avant que ce fournisseur commence à soumettre des attributs de jeton en vue d'une fusion.

### 11.13 MergeTokensConfirm

Une unité **MergeTokensConfirm** répond à une précédente demande **MergeTokensRequest**. Elle correspond au même ensemble d'identificateurs de jeton et à un sous-ensemble de ses attributs. Les attributs de jeton non intégrés dans le domaine fusionné sont signalés comme étant des identificateurs de jeton à purger.

**Tableau 11-13/T.125 – Unité MCSPDU de type MergeTokensConfirm**

Contenu	Source	Collecteur
Jetons de fusion	Fournisseur supérieur	Intermédiaires
Identificateurs de jeton de purge	Fournisseur supérieur	Précédent supérieur

Les identificateurs de jeton acceptés correspondent aux attributs qui ont été insérés dans la base de données au niveau du fournisseur de service MCS supérieur. Les fournisseurs intermédiaires doivent mettre à jour leur base de données en conséquence.

Les jetons à purger du domaine inférieur ne sont énumérés que par leurs identificateurs. Si les mêmes identificateurs de jeton sont utilisés dans le domaine supérieur, ils sont laissés inchangés. Les fournisseurs intermédiaires doivent expédier les identificateurs de jeton purgés sans les modifier.

Les fournisseurs de service MCS doivent acheminer les confirmations **MergeTokensConfirm** vers la source de la demande **MergeTokensRequest** précédente, en appliquant le principe qu'il existe une seule réponse à chaque demande. La confirmation **MergeTokensConfirm** retourne vers le précédent supérieur qui avait émis la demande **MergeTokensRequest**. Ensuite, il est possible de ne pas tenir compte des jetons fusionnés, car ils sont restés dans la base de données en attendant la réponse. Les identificateurs de jeton purgés doivent être supprimés, car ils doivent faire l'objet d'une indication **PurgeTokensIndication**.

Les fournisseurs de service MCS intermédiaires doivent confirmer que les identificateurs d'utilisateur annoncés dans des attributs de jeton sont attribués au sous-arbre vers lequel la confirmation **MergeTokensConfirm** est acheminée. Si certains identificateurs d'utilisateur ont été détachés et réattribués depuis la validation de la demande **MergeTokensRequest** précédente, un fournisseur intermédiaire doit émettre, pour ces identificateurs, une demande **DetachUserRequest** avec le code de cause *canal purgé*, faisant de ces identificateurs des victimes de la fusion de domaines. Si un identificateur de jeton non inhibé se trouve libéré à cause de cette suppression d'identificateurs d'utilisateur, il doit être transféré dans l'ensemble purgé.

### 11.14 PurgeTokensIndication

L'unité d'indication **PurgeTokensIndication** est produite chez un précédent fournisseur supérieur dès réception d'une confirmation **MergeTokensConfirm**. Elle est diffusée en aval et supprime l'utilisation d'identificateurs de jeton spécifiés par des fournisseurs subordonnés.

**Tableau 11-14/T.125 – Unité MCSPDU de type PurgeTokensIndication**

Contenu	Source	Collecteur
Identificateurs de jeton de purge	Supérieur précédent	Subordonnés

L'effet de la purge d'un jeton est grave: elle provoque l'envoi d'une indication MCS-DETACH-USER à tout utilisateur ayant saisi, inhibé ou recevant un des identificateurs de jeton. Un fournisseur doit

mettre en œuvre cette procédure en émettant une demande **DetachUserRequest** pour le compte des utilisateurs touchés, avec la cause *jeton purgé*.

NOTE – On prévoit que la Recommandation T.122 puisse être révisée pour permettre une indication MCS-TOKEN-RELEASE dans cette situation. Cela permettrait à l'utilisateur touché de rester rattaché bien que son droit d'utilisation du jeton ait été retiré.

### 11.15 DisconnectProviderUltimatum

L'unité **DisconnectProviderUltimatum** est produite par une demande MCS-DISCONNECT-PROVIDER. A son tour, cette unité produit une primitive d'indication MCS-DISCONNECT-PROVIDER à l'autre extrémité d'une connexion MCS. L'unité **DisconnectProviderUltimatum** oblige le récepteur à couper la connexion MCS qui l'a acheminée.

L'unité **DisconnectProviderUltimatum** peut également être produite par un fournisseur de service MCS qui détecte un état d'erreur comme l'existence d'une boucle dans la hiérarchie d'un domaine. En ces occurrences, la cause est autre que *sur demande de l'utilisateur*.

**Tableau 11-15/T.125 – Unité MCSPDU de type DisconnectProviderUltimatum**

Contenu	Source	Collecteur
Cause	Fournisseur demandeur	Indication

### 11.16 RejectMCSPDUUltimatum

L'unité **RejectMCSPDUUltimatum** est produite lorsqu'un fournisseur de service MCS reçoit une unité MCSPDU non valide ou détecte une erreur de protocole de communication MCS. Cette unité invite le fournisseur homologue, à l'autre extrémité d'une connexion MCS, à se déconnecter parce qu'il n'est pas certain que l'on puisse opérer un rétablissement à partir d'une situation qui ne devrait pas se produire.

L'unité **RejectMCSPDUUltimatum** diagnostique l'erreur et renvoie une portion initiale de l'unité TSDU fautive, normalement autant d'octets que pourra en recevoir une unité MCSPDU de longueur maximale. Le fournisseur récepteur a la possibilité de se déconnecter ou de poursuivre.

**Tableau 11-16/T.125 – Unité MCSPDU de type RejectMCSPDUUltimatum**

Contenu	Source	Collecteur
Diagnostic	Fournisseur rejetant	Fournisseur rejeté
Octets initiaux	Fournisseur rejetant	Fournisseur rejeté

### 11.17 AttachUserRequest

L'unité **AttachUserRequest** est produite par une demande MCS-ATTACH-USER. Elle remonte vers le fournisseur de service MCS supérieur, qui retourne une réponse de type **AttachUserConfirm**. Si la limite de domaine concernant le nombre d'identificateurs d'utilisateur le permet, un nouvel identificateur d'utilisateur est produit.

**Tableau 11-17/T.125 – Unité MCSPDU de type AttachUserRequest**

Contenu	Source	Collecteur
(Néant)	–	–

L'unité **AttachUserRequest** ne contient pas d'informations autres que son type d'unité MCSPDU. Le domaine auquel l'utilisateur se rattache est déterminé par la connexion MCS qui achemine cette MCSPDU. La seule caractéristique initiale de l'identificateur d'utilisateur produit est le fait qu'elle est unique.

Un fournisseur de service MCS doit tenir un registre de toutes les unités **AttachUserRequest** reçues sans réponse et des connexions MCS par lesquelles ces unités sont arrivées, de manière qu'une confirmation **AttachUserConfirm** puisse être réexpédiée en réponse à la même source. Pour répartir équitablement les réponses, il convient que chaque fournisseur constitue à cette fin une file d'attente de type premier entré-premier sorti.

### 11.18 AttachUserConfirm

L'unité **AttachUserConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une demande **AttachUserRequest**. Renvoyée au fournisseur demandeur, cette confirmation engendre une confirmation MCS-ATTACH-USER.

**Tableau 11-18/T.125 – Unité MCSPDU de type AttachUserConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur (sur option)	Fournisseur supérieur	Confirmation

L'unité **AttachUserConfirm** contient un identificateur d'utilisateur si et seulement si le résultat est favorable. Les fournisseurs qui reçoivent une confirmation **AttachUserConfirm** favorable doivent introduire l'identificateur d'utilisateur dans leur base de données.

Les fournisseurs de service MCS doivent acheminer l'unité **AttachUserConfirm** vers la source d'une demande **AttachUserRequest** antérieure, en appliquant le principe qu'il existe une seule réponse par demande. Un fournisseur qui envoie une unité **AttachUserConfirm** doit enregistrer la connexion MCS descendante à laquelle le nouvel identificateur d'utilisateur est ainsi attribué, de manière qu'il puisse valider cet identificateur lorsqu'il réapparaîtra dans d'autres demandes.

### 11.19 DetachUserRequest

L'unité **DetachUserRequest** est produite par une demande MCS-DETACH-USER. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui supprime l'utilisateur de sa base de données et diffuse une indication **DetachUserIndication** pour aviser les autres fournisseurs du changement.

**Tableau 11-19/T.125 – Unité MCSPDU de type DetachUserRequest**

Contenu	Source	Collecteur
Cause	Fournisseur demandeur	Fournisseur supérieur
Identificateurs d'utilisateur	Fournisseur demandeur	Fournisseur supérieur

Une demande MCS-DETACH-USER produit une unité **DetachUserRequest** contenant la cause *demandé par l'utilisateur* et un seul identificateur d'utilisateur.

L'unité **DetachUserRequest** peut aussi être produite par un fournisseur de service MCS lorsqu'une connexion MCS descendante est coupée. A ce point, tous les utilisateurs contenus dans le sous-arbre en question sont perdus et doivent être signalés comme étant détachés avec la cause *domaine déconnecté*. Si des attributions d'identificateur d'utilisateur sont en instance, on peut laisser dans le sous-arbre déconnecté une réponse ultérieure, de type **MergeChannelsConfirm** ou **AttachUserConfirm**, sans adresse de réacheminement à sa source. Un fournisseur confronté à cette situation doit également produire une demande **DetachUserRequest** pour supprimer les identificateurs d'utilisateur dont l'attribution n'est pas possible.

Les fournisseurs qui reçoivent des demandes **DetachUserRequest** doivent valider les identificateurs d'utilisateur contenus dans ces unités pour faire en sorte que ces identificateurs soient légitimement attribués au sous-arbre d'origine. Les identificateurs d'utilisateur non valides doivent être supprimés. S'il ne reste plus d'identificateurs d'utilisateur, il ne sera pas tenu compte d'une demande **DetachUserRequest**.

Les identificateurs d'utilisateur contenus dans l'unité **DetachUserRequest** ne doivent pas être supprimés de la base de données avant qu'un fournisseur reçoive une indication **DetachUserIndication**. Cela assure la cohérence avec le fournisseur de service MCS supérieur.

NOTE – Si plusieurs priorités de données sont mises en œuvre dans un domaine MCS, l'indication **DetachUserIndication** peut arriver chez un fournisseur donné avant les données déjà envoyées par le même utilisateur, mais avec une priorité plus basse. Ce protocole n'empêche pas les données d'être acheminées jusqu'à un point de rattachement, même si cela a lieu après qu'une indication **DetachUserIndication** a signalé que l'expéditeur a été détaché.

## 11.20 DetachUserIndication

Une unité **DetachUserIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une primitive **DetachUserRequest**. Cette unité fait l'objet d'une diffusion en aval à tous les autres fournisseurs. Elle produit des indications MCS-DETACH-USER à tous les rattachements.

A un point de rattachement survivant, l'unité **DetachUserIndication** produit une indication MCS-DETACH-USER pour chaque identificateur d'utilisateur qu'elle contient, que l'utilisateur notifié ait ou non déjà appris l'existence d'un utilisateur détaché.

Dès réception d'une indication **DetachUserIndication** contenant son propre identificateur d'utilisateur, un rattachement MCS cesse d'exister. Tous les canaux auxquels il n'est plus adhérent à la suite du détachement d'un utilisateur doivent être quittés via l'unité **ChannelLeaveRequest**.

**Tableau 11-20/T.125 – Unité MCSPDU de type DetachUserIndication**

Contenu	Source	Collecteur
Cause	Fournisseur supérieur	Indication
Identificateurs d'utilisateur	Fournisseur supérieur	Indication

Les fournisseurs qui reçoivent une indication **DetachUserIndication** doivent supprimer de leur base de données les identificateurs d'utilisateur spécifiés, y compris en tant que gestionnaire ou utilisateur admis d'un canal privé.

Les canaux privés qui étaient gérés par un utilisateur détaché doivent être supprimés s'il n'y a pas d'autres utilisateurs admis. Si d'autres utilisateurs restent attachés, la suppression du gestionnaire aura pour conséquence que le fournisseur supérieur assurera la diffusion multiple d'une indication **ChannelDisbandIndication**. Si l'ensemble des utilisateurs admis à un canal privé demeure vide et que le gestionnaire ne réside pas dans le sous-arbre, l'identificateur de canal doit être supprimé de la base de données. Sinon, en cas de perte d'adhésion à un canal privé par suite d'un détachement d'utilisateurs, un fournisseur doit produire une demande **ChannelLeaveRequest** correspondante.

Un fournisseur qui diffuse une indication **DetachUserIndication** doit calculer, pour chaque canal privé visé et pour chaque sous-arbre de destination, si des éléments subséquents du sous-arbre contiennent des rattachements admis pour le canal privé. Si tel n'est pas le cas, un fournisseur doit conclure que le fournisseur subordonné correspondant n'est plus rattaché au canal privé et doit immédiatement mettre à jour la base de données à cet effet, sans attendre de demande **ChannelLeaveRequest**.

L'état de tous les jetons saisis, en cession ou inhibés par un utilisateur détaché doit être ajusté en conséquence. La suppression d'un destinataire de jeton prévu doit avoir pour conséquence que le fournisseur supérieur produise une confirmation **TokenGiveConfirm** défavorable à destination du donneur de ce jeton, à moins qu'il n'ait libéré le jeton ou qu'il se soit lui-même détaché.

### 11.21 ChannelJoinRequest

L'unité **ChannelJoinRequest** est produite par une demande MCS-CHANNEL-JOIN. Si elle est valide, elle remonte jusqu'à atteindre un fournisseur de service MCS possédant assez d'informations pour émettre une réponse **ChannelJoinConfirm**. Ce fournisseur peut être le fournisseur de service MCS supérieur.

**Tableau 11-21/T.125 – Unité MCSPDU de type ChannelJoinRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur plus élevé
Identificateur de canal	Demande	Fournisseur plus élevé

Le fournisseur de service MCS qui reçoit la primitive de demande fournit l'identificateur d'utilisateur du rattachement MCS initiateur. Les fournisseurs qui recevront la demande **ChannelJoinRequest** par la suite devront valider cet identificateur d'utilisateur pour s'assurer qu'il est légitimement attribué au sous-arbre d'origine. Si l'utilisateur est non valide, il ne doit pas être tenu compte de cette unité MCSPDU.

NOTE – Cette procédure tient compte de la possibilité qu'une unité **ChannelJoinRequest** ascendante croise une unité descendante de purge d'identificateur de l'utilisateur initiateur. Un fournisseur qui reçoit d'abord une unité **PurgeChannelsIndication** peut recevoir peu après une demande **ChannelJoinRequest** contenant un identificateur d'utilisateur non valide. Il s'agit d'une occurrence normale, qui ne justifie pas le rejet de l'unité MCSPDU.

La demande **ChannelJoinRequest** peut remonter jusqu'à un fournisseur de service MCS qui possède, dans sa base de données, l'identificateur de canal demandé. Un tel fournisseur, pour rester en accord avec le fournisseur de service MCS supérieur, décidera si cette demande doit être suivie d'effet. S'il convient que la demande échoue, le fournisseur doit émettre une confirmation **ChannelJoinConfirm** défavorable. S'il convient que la demande soit suivie d'effet et que le fournisseur ait déjà adhéré au même canal, le fournisseur doit émettre une confirmation **ChannelJoinConfirm** favorable. Dans ces deux cas, la demande MCS-CHANNEL-JOIN se réalisera sans nécessairement passer par le fournisseur de service MCS supérieur. Dans le cas contraire, s'il y a lieu que la demande soit suivie d'effet mais que le canal n'a pas encore fait l'objet de l'adhésion correspondante, un fournisseur doit envoyer la demande **ChannelJoinRequest** en amont.

Si la demande **ChannelJoinRequest** remonte jusqu'au fournisseur de service MCS supérieur, l'identificateur de canal demandé peut être zéro: cette information ne se trouve dans aucune base de données, car il s'agit d'un identificateur non valide. Si la limite de domaine imposée au nombre de canaux utilisés le permet, un nouvel identificateur de canal attribué doit être émis et renvoyé dans une unité **ChannelJoinConfirm** favorable. Si l'identificateur de canal demandé se trouve dans l'ensemble des canaux statiques et que la limite de domaine imposée au nombre de canaux utilisés le permet, cet identificateur de canal doit être introduit dans la base de données et être également renvoyé dans une unité **ChannelJoinConfirm** favorable.

Sinon, la demande ne sera suivie d'effet que si l'identificateur de canal se trouve déjà dans la base de données du fournisseur de service MCS supérieur. Un canal d'identificateur d'utilisateur ne peut recevoir d'adhésion que de cet utilisateur. Seuls des utilisateurs préalablement admis par le gestionnaire d'un identificateur de canal privé peuvent adhérer à celui-ci. Tout utilisateur peut adhérer à un identificateur de canal attribué.

## 11.22 ChannelJoinConfirm

L'unité **ChannelJoinConfirm** est produite chez un fournisseur de service MCS de niveau plus élevé dès la réception de la demande **ChannelJoinRequest**. Renvoyée au fournisseur demandeur, elle produit une confirmation MCS-CHANNEL-JOIN.

**Tableau 11-22/T.125 – Unité MCSPDU de type ChannelJoinConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur plus élevé	Confirmation
Initiateur	Fournisseur plus élevé	Acheminement d'unité MCSPDU
Demandé	Fournisseur plus élevé	Confirmation
Identificateur de canal (sur option)	Fournisseur plus élevé	Confirmation

L'unité **ChannelJoinConfirm** ne contient un identificateur de canal d'adhésion que si et seulement si le résultat est favorable.

La demande d'identificateur de canal est la même que dans l'unité **ChannelJoinRequest**. Cela aide le rattachement initiateur à établir une relation entre une demande antérieure et une confirmation MCS-CHANNEL-JOIN. Etant donné que l'unité **ChannelJoinRequest** n'a pas besoin de remonter jusqu'au fournisseur supérieur, les confirmations peuvent se produire dans le désordre.

Si le résultat est favorable, l'unité **ChannelJoinConfirm** opère l'adhésion du fournisseur de service MCS récepteur au canal spécifié. Ensuite, les fournisseurs plus élevés doivent acheminer vers ce récepteur toutes données envoyées par des utilisateurs sur ce canal. Un fournisseur doit maintenir son adhésion à un canal tant qu'un de ses rattachements ou de ses fournisseurs subordonnés la conserve. Pour sortir du canal, un fournisseur doit émettre une demande **ChannelLeaveRequest**.

Les fournisseurs qui reçoivent une confirmation **ChannelJoinConfirm** favorable doivent introduire dans leur base de données l'identificateur du canal. Si celui-là ne s'y trouve pas déjà, il doit être affecté au type statique ou attribué, selon sa catégorie.

L'unité **ChannelJoinConfirm** doit être envoyée en direction de l'identificateur d'utilisateur initiateur. Si cet identificateur ne peut pas être atteint parce qu'il n'existe plus de connexion MCS, le fournisseur doit décider s'il est justifié que cet identificateur conserve son adhésion au canal. Si ce n'est pas le cas, il doit émettre une demande de sortie de canal, **ChannelLeaveRequest**.

### 11.23 ChannelLeaveRequest

L'unité **ChannelLeaveRequest** est produite par un fournisseur de service MCS qui souhaite se retirer d'un ensemble de canaux. La cause peut être une demande de sortie de canal MCS-CHANNEL-LEAVE issue du dernier rattachement ayant adhéré à un canal. La demande **ChannelLeaveRequest** continue à remonter dans la hiérarchie si des fournisseurs plus élevés perdent aussi, par voie de conséquence, leur raison de faire l'objet d'une adhésion.

Les fournisseurs qui reçoivent l'unité **ChannelLeaveRequest** doivent arrêter d'acheminer, vers la connexion MCS qui l'a transportée, toute donnée envoyée par des utilisateurs sur les canaux spécifiés. Lorsque le dernier rattachement ou le dernier fournisseur subordonné sort d'un canal, un fournisseur de service MCS doit émettre une demande **ChannelLeaveRequest** correspondante.

**Tableau 11-23/T.125 – Unité MCSPDU de type ChannelLeaveRequest**

Contenu	Source	Collecteur
Identificateurs de canal	Fournisseur demandeur	Fournisseur plus élevé

### 11.24 ChannelConveneRequest

L'unité **ChannelConveneRequest** est produite par une demande MCS-CHANNEL-CONVENE. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui retourne une réponse **ChannelConveneConfirm**. Si la limite de domaine imposée au nombre d'identificateurs de canal le permet, un nouvel identificateur de canal privé est produit.

L'unité **ChannelConveneRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour la demande **ChannelJoinRequest**.

Le demandeur devient gestionnaire du canal privé. Initialement, aucun utilisateur n'a adhéré à ce canal et son gestionnaire y est le seul utilisateur admis.

**Tableau 11-24/T.125 – Unité MCSPDU de type ChannelConveneRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur

### 11.25 ChannelConveneConfirm

L'unité **ChannelConveneConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de la demande **ChannelConveneRequest**. Renvoyée vers le fournisseur demandeur, cette unité produit une confirmation MCS-CHANNEL-CONVENE.

**Tableau 11-25/T.125 – Unité MCSPDU de type ChannelConveneConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur	Fournisseur supérieur	Acheminement d'unité MCSPDU
Identificateur de canal (sur option)	Fournisseur supérieur	Confirmation

L'unité **ChannelConveneConfirm** contient un identificateur de canal privé si et seulement si le résultat confirmé est favorable.

Les fournisseurs qui reçoivent une confirmation **ChannelConveneConfirm** favorable doivent introduire l'identificateur de canal dans leur base de données en tant que canal privé ayant comme gestionnaire l'identificateur de l'utilisateur initiateur.

L'unité **ChannelConveneConfirm** doit être envoyée en direction de l'identificateur d'utilisateur initiateur. Si cet identificateur ne peut pas être atteint parce qu'il n'existe plus de connexion MCS, il n'est besoin d'aucune action particulière car l'unité **DetachUserIndication** doit toujours arriver ultérieurement pour signaler que l'initiateur s'est détaché. Comme celui-ci est le gestionnaire du canal, l'identificateur de ce canal sera supprimé de la base de données.

### 11.26 ChannelDisbandRequest

L'unité **ChannelDisbandRequest** est produite par une demande MCS-CHANNEL-DISBAND. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui supprime l'identificateur du canal privé et produit une indication **ChannelDisbandIndication**.

**Tableau 11-26/T.125 – Unité MCSPDU de type ChannelDisbandRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de canal	Demande	Fournisseur supérieur

L'unité **ChannelDisbandRequest** peut également être produite à la propre initiative d'un fournisseur de service MCS pour dissoudre un canal.

L'unité **ChannelDisbandRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé pour veiller à ce qu'il soit légitimement attribué au sous-arbre d'origine. Si l'initiateur ne correspond pas au gestionnaire du canal privé, tel qu'enregistré dans la base de données, il ne doit pas être tenu compte de l'unité MCSPDU.

### 11.27 ChannelDisbandIndication

L'unité **ChannelDisbandIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception de la demande **ChannelDisbandRequest**. Elle fait l'objet d'une diffusion multipoint en aval aux fournisseurs qui contiennent dans leur sous-arbre le gestionnaire ou un utilisateur admis. Elle produit des indications MCS-CHANNEL-EXPEL qui sont envoyées aux utilisateurs admis avec la cause *canal dissous*.

**Tableau 11-27/T.125 – Unité MCSPDU de type ChannelDisbandIndication**

Contenu	Source	Collecteur
Identificateur de canal	Fournisseur supérieur	Indication

L'unité **ChannelDisbandIndication** doit aussi être produite par le fournisseur de service MCS supérieur lorsque le gestionnaire d'un canal privé est détaché.

Les fournisseurs qui reçoivent l'unité **ChannelDisbandIndication** doivent supprimer le canal de leur base de données.

### 11.28 ChannelAdmitRequest

L'unité **ChannelAdmitRequest** est produite par une demande MCS-CHANNEL-ADMIT. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui admettra les utilisateurs spécifiés dans le canal privé et assurera la diffusion multipoint de l'indication **ChannelAdmitIndication** pour aviser les fournisseurs dans le sous-arbre desquels ces utilisateurs résident.

L'unité **ChannelAdmitRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelDisbandRequest**.

Les autres identificateurs d'utilisateur contenus dans l'unité **ChannelAdmitRequest**, représentant des utilisateurs à admettre, doivent être validés au niveau du fournisseur de service MCS supérieur qui est le seul à avoir connaissance de l'entière population des utilisateurs. Les identificateurs non valides doivent être omis de l'indication **ChannelAdmitIndication** résultante.

**Tableau 11-28/T.125 – Unité MCSPDU de type ChannelAdmitRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de canal	Demande	Fournisseur supérieur
Identificateurs d'utilisateur	Demande	Fournisseur supérieur

Les identificateurs d'utilisateur contenus dans l'unité **ChannelAdmitRequest** ne doivent pas être admis dans le canal privé avant qu'un fournisseur ait reçu l'indication **ChannelAdmitIndication**. Cela assure la cohérence avec le fournisseur de service MCS supérieur.

### 11.29 ChannelAdmitIndication

L'unité **ChannelAdmitIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception de la demande **ChannelAdmitRequest**. Elle est diffusée en aval vers les fournisseurs qui contiennent dans leur sous-arbre un utilisateur nouvellement admis. Elle produit des indications MCS-CHANNEL-ADMIT aux rattachements touchés.

Les fournisseurs qui reçoivent l'indication **ChannelAdmitIndication** doivent normalement mettre à jour le canal dans leur base de données pour admettre les utilisateurs spécifiés qui résident dans leur sous-arbre. Si toutefois un fournisseur est le précédent supérieur d'un domaine inférieur en cours de fusion à la suite d'une demande MCS-CONNECT-PROVIDER, ce fournisseur peut refuser l'admission en émettant la demande **DetachUserRequest** pour les identificateurs d'utilisateur touchés, avec la raison *canal purgé*.

**Tableau 11-29/T.125 – Unité MCSPDU de type ChannelAdmitIndication**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Indication
Identificateur de canal	Fournisseur supérieur	Indication
Identificateurs d'utilisateur	Fournisseur supérieur	Acheminement d'unité MCSPDU

### 11.30 ChannelExpelRequest

L'unité **ChannelExpelRequest** est produite par une demande MCS-CHANNEL-EXPEL. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui expulsera les utilisateurs spécifiés du canal privé et multidiffusera une indication **ChannelExpelIndication** pour aviser les fournisseurs dans le sous-arbre desquels ces utilisateurs résident.

L'unité **ChannelExpelRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelDisbandRequest**.

Les autres identificateurs d'utilisateur contenus dans l'unité **ChannelExpelRequest**, représentant des utilisateurs à expulser, doivent être validés au niveau du fournisseur de service MCS supérieur, qui est le seul à connaître l'ensemble complet des utilisateurs admis. Ceux qui n'ont pas été admis ne sont pas insérés dans l'indication **ChannelExpelIndication** résultante.

Les identificateurs d'utilisateur contenus dans l'unité **ChannelExpelRequest** ne doivent pas être expulsés du canal privé avant qu'un fournisseur ait reçu une indication **ChannelExpelIndication**. Cela assure la cohérence avec le fournisseur de service MCS supérieur.

**Tableau 11-30/T.125 – Unité MCSPDU de type ChannelExpelRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de canal	Demande	Fournisseur supérieur
Identificateurs d'utilisateur	Demande	Fournisseur supérieur

### 11.31 ChannelExpelIndication

L'unité **ChannelExpelIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une unité **ChannelExpelRequest**. Elle est multidiffusée vers les fournisseurs dont le sous-arbre contient un utilisateur expulsé. Elle produit des indications MCS-CHANNEL-EXPEL aux rattachements touchés, avec la cause *demandé par l'utilisateur*.

**Tableau 11-31/T.125 – Unité MCSPDU de type ChannelExpelIndication**

Contenu	Source	Collecteur
Identificateur de canal	Fournisseur supérieur	Indication
Identificateurs d'utilisateur	Fournisseur supérieur	Acheminement d'unité MCSPDU

Les fournisseurs qui reçoivent l'indication **ChannelExpelIndication** doivent mettre à jour le canal enregistré dans leur base de données en supprimant les utilisateurs spécifiés de l'ensemble des utilisateurs admis dans ce canal. Si l'ensemble des utilisateurs admis dans un canal privé devient vide et que le gestionnaire ne réside pas dans le sous-arbre, l'identificateur de canal doit être supprimé de la base de données. Sinon, si le canal ne fait plus l'objet d'adhésions à la suite d'expulsions, un fournisseur doit émettre une demande **ChannelLeaveRequest** correspondante.

Un fournisseur qui expédie une unité **ChannelExpelIndication** doit calculer, pour chaque sous-arbre de destination, si celui-ci contiendra, après l'opération, des rattachements admis dans le canal privé. S'il n'y en a pas, le fournisseur en conclut que le fournisseur subordonné correspondant n'adhère plus au canal privé. Il doit mettre immédiatement à jour sa base de données en conséquence, sans attendre de demande **ChannelLeaveRequest**.

### **11.32 SendDataRequest**

L'unité **SendDataRequest** est produite par une demande MCS-SEND-DATA. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur. En cours de route, des fournisseurs peuvent en tirer une indication **SendDataIndication** de contenu identique, qu'ils multidiffuseront en aval.

L'unité **SendDataRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si l'identificateur de canal est répertorié dans la base de données du fournisseur de service MCS récepteur en tant que canal privé et que l'initiateur de l'unité **SendDataRequest** n'est pas un utilisateur admis, il ne doit pas être tenu compte de l'unité MCSPDU.

L'unité **SendDataRequest** contient le niveau de fiabilité qui était spécifié par la demande MCS-SEND-DATA originale. Si les données ne sont pas fiables, l'identificateur de référence de domaine est également spécifié.

La connexion de transport initiale ou additionnelle qui achemine la demande **SendDataRequest** doit correspondre à sa priorité de transfert de données, compte tenu du nombre de priorités mises en œuvre dans le domaine. Les unités MCSPDU arrivant sur une connexion MCS par une connexion de transport erronée doivent être rejetées.

**Tableau 11-32/T.125 – Unité MCSPDU de type SendDataRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur plus élevé
Identificateur de canal	Demande	Fournisseur plus élevé
Fiabilité	Demande	Fournisseur plus élevé
Identificateur de référence de domaine	Demande	Fournisseur plus élevé
Priorité des données	Demande	Fournisseur plus élevé
Segmentation	Fournisseur demandeur	Fournisseur plus élevé
Longueur totale des données	Fournisseur demandeur	Fournisseur plus élevé
Données d'utilisateur	Demande	Fournisseur plus élevé

Les fanions de segmentation *début* et *fin* doivent être insérés par un fournisseur pour montrer la relation entre les données d'utilisateur contenues dans l'unité **SendDataRequest** et les frontières d'une unité de données du service MCS. Les fournisseurs ont la possibilité de segmenter et de réassembler des unités MCSPDU qui font partie de la même unité de données du service MCS, du moment que ces opérations n'ont pas d'incidence sur l'intégrité des données d'utilisateur. Une telle manipulation ne devrait cependant apporter que peu d'avantages, car la longueur maximale d'une unité MCSPDU est constante dans l'ensemble d'un domaine.

Pour une unité **SendDataRequest** donnée, si le fanion de segmentation *début* est au niveau *true* et que le fanion *fin* est au niveau *false*, la longueur totale des données segmentées de l'utilisateur sera fournie. Ce champ est uniquement présent si les nœuds de source et de collecteur font tous deux partie du sommet V3.

Un fournisseur doit générer à partir d'une demande **SendDataRequest** une indication **SendDataIndication** ayant le même contenu, puis la transmettre à tous les fournisseurs qui ont adhéré au canal spécifié, à l'exception du fournisseur subordonné qui a transmis la demande **SendDataRequest** en amont. Il doit aussi transmettre la demande **SendDataRequest** en amont, à moins que le canal ne soit répertorié dans sa base de données comme étant un identificateur d'utilisateur résidant dans son sous-arbre.

### 11.33 SendDataIndication

L'unité **SendDataIndication** est produite par un fournisseur de service MCS de niveau plus élevé dès réception d'une unité **SendDataRequest**. Elle est multidiffusée en aval et produit des indications MCS-SEND-DATA à tous les rattachements qui ont adhéré au canal.

**Tableau 11-33/T.125 – Unité MCSPDU de type SendDataIndication**

Contenu	Source	Collecteur
Initiateur	Fournisseur plus élevé	Indication
Identificateur de canal	Fournisseur plus élevé	Indication
Fiabilité	Fournisseur plus élevé	Indication
Identificateur de référence de domaine	Fournisseur plus élevé	Indication
Priorité des données	Fournisseur plus élevé	Indication
Segmentation	Fournisseur plus élevé	Fournisseur indicateur
Longueur totale des données	Fournisseur plus élevé	Fournisseur indicateur
Données d'utilisateur	Fournisseur plus élevé	Indication

La connexion de transport initiale ou additionnelle qui achemine l'indication **SendDataIndication** doit correspondre à sa priorité de transfert de données, compte tenu du nombre de priorités mises en œuvre dans un domaine. Les unités MCSPDU arrivant sur une connexion MCS par une connexion de transport erronée doivent être rejetées.

Les fanions de segmentation *début* et *fin* permettent de réassembler les données d'utilisateur pour constituer une unité complète de données du service MCS. Ces fanions doivent être interprétés dans le contexte d'unités MCSPDU **SendDataIndication** provenant du même utilisateur sur le même canal et avec la même priorité. Un train de fragments à réassembler peut être entrelacé avec d'autres unités MCSPDU et avec des données issues d'autres utilisateurs sur des canaux ayant d'autres priorités.

Pour une unité **SendDataIndication** donnée, si le fanion de segmentation *début* est au niveau *true* et que le fanion *fin* est au niveau *false*, la longueur totale des données segmentées de l'utilisateur sera fournie. Ce champ est uniquement présent si les nœuds de source et de collecteur font tous deux partie du sommet V3.

La manière dont les unités de données du service sont indiquées à des utilisateurs MCS rattachés relève d'une décision de mise en œuvre locale. Une possibilité consiste à remettre chaque unité MCSPDU sous forme d'une unité de données d'interface distincte, fanions de segmentation inclus. D'autres solutions, visant à effectuer le réassemblage au niveau du fournisseur récepteur, doivent normalement tenir compte de grandes unités de données du service et de l'ordre relatif dans lequel ces unités commencent à arriver.

Les fournisseurs qui reçoivent l'unité **SendDataIndication** doivent l'expédier à tous les subordonnés qui ont adhéré au canal.

### **11.34 UniformSendDataRequest**

L'unité **UniformSendDataRequest** est produite par une demande MCS-UNIFORM-SEND-DATA. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS, qui en tirera une unité **UniformSendDataIndication** de contenu identique et qui la multidiffusera en aval.

**Tableau 11-34/T.125 – Unité MCSPDU de type UniformSendDataRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de canal	Demande	Fournisseur supérieur
Fiabilité	Demande	Fournisseur supérieur
Identificateur de référence de domaine	Demande	Fournisseur supérieur
Priorité des données	Demande	Fournisseur supérieur
Segmentation	Fournisseur demandeur	Fournisseur supérieur
Longueur totale des données	Fournisseur demandeur	Fournisseur supérieur
Données d'utilisateur	Demande	Fournisseur supérieur

L'unité **UniformSendDataRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si l'identificateur de canal est répertorié dans la base de données du fournisseur de service MCS récepteur en tant que canal privé et que l'initiateur de la demande **UniformSendDataRequest** n'est pas un utilisateur admis, il ne doit pas être tenu compte de l'unité MCSPDU.

La connexion de transport initiale ou additionnelle qui achemine l'unité **UniformSendDataRequest** doit être adaptée à sa priorité de données, compte tenu du nombre de priorités mises en œuvre dans le domaine. Les unités MCSPDU arrivant sur une connexion MCS par une connexion de transport erronée doivent être rejetées.

Les fanions de segmentation *début* et *fin* doivent être insérés par un fournisseur pour montrer la relation entre les données d'utilisateur contenues dans l'unité **UniformSendDataRequest** et les frontières d'une unité de données du service MCS. Les fournisseurs ont la possibilité de segmenter et de réassembler des unités MCSPDU qui font partie de la même unité de données du service MCS, du moment que ces opérations n'ont pas d'incidence sur l'intégrité des données d'utilisateur. Une telle manipulation ne devrait cependant apporter que peu d'avantages, car la longueur maximale d'une unité MCSPDU est constante dans l'ensemble d'un domaine.

Pour une unité **UniformSendDataRequest** donnée, si le fanion de segmentation *début* est au niveau *true* et que le fanion *fin* est au niveau *false*, la longueur totale des données segmentées de l'utilisateur sera fournie. Ce champ est uniquement présent si les nœuds de source et de collecteur font tous deux partie du sommet V3.

Le fournisseur de service MCS supérieur doit tirer de l'unité **UniformSendDataRequest** une unité **UniformSendDataIndication** ayant le même contenu.

### 11.35 UniformSendDataIndication

L'unité **UniformSendDataIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une demande **UniformSendDataRequest**. Elle est multidiffusée en aval et produit des indications MCS-UNIFORM-SEND-DATA à tous les rattachements qui ont adhéré au canal.

**Tableau 11-35/T.125 – Unité MCSPDU de type UniformSendDataIndication**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Indication
Identificateur de canal	Fournisseur supérieur	Indication
Fiabilité	Fournisseur supérieur	Indication
Identificateur de référence de domaine	Fournisseur supérieur	Indication
Priorité des données	Fournisseur supérieur	Indication
Segmentation	Fournisseur supérieur	Fournisseur indicateur
Longueur totale des données	Fournisseur supérieur	Fournisseur indicateur
Données d'utilisateur	Fournisseur supérieur	Indication

La connexion de transport initiale ou additionnelle qui achemine l'indication **UniformSendDataIndication** doit correspondre à sa priorité de données, compte tenu du nombre de priorités mises en œuvre dans un domaine. Les unités MCSPDU arrivant sur une connexion MCS par une connexion de transport erronée doivent être rejetées.

Les fanions de segmentation *début* et *fin* permettent de réassembler les données d'utilisateur pour constituer une unité complète de données du service MCS. Ces fanions doivent être interprétés dans le contexte d'unités MCSPDU d'indication **UniformSendDataIndication** provenant du même utilisateur sur le même canal et avec la même priorité. Un train de fragments à réassembler peut être entrelacé avec d'autres unités MCSPDU et avec des données issues d'autres utilisateurs sur d'autres canaux ayant d'autres priorités.

Pour une unité **UniformSendDataIndication** donnée, si le fanion de segmentation *début* est au niveau *true* et que le fanion *fin* est au niveau *false*, la longueur totale des données segmentées de l'utilisateur sera fournie. Ce champ est uniquement présent si les nœuds de source et de collecteur font tous deux partie du sommet V3.

La manière dont les unités de données du service sont indiquées aux utilisateurs MCS rattachés relève d'une décision de mise en œuvre locale.

Les fournisseurs qui reçoivent l'unité **UniformSendDataIndication** doivent l'expédier à tous les subordonnés qui ont adhéré au canal.

### 11.36 TokenGrabRequest

L'unité **TokenGrabRequest** est produite par une demande MCS-TOKEN-GRAB. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur qui retourne une réponse **TokenGrabConfirm**.

**Tableau 11-36/T.125 – Unité MCSPDU de type TokenGrabRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur

L'unité **TokenGrabRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si le jeton est libre et que la limite de domaine quant au nombre de jetons utilisés le permet, ce jeton doit passer à l'état saisi. Si le jeton n'est inhibé que par l'utilisateur demandeur, il doit passer à l'état saisi. Sinon, l'état du jeton ne doit pas changer.

### 11.37 TokenGrabConfirm

L'unité **TokenGrabConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de l'unité **TokenGrabRequest**. Réacheminée vers le fournisseur demandeur, elle produit une confirmation MCS-TOKEN-GRAB.

**Tableau 11-37/T.125 – Unité MCSPDU de type TokenGrabConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur	Fournisseur supérieur	Acheminement d'unité MCSPDU
Identificateur de jeton	Fournisseur supérieur	Confirmation
Etat du jeton	Fournisseur supérieur	Fournisseurs intermédiaires

Le résultat doit être *favorable* si le jeton était préalablement libre ou s'il est le résultat d'une conversion de l'état inhibé à l'état saisi par le même utilisateur. Les autres résultats auront les valeurs *jetons trop nombreux* et *jeton non disponible*. Ce dernier cas s'applique à un jeton déjà saisi par le demandeur, ce qui peut être détecté par examen de l'état du jeton.

Les fournisseurs qui reçoivent l'unité **TokenGrabConfirm** doivent mettre à jour l'état du jeton dans leur base de données en fonction de l'état signalé.

L'unité **TokenGrabConfirm** doit être envoyée en direction de l'identificateur d'utilisateur initiateur. Si cet identificateur ne peut pas être atteint parce qu'il n'existe plus de connexion MCS, il n'est besoin d'aucune action particulière, car l'unité **DetachUserIndication** doit toujours arriver ultérieurement pour signaler que l'initiateur s'est détaché. Cette opération libérera l'identificateur de jeton dans la base de données.

### 11.38 TokenInhibitRequest

L'unité **TokenInhibitRequest** est produite par une demande MCS-TOKEN-INHIBIT. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur qui retourne une réponse **TokenInhibitConfirm**.

**Tableau 11-38/T.125 – Unité MCSPDU de type TokenInhibitRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur

L'unité **TokenInhibitRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si le jeton est libre et que la limite de domaine quant au nombre de jetons utilisé le permet, ce jeton doit passer à l'état inhibé. Si le jeton est saisi par l'utilisateur demandeur, il doit devenir inhibé. Si le jeton est déjà inhibé, le demandeur doit être ajouté à l'ensemble des inhibiteurs. Sinon, l'état du jeton ne doit pas changer.

### 11.39 TokenInhibitConfirm

L'unité **TokenInhibitConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de l'unité **TokenInhibitRequest**. Réacheminée vers le fournisseur demandeur, elle produit une confirmation MCS-TOKEN-INHIBIT.

Le résultat doit être *favorable* si le jeton était préalablement libre ou inhibé ou s'il est le résultat d'une conversion de l'état saisi à l'état inhibé par le même utilisateur. Les autres résultats auront les valeurs *jetons trop nombreux* et *jeton non disponible*.

**Tableau 11-39/T.125 – Unité MCSPDU de type TokenInhibitConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur	Fournisseur supérieur	Acheminement d'unité MCSPDU
Identificateur de jeton	Fournisseur supérieur	Confirmation
Etat du jeton	Fournisseur supérieur	Fournisseurs intermédiaires

Les fournisseurs qui reçoivent l'unité **TokenInhibitConfirm** doivent mettre à jour l'état du jeton dans leur base de données en fonction de l'état signalé.

Cette unité MCSPDU est acheminée de la même manière que l'unité **TokenGrabConfirm**.

### 11.40 TokenGiveRequest

L'unité **TokenGiveRequest** est produite par une demande MCS-TOKEN-GIVE. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui retourne une indication **TokenGiveIndication** ou une confirmation **TokenGiveConfirm** défavorable.

**Tableau 11-40/T.125 – Unité MCSPDU de type TokenGiveRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur
Destinataire	Demande	Fournisseur supérieur

L'unité **TokenGiveRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si le jeton est saisi par le demandeur et que le destinataire prévu existe, l'unité **TokenGiveIndication** doit être transmise au destinataire. Sinon, la demande échoue, l'état du jeton ne change pas et une confirmation **TokenGiveConfirm** est envoyée au demandeur avec le résultat *jeton non détenu* ou *utilisateur inconnu*.

### 11.41 TokenGiveIndication

L'unité **TokenGiveIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une unité **TokenGiveRequest**. Acheminée jusqu'au destinataire prévu, elle produit une indication MCS-TOKEN-GIVE.

Les fournisseurs qui reçoivent l'unité **TokenGiveIndication** doivent normalement mettre à jour l'identificateur de jeton dans leur base de données pour indiquer qu'il est en phase de cession

d'initiateur à destinataire. Mais si un fournisseur est le précédent supérieur d'un domaine inférieur en cours de fusion à la suite d'une demande MCS-CONNECT-PROVIDER, ce fournisseur peut rejeter le jeton offert en émettant une réponse **TokenGiveResponse** avec la cause *fusion de domaines*.

**Tableau 11-41/T.125 – Unité MCSPDU de type TokenGiveIndication**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Indication
Identificateur de jeton	Fournisseur supérieur	Indication
Destinataire	Fournisseur supérieur	Acheminement d'unité MCSPDU

L'unité **TokenGiveIndication** doit être envoyée en direction de l'identificateur d'utilisateur destinataire. Si cet identificateur ne peut pas être atteint parce qu'il n'existe plus de connexion MCS, il n'est besoin d'aucune action particulière car l'unité **DetachUserIndication** doit arriver ultérieurement pour signaler que le destinataire s'est détaché. Cette opération libérera l'identificateur de jeton dans la base de données.

#### 11.42 TokenGiveResponse

L'unité **TokenGiveResponse** est produite par une réponse MCS-TOKEN-GIVE. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui produit une unité **TokenGiveConfirm** afin d'informer le donneur de jeton au sujet de l'issue.

**Tableau 11-42/T.125 – Unité MCSPDU de type TokenGiveResponse**

Contenu	Source	Collecteur
Résultat	Réponse	Fournisseur supérieur
Destinataire	Fournisseur répondeur	Fournisseur supérieur
Identificateur de jeton	Fournisseur répondeur	Fournisseur supérieur

Un résultat de type *favorable* signifie que le destinataire accepte le jeton proposé.

L'identificateur d'utilisateur du rattachement MCS répondeur est communiqué par le fournisseur de service MCS qui reçoit la primitive de réponse. Les fournisseurs qui recevront des réponses **TokenGiveResponse** ultérieures devront valider l'identificateur d'utilisateur pour faire en sorte qu'il soit légitimement attribué au sous-arbre d'origine. Si l'identificateur d'utilisateur n'est pas valide, il ne doit pas être tenu compte de l'unité MCSPDU.

Si l'identificateur de jeton n'est pas indiqué dans la base de données du fournisseur comme étant transmis au destinataire, il ne doit pas être tenu compte de l'unité MCSPDU. Si l'identificateur de jeton est encore saisi par le donneur, son état doit être mis à jour pour indiquer qu'il est saisi par le destinataire lorsque le résultat est favorable; sinon, l'état doit revenir à *saisi par le donneur* ou être supprimé de la base de données, selon que le donneur réside ou non dans le sous-arbre du fournisseur. Si l'identificateur de jeton a entre-temps été libéré par le donneur et que le résultat n'est pas favorable, ce jeton doit être supprimé de la base de données du fournisseur.

Si l'unité MCSPDU n'est pas non valide et qu'il en est tenu compte, elle doit être expédiée en amont. Le fournisseur de service MCS supérieur doit donner suite à l'unité **TokenGiveResponse** comme spécifié ci-dessus. De plus, si le donneur n'a pas déjà libéré le jeton, le fournisseur supérieur doit émettre une confirmation **TokenGiveConfirm** contenant le même résultat que l'unité **TokenGiveResponse**.

### 11.43 TokenGiveConfirm

L'unité **TokenGiveConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de l'unité **TokenGiveResponse**. Réacheminée vers le fournisseur demandeur, elle produit une confirmation MCS-TOKEN-GIVE.

**Tableau 11-43/T.125 – Unité MCSPDU de type TokenGiveConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur	Fournisseur supérieur	Acheminement d'unité MCSPDU
Identificateur de jeton	Fournisseur supérieur	Confirmation
Etat du jeton	Fournisseur supérieur	Fournisseurs intermédiaires

L'unité **TokenGiveConfirm** doit également être produite par le fournisseur de service MCS supérieur dès réception d'une unité **TokenGiveRequest** si un jeton ne peut pas être offert au destinataire prévu. Cette opération remplace celle de production d'une unité **TokenGiveIndication**. Une unité **TokenGiveConfirm** doit également être produite avec le résultat *utilisateur inconnu* si le destinataire est détaché avant d'avoir reçu l'unité **TokenGiveResponse**.

Les fournisseurs qui reçoivent l'unité **TokenGiveConfirm** doivent mettre à jour l'état du jeton dans leur base de données en fonction de l'état signalé.

Cette unité MCSPDU est acheminée de la même manière que l'unité **TokenGrabConfirm**.

### 11.44 TokenPleaseRequest

L'unité **TokenPleaseRequest** est produite par une demande MCS-TOKEN-PLEASE. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui multidiffuse une unité **TokenPleaseIndication** afin d'alerter les utilisateurs actuels du jeton.

L'unité **TokenPleaseRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

**Tableau 11-44/T.125 – Unité MCSPDU de type TokenPleaseRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur

### 11.45 TokenPleaseIndication

L'unité **TokenPleaseIndication** est produite au niveau du fournisseur de service MCS supérieur dès réception d'une unité **TokenPleaseRequest**. Elle est multidiffusée en aval et produit des indications MCS-TOKEN-PLEASE.

Les fournisseurs qui reçoivent l'unité **TokenPleaseIndication** doivent l'envoyer à tous les subordonnés dont le sous-arbre contient un utilisateur qui a saisi, a inhibé ou va recevoir le jeton spécifié.

**Tableau 11-45/T.125 – Unité MCSPDU de type TokenPleaseIndication**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Indication
Identificateur de jeton	Fournisseur supérieur	Indication

#### 11.46 TokenReleaseRequest

L'unité **TokenReleaseRequest** est produite par une demande MCS-TOKEN-RELEASE. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui retourne en réponse une réponse **TokenReleaseConfirm**.

**Tableau 11-46/T.125 – Unité MCSPDU de type TokenReleaseRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur

L'unité **TokenReleaseRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

Si le jeton est saisi par le demandeur, il doit devenir libre. S'il est inhibé, le demandeur doit être supprimé de l'ensemble des inhibiteurs; si cet ensemble devient vide, le jeton devient libre. Si le jeton est en cours de cession par le demandeur, il doit prendre un état intermédiaire distinct: *donné au destinataire prévu*, en attendant la réception d'une réponse **TokenGiveResponse**. Sinon, l'état du jeton ne doit pas changer.

#### 11.47 TokenReleaseConfirm

L'unité **TokenReleaseConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de l'unité **TokenReleaseRequest**. Réacheminée vers le fournisseur demandeur, elle produit une confirmation MCS-TOKEN-RELEASE.

**Tableau 11-47/T.125 – Unité MCSPDU de type TokenReleaseConfirm**

Contenu	Source	Collecteur
Résultat	Fournisseur supérieur	Confirmation
Initiateur	Fournisseur supérieur	Acheminement d'unité MCSPDU
Identificateur de jeton	Fournisseur supérieur	Confirmation
Etat du jeton	Fournisseur supérieur	Fournisseurs intermédiaires

Le résultat doit être *favorable* si le jeton a été saisi ou inhibé par le demandeur ou si celui-ci était en train de le donner. L'autre résultat possible est *jeton non détenu*.

Les fournisseurs qui reçoivent l'unité **TokenReleaseConfirm** doivent mettre à jour l'état du jeton dans leur base de données en fonction de l'état signalé.

Cette unité MCSPDU est acheminée de la même manière que l'unité **TokenGrabConfirm**.

### 11.48 TokenTestRequest

L'unité **TokenTestRequest** est produite par une demande MCS-TOKEN-TEST. Si elle est valide, elle remonte jusqu'au fournisseur de service MCS supérieur, qui retourne une réponse **TokenTestConfirm**.

L'unité **TokenTestRequest** contient l'identificateur d'utilisateur initiateur, qui doit être validé comme expliqué pour l'unité **ChannelJoinRequest**.

**Tableau 11-48/T.125 – Unité MCSPDU de type TokenTestRequest**

Contenu	Source	Collecteur
Initiateur	Fournisseur demandeur	Fournisseur supérieur
Identificateur de jeton	Demande	Fournisseur supérieur

### 11.49 TokenTestConfirm

L'unité **TokenTestConfirm** est produite au niveau du fournisseur de service MCS supérieur dès réception de l'unité **TokenTestRequest**. Réacheminée vers le fournisseur demandeur, elle produit une confirmation MCS-TOKEN-TEST.

Les fournisseurs qui reçoivent l'unité **TokenTestConfirm** devraient constater que l'état du jeton dans leur base de données est conforme à l'état signalé.

Cette unité MCSPDU est acheminée de la même manière que l'unité **TokenGrabConfirm**.

**Tableau 11-49/T.125 – Unité MCSPDU de type TokenTestConfirm**

Contenu	Source	Collecteur
Initiateur	Fournisseur supérieur	Unité MCSPDU
Identificateur de jeton	Fournisseur supérieur	Confirmation
Etat du jeton	Fournisseur supérieur	Confirmation

### 11.50 CapabilitiesNotificationRequest

L'unité **CapabilitiesNotificationRequest** est une unité PDU ascendante qui est utilisée pour avertir les nœuds supérieurs de la hiérarchie des modifications désirées à la liste des capacités du sommet ou pour avertir le sommet d'un changement d'état des connexions du nœud V2. Il arrive parfois qu'il soit transmis par des nœuds intermédiaires qui détectent la perte d'une connexion descendante. Tout nœud qui détecte l'établissement ou la perte d'une connexion avec un nœud V2 peut envoyer cette unité PDU.

**Tableau 11-50/T.125 – Unité MCSPDU de type CapabilitiesNotificationRequest**

Contenu	Source	Collecteur
Nœud V2 présent	Fournisseur demandeur	Fournisseur supérieur
Liste d'ajout	Fournisseur demandeur	Fournisseur supérieur
Liste de suppression	Fournisseur demandeur	Fournisseur supérieur

Le contenu de l'unité **CapabilitiesNotificationRequest** est le suivant:

*nœud V2 présent*: ce fanion est remis à 1 lorsqu'un nœud V2 est connecté au sommet V3. A ce moment, ce fanion est réinitialisé du fait que le nœud V2 se déconnecte;

*liste d'ajout*: il s'agit d'une liste des capacités de demande (décrites au 11.50.1) dont un nœud donné demande l'ajout à la liste des capacités du sommet. Ce champ est facultatif du fait que certaines unités **CapabilitiesNotificationRequest** produites servent à ajouter des capacités à la liste;

*liste de suppression*: il s'agit d'une liste des capacités de demande dont un nœud donné demande la suppression de la liste des capacités du sommet. Ce champ est facultatif du fait que certaines unités **CapabilitiesNotificationRequest** produites servent à supprimer des capacités de la liste.

### 11.50.1 Capacité de demande

Le tableau ci-dessous résume le contenu de la capacité de demande. Les unités de liste d'ajout et de liste de suppression d'une unité PDU **CapabilitiesNotificationRequest** sont constituées de capacités de demande.

Identificateur de capacité
Classe de capacité
Indicateur de participation

*identificateur de capacité*: il s'agit d'une identification assignée à la capacité. Cette identification peut être normalisée (documentée dans T.125) ou non;

*classe de capacité*: elle indique si une valeur est associée à la capacité, ainsi que le traitement associé à la valeur en question. Toutes les capacités sont de classe MIN, MAX ou NULLE;

*indicateur de participation*: ce champ indique le degré de participation requis des autres nœuds V3 du sommet. Les valeurs possibles de ce champ sont "partielle" ou "globale".

### 11.51 CapabilitiesNotificationIndication

L'unité **CapabilitiesNotificationIndication** est une unité PDU descendante qui est utilisée pour avertir les nœuds d'un changement des capacités du sommet ou pour avertir le sommet d'un changement d'état des connexions du nœud V2. Cette unité PDU est envoyée par le fournisseur de sommet ou par tout nœud intermédiaire qui a besoin d'informer un nœud inférieur d'un changement.

**Tableau 11-51/T.125 – Unité MCSPDU de type CapabilitiesNotificationIndication**

Contenu	Source	Collecteur
Nœud V2 présent	Fournisseur supérieur	Subordonné
Liste d'ajout	Fournisseur supérieur	Subordonné
Liste de suppression	Fournisseur supérieur	Subordonné

Le contenu de l'unité **CapabilitiesNotificationIndication** est le suivant:

*nœud V2 présent*: ce fanion est remis à 1 lorsqu'un nœud V2 est connecté au sommet V3. A ce moment, ce fanion est réinitialisé du fait que tous les nœuds V2 sont déconnectés;

*liste d'ajout*: il s'agit d'une liste des capacités d'indication (décrites au 11.51.1) qu'un nœud donné décide qu'il est nécessaire d'ajouter à la liste des capacités du sommet. Ce champ est facultatif du fait que ce ne sont pas toutes les unités **CapabilitiesNotificationIndication** qui permettront d'ajouter des capacités à la liste;

*liste de suppression*: il s'agit d'une liste des capacités d'indication qu'un nœud donné décide qu'il est nécessaire de supprimer de la liste des capacités du sommet. Ce champ est facultatif du fait que ce ne sont pas toutes les unités **CapabilitiesNotificationIndication** produites qui serviront à supprimer des capacités de la liste.

### 11.51.1 Capacité d'indication

Le tableau ci-dessous résume le contenu de la capacité d'indication. Les unités de liste d'ajout et de liste de suppression d'une unité PDU **CapabilitiesNotificationIndication** sont constituées de capacités d'indication.

Identificateur de capacité
Classe de capacité
Pris en charge par le fournisseur de sommet
Pris en charge par les nœuds intermédiaires

*identificateur de capacité*: il s'agit d'une identification assignée à la capacité. Cette identification peut être normalisée (documentée dans T.125) ou non;

*classe de capacité*: elle indique si une valeur est associée à la capacité, ainsi que le traitement associé à la valeur en question. Toutes les capacités sont de classe MIN, MAX ou NULLE;

*pris en charge par le fournisseur de sommet*: ce fanion indique si le fournisseur de sommet V3 prend cette capacité en charge;

*pris en charge par les nœuds intermédiaires*: ce fanion indique si les nœuds intervenants du chemin vers le fournisseur de sommet prennent cette capacité en charge. En l'absence de nœuds intervenants, c'est-à-dire lorsqu'un nœud est directement connecté au fournisseur de sommet, ce fanion est remis à 1.

## 12 Base de données de fournisseur de service MCS

### 12.1 Copie hiérarchique

Bien qu'un fournisseur de service MCS puisse couvrir plusieurs domaines, il dessert chacun d'entre eux indépendamment. Il tient à jour une base de données logiquement distincte pour chaque domaine afin d'y enregistrer l'état de canal et les ressources de jeton utilisées. La description ci-après est conçue dans le cadre d'un domaine unique.

Les ressources du service MCS qu'il y a lieu de gérer dans un domaine sont les identificateurs de canal et les identificateurs de jeton. Les identificateurs d'utilisateur sont un sous-ensemble des identificateurs de canal. Les paramètres de domaine imposent une limite au nombre d'identificateurs de chaque catégorie que l'on peut utiliser simultanément. Cela permettra à un fournisseur de calculer la quantité de mémoire nécessaire pour sa base de données dans le cas plus défavorable d'un domaine complètement utilisé.

Dans la hiérarchie d'un domaine, les identificateurs utilisés au niveau de tout fournisseur de service MCS donné se réduisent au sous-ensemble de ceux qui sont utilisés au niveau de son fournisseur immédiatement supérieur. Les renseignements relatifs à un identificateur sont enregistrés au point où ils peuvent être utilisés pour prendre en charge des services MCS mettant en œuvre cet identificateur. Un enregistrement plus étendu d'informations entraînerait des coûts supplémentaires dans le trafic des unités MCSPDU afin de tenir les renseignements à jour. Comme les informations enregistrées au niveau d'un fournisseur sont cohérentes avec celles qui sont enregistrées au niveau de fournisseurs plus élevés, dans les limites du temps de propagation des unités MCSPDU, on peut considérer que la base de données du fournisseur est partiellement copiée dans la hiérarchie du domaine.

Les paramètres de domaine deviennent fixes et immuables lors de l'établissement de la première connexion MCS de ce domaine. Un fournisseur qui n'a pas la capacité correspondant au nombre maximal d'identificateurs spécifié dans chaque catégorie peut négocier son adhésion à un domaine sous un faux prétexte. Il peut supposer que, dans sa position inférieure dans la hiérarchie, il ne sera pas appelé à conserver plus qu'une fraction de la base de données totale. Un tel fournisseur peut prendre en charge des rattachements et des subordonnés sans la gamme complète de services MCS que ces rattachements et subordonnés peuvent escompter. Néanmoins, tant que sa capacité n'est pas effectivement dépassée, un tel fournisseur peut apparaître comme un membre à part entière du domaine. Cette stratégie peut convenir à des nœuds terminaux à besoins limités.

Les identificateurs sont d'abord mis en application au niveau du fournisseur de service MCS supérieur. Ils sont mis en application au niveau de fournisseurs subordonnés au moyen d'un flux descendant et sélectif d'unités MCSPDU. La plupart de ces identificateurs sont supprimés de la même manière, de haut en bas. Il y a nécessairement des intervalles pendant lesquels un fournisseur enregistre comme utilisé un identificateur que ses supérieurs n'utilisent plus, parce que l'unité MCSPDU qui supprime cet identificateur est encore en transit. Ce n'est cependant pas une situation durable. Les unités MCSPDU de commande sont reçues et traitées dans l'ordre où elles ont été envoyées. Les conséquences du traitement d'une unité MCSPDU, y compris sa création ou sa suppression d'identificateurs de canal ou de jeton, prennent effet avant que l'attention passe à l'événement d'entrée suivant.

Une exception à la règle du paragraphe précédent est la suppression d'identificateurs de canal statique et de canal attribué. Bien que mis en usage par un flux descendant d'unités **ChannelJoinConfirm**, ces identificateurs de canal sont supprimés dans l'ordre contraire: de bas en haut. Plus précisément, ils sont supprimés lorsqu'une accumulation de demandes MCS-CHANNEL-LEAVE issues de rattachements se combine avec des unités MCSPDU de type **ChannelLeaveRequest** issues de fournisseurs subordonnés pour effectuer une sortie de canal sans adhésion. De telles transitions justifient l'envoi plus en amont d'unités **ChannelLeaveRequest**. Dans ces deux cas, les identificateurs de canal enregistrés comme étant en usage formeront donc un sous-ensemble exact des identificateurs de canal en usage au niveau du fournisseur plus élevé. Il s'agit d'un corollaire obligé des optimisations conçues pour accélérer la gestion des canaux avant un transfert de données.

Les identificateurs de canal sont mis en usage par les unités **MergeChannelsConfirm**, **AttachUserConfirm**, **ChannelJoinConfirm**, **ChannelConveneConfirm** et **ChannelAdmitIndication**; ils sont supprimés par les unités **MergeChannelsConfirm**, **PurgeChannelsIndication**, **DetachUserIndication**, **ChannelLeaveRequest**, **ChannelDisbandIndication** et **ChannelExpelIndication**. Les identificateurs de jeton sont mis en usage par les unités **MergeTokensConfirm**, **TokenGrabConfirm**, **TokenInhibitConfirm** et **TokenGiveIndication**; ils sont supprimés par les unités **MergeTokensConfirm**, **PurgeTokensIndication**, **TokenReleaseConfirm**, **TokenGiveResponse** et **TokenGiveConfirm**. Lorsqu'un identificateur est mis en usage au niveau d'un fournisseur donné, l'unité MCSPDU qui en

est la cause peut être envoyée à zéro, un, plusieurs ou tous les fournisseurs subordonnés. L'utilisation d'un identificateur peut augmenter ou diminuer progressivement, par exemple si des utilisateurs individuels sont admis dans un canal privé et en sont expulsés. Lorsqu'un identificateur est supprimé d'un fournisseur donné, l'unité MCSPDU qui effectue cette opération est envoyée à tous les subordonnés qui peuvent encore être en train d'enregistrer cet identificateur comme étant en usage.

L'utilisation d'un identificateur est finalement liée aux actions sur un canal ou sur un jeton par un utilisateur rattaché au domaine (malgré un certain retard dû, comme cela a été expliqué, à la communication des changements par l'envoi d'unités MCSPDU). Les identificateurs enregistrés de manière stable comme étant en usage au niveau d'un fournisseur de service MCS donné sont ceux qui sont activement employés par un utilisateur se trouvant dans le sous-arbre de ce fournisseur. Il en découle que ces identificateurs forment un sous-ensemble de ceux qui ont été enregistrés de manière stable au niveau d'un fournisseur plus élevé quelconque.

La suppression d'un identificateur d'utilisateur a pour effet secondaire de supprimer les identificateurs de canal et les identificateurs de jeton dont il est l'unique utilisateur dans un sous-arbre.

Les sous-paragraphes suivants spécifient les critères permettant de considérer que des identificateurs de canal et des identificateurs de jeton sont en usage.

## **12.2 Informations relatives aux canaux**

Les quatre types de canaux possèdent des critères correspondants pour déterminer si un point de rattachement donné est considéré comme utilisant l'identificateur de canal et donc s'il doit être représenté dans une base de données de fournisseur:

- a) un identificateur de canal statique (dans l'intervalle 1 .. 1000) est en usage si l'utilisateur a adhéré à ce canal avec une confirmation MCS-CHANNEL-JOIN contenant une valeur favorable et s'il n'est pas sorti de ce canal par une demande ou une indication MCS-CHANNEL-LEAVE;
- b) un canal d'identification d'utilisateur est en usage s'il a été attribué à l'utilisateur par une confirmation MCS-ATTACH-USER contenant une valeur favorable et si cet utilisateur n'a pas été détaché par une demande ou une indication MCS-DETACH-USER;
- c) un identificateur de canal privé est en usage si l'utilisateur a créé le canal avec une confirmation MCS-CHANNEL-CONVENE favorable ou s'il y a été admis avec une indication MCS-CHANNEL-ADMIT et n'en a pas été expulsé par une indication MCS-CHANNEL-EXPEL; le canal ne doit pas non plus avoir été dissous par une demande ou une indication MCS-CHANNEL-DISBAND;
- d) un identificateur de canal attribué est en usage si l'utilisateur a adhéré au canal avec une confirmation MCS-CHANNEL-JOIN favorable et qu'il n'en est pas sorti par une demande ou une indication MCS-CHANNEL-LEAVE.

Les informations suivantes doivent être enregistrées pour un identificateur de canal en usage:

- a) le type de canal qu'il représente (statique, d'identification d'utilisateur, privé ou attribué);
- b) les rattachements et connexions MCS avec des fournisseurs subordonnés qui ont adhéré au canal;
- c) dans le cas d'un canal d'identification d'utilisateur, le sens d'acheminement correspondant, soit celui du rattachement MCS local auquel l'identificateur d'utilisateur est attribué ou la connexion MCS descendante vers un fournisseur subordonné dans le sous-arbre où l'utilisateur réside;

- d) dans le cas d'un identificateur de canal privé, l'identificateur d'utilisateur du gestionnaire qui l'a constitué (que ce gestionnaire se trouve ou non dans le sous-arbre du fournisseur) et l'ensemble de tous les identificateurs d'utilisateur contenus dans le sous-arbre du fournisseur et admis dans ce canal.

Les informations enregistrées pour les identificateurs de canal sont employées comme expliqué au paragraphe 10 afin de valider les unités MCSPDU de demande et pour acheminer des unités MCSPDU d'indication et de confirmation.

### 12.3 Informations relatives aux jetons

Les transitions d'état d'un identificateur de jeton sont représentées à la Figure 12-1. Les abréviations suivantes s'appliquent:

DUin – DetachUserIndication

TGcf – TokenGrabConfirm

Tlcf – TokenInhibitConfirm

TRcf – TokenReleaseConfirm

TVin – TokenGiveIndication

TVrs – TokenGiveResponse

TVcf – TokenGiveConfirm

Un identificateur de jeton individuel peut être *saisi* par un utilisateur unique ou être *inhibé* par un ou plusieurs utilisateurs. L'action de l'unité **TokenGiveIndication** fait passer l'état du jeton à *donnant* sur la branche d'une hiérarchie de domaine conduisant du fournisseur de service MCS supérieur au destinataire prévu. Cet état se dégrade en *inaccessible* si le destinataire se détache avant que son fournisseur ait répondu par une unité **TokenGiveResponse**. Il passe à *donné* si, au contraire, le donneur libère explicitement le jeton ou se détache. Au cours de la cession d'un jeton, la branche d'une hiérarchie de domaine qui part du donneur coupe la branche menant vers le destinataire au moins au niveau du fournisseur de service MCS supérieur. L'état du jeton passe de *saisi* à *donnant* puis, le cas échéant, à *non donnable* ou à *donné*, mais seulement à partir de cette intersection.

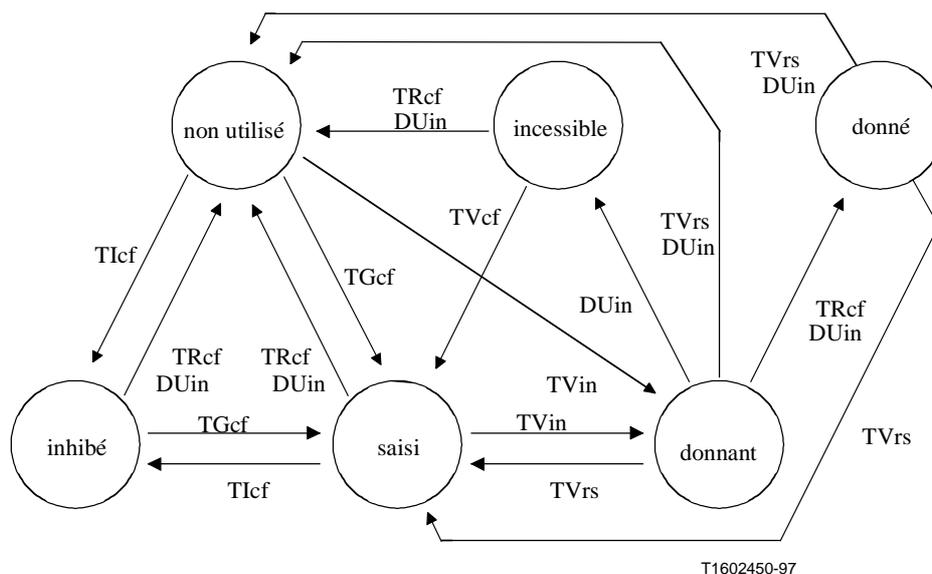


Figure 12-1/T.125 – Transitions d'état d'un identificateur de jeton

L'utilisateur d'un jeton a avec celui-ci une relation de saisisseur, d'inhibiteur, de destinataire ou à la fois de saisisseur et de destinataire (lorsqu'il se donne à lui-même un jeton):

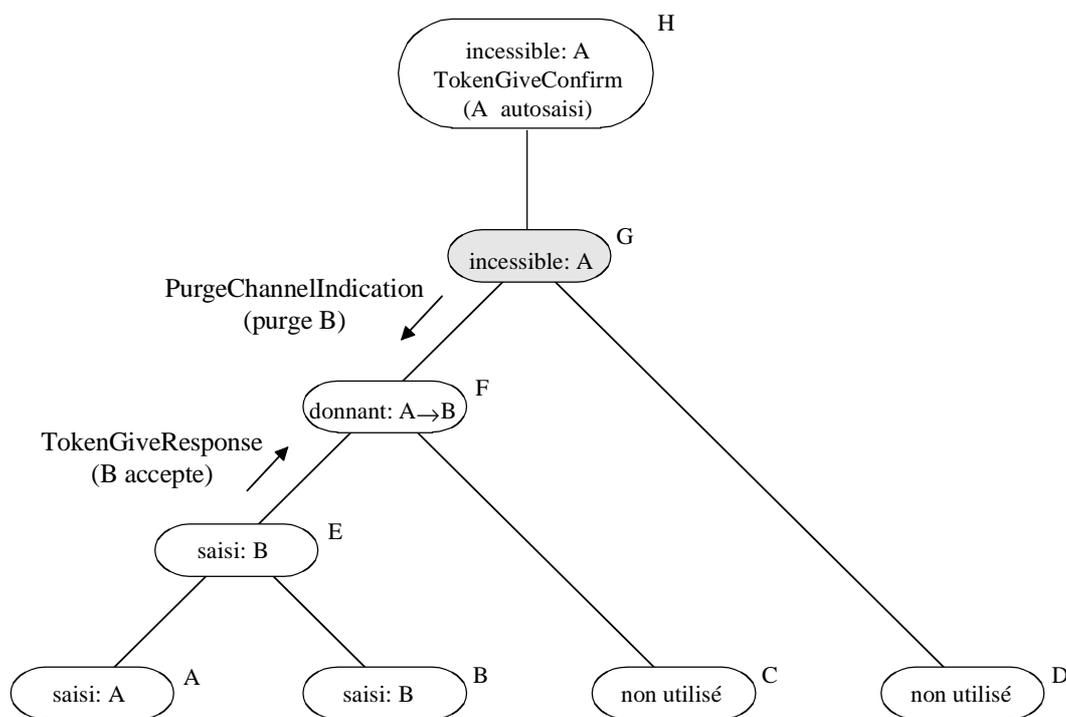
- a) l'utilisateur est un saisisseur s'il a saisi un jeton avec une confirmation MCS-TOKEN-GRAB contenant une valeur favorable et qu'il ne l'a pas libéré par une demande MCS-TOKEN-RELEASE ou par une confirmation MCS-TOKEN-GIVE favorable ni transformé avec une confirmation MCS-TOKEN-INHIBIT favorable ou avec une réponse MCS-TOKEN-GIVE s'il a accepté un jeton proposé;
- b) l'utilisateur est un inhibiteur s'il a saisi un jeton avec une confirmation MCS-TOKEN-INHIBIT favorable et qu'il ne l'a pas libéré avec une demande MCS-TOKEN-RELEASE ni transformé avec une confirmation MCS-TOKEN-GRAB favorable;
- c) l'utilisateur est un destinataire si un jeton lui a été proposé par une indication MCS-TOKEN-GIVE et qu'il n'a pas libéré ce jeton avec une réponse MCS-TOKEN-GIVE défavorable.

Ces informations doivent être enregistrées comme suit pour un identificateur de jeton en usage:

- a) l'état de l'identificateur de jeton au niveau du fournisseur de service MCS (cet état n'est pas nécessairement le même qu'au niveau du fournisseur supérieur);
- b) si le jeton est saisi ou inaccessible, l'identificateur d'utilisateur du saisisseur dans le sous-arbre du fournisseur;
- c) si le jeton est donnant, l'identificateur d'utilisateur du saisisseur (que celui-ci soit ou non dans le sous-arbre du fournisseur);
- d) si le jeton est donnant ou donné, l'identificateur d'utilisateur du destinataire dans le sous-arbre du fournisseur;
- e) si le jeton est inhibé, l'ensemble de tous les identificateurs d'utilisateur se trouvant dans le sous-arbre du fournisseur, qui ont inhibé le jeton.

Les informations enregistrées pour les identificateurs de jeton en usage sont employées comme expliqué au paragraphe 10 pour valider les unités MCSPDU de réponse et pour acheminer les unités MCSPDU d'indication.

L'état d'un identificateur de jeton au niveau d'un fournisseur subordonné n'a pas besoin d'être identique à ce qu'il serait au niveau du fournisseur de service MCS supérieur. Cela est dû au fait qu'un donneur de jeton ne traite généralement pas les unités **TokenGiveIndication** ou **TokenGiveResponse** et qu'un destinataire ne traite généralement pas une confirmation **TokenReleaseConfirm** de donneur. La Figure 12-2 montre les états qui peuvent apparaître lors d'une interaction complexe entre jetons.



T1602460-97

**Figure 12-2/T.125 – États d'un jeton pouvant apparaître lors d'une interaction complexe**

La figure concerne un seul identificateur de jeton dans la base de données des fournisseurs A à H. Un cas de figure plausible est celui où le jeton a été donné par l'utilisateur A, lui-même rattaché au fournisseur A, à l'utilisateur B rattaché au fournisseur B. Avant que l'utilisateur B puisse répondre, le fournisseur G a toutefois connecté le domaine à un nouveau fournisseur supérieur H et a commencé une fusion. Le nouveau domaine, incapable d'accepter l'utilisateur B en raison d'un conflit entre identificateurs de canal, a commencé sa purge par l'intermédiaire de l'unité **PurgeChannelsIndication**. Cette unité MCSPDU est représentée après avoir effectué une partie de son itinéraire passant par les fournisseurs G et D. Le fournisseur G, par conséquent, a adapté l'état de son jeton de *donnant* à *incessible*. Puis il a fusionné le jeton, avec cet état, dans le nouveau domaine. Le nouveau fournisseur supérieur H, auquel a été présenté un jeton avec cet état, a mis en file d'attente de transmission une confirmation **TokenGiveConfirm** de valeur défavorable afin de renvoyer le jeton à l'utilisateur A. A l'instant de la description, le jeton a aussi été finalement accepté par l'utilisateur B et une unité **TokenGiveResponse** est en train de remonter la branche destinataire de la hiérarchie de domaine: elle fait passer l'état du jeton, à chaque fournisseur, de *donnant* à *saisi* par B. Deux unités MCSPDU convergent donc vers le fournisseur F et celle qui arrive la première détermine l'état suivant du jeton à ce point: soit un retour à l'état *saisi* par A ou le maintien transitoire à *saisi* par B puis de nouveau l'état *non utilisé* une fois que l'utilisateur B s'est détaché. Dans un cas comme dans l'autre, l'état du jeton se stabilise à *saisi* par A lorsque le nouveau fournisseur supérieur envoie la confirmation **TokenGiveConfirm** qu'il tient en instance.

## 13 Éléments de procédure

### 13.1 Ordonnement des unités MCSPDU

Les unités MCSPDU de commande restent en séquence entre toute paire de fournisseurs MCS parce qu'elles se déplacent sur une même connexion de transport: la connexion de transport initiale d'une

connexion MCS. Les fournisseurs MCS doivent traiter les unités MCSPDU reçues et envoyer toutes les unités MCSPDU de sortie résultantes dans le même ordre. Cela s'applique aux unités MCSPDU qui sont simplement expédiées en amont ou en aval de la hiérarchie du domaine ainsi qu'aux unités MCSPDU qui sont transformées, comme des demandes ou des réponses devenant des indications ou des confirmations. L'ordonnancement des unités MCSPDU doit être conservé dans le cadre d'un fournisseur de service MCS s'il est nécessaire de mettre des informations de sortie en file d'attente pour transmission ultérieure en raison de la contre-pression de la commande de flux sur une connexion de transport.

Les unités MCSPDU de données ayant des priorités différentes n'ont pas besoin de rester en séquence. Au contraire, l'avantage de priorités relatives n'est obtenu que lorsque les données de priorité supérieure sont mises devant les données de priorité inférieure. C'est-à-dire qu'il y a lieu de les transmettre sur des connexions de transport distinctes et de les mettre en file d'attente séparément chez chaque fournisseur de service MCS. Si le nombre de priorités de données mises en œuvre dans un domaine est inférieur à la valeur maximale, un moins grand nombre de connexions de transport sera disponible, mais les fournisseurs qui décident de faire ainsi peuvent toujours gérer des files d'attente distinctes à l'interne. On tire ainsi parti de certains avantages de la priorité relative, mais pas de tous.

Un fournisseur de service MCS doit respecter une certaine priorité pour la séquence des unités MCSPDU de données transmises. Il s'agit là d'une restriction plus sévère que celle qui est imposée par la Recommandation T.122, qui ne garantit que l'ordonnancement des unités de données du service transmises avec une priorité donnée sur le même canal de destination.

Les unités MCSPDU de commande et de données qui ont une priorité absolue sont transmises sur la même connexion de transport initiale. Elles doivent faire l'objet d'une attention égale de la part d'un fournisseur de service MCS. Les données de priorité inférieure peuvent venir après. Des indications de commande progressant en tête peuvent arriver avant les données de priorité inférieure qui ont été transmises effectivement les premières.

### **13.2 Commande du flux d'entrée**

Un fournisseur de service MCS se trouve parfois devant des objectifs contradictoires: continuer à faire avancer rapidement des données dans un domaine malgré des blocages transitoires touchant certains récepteurs, donner aux émetteurs un accès équitable à la largeur de bande disponible et empêcher un quelconque correspondant de se faire largement distancer par des homologues qui reçoivent les mêmes données multidiffusées. Le service de communication multipoint (MCS) est un service fiable qui préserve l'intégrité des données d'utilisateur. Comme un fournisseur est limité dans sa capacité d'enregistrer des unités MCSPDU lorsque celles-ci ne peuvent pas être transmises immédiatement, ce fournisseur doit avoir la possibilité de se protéger en refusant de continuer à recevoir des données. Bien que les détails de l'interface avec les services de transport relèvent de décisions locales, l'effet abstrait de ce refus doit être que des unités TSDU entrantes sont conservées dans des pipelines de connexions de transport, intactes et en séquence, pour réception ultérieure dès que la commande de flux le permettra. Lorsque ces pipelines de connexion de transport se remplissent, des fournisseurs MCS distants peuvent se trouver devant une contre-pression bloquant leurs émissions de nouvelles unités MCSPDU. Ils peuvent dans ce cas invoquer un système de protection analogue.

La commande de flux n'est pas décrite explicitement dans le protocole MCS. Il s'agit d'une fonction des couches inférieures qu'il serait dispendieux de dédoubler. Il en découle qu'il est difficile de percevoir, par l'intermédiaire d'une connexion de transport mise à contribution, si un fournisseur distant se protège contre de nouvelles entrées. Pour répondre au mieux à des objectifs contradictoires, on peut recommander les politiques décrites ci-après.

Un fournisseur de service MCS peut accorder à chaque connexion de transport entrante un quota fixe de tampons qu'il peut remplir d'unités MCSPDU avant que la contre-pression soit appliquée. Chaque tampon est traité comme spécifié dans le présent protocole, puis affecté à un accès de sortie vers zéro, une ou plusieurs connexions de transport sortantes. La sortie peut se produire immédiatement ou être retardée parce qu'un pipeline de transport est plein. Avant que son contenu soit envoyé en sortie vers la dernière connexion de transport, un tampon se charge en fonction du quota d'entrée de la connexion de transport qui a transporté celui-ci. Une fois le tampon vidé vers toutes les connexions de transport requises, il est réinitialisé pour se recharger en fonction de ce quota. Lorsqu'un quota est épuisé, les nouvelles entrées sur la connexion de transport correspondante sont arrêtées. Les quotas d'entrée peuvent être déterminés compte tenu du fait qu'une connexion de transport fait ou non partie d'une connexion MCS ascendante ou descendante et du niveau de priorité de données qu'elle représente.

La mise en mémoire-tampon peut atténuer une disparité de débits entre émetteurs et récepteurs. Un quota imposé aux entrées peut empêcher une certaine connexion de transport de monopoliser les ressources. Elle peut également imposer une limite à la mesure dans laquelle deux récepteurs des mêmes données multidiffusées peuvent être asynchrones. Le procédé recommandé n'est cependant pas assez élaboré pour prévoir tous les cas d'utilisation. Il peut parfois ralentir le débit de transfert des données à travers un domaine lorsqu'il existe d'autres solutions acceptables. La découverte de meilleures politiques de commande de flux (mises en œuvre localement et ne nécessitant pas de communication additionnelle d'unités MCSPDU) pourra être un moyen de différencier des produits.

### 13.3 Application du débit

Contrairement à la commande du débit d'entrée, le protocole MCS développe quelques moyens d'appliquer le débit utile. Tout d'abord, le débit qui est appliqué est un paramètre de domaine qui est négocié par l'intermédiaire de la demande MCS-CONNECT-PROVIDER. En deuxième lieu, l'intervalle de temps pendant lequel le débit est surveillé avant de prendre une mesure contraire est communiqué par chaque fournisseur de service MCS à son supérieur par l'intermédiaire de l'unité **ErectDomainRequest**. La description de l'intervalle d'application du débit nécessite que les fournisseurs aient en commun certains principes de comportement. Il n'en demeure pas moins que l'application reste une technique heuristique qui laisse place à l'invention.

L'application d'un débit d'entrée minimal à chaque récepteur est une option offerte aux applications de commande, qui établissent les connexions MCS d'un domaine. Cette option est choisie par l'intermédiaire du paramètre de domaine relatif à l'application du débit, qui est indiqué en octets par seconde. Bien qu'il paraisse évident qu'il n'y a pas lieu d'autoriser un correspondant à fonctionner à une vitesse arbitrairement lente et à faire ainsi obstacle au transfert de données avec d'autres correspondants, le risque inverse est de chercher à imposer trop strictement un débit.

Les configurations complexes à émetteurs multiples posent un problème. Les types de contre-pression auxquels une politique d'application de débit réagit peuvent ne pas provenir d'un seul récepteur de vitesse anormalement lente. Tout d'abord, les unités MCSPDU descendantes sont en concurrence de traitement avec les unités MCSPDU qui remontent mais dont une image redescend, comme l'unité **SendDataIndication**. Le flux descendant rencontré par l'intermédiaire d'un fournisseur homologue peut donc n'atteindre qu'une fraction de la largeur de bande nominale et peut varier dynamiquement selon le nombre d'autres connexions et rattachements à cet homologue. En deuxième lieu, seules les unités MCSPDU de priorité absolue sont censées transiter en continu sur une connexion MCS. Des unités de moindre priorité peuvent tout à fait être bloquées par un fournisseur homologue pendant de longues périodes à cause de l'intensité du trafic provenant d'autres sources avec des priorités plus élevées. Finalement, une grande variation de débit instantané peut se produire si la mesure tient compte du temps pendant lequel une unité MCSPDU bloquée doit attendre avant d'être acceptée sur une connexion MCS. Ce temps peut dépendre, entre autres

variables, du nombre et de l'ordre des unités MCSPDU issues d'autres sources et mises en file d'attente chez le fournisseur homologue.

Néanmoins, l'application du débit est une option intéressante dans les cas réels où l'on sait que les configurations de transfert de données sont plus uniformes. Il faut l'interpréter comme exigeant un envoi minimal d'unités MCSPDU vers chaque rattachement MCS direct et vers chaque connexion MCS descendante dans un intervalle de temps qui sera spécifié par le fournisseur de service MCS imposant son débit. Chaque unité MCSPDU sortante, de commande aussi bien que de données, doit compter pour le débit comme si elle avait la longueur maximale autorisée par les paramètres de domaine. La sortie d'une unité MCSPDU vers un rattachement MCS implique l'acheminement de la primitive d'indication ou de confirmation correspondante. La sortie vers une connexion MCS descendante implique l'absence de contre-pression à l'interface avec le service de transport et l'acceptation de cette unité dans le pipeline de la connexion de transport correspondante.

La sortie doit être contrôlée tant qu'au moins une unité MCSPDU est, quelle que soit la priorité des données, en file d'attente de transmission vers un rattachement donné ou une connexion descendante donnée. Chaque fois que les files sont vides, le contrôle doit s'arrêter et aucune mesure d'application de débit ne doit être prise. En même temps, aucune accumulation de droits n'est ouverte par un bon comportement afin de compenser de futures défaillances. Le contrôle doit reprendre dès qu'une contre-pression empêche une unité MCSPDU d'être envoyée en sortie et oblige au contraire à la mettre en file d'attente. Lorsqu'au moins une unité MCSPDU reste en file d'attente, le nombre d'unités effectivement envoyées en sortie doit être compté sur un intervalle de temps déterminé.

Chaque fournisseur de service MCS doit choisir un intervalle d'application de débit. Cet intervalle doit être assez long pour qu'au moins une unité MCSPDU de longueur maximale puisse être envoyée en sortie au débit minimal. Un fournisseur de service MCS doit signaler à son supérieur l'intervalle choisi et tous changements ultérieurement apportés à cette durée par un envoi d'unité **ErectDomainRequest** en amont. Un fournisseur doit se protéger contre le rattachement MCS ou la connexion MCS descendante fautifs à la fin de tout intervalle pendant lequel le débit contrôlé n'arrive pas à atteindre les valeurs prévues. Ce fournisseur doit alors détacher l'utilisateur ou couper la connexion.

Des fournisseurs de niveau élevé dans la hiérarchie peuvent régler leur intervalle d'application de débit de manière qu'il soit plus long que celui de chacun de leurs subordonnés, plus une certaine marge pour le temps de réaction. Le but est d'inciter à ce que la mesure d'application du débit soit prise à partir du fournisseur de niveau le moins élevé qui a la possibilité de détecter un problème. Lorsqu'un contrevenant est éliminé à la fin d'un intervalle correspondant à un niveau inférieur, les fournisseurs de niveau plus élevé doivent normalement avoir assez de temps pour percevoir le rétablissement du débit à des niveaux adéquats. S'ils agissent trop rapidement, ils peuvent pénaliser un plus grand sous-arbre que nécessaire et interrompre des utilisateurs non fautifs dans le domaine.

Il y a lieu que les applications de régulation qui déterminent les paramètres de domaine soient prudentes dans leurs exigences et qu'elles s'attendent que le débit pourra occasionnellement chuter en périodes d'intenses stimulations d'application. Si leur objet est simplement d'assurer une protection à l'encontre des récepteurs qui arrêtent d'accepter toute entrée que ce soit, ces applications peuvent fixer à une valeur très basse le débit minimal. Connaissant la longueur maximale des unités MCSPDU et le débit utile appliqué, les régulateurs peuvent calculer l'intervalle minimal qui devra toujours s'écouler avant qu'un blocage quelconque soit détecté et résolu.

### 13.4 Configuration de domaine

La Recommandation T.122 n'indique aucun mécanisme permettant de configurer l'ensemble des domaines pris en charge par un fournisseur de service MCS. Il faut considérer cela comme relevant d'une décision locale, dont la normalisation pourra faire l'objet d'un complément d'étude. Le présent

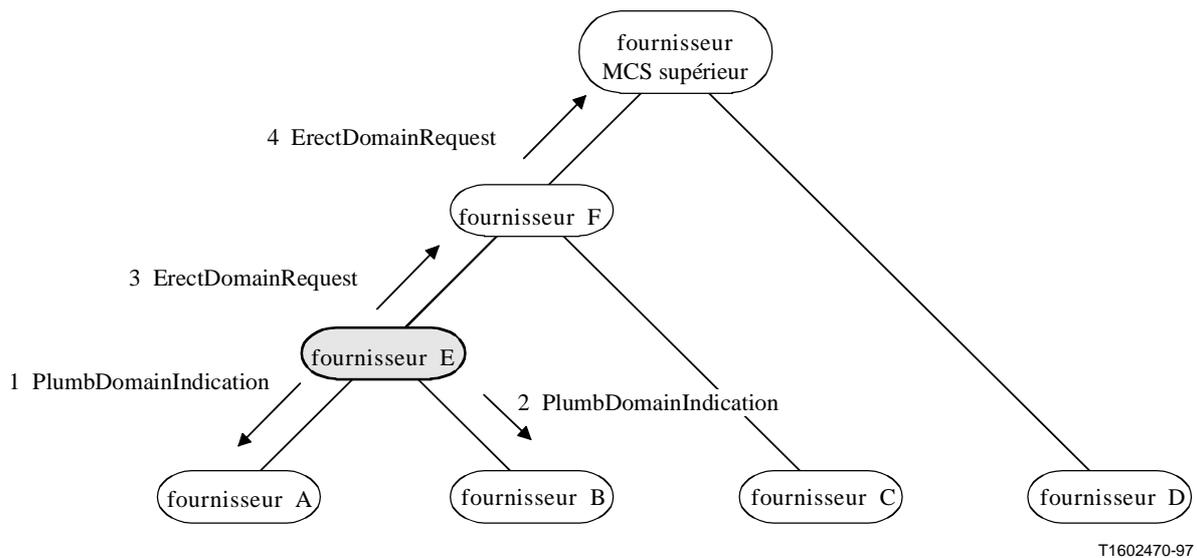
protocole part du principe qu'un fournisseur de service MCS reconnaîtra certains sélecteurs de domaine comme étant valides et d'autres comme non valides. Il prévoit que les sélecteurs de domaine soient communiqués dans le cadre de l'établissement d'une connexion MCS.

Un fournisseur de service MCS participe implicitement à la négociation des paramètres de domaine. Qu'il soit du côté appelant ou du côté appelé, il contraint la gamme des valeurs paramétriques autorisées en fonction des limites de sa configuration. Un fournisseur de service MCS doit suspendre la possibilité de négocier les paramètres d'un domaine une fois qu'un utilisateur quelconque s'est rattaché à ce domaine ou que la première connexion MCS a été établie.

### 13.5 Fusion de domaines

Les domaines sont fusionnés à la suite d'une demande MCS-CONNECT-PROVIDER. Si l'on souhaite faire en sorte qu'un domaine ou un autre soit vide en l'occurrence, cela ne complique pas beaucoup une fusion. Dans le cas le plus général, il faut toutefois tenir compte d'une mise à jour de la base de données au niveau du fournisseur supérieur restant de manière qu'elle reprenne le contenu de la base de données du précédent fournisseur supérieur, et prévoir la résolution de tout conflit pouvant apparaître. Le paragraphe 10 donne plus de détails à ce sujet.

Pour faciliter la compréhension, la séquence des Figures 13-1 à 13-4 montre un exemple de fusion de domaines. Le fournisseur E y représente un ancien supérieur qui a adhéré à un nouveau domaine au moyen de la connexion MCS indiquée, qui le relie à un fournisseur intermédiaire F. La connexion MCS peut avoir été établie par le fournisseur E ou par le fournisseur F.



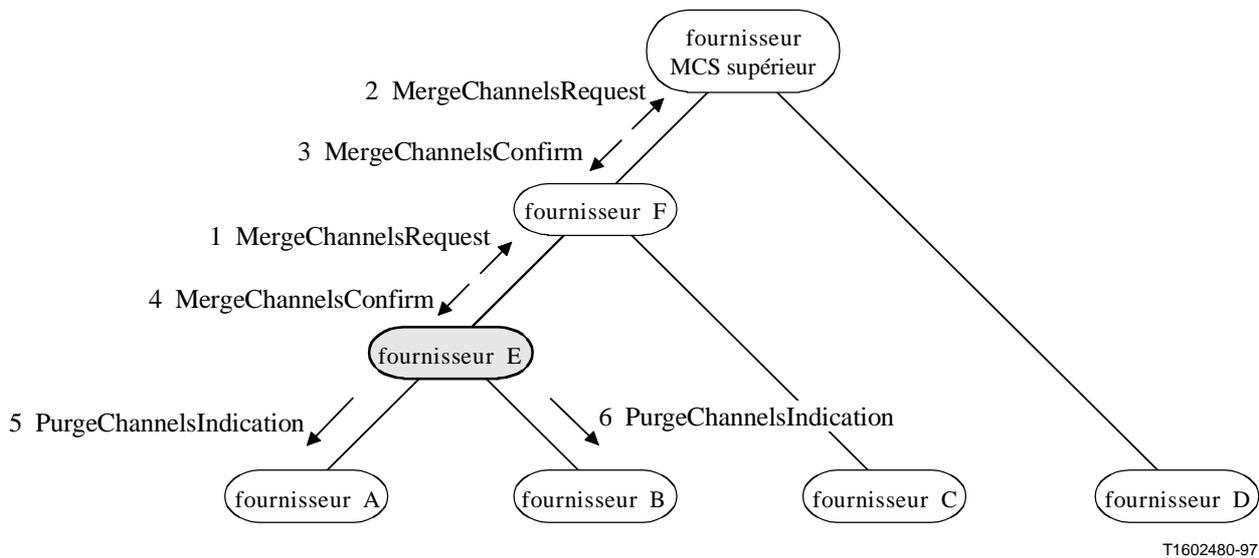
**Figure 13-1/T.125 – Première phase de la fusion de domaines – Établissement de la hiérarchie**

Le fournisseur E, qui se trouve à l'extrémité inférieure d'une connexion ascendante, prend la responsabilité d'exécuter la fusion. Comme il est risqué d'entreprendre une autre activité pendant que la base de données est en consultation, le fournisseur E arrête d'accepter des entrées issues de son sous-arbre. Toute unité MCSPDU déjà en transit depuis les fournisseurs A et B – ou de nouveaux fournisseurs créés avant l'achèvement de la fusion – sera sauvegardée et traitée ultérieurement. Le flux descendant d'unités MCSPDU n'est toutefois pas gêné, qu'il ait été produit par le fournisseur E ou expédié de plus haut.

En attendant que la fusion soit effectuée, les seules unités MCSPDU de confirmation que le fournisseur E recevra seront des unités **MergeChannelsConfirm** et **MergeTokensConfirm** étant donné que les demandes d'utilisateur ne sont pas autorisées à remonter dans la hiérarchie. Ces unités confirmeront ou purgeront les identificateurs de canaux et de jetons déjà en usage. Des unités de type **PurgeChannelsIndication**, **PurgeTokensIndication**, **DetachUserIndication** et **ChannelDisbandIndication** peuvent également arriver du domaine de niveau plus élevé et peuvent supprimer des identificateurs de canaux et de jetons issus du domaine de niveau moins élevé dont la fusion a été confirmée à titre individuel. Les identificateurs non confirmés du domaine moins élevé sont protégés contre la suppression, car ils ont encore une signification différente de celle de leurs homologues du domaine plus élevé. L'unité **PlumbDomainIndication** doit toujours être suivie d'effet afin d'appliquer la limite de hauteur de domaine. Les autres indications peuvent ne pas être appliquées par le fournisseur E pendant la réalisation de la fusion. Il n'est en particulier pas obligatoire que des transferts de données s'effectuent entre domaines antérieurement séparés, en attendant que la fusion soit terminée; ces données ne peuvent pas passer avant que les canaux qui les acheminent aient au moins été réorganisés. Il peut être gênant que deux unités d'indication puissent mettre en usage de nouveaux identificateurs. Afin que la base de données reste cohérente, le fournisseur E doit réagir comme spécifié au paragraphe 10 s'il refuse une unité **ChannelAdmitIndication** ou **TokenGiveIndication**.

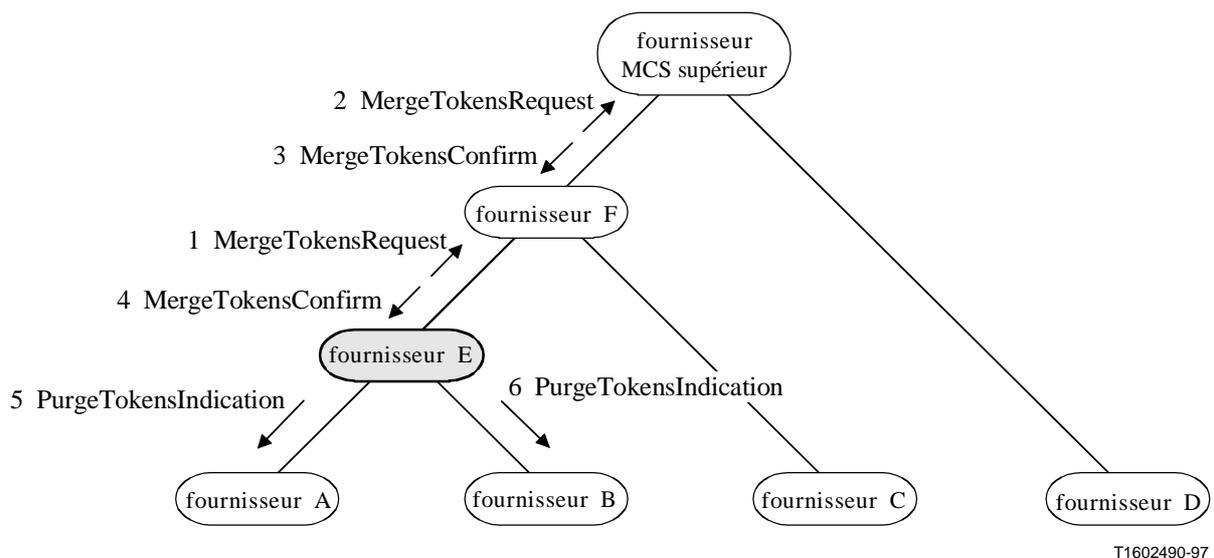
Les premières actions du fournisseur E sont d'envoyer en aval une unité **PlumbDomainIndication** afin de s'assurer que la plus récente connexion MCS n'a pas créé de boucle qui contreviendrait au principe que chaque hiérarchie de domaine ne possède qu'un seul fournisseur supérieur, et d'envoyer en amont une unité **ErectDomainRequest** pour signaler sa hauteur existante et l'intervalle d'application de débit. Le fournisseur F, passant ainsi de la hauteur 2 à la hauteur 3, relaie la demande **ErectDomainRequest** vers le fournisseur supérieur, qui passe alors à la hauteur 4.

Au cours de la deuxième étape de la fusion de domaines, le fournisseur E envoie en amont autant d'instances de l'unité **MergeChannelsRequest** qu'il en faut pour insérer les identificateurs d'utilisateur dans sa base de données. Les identificateurs d'utilisateur qui n'entrent pas en conflit avec le domaine supérieur sont confirmés. Les autres sont purgés dans un nombre égal d'instances de l'unité **MergeChannelsConfirm**. Le fournisseur E produit l'unité **PurgeChannels Indication** à partir de la confirmation **MergeChannelsConfirm** afin de signaler toute purge dans l'ensemble de son sous-arbre. Cette phase se termine lorsque tous les identificateurs d'utilisateur ont été soit explicitement confirmés, soit purgés.



**Figure 13-2/T.125 – Deuxième phase de la fusion de domaines – Fusion des canaux d'identification d'utilisateur**

La troisième phase ressemble à la deuxième, mais concerne les identificateurs de jeton plutôt que les canaux d'identification d'utilisateur, et utilise un ensemble parallèle d'unités MCSPDU. Si les identificateurs d'utilisateur n'ont pas été fusionnés d'abord, des portions d'une ultérieure unité **MergeTokensRequest** pourront être refusées parce que non valides et les identificateurs de jeton touchés seront purgés sans nécessité. Dans le cas d'un jeton inhibé, l'ensemble entier des utilisateurs inhibeurs pourra ne pas s'intégrer dans une seule unité MCSPDU. Le fournisseur E attend confirmation du premier sous-ensemble qu'il envoie en amont, avant d'envoyer de nouveau le jeton inhibé avec les identificateurs d'utilisateur restants. Cela assure une protection contre le refus du premier ensemble en raison de trop nombreux identificateurs de jeton en usage alors que les autres unités seraient prises en charge ultérieurement, ce qui corromprait la base de données. Cette phase se termine lorsque tous les identificateurs de jeton ont été soit explicitement confirmés, soit purgés.



**Figure 13-3/T.125 – Troisième phase de la fusion de domaines – Fusion des identificateurs de jeton**

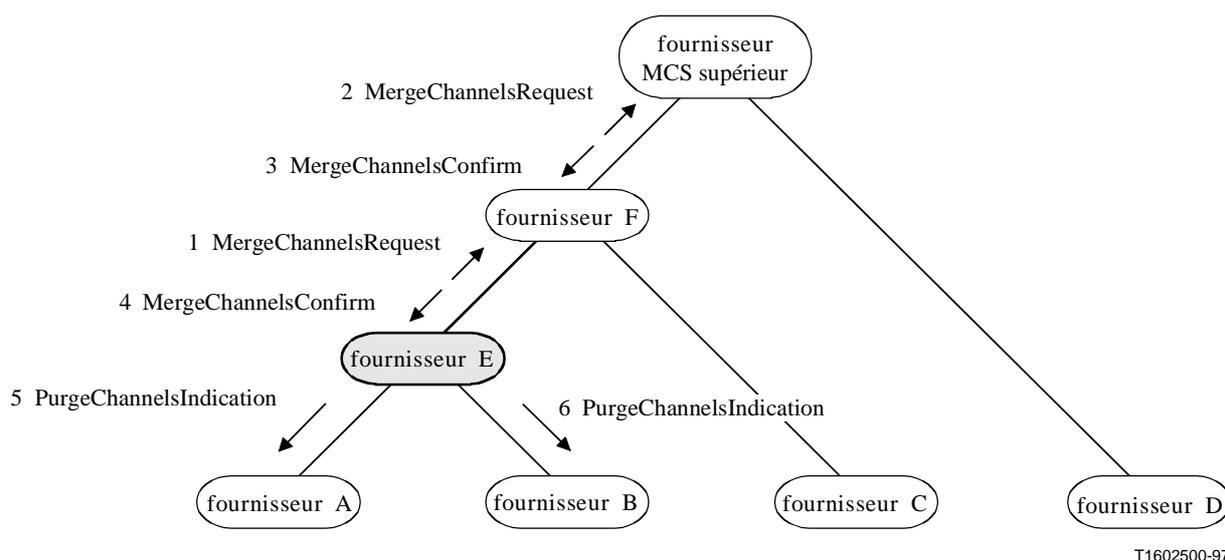
La quatrième phase fait intervenir les mêmes unités MCSPDU que la deuxième phase, mais avec des identificateurs de canal différents. Les canaux d'identification d'utilisateur ayant été pris en compte, il ne reste que les identificateurs de canaux statiques, privés et attribués. Si les identificateurs d'utilisateur n'ont pas été fusionnés en premier, des portions d'unités **MergeChannelsRequest** ultérieures pourront être refusées parce que non valides et les identificateurs de canal touchés seront purgés sans nécessité. Dans le cas d'un canal privé, la totalité des utilisateurs admis par le gestionnaire de canaux pourra ne pas s'intégrer dans une seule unité MCSPDU. Le fournisseur E attend confirmation du premier sous-ensemble qu'il envoie en amont, avant d'envoyer de nouveau le canal privé avec les identificateurs d'utilisateur restants. Cela assure une protection contre le refus du premier ensemble en raison de trop nombreux identificateurs de canal en usage alors que les autres unités seraient prises en charge ultérieurement, ce qui corromprait la base de données. Cette phase se termine, de même que la fusion de domaines, lorsque tous les identificateurs de canal restants ont été soit explicitement confirmés, soit purgés.

NOTE – La fusion des identificateurs de jeton en premier rend leur possession exclusive plus intéressante dans la mesure où cela supprime les conflits entre flux de données. Autrement, des données peuvent se perdre entre des domaines, lors de la confirmation des identificateurs de canal, avant que le conflit soit révélé dans le cadre d'une purge de jetons.

### 13.6 Déconnexion de domaine

Lorsqu'une connexion MCS ascendante est coupée, un fournisseur de service MCS doit éliminer son sous-arbre du domaine en détachant tous ses rattachements MCS directs et en coupant toutes ses autres connexions MCS. Le fournisseur touché ne peut en général pas établir de domaine résiduel dans son propre sous-arbre, car il ne possède pas d'enregistrement d'unités MCSPDU de demande qui auraient été envoyées en amont, pour lesquelles il ne recevra jamais d'unité MCSPDU de confirmation correspondante.

Lorsqu'une connexion MCS descendante est coupée, un fournisseur de service MCS doit produire des unités MCSPDU de type **DetachUserRequest** pour tous les utilisateurs résidant dans cette portion de son sous-arbre, en donnant comme cause: *domaine déconnecté*.



**Figure 13-4/T.125 – Quatrième phase de la fusion de domaines – Fusion des identificateurs de canal restants**

### 13.7 Attribution des identificateurs de canal

Les identificateurs de canal se trouvant dans la gamme de 1001 et plus sont attribués dynamiquement au niveau du fournisseur de service MCS supérieur pendant le traitement des primitives de demande MCS-ATTACH-USER, MCS-CHANNEL-JOIN de valeur zéro et MCS-CHANNEL-CONVENE. Il n'est pas prescrit que les valeurs attribuées correspondent à une quelconque configuration particulière. Il est en revanche souhaitable que les valeurs soient dispersées aléatoirement dans la gamme autorisée. Cela augmentera la probabilité que deux domaines fonctionnant indépendamment l'un de l'autre pendant une période notable puissent être ultérieurement fusionnés sans conflits quant aux attributions de leurs identificateurs de canal respectifs. Cette précaution protégera également contre une remise en circuit trop précoce des identificateurs à l'intérieur d'un même domaine lorsqu'ils sont libérés d'un usage puis réattribués à un autre. Les applications faisant appel au service MCS devront avoir le temps de s'adapter à la disparition d'un identificateur de canal ou d'utilisateur avant qu'il revienne sous une autre forme "d'incarnation".

Dans les situations où la fusion cohérente de domaines actifs est requise, on peut éviter des conflits en sélectionnant des identificateurs de canaux à l'intérieur d'un sous-ensemble unique à chaque fournisseur. On peut créer de tels sous-ensembles en subdivisant en plusieurs bandes les identificateurs de canaux touchés dynamiquement de 1001 à 65535. En outre, à l'intérieur d'un de ces sous-ensembles, on peut affecter des identificateurs en séquence dans un sens, à partir du sommet ou de la racine. Les fournisseurs qui font appel à l'affectation par sous-ensembles doivent toujours respecter tous les aspects du présent protocole, y compris la procédure de fusion de domaines. Ils doivent ainsi tenir compte des fournisseurs homologues qui n'ont peut-être pas de sous-ensembles particuliers.

NOTE – La manière dont un fournisseur choisit un sous-ensemble à partir duquel il effectuera des affectations relève d'une décision locale. Des conférences préorganisées peuvent être réunies par un système de gestion de réseau.

Au contraire des identificateurs de canal, les identificateurs de jeton ne sont pas attribués et la ligne de démarcation de valeur 1001 n'a pas de signification pour eux. Un identificateur de jeton libre, possédant une valeur donnée, est statiquement disponible à tout moment pour être saisi ou inhibé, sous la seule réserve d'une limite de domaine quant au nombre total de jetons utilisés en même temps.

### 13.8 Etat des jetons

L'état des jetons est défini formellement au paragraphe 7. Il est signalé en tant que composant des unités MCSPDU de confirmation relatives aux jetons. Il sert à mettre à jour l'enregistrement concernant un identificateur de jeton dans la base de données des fournisseurs subordonnés. L'état des jetons n'est pas nécessairement signalé directement par l'intermédiaire d'une primitive de confirmation à l'utilisateur initiateur, mais il peut être signalé indirectement au moyen d'une valeur de résultat.

Lorsque plus d'une seule valeur de statut de jeton décrit la relation entre un utilisateur donné et un identificateur de jeton spécifié, l'ordre de préférence doit être le suivant. La valeur d'état *autodestinataire* doit être signalée la première, si elle peut être insérée, afin de rappeler aux utilisateurs qu'ils doivent toujours répondre à une indication de cession de jeton MCS-TOKEN-GIVE. La valeur préférée ensuite est *autodonnant*, afin de rappeler aux utilisateurs qu'ils sont engagés dans une opération non terminée, puis *autosaisi* et *auto-inhibé*. Les valeurs d'état restantes viennent à la fin: elles rendent compte de l'état actuel du jeton résultant de son seul usage par d'autres correspondants.

Un fournisseur de service MCS se fonde sur la préférence spécifiée des valeurs d'état de jeton pour mettre à jour correctement ces états dans sa base de données.

### 13.9 Notification de capacités

Grâce aux mécanismes d'extensibilité intégrés au protocole V3, il est possible de mettre des capacités en œuvre dans le service MCS en ajoutant des éléments aux unités PDU T.125, ou même en ajoutant de nouvelles unités PDU, sans recourir à une nouvelle version du protocole. Un nœud donné peut influencer sur les paramètres des unités PDU qu'il comprend et relayer simplement les paramètres non reconnus sans les modifier.

Afin d'exercer une capacité donnée, les nœuds qui constituent le domaine peuvent devoir satisfaire à certaines conditions. Ces conditions, ou règles, pourraient notamment porter sur le degré de participation des autres nœuds, ou encore sur la présence ou l'absence de nœuds de la version 2. Par exemple, une certaine capacité peut exiger la participation de tous les nœuds du domaine afin de pouvoir s'exercer. Par ailleurs, une autre capacité peut n'exiger la participation que de deux nœuds ou plus pour qu'une capacité soit utile. En d'autres termes, un ensemble de règles est associé à chaque capacité, et un nœud donné doit prendre une décision quant à l'exercice de la capacité selon ces règles et l'état du domaine.

Les unités PDU **CapabilitiesNotificationRequest** et **CapabilitiesNotificationIndication** offrent le moyen de faciliter la notification de capacités dans un sommet V3. L'unité PDU **CapabilitiesNotificationRequest** véhicule les capacités que divers nœuds veulent ajouter à la liste de capacités du sommet ou supprimer de cette liste. Cette unité PDU est également utilisée pour avertir d'autres nœuds des changements d'état touchant les connexions V2. L'unité PDU **CapabilitiesNotificationIndication** permet de modifier la liste de capacités du sommet et de fournir de l'information sur l'état des connexions V2. Dans le reste du présent sous-paragraphe, une unité PDU **CapabilitiesNotificationRequest** sera appelée "**Demande**", et une unité PDU **CapabilitiesNotificationIndication** sera appelée "**Indication**".

#### 13.9.1 Identificateurs de capacité

Une capacité est un comportement de niveau supérieur du système, qui requiert vraisemblablement la coopération de deux nœuds ou plus. A mesure que des capacités se développent, un identificateur de capacité leur est assigné. Cet identificateur, qui a pu être normalisé, est décrit dans la T.125. Un identificateur de capacité désigne ainsi un ensemble de paramètres d'unité PDU (pas nécessairement dans la même unité PDU), peut-être un ensemble de valeurs pour ces paramètres et le traitement requis de ces unités PDU par les nœuds participants.

#### 13.9.2 Sommets de la version 3

On ne peut pas s'attendre à ce qu'une hiérarchie de domaines comprenne uniquement des nœuds V3. En raison de la stipulation de T.125 (révisée) selon laquelle la version du protocole doit être négociée connexion par connexion, il est possible qu'un domaine soit constitué d'une combinaison de nœuds V2 et V3. La T.125 (révisée) stipule aussi qu'un nœud à capacité V3 doit servir de source et de collecteur uniquement pour des unités PDU V2 si sa connexion ascendante négocie avec son nœud supérieur selon le protocole V2. Ces nœuds sont équivalents aux nœuds V2 et sont considérés comme tels pour le reste du sous-paragraphe 13.9.

Un sommet V3 est un groupe de nœuds V3 connectés de façon contiguë et situés au sommet de la hiérarchie de domaines. Le fournisseur supérieur du domaine doit être un nœud V3 pour qu'un sommet V3 puisse exister. Les nœuds qui constituent le sommet sont capables d'échanger de l'information sur les capacités, mais ils ne peuvent pas échanger cette information avec l'un ou l'autre des nœuds V2 du domaine. Cela peut ou non nuire à l'utilisation d'une capacité donnée. A l'intérieur du sommet V3, le fournisseur supérieur du domaine est appelé "fournisseur de sommet V3".

### **13.9.3 Règles de sélection de capacité**

Il existe plusieurs facteurs influant sur le fait qu'une capacité donnée doit ou non être exercée dans un sommet V3. Ces facteurs constituent essentiellement la mesure de la participation des autres nœuds à cette capacité. Supposons par exemple qu'une capacité exige que tous les nœuds du domaine effectuent simultanément une tâche donnée. Une unité PDU pourrait être échangée entre tous les nœuds qui contenaient une référence temporelle quant au moment où la tâche en question devait être effectuée. Pour ce faire, tous les nœuds du domaine devraient être participants, par définition. Cela ne serait pas possible si le domaine contenait soit un nœud V2, soit un nœud V3 dans lequel cette capacité ne serait pas mise en œuvre. Par ailleurs, certaines capacités peuvent ne pas exiger une participation à l'échelle d'un domaine ou à l'échelle d'un sommet.

#### **13.9.3.1 Présence de nœuds V2 dans le domaine**

Tout nœud V3 d'un sommet sera tenu au courant du fait qu'il dispose ou non d'une connexion descendante vers un nœud V2. Grâce aux modifications proposées de protocole V3, un nœud V3 pourra avertir tous les nœuds V3 du sommet si un nœud V2 est connecté au sommet. Certaines capacités peuvent être inutiles dans un domaine où des nœuds V2 sont présents. Les autres capacités peuvent n'être aucunement touchées. En sachant si un nœud V2 est connecté au sommet, un nœud particulier peut prendre une décision quant à l'exercice d'une capacité donnée.

#### **13.9.3.2 Participation de nœud V3**

Pour pouvoir s'exercer, les capacités exigeront un certain degré de participation de la part des nœuds du sommet V3. Chaque capacité sera catégorisée par sa participation "globale" ou "partielle". La participation globale signifie que chaque nœud du sommet doit participer pour que la capacité puisse s'exercer. La participation partielle exige la coopération de deux ou plusieurs nœuds. Cette exigence de participation se traduit par un "indicateur de participation". Les capacités à indicateur de participation partielle sont appelées "capacités partielles". Les capacités à indicateur de participation globale sont appelées "capacités globales". En connaissant l'indicateur de participation correspondant à une capacité donnée, le fournisseur de sommet peut décider si la capacité en question doit être disponible au sommet.

#### **13.9.3.3 Nœuds V3 intervenants**

En raison des marques d'extension ASN.1, il est possible qu'un nœud V3 reçoive une unité PDU avec des composantes que le nœud ne reconnaît pas. Le nœud reconnaîtra toutefois l'unité PDU à laquelle ces composantes sont associées et relaiera les composantes non reconnues avec l'unité PDU sans les modifier.

Outre l'extension des unités PDU existantes, le protocole V3 permet l'addition de nouvelles unités PDU. Un problème se pose toutefois du fait que les nœuds qui ne reconnaissent pas les nouvelles unités PDU ne savent absolument pas comment les acheminer. Dans cette situation, il serait avantageux qu'un nœud ayant servi de source pour une telle unité PDU sache si les nœuds intervenants sur le trajet vers le fournisseur de sommet V3 peuvent prendre en charge l'unité PDU. Dans l'affirmative, le nœud émetteur pourrait supposer que chaque nœud sur le trajet saurait comment traiter l'unité PDU. Les modifications proposées au protocole V3 portent notamment sur une ressource par laquelle un nœud quelconque, pour chaque capacité, est informé du fait que la capacité est prise en charge par les nœuds intervenants.

#### **13.9.3.4 Classe de capacité**

Une fois qu'il a été décidé qu'une certaine capacité doit être mise à la disposition du sommet V3, il doit exister une ressource permettant au besoin d'associer une valeur à cette capacité. C'est ce que permet la classe de capacité. Chaque capacité serait classée comme MIN, MAX ou NULLE. Pour les

capacités de la classe MIN, le fournisseur de sommet V3 est chargé de déterminer la valeur minimale à utiliser pour une capacité en sélectionnant la valeur minimale demandée pour chaque nœud participant. De même pour les capacités de la classe MAX, le fournisseur de sommet V3 est chargé de déterminer la valeur maximale à utiliser. Aucune valeur n'est associée à une capacité de classe NULLE.

Une fois qu'une capacité a été normalisée, il est prévu que la classe de capacité associée soit également normalisée. Chaque fois qu'une capacité donnée est demandée d'un nœud de niveau inférieur, le fournisseur de sommet ne doit traiter qu'une seule classe de capacité associée à la capacité en question. Si le fournisseur de sommet reçoit une demande de capacité dont la classe de capacité diffère de celle d'une demande originale, le fournisseur de sommet doit considérer qu'il s'agit d'une condition d'erreur et ne pas tenir compte de la demande.

#### 13.9.4 Etablissement du fournisseur de sommet V3

Chaque sommet V3 est initialement établi par un seul nœud V3. Le nœud V3 sera le "fournisseur supérieur" d'un domaine à un seul nœud. Ce fournisseur de sommet ne connaît que ses propres capacités.

Chacune des capacités du fournisseur de sommet exige une participation partielle ou globale à l'intérieur de ce sommet à un nœud. Aucune des capacités partielles ne figure dans la liste de capacités du sommet, car elles requièrent, par définition, la participation de deux ou plusieurs nœuds. Il reste à déterminer si les capacités exigeant une participation globale doivent figurer dans la liste. On peut avancer que "tous les nœuds" du sommet sont des nœuds participants, et donc que les capacités globales de tous les nœuds doivent figurer dans la liste du sommet. Si l'une ou l'autre de ces capacités globales peut s'exercer, il doit évidemment exister une capacité n'exigeant la participation que d'un seul nœud. De telles capacités dépassent la portée de la présente Recommandation. Si un seul nœud permet d'exercer une capacité quelconque, ce protocole ne s'applique pas à la capacité en question. C'est pourquoi la liste de capacités du sommet est considérée comme vide lorsque le sommet est constitué d'un seul nœud V3.

#### 13.9.5 Extension du sommet V3

Une fois que le fournisseur de sommet V3 a été établi, le sommet s'étend à mesure que d'autres nœuds V3 se connectent au fournisseur de sommet. Un nouveau nœud feuille génère une **Demande** destinée au fournisseur de sommet. Cette unité PDU contient les capacités et les aspects de ces capacités, que le nœud feuille désire voir figurer sur la liste de capacités du sommet.

A partir de cette unité PDU, le fournisseur de sommet peut effectuer deux choses. Premièrement, il peut formuler de l'information d'état sur les capacités demandées par la nouvelle connexion. Deuxièmement, à partir de l'information d'état à l'échelle du sommet, il peut exercer un jugement sûr quant aux capacités qui devraient apparaître dans la liste de capacités du sommet. Une **Indication** est ensuite générée et transmise vers le bas afin d'informer le nœud feuille du contenu de la liste du sommet, ainsi que de l'état des nœuds V2 connectés au sommet.

De cette façon, le sommet V3 peut continuer de se développer. A mesure que de nouveaux nœuds feuilles se connectent aux nœuds intermédiaires, une **Demande** est générée et envoyée. L'information contenue dans cette unité PDU permettra au nœud intermédiaire de mettre à jour son information d'état. A partir de l'information d'état correspondant à son sous-arbre et du contenu courant de la liste de capacités du sommet, le nœud intermédiaire prendra les mesures qui s'imposent, qui pourraient entraîner la transmission d'une **Demande** en amont ou la transmission d'**Indications** en aval. Une **Demande** peut éventuellement se rendre jusqu'au fournisseur de sommet. Si le fournisseur de sommet établit que la liste de capacités du sommet doit être modifiée, il générera et transmettra une **Indication** en aval afin d'informer les autres nœuds du sommet.

### 13.9.6 Traitement des unités PDU à un nœud feuille

Une fois qu'un nœud feuille V3 établit une connexion à un nœud V3 supérieur, il doit informer le nœud supérieur des capacités qu'il prend en charge. Pour chaque capacité demandée, le nœud feuille spécifiera l'identificateur et la classe de capacité. S'il est normalisé, l'identificateur spécifié est celui qui est assigné à la capacité dans T.125. Si la classe appartient à la catégorie MIN ou MAX, le nœud feuille doit également spécifier une valeur associée. Si la classe est NULLE, aucune valeur n'est spécifiée.

De plus, le nœud feuille doit spécifier l'indicateur de participation pour chaque capacité. Pour toute capacité partielle, le nœud feuille spécifie la valeur 1, indiquant au nœud supérieur que seul le nœud feuille lui-même a demandé la capacité inférieure à cette nouvelle connexion.

Ces capacités sont regroupées dans la composante Liste d'ajout (AddList) d'une **Demande** unique. Le fanion V2NodePresent sera remis à 0 dans la **Demande**. Comme il s'agit d'un nœud feuille dont l'adhésion est nouvelle, aucun nœud V2 ne lui est connecté. La composante Liste de suppression (RemoveList) sera également vide. Après le regroupement, cette unité PDU est transmise au nœud supérieur.

Le nœud feuille nouvellement connecté ne connaît aucunement le contenu de la liste de capacités du sommet. Le nœud peut s'attendre à être informé du contenu de cette liste par une **Indication** unique provenant du nœud supérieur. Le contenu de la liste se trouvera dans la composante Liste d'ajout de l'**Indication**. La Liste de suppression sera vide. Le fanion V2NodePresent indiquera au nœud feuille si un nœud V2 est à ce moment connecté à un nœud quelconque du sommet. A partir de ce point et en aval, le nœud feuille sera informé de toute modification apportée à la liste du sommet par une **Indication** provenant du nœud supérieur. Si l'état des connexions aux nœuds V2 change, le nœud feuille en sera également informé par une **Indication**.

Pour chaque capacité de la composante Liste d'ajout d'une **Indication** reçue, le nœud feuille sera informé du fait que la capacité est ou non prise en charge par le fournisseur de sommet et que la capacité en question est ou non prise en charge par chaque nœud intervenant entre le nœud feuille et le fournisseur de sommet. Avec cette information, le nœud feuille peut décider d'exercer ou non une capacité donnée.

Tant qu'un nœud demeure un nœud feuille, il n'a plus jamais besoin d'envoyer une **Demande**. Au moment où un autre nœud (V2 ou V3) se connecte en amont à ce nœud, il devient un nœud intermédiaire. Le traitement des unités PDU à un nœud intermédiaire est décrit au 13.9.7.

### 13.9.7 Unités PDU de traitement à un nœud intermédiaire

Lorsqu'un nœud intermédiaire reçoit une **Demande** d'une connexion descendante, il met d'abord à jour l'information d'état correspondant à la connexion associée. Le nœud adresse ensuite chaque capacité des composantes Liste d'ajout et Liste de suppression, ainsi que le fanion V2NodePresent. Bien que plusieurs capacités exigeant une action puissent arriver dans une **Demande** unique, elles sont traitées indépendamment. Par conséquent, le nœud intermédiaire peut avoir besoin de transmettre plusieurs éléments d'information à des nœuds adjacents. Le nœud combinera ces actions pour que pas plus d'une unité PDU ne soit transmise à une connexion quelconque par suite du traitement de la **Demande** originale.

#### 13.9.7.1 Tenue à jour de l'information d'état

Les nœuds intermédiaires tiennent à jour un ensemble d'information d'état pour chaque connexion descendante. Dès la réception d'une **Demande** provenant d'un sous-nœud, le nœud intermédiaire met à jour l'information d'état associée à la connexion en question. Deux listes font partie de cette information. La première est constituée de toutes les capacités partielles demandées par les nœuds inférieurs à la connexion en question. Chacune de ces capacités est étiquetée par un compte de 1

ou 2. La valeur 1 signifie que la capacité en question n'a été demandée que par un seul nœud inférieur à la connexion. La valeur 2 signifie que deux nœuds ou plus ont demandé la capacité.

La deuxième liste est constituée d'un ensemble de capacités globales. Comparativement aux capacités partielles, les capacités globales figurent dans la liste uniquement si elles ont été demandées par chacun des nœuds inférieurs à la connexion. Pour toute capacité figurant dans l'une ou l'autre des deux listes, le nœud intermédiaire suit également la classe de la capacité ainsi que toute valeur associée à la classe.

Enfin, un fanion est tenu à jour pour chaque connexion descendante. Ce fanion indique si un nœud V2 est connecté au sommet inférieur à la connexion en question.

L'algorithme de mise à jour de l'information d'état est simple. Si le nœud intermédiaire reçoit une demande d'ajout d'une capacité qui ne figure pas dans l'une ou l'autre des deux listes, la capacité est ajoutée à la liste appropriée, selon l'indicateur de participation. Si une demande est reçue en vue de supprimer une capacité qui figure dans l'une des deux listes, la capacité est éliminée de la liste en question.

Si le nœud intermédiaire reçoit une demande d'ajout d'une capacité qui figure déjà dans l'une des deux listes, le traitement dépend de la classe et de l'indicateur de participation de la capacité. Dans le cas de la classe MIN ou MAX, la valeur de la classe dans l'information d'état est mise à jour lorsqu'elle diffère de celle de la **Demande**. Dans le cas d'un indicateur de participation partielle, la valeur associée de l'information d'état est mise à jour lorsqu'elle est différente.

Enfin, si le nœud intermédiaire reçoit une **Demande** dans laquelle le fanion V2NodePresent est présent, la valeur de ce fanion est comparée à celle du fanion de l'information d'état. En cas de différence, l'information d'état est mise à jour.

### 13.9.7.2 Traitement de l'unité PDU CapabilitiesNotificationRequest

Un nœud intermédiaire peut transmettre une **Demande** dans les conditions suivantes:

- 1) le nœud reçoit une **Demande** d'une connexion descendante;
- 2) le nœud détecte la perte d'une connexion V3;
- 3) le nœud détecte une connexion directe/déconnexion par rapport à un nœud V2.

Les sous-paragraphes ci-dessous décrivent comment un nœud intermédiaire traite ces conditions.

#### 13.9.7.2.1 Capacités globales dans la Liste d'ajout

Lorsqu'un nœud intermédiaire reçoit une **Demande** d'ajout d'une capacité globale, le nœud examine sa copie locale de la liste de capacités du sommet afin de déterminer si la capacité demandée figure déjà dans la liste. L'action suivante effectuée par le nœud dépend du fait que la capacité se trouve ou non dans la liste.

Lorsque la capacité ne figure pas dans la liste, le nœud intermédiaire doit déterminer s'il s'agit de la première **Demande** provenant de la connexion. Si tel est le cas, cela signifie que la **Demande** provient d'un nœud feuille nouvellement connecté et qu'aucune action ultérieure n'est requise. Sinon, il peut être nécessaire de transmettre une **Demande** d'ajout de la capacité, selon l'information d'état de l'une ou l'autre des connexions descendantes. Si le nœud intermédiaire peut, à partir de son information d'état, déterminer que la capacité est maintenant prise en charge par tous les nœuds de son sous-arbre, la **Demande** est alors transmise. Cette action pourrait éventuellement entraîner l'ajout de la capacité à la liste du sommet.

Le fait que la capacité demandée figure dans la liste implique que chaque nœud du sommet prend en charge la capacité en question. Le traitement ultérieur dépend de la classe de la capacité. Si la capacité est de la classe MIN, le nœud intermédiaire doit déterminer si la valeur associée est la

moindre des valeurs demandées par chaque nœud de ce sous-arbre pour la capacité en question. De même, si la capacité est de la classe MAX, le nœud intermédiaire doit déterminer si la valeur associée est la plus élevée des valeurs demandées par les autres nœuds. Si l'une ou l'autre de ces deux conditions est satisfaite, le nœud intermédiaire transmettra une **Demande** en amont, pour demander d'ajouter la capacité avec sa valeur associée. Il sera ainsi possible d'indiquer au nœud supérieur s'il existe une valeur MIN inférieure ou une valeur MAX supérieure dans son sous-arbre pour la capacité en question. Si la capacité est de la classe NULLE, aucune **Demande** ne sera requise. Enfin, le nœud intermédiaire produira une **Indication** à transmettre au nœud inférieur pour l'avertir d'ajouter la capacité (et sa valeur associée, le cas échéant) à sa copie locale de la liste de capacités du sommet.

Lorsqu'un nœud intermédiaire reçoit une **Demande** d'un nœud feuille nouvellement connecté, le nœud intermédiaire peut être forcé d'effectuer une action en raison de capacités globales NON demandées par le nœud feuille. La première demande provenant d'un nœud feuille contient la liste de toutes les capacités globales prises en charge par le nœud en question. Si le nœud intermédiaire dispose, dans sa copie locale de la liste du sommet, de capacités globales non demandées par le nœud feuille, alors le nœud feuille ne prend pas ces capacités en charge. Par définition, ces capacités ne sont plus à la disposition du sommet et doivent être supprimées de la liste du sommet. A ce moment, le nœud intermédiaire produit une **Demande** et indique cette capacité dans la Liste de suppression. Cette unité PDU est ensuite transmise en amont, informant le nœud supérieur que la capacité n'est plus prise en charge dans le sous-arbre. Si le nœud intermédiaire est associé à d'autres connexions descendantes, il produira une **Indication** à transmettre en aval. Cette **Indication** contiendra la capacité indiquée dans la Liste de suppression. Enfin, le nœud intermédiaire supprimera la capacité de sa propre copie de la liste de capacités du sommet.

#### 13.9.7.2.2 Capacités globales dans la Liste de suppression

La réception par un nœud intermédiaire d'une **Demande** de suppression d'une capacité globale implique que, quelque part en dessous de la connexion, un nœud qui ne prend pas la capacité en charge a adhéré au sommet. Cette capacité ne doit maintenant plus être à la disposition du sommet. Le traitement nécessaire dans ce cas-ci est simple. Le nœud génère une **Demande** afin que la capacité soit supprimée, et il la transmet en amont. S'il existe d'autres connexions descendantes, le nœud intermédiaire produit une **Indication** demandant que la capacité soit supprimée, et il la transmet à toutes les connexions descendantes (sauf la connexion d'où provenait la **Demande**). Le nœud la supprime ensuite de la copie locale de la liste de capacités du sommet.

#### 13.9.7.2.3 Capacités partielles dans la Liste d'ajout

Si un nœud intermédiaire reçoit une **Demande** d'ajout de capacité partielle, le nœud doit déterminer s'il s'agit de la première demande, de la deuxième demande ou d'une demande ultérieure à la deuxième pour la capacité. C'est ce qu'il est possible de déterminer à partir de l'information d'état du nœud intermédiaire.

S'il s'agit de la première demande, le nœud intermédiaire produira une **Demande** d'ajout de la capacité. La valeur fournie avec l'indicateur de participation partielle sera de 1, ce qui signifiera qu'une seule demande a été effectuée pour la capacité en question dans le sous-arbre du nœud intermédiaire. S'il s'agit de la deuxième demande, le nœud intermédiaire produira une **Demande** d'ajout de la capacité, et la valeur 2 sera fournie pour l'indicateur de participation partielle. Si le nœud intermédiaire détermine que cette capacité a déjà été demandée au moins deux fois, aucune **Demande** n'est nécessaire.

Ensuite, le nœud intermédiaire doit examiner sa copie locale de la liste du sommet pour voir si la capacité se trouve dans la liste. Si tel est le cas, le nœud produira une **Indication** avec la capacité de la composante Liste d'ajout et il la transmettra à la connexion demandeuse.

Le nœud intermédiaire doit également tenir compte de la classe de capacité. Cet aspect du traitement s'apparente à celui expliqué précédemment pour les capacités globales. Si une capacité présente la valeur MIN la plus faible du sous-arbre ou la valeur MAX la plus élevée du sous-arbre, le nœud supérieur doit alors en être informé par une **Demande**.

#### 13.9.7.2.4 Capacités partielles dans la Liste de suppression

Lorsqu'un nœud intermédiaire reçoit une **Demande** de suppression d'une capacité partielle, le nœud doit déterminer si la suppression influencerait sur le nombre de demandes pour la capacité dans le sous-arbre. Si tel est le cas, le nœud intermédiaire doit produire une **Demande** pour la capacité en question afin que la capacité soit ajoutée avec un indicateur de participation partielle de 1 ou que la capacité soit supprimée complètement, selon le changement d'état.

Si le nœud intermédiaire peut déterminer l'effet direct sur la liste du sommet, le nœud peut adapter sa propre copie locale de la liste et produire des **Indications** pour avertir les autres connexions descendantes. Si le nœud ne peut pas déterminer l'effet, tout réglage s'effectuera éventuellement à partir d'un nœud supérieur pouvant déterminer l'effet.

#### 13.9.7.2.5 Traitement du fanion V2Node Present

Bien que cela ne soit pas nécessaire, un nœud intermédiaire recevra en pratique une **Demande** qui contient un fanion V2NodePresent ou une liste de capacités (Liste d'ajout ou Liste de suppression), mais habituellement pas les deux.

Pour chaque connexion descendante, un nœud intermédiaire assure la mise à jour de l'état de connexion d'un nœud V2 au sommet en dessous de la connexion. La première fois qu'un nœud intermédiaire apprend qu'un nœud V2 s'est connecté à partir du bas, y compris directement au nœud lui-même, il génère une **Demande** contenant seulement le fanion V2NodePresent, qui est remis à 1. Cette **Demande** est ensuite transmise en amont. Une **Indication** de contenu identique sera transmise à toutes les connexions descendantes, sauf pour la connexion qui pourrait avoir produit la **Demande** originale. De cette façon, chaque nœud du sommet apprend l'existence du nœud V2. Chaque nœud peut en tenir compte au moment de décider d'exercer ou non une capacité donnée.

De même, un nœud intermédiaire peut recevoir une **Demande** informant le nœud que toutes les connexions V2 précédentes ont été coupées du sous-arbre. Avec chacune de ces **Demandes**, le nœud intermédiaire peut déterminer si un nœud V2 est maintenant connecté à son sous-arbre. Si d'autres existent encore, le nœud intermédiaire ne fait rien. Si tous les nœuds V2 sont déconnectés, le nœud intermédiaire produit une **Demande** avec le fanion V2NodePresent à 0, et il la transmet en amont.

Si tous les nœuds V2 connectés au sommet se déconnectent, le fournisseur de sommet le détectera en veillant à ce qu'aucune connexion descendante ne contienne un nœud V2 dans son sous-arbre et qu'aucun nœud V2 ne soit connecté directement au fournisseur de sommet. Dans ces conditions, le fournisseur de sommet produit une **Indication** avec fanion V2NodePresent réinitialisé, et il la transmet dans tout le sommet.

#### 13.9.7.2.6 Traitement des déconnexions de nœud V3

Lorsqu'un nœud donné du sommet détecte qu'une connexion descendante a été coupée, toute l'information d'état relative à cette connexion est ignorée. L'information d'état relative aux autres connexions descendantes pourrait, le cas échéant, influencer sur le contenu de la liste de capacités du sommet.

Si l'une quelconque des capacités globales est prise en charge par chaque nœud du sommet, à l'exception d'un ou de plusieurs nœuds en dessous de la connexion coupée, les capacités en question peuvent maintenant être ajoutées à la liste du sommet. Si le nombre de demandes d'une capacité partielle du sommet chute à moins de deux par suite de la coupure de nœuds en dessous de la

connexion coupée, les capacités en question doivent être supprimées du sommet. Le processus d'adaptation à la connexion coupée commence par le nœud qui détecte la coupure de la connexion. Le nœud assure le traitement de la situation comme s'il s'agissait d'une **Demande** reçue pour que chaque capacité de l'information d'état visant la connexion soit supprimée. La connexion assurerait ensuite le traitement de la façon décrite au 13.9.7.2.

### 13.9.7.3 Traitement de l'unité PDU CapabilitiesNotificationIndication

Il arrive parfois qu'un nœud intermédiaire reçoive une **Indication** de sa connexion ascendante. L'indication informe le nœud intermédiaire d'ajouter des capacités à la liste du sommet ou d'en supprimer. Ces modifications seront apportées à la liste de sommet "locale", et l'indication sera dirigée vers chaque connexion descendante. S'il s'agit de la première **Indication** envoyée sur une connexion, le nœud intermédiaire transmettra simplement la liste de sommet complète par l'intermédiaire de la composante Liste d'ajout de l'**Indication**. Si un nœud intermédiaire ne prend pas en charge une capacité, le nœud doit réinitialiser le fanion qui indique la prise en charge du nœud intermédiaire.

### 13.9.7.4 Traitement des unités PDU au fournisseur de sommet

La façon dont le fournisseur de sommet traite les unités PDU est presque identique à celle d'un nœud intermédiaire. Dès la réception d'une **Demande** provenant d'une connexion descendante, le fournisseur de sommet mettra à jour l'information d'état de la connexion associée comme pour un nœud intermédiaire, de la façon décrite au 13.9.7.1. Le fournisseur de sommet traite les capacités globales et les capacités partielles comme pour les nœuds intermédiaires, avec l'exception évidente qu'aucune **Demande** n'est envoyée à un nœud supérieur. Avec chaque demande d'ajout ou de suppression de capacité, le fournisseur de sommet décide si la liste du sommet doit être modifiée. Si tel est le cas, le fournisseur de sommet produira une **Indication** contenant les capacités appropriées à ajouter ou à supprimer, et il transmettra l'unité PDU en aval par les connexions appropriées.

Ce protocole a été conçu afin que chaque nœud déterminant qu'il est nécessaire de modifier la liste de capacités du sommet déclenche la modification en avertissant ses connexions descendantes ainsi que son nœud supérieur. De cette façon, la liste du sommet est mise à jour par l'intermédiaire du sommet.

Lorsqu'un nœud ne dispose pas de suffisamment d'information pour déterminer si une modification peut être apportée à la liste du sommet, il envoie une **Demande** vers le prochain nœud de niveau supérieur, qui peut à son tour être à même de prendre une décision. Toutes les décisions relatives à cette question peuvent ultimement être résolues par le fournisseur de sommet.

## 13.10 Arbitrage des versions de protocole

La présente Recommandation définit trois versions de protocole (voir détails au paragraphe 9). Toutefois, la Recommandation T.120 spécifie que la version 1 est désuète. Un nœud peut refuser de prendre en charge la version 1 en exigeant la version 2 (ou une version supérieure) durant l'établissement de la connexion (si un nœud éloigné prend en charge *seulement* la version 1, la connexion peut être rejetée). Dans le cas des nœuds qui veulent prendre en charge la version 1 à des fins de compatibilité amont, il est proposé de n'autoriser aucun mélange d'autres versions de protocole du même domaine [selon les règles de la Recommandation T.125 (1994)].

La Recommandation T.120 sera (ou a été) modifiée afin de permettre les versions 2 ou 3. Le paramètre de numéro de la version de protocole des paramètres de domaine sera utilisé pour négocier la capacité des unités PDU de la version 3 à servir de source et de collecteur. Un nœud qui met en œuvre la version 3 doit également pouvoir servir de source et de collecteur pour les unités PDU de la version 2.

Le paramètre de numéro de version du protocole est négocié indépendamment pour chaque connexion (avec une restriction, décrite ci-dessous). Chaque nœud de la version 3 doit tenir à jour l'information concernant les versions de protocole sélectionnées par ses connexions ascendantes et descendantes. La traduction entre versions doit être fournie pour tous les nœuds autres que feuilles qui mettent en œuvre la version 3 du protocole (voir détails au 13.11).

Un nœud capable de prendre en charge la version 3 et qui a négocié l'utilisation de la version 2 pour sa connexion ascendante négociera *seulement* l'utilisation de la version 2 pour ses connexions descendantes. Cela a pour effet de concentrer l'utilisation de la version 3 au sommet de la hiérarchie de domaine (cette partie de l'utilisation de la version 3 renvoie à ce qu'on appelle "Sommet V3").

Un nœud capable de prendre en charge la version 3 et qui a négocié l'utilisation de la version 3 pour sa connexion ascendante tentera de négocier l'utilisation de la version 3 pour ses connexions descendantes (bien que la version 2 soit acceptée au besoin).

### **13.11 Interfonctionnement des versions de protocole**

La traduction de la version 2 à la version 3 s'effectue par la copie des paramètres d'une unité PDU de la version 2 dans une unité PDU équivalente de la version 3.

La traduction de la version 3 à la version 2 s'effectue par la copie des paramètres d'une unité PDU de la version 3 à une unité PDU équivalente de la version 2. Les paramètres de la version 3 qui ne correspondent pas à des paramètres valides de la version 2 sont ignorés. Les unités PDU de la version 3 sans équivalent de la version 2 sont également ignorées.

Lorsqu'un nœud reçoit une unité PDU de la version 3 qu'il ne reconnaît pas, l'unité PDU est ignorée (il ne s'agit pas d'une erreur).

Lorsqu'un nœud reçoit une unité PDU de la version 3 *qu'il reconnaît*, l'unité PDU est traitée normalement. Si l'unité PDU contient des paramètres que le récepteur ne reconnaît pas, il n'en est tout simplement pas tenu compte. Si l'unité PDU doit être acheminée vers un autre nœud utilisant la version 3 du protocole, tous les paramètres sont acheminés sans modification (y compris ceux qui n'ont pas été reconnus localement).

Considérations visant l'utilisation de paramètres non normalisés dans des conférences multiprotocoles:

NOTE 1 – Avant d'utiliser des paramètres non normalisés, le responsable de la mise en œuvre doit s'assurer que les nœuds de la version 2 et de la version 3 peuvent interfonctionner correctement. Aucune information comprenant une unité PDU de la version 3 sous la forme de paramètre non normalisé, ou un paramètre normalisé au-delà du marqueur d'extension, ne peut être livré à un nœud de la version 2 par l'intermédiaire d'une unité PDU de la version 2. Les paramètres non normalisés s'utilisent donc seulement avec les extensions dont l'absence n'influera pas sur l'exploitation de base aux nœuds V2 ou V3 (exploitation avec sécurité intégrée). Les procédures précises que peut utiliser une unité MCU ou d'autres nœuds intervenants doivent être examinées en détail afin de garantir la réussite de l'interfonctionnement.

NOTE 2 – Il importe de se rappeler que l'utilisation de paramètres non normalisés constitue seulement une solution provisoire en attendant que des fonctions équivalentes aient été normalisées.

NOTE 3 – La présente Recommandation décrit un mécanisme d'échange d'information entre nœuds visant à confirmer des capacités non normalisées disponibles à un sommet de la version 3. Les aspects du dimensionnement doivent être envisagés, la probabilité étant réduite que, dans les conférences les plus importantes, l'une ou l'autre des capacités non normalisées s'avère commune.

NOTE 4 – Le coût du traitement de la traduction entre unités PDU V2 et V3 doit être envisagé. Outre le fait qu'une unité MCU doit générer des unités PDU de la version 2 en éliminant certains paramètres d'une unité PDU équivalente de la version 3, en plus de produire des unités PDU de la version 3 en générant des paramètres provenant d'une unité PDU équivalente de la version 2, le coût de l'inclusion effective de

paramètres non normalisés dans des unités PDU est relativement élevé. Le codage minimal semble faire appel à environ 8 octets, à l'exclusion des données véhiculées dans le paramètre. Cette longueur est égale à celle de l'en-tête MCS des paramètres normalisés dans une unité PDU de données et impose une consommation supplémentaire de largeur de bande. A la réception d'un quelconque paramètre non normalisé, un nœud de la version 3 doit analyser sa clé et examiner ensuite les codes du fabricant pour déterminer sa signification, si elle est connue. Ce processus consomme de la puissance de traitement, ce qu'il importe de réduire autant que possible. Le présent sous-paragraphe spécifie les procédures que doivent suivre les nœuds qui ne comprennent pas les paramètres non normalisés.

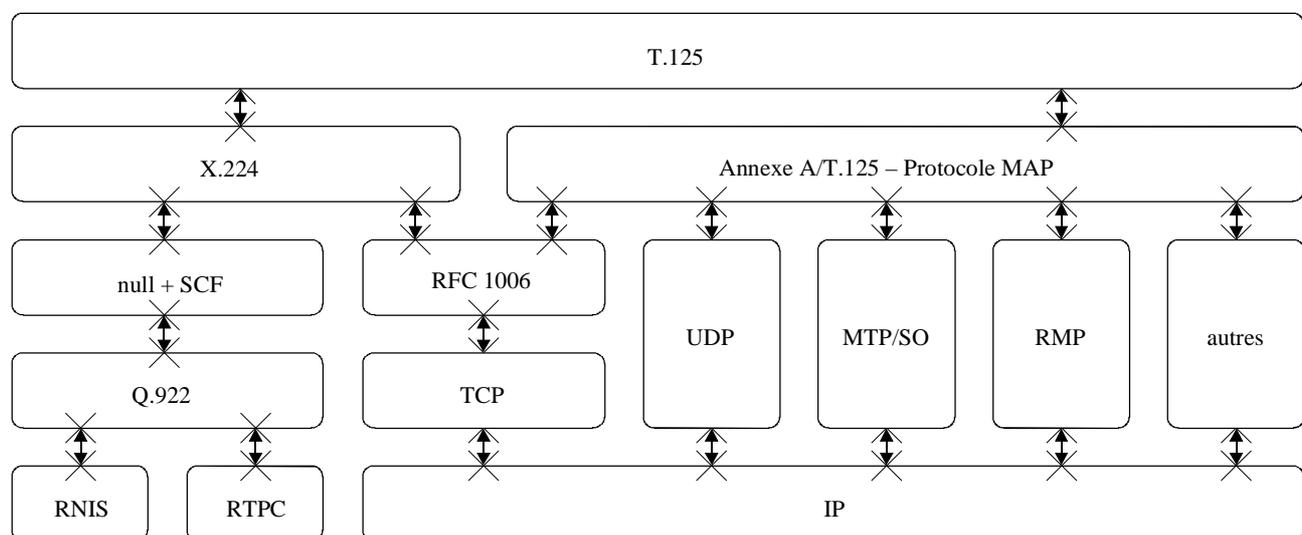
## ANNEXE A

### Protocole d'adaptation de multidiffusion

#### A.1 Domaine d'application

La présente annexe définit un protocole qui permet l'utilisation de services multidiffusion dans une conférence T.120. Ce protocole est indépendant du genre de réseau dans lequel il est utilisé. Les questions propres à un réseau particulier, comme le format d'adresse et l'attribution d'adresse de groupe de multidiffusion, dépassent la portée de la présente annexe. Ce protocole permet aussi le transfert de données non fiables.

La Figure A.1-1 montre comment le protocole d'adaptation de multidiffusion (MAP) s'intègre au modèle T.120. Le protocole d'adaptation de multidiffusion prend en charge la traduction des services T.122 en services multidiffusion. Divers protocoles de livraison de données comme TCP, UDP, MTP/SO et RMP peuvent figurer sous le protocole d'adaptation de multidiffusion (MTP/SO et RMP sont des exemples de propositions en vue de la multidiffusion fiable au moyen de réseaux IP).



T1600670-97

**Figure A.1-1/T.125 – Modèle T.120 montrant la relation entre le protocole d'adaptation de multidiffusion et les autres couches**

#### A.2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions de la présente annexe. Au moment de

la publication, les éditions indiquées étaient en vigueur. Toute Recommandation ou autre Norme est sujette à révision; tous les utilisateurs de la présente annexe sont donc invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO tiennent à jour des registres des Normes internationales en vigueur. Le Secrétariat de l'UIT tient à jour une liste des Recommandations UIT-T en vigueur.

- Recommandation T.35 du CCITT (1991), *Procédure d'attribution des codes définis par le CCITT dans le cas des moyens non normalisés.*
- Recommandation UIT-T T.120 (1996), *Protocoles de données pour conférence multimédia.*
- Recommandation UIT-T T.122 (1998), *Service de communication multipoint – Définition du service.*
- Recommandation UIT-T T.123 (1996), *Piles protocolaires de données propres au réseau pour conférences multimédias.*
- Recommandation UIT-T X.214 (1995) | ISO/CEI 8072:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition du service de transport.*
- Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*

### A.3 Définitions

La présente annexe définit les termes suivants:

**A.3.1 flux de données:** train de données transmis par un émetteur à tous les récepteurs intéressés. Les flux de données sont identifiés de manière unique au moyen des cinq champs suivants: identificateur d'émetteur; identificateur de canal; niveau de fiabilité; priorité et type de données. Il existe un compteur de séquence distinct pour chaque flux de données, et chaque flux peut être acheminé différemment de tous les autres flux de données.

**A.3.2 type de données:** toutes les unités PDU de données MCS sont non uniformes (ordonnées à la source), uniformes (ordonnées d'ensemble), ou encore sont des données obtenues par mandat (des données non uniformes envoyées par multidiffusion par le fournisseur de groupe de multidiffusion au nom de l'expéditeur).

**A.3.3 connexion MAP:** il s'agit d'une connexion MCS entre deux nœuds établie au moyen du protocole MAP. Par ailleurs, une connexion MAP est constituée de une ou plusieurs connexions de transport. L'établissement d'une connexion MAP n'exige pas que des nœuds adjacents disposent de protocoles multidiffusion communs.

**A.3.4 connexion MCS:** il s'agit d'une connexion entre deux fournisseurs de service MCS qui sont adjacents dans la hiérarchie du domaine. Elle peut être établie avec le protocole MAP ou directement au moyen d'une des piles de protocole définies dans la Recommandation T.123.

**A.3.5 métacanal:** il s'agit d'un canal MCS ayant un niveau de fiabilité donné et dans une gamme de priorités donnée. Le protocole MAP assigne des groupes multidiffusion à des métacanaux.

**A.3.6 multidiffusion:** ce terme désigne la possibilité d'envoyer en une seule opération un paquet de données à plus d'un destinataire.

**A.3.7 groupe de multidiffusion:** il s'agit d'un groupe de récepteurs qui ont exprimé leur intérêt à recevoir certaines données par multidiffusion. Ce groupe est identifié de manière unique au moyen d'une adresse de groupe de multidiffusion.

**A.3.8 adresse de groupe de multidiffusion:** il s'agit d'une adresse réseau qui identifie de manière unique un groupe de multidiffusion. Les données envoyées à cette adresse sont livrées, par multidiffusion, à tous les nœuds accessibles qui sont joints au groupe de multidiffusion.

**A.3.9 fournisseur de groupe de multidiffusion:** il s'agit d'un nœud qui ne dispose pas d'une connexion MAP amont, ou qui dispose d'une connexion MAP amont mais qui n'a pas de protocoles de multidiffusion en commun avec le nœud en amont. Le fournisseur de groupe de multidiffusion est responsable de l'association de métacanaux avec des groupes de multidiffusion. Lorsque cette association est établie, les données transmises au métacanal peuvent être livrées par multidiffusion.

**A.3.10 îlot multidiffusion:** il s'agit d'un groupe de connexions MAP contiguës à l'intérieur d'un domaine MCS qui ont convenu d'utiliser un protocole de transport multidiffusion particulier pour un métacanal donné.

**A.3.11 émetteur multidiffusion:** il s'agit du nœud dans un îlot multidiffusion qui émet un flux de données au moyen d'un service de multidiffusion. Pour les données non uniformes, il s'agit du nœud de point d'entrée. Dans le cas des données uniformes et des données obtenues par mandat, il s'agit du fournisseur de groupe de multidiffusion.

**A.3.12 point d'entrée:** il s'agit du nœud dans un îlot multidiffusion où un flux de données particulier pénètre dans l'îlot.

**A.3.13 niveau de fiabilité:** il s'agit du degré d'assurance offert par le transfert de données. Le protocole MAP prend en charge deux niveaux de fiabilité: entièrement fiable (remise garantie) et non fiable (aucune garantie).

**A.3.14 numéro séquentiel:** il s'agit d'un numéro récurrent associé à chaque flux de données. Ce nombre est utilisé pour le réassemblage de données non fiables, et pour faciliter la réalisation de transitions transparentes entre le transfert de données monodiffusion et le transfert multidiffusion.

**A.3.15 connexion de transport (TC, *transport connection*):** il s'agit d'une connexion monodiffusion faisant appel à une des piles de protocole sous le protocole MAP.

**A.3.16 protocole de transport:** ce sont les protocoles utilisés par le protocole MAP pour effectuer la livraison de données. Il existe quatre types de protocoles de transport: monodiffusion fiable, multidiffusion fiable, monodiffusion non fiable et multidiffusion non fiable. Les nœuds dans le domaine arbitrent les protocoles de transport avant de les utiliser.

## A.4 Abréviations

La présente annexe utilise les abréviations suivantes:

ATM	mode de transfert asynchrone ( <i>asynchronous transfer mode</i> )
IP	protocole Internet ( <i>internet protocol</i> )
MAP	protocole d'adaptation de multidiffusion ( <i>multicast adaptation protocol</i> )
MAPPDU	protocole d'adaptation de multidiffusion – Unité de données protocolaires ( <i>multicast adaptation protocol – protocol data unit</i> )
MAPSAP	protocole d'adaptation de multidiffusion – Point d'accès au service ( <i>multicast adaptation protocol – service access point</i> )
MCS	service de communication multipoint ( <i>multipoint communication service</i> )

MTP/SO	protocole de transport multidiffusion/auto-organisateur ( <i>multicast transport protocol/self-organizing</i> )
PDU	unité de données protocolaires ( <i>protocol data unit</i> )
RMP	protocole multidiffusion fiable ( <i>reliable multicast protocol</i> )
RNIS	réseau numérique à intégration de services
RTPC	réseau téléphonique public commuté
SCF	fonction synchronisation et convergence ( <i>synchronization and convergence function</i> )
TCP	protocole de contrôle de transmission ( <i>transmission control protocol</i> )
UDP	protocole datagramme utilisateur ( <i>user datagram protocol</i> )

## A.5 Aperçu général

Lorsqu'il fait appel aux protocoles de transport prenant en charge seulement les monodiffusions, le protocole T.125 crée des connexions point-à-point afin de "construire" une conférence multipoint. Les flux de données passent par ces connexions, suivant la hiérarchie des domaines, jusqu'à ce qu'ils atteignent tous les destinataires prévus. Dans les réseaux capables de prendre en charge de vrais services multidiffusion, une telle démarche représente des pertes considérables car les données peuvent parfois être transmises à plusieurs reprises par les mêmes dispositifs matériels. La présente annexe décrit un protocole qui permet le transfert de données multidestinataire dans une conférence T.120. Ce protocole se nomme le protocole d'adaptation de multidiffusion (MAP).

Le protocole MAP est complètement indépendant du réseau sous-jacent. Il est isolé du réseau par un protocole de transport conforme aux exigences de service définies au 5.2. Certains exemples de la présente annexe renvoient à des protocoles qui sont mis en œuvre au-dessus du Protocole Internet (IP), mais le protocole MAP peut aussi être utilisé dans d'autres types de réseaux multidiffusion, comme les réseaux multidiffusion ATM ou les réseaux multidiffusion à relais de trames.

Afin de gérer les multidiffusions, le protocole MAP doit connaître d'autres éléments d'information que les seules connexions monodiffusion. Le protocole MAP doit aussi connaître les domaines et les canaux. Cette information lui permet de grouper de manière adéquate les connexions monodiffusion et d'utiliser le principe de la multidiffusion de la manière la plus efficace. La gestion des domaines et des canaux fait encore partie du service MCS, mais cette information est partagée avec le protocole MAP lorsque besoin est.

A un niveau élevé, le problème que pose l'utilisation du principe de la multidiffusion en T.120 peut se subdiviser en trois parties:

- 1) *attribution des groupes de multidiffusion*: l'utilisation de la multidiffusion a pour condition préalable l'attribution d'une adresse de groupe de multidiffusion qui n'est pas en cours d'utilisation par quiconque dans le réseau pouvant être atteint. Le protocole MAP détermine le nœud qui attribue l'adresse. La manière dont les attributions sont faites dépasse toutefois la portée de la définition du protocole MAP (elle fait l'objet des travaux d'autres groupes de normalisation);
- 2) *distribution des groupes de multidiffusion*: lorsqu'une adresse de groupe de multidiffusion est attribuée, il est nécessaire de la distribuer aux nœuds qui en auront besoin. Le protocole MAP traite la distribution des adresses de groupe de multidiffusion;
- 3) *gestion du transfert de données en multidiffusion*: il n'est pas suffisant d'envoyer l'adresse d'un groupe de multidiffusion et de dire à tous de commencer à l'utiliser. De nombreuses raisons expliquent pourquoi le principe de la multidiffusion peut ne pas fonctionner uniformément dans l'intégralité d'un domaine. Il est donc nécessaire de "tester" le réseau

sous-jacent afin de détecter qui peut recevoir des données multidiffusion (et de quelle provenance). Le protocole MAP effectue ce test et utilise le principe de la multidiffusion dans tous les cas où il le peut. Si pour une raison quelconque un nœud donné ne peut recevoir de données au moyen d'un service de multidiffusion, les données sont livrées à ce nœud par monodiffusion.

Le protocole MAP traite aussi la sélection dynamique des protocoles de transport monodiffusion et multidiffusion. Ces protocoles font l'objet d'un arbitrage automatique entre deux nœuds qui communiquent avec le protocole MAP. Lorsque ces protocoles sont choisis, le protocole MAP peut faciliter la livraison efficace de données au sein du domaine. Le protocole MAP est une couche séparée au-dessus de ces protocoles de transport.

### **A.5.1 Utilisation du principe de la multidiffusion**

Le protocole MAP prend en charge la transmission de données à deux niveaux de fiabilité différents, ce qui joue un rôle direct dans l'utilisation du principe de la multidiffusion. Des niveaux de fiabilité différents entraîneront le choix de protocoles de transport différents, et donc de groupes de multidiffusion différents. Le protocole MAP peut aussi utiliser des groupes de multidiffusion différents pour différentes gammes de priorités (afin que les flux soient contrôlés indépendamment pour des priorités différentes). La présente annexe utilise l'expression *métacanal* pour désigner un canal MCS ayant un niveau de fiabilité donné, dans une gamme de priorités donnée.

Afin de permettre l'utilisation de la multidiffusion, le protocole MAP assigne un groupe de multidiffusion à chaque métacanal dans chaque îlot multidiffusion (les îlots multidiffusion sont décrits en détail au A.5.2). Les canaux d'identification d'utilisateur, auxquels ne sont pas assignés des groupes de multidiffusion, représentent une exception à ce principe (il n'y a qu'un seul récepteur potentiel écoutant un canal d'identification d'utilisateur, ce qui contredit le principe de la multidiffusion).

Le protocole MAP peut faire preuve d'une grande souplesse en ce qui concerne l'assignation des groupes. Ainsi, il peut assigner un même groupe de multidiffusion à chacun des métacanaux (afin de réduire le nombre de groupes de multidiffusion). Il pourrait aussi assigner un groupe de multidiffusion différent à chaque métacanal (de manière à refléter le fait que chacun des métacanaux peut comporter un ensemble différent de récepteurs intéressés). Ces assignations de groupe sont contrôlées par un ensemble de politiques qui pourront être adaptées selon l'évolution des technologies réseau.

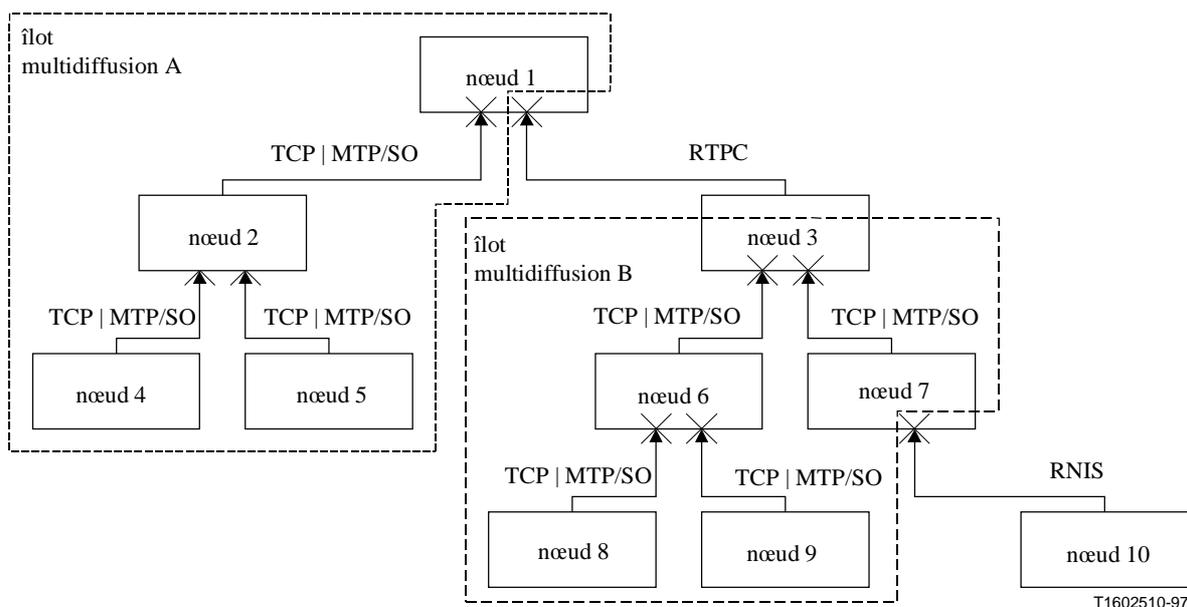
A la transmission de données, le protocole MAP utilisera les paramètres de canal, de niveau de fiabilité et de priorité pour choisir le groupe de multidiffusion approprié. Cela permet que les données soient transmises directement aux destinataires prévus, sans qu'il y ait propagation dans la hiérarchie du domaine (lorsque la technologie de multidiffusion fonctionne).

### **A.5.2 Îlots multidiffusion**

La multidiffusion ne peut pas toujours être utilisée pour livrer des données à chacun des nœuds d'un domaine. Certains nœuds peuvent être connectés au domaine par l'intermédiaire de réseaux qui ne peuvent pas assurer des services de multidiffusion (comme le RTPC ou le RNIS). Il est aussi possible que deux groupes de nœuds capables de multidiffusion soient séparés dans la hiérarchie du domaine par une connexion incapable d'assurer la multidiffusion. Il est donc nécessaire de reconnaître l'existence des "îlots multidiffusion" dans un domaine.

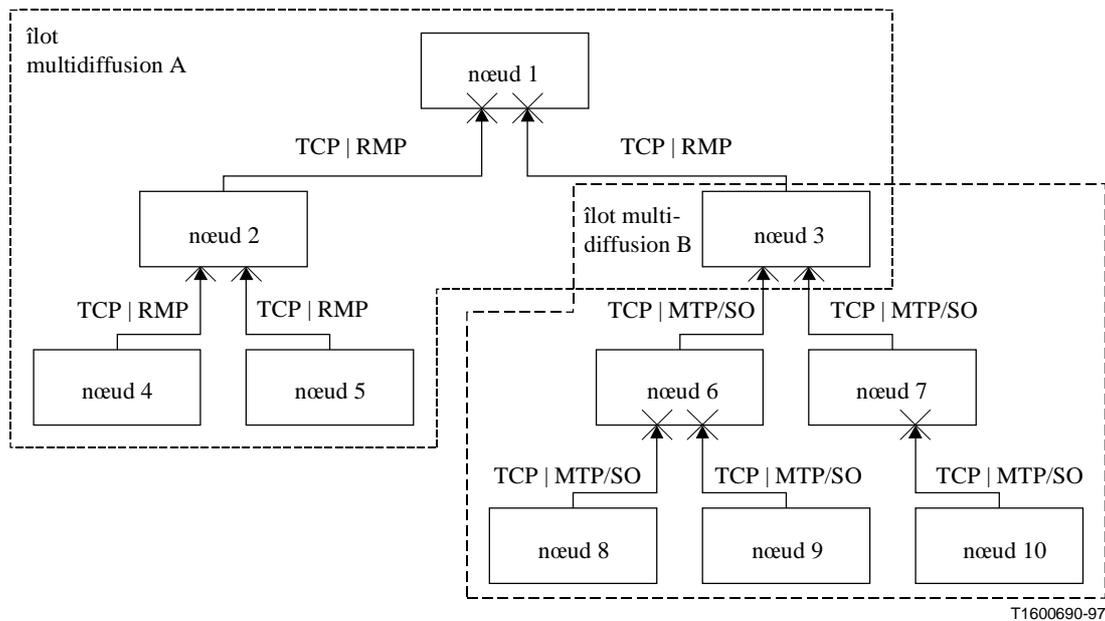
Un îlot multidiffusion est un groupe de connexions MAP contiguës dans un domaine MCS qui ont convenu d'utiliser un protocole de transport multidiffusion particulier pour un métacanal donné.

La Figure A.5-1 illustre une conférence de 10 nœuds. Chacune des connexions constituant la conférence est étiquetée avec le ou les protocoles de transport que les nœuds adjacents ont convenu d'utiliser pour le transfert de données. "RTPC" et "RNIS" désignent les piles de protocole définies dans la Recommandation T.123. "TCP | MTP/SO" représente une connexion MAP qui utilise le protocole TCP pour le transport monodiffusion, et MTP/SO pour le transport multidiffusion (pour un métacanal donné). Les nœuds 1, 2, 4 et 5 sont dans l'îlot multidiffusion A, alors que les nœuds 3, 6, 7, 8 et 9 sont dans l'îlot multidiffusion B. Ces îlots sont séparés parce que la connexion entre les nœuds 1 et 3 appartient au RTPC (cette connexion ne peut être utilisée pour transmettre l'information de commande de multidiffusion).



**Figure A.5-1/T.125 – Îlots multidiffusion séparés par des connexions non compatibles avec la multidiffusion**

Les îlots multidiffusion peuvent aussi être séparés suite à la sélection de protocoles de transport multidiffusion différents pour un métacanal donné. La Figure A.5-2 illustre une conférence ayant la même hiérarchie de domaine que la Figure A.5-1, mais avec des sélections de protocoles différentes. Les nœuds 1, 2, 3, 4 et 5 sont dans l'îlot multidiffusion A et ils ont convenu d'utiliser RMP pour le transport multidiffusion. Les nœuds 3, 6, 7, 8, 9 et 10 sont dans l'îlot multidiffusion B et ils ont convenu d'utiliser MTP/SO pour le transport multidiffusion. Il est à noter que le nœud 3 appartient aux deux îlots multidiffusion, agissant à titre de pont entre ces deux îlots.



**Figure A.5-2/T.125 – Îlots multidiffusion séparés par des sélections de protocole multidiffusion**

La sélection de protocole est effectuée de manière différente pour les protocoles monodiffusion et multidiffusion. Les protocoles monodiffusion sont choisis *par connexion MCS*. Les protocoles monodiffusion choisis sont utilisés afin de transférer des données entre des nœuds adjacents dans la hiérarchie de domaine MCS. Il est parfaitement valable que des connexions différents utilisent différents protocoles monodiffusion. Les protocoles multidiffusion sont choisis *par métacanal*. Lorsqu'un métacanal est mappé pour la multidiffusion, chaque nœud MAP choisit les protocoles multidiffusion qui seront utilisés par les nœuds qui lui sont inférieurs. Ce processus de sélection définit les limites de l'îlot multidiffusion *pour ce métacanal*. Il est possible que les limites d'un îlot multidiffusion varient pour divers métacanaux.

En principe, chaque nœud doit choisir les protocoles multidiffusion qui donneront lieu à un plus petit nombre d'îlots multidiffusion de plus grande taille. Lorsque c'est possible, chaque nœud doit choisir le protocole multidiffusion qui a été choisi par le nœud qui lui est supérieur. Dans la Figure A.5-2, les nœuds 6 et 7 doivent avoir indiqué qu'ils ne peuvent prendre en charge le protocole RMP, sinon le nœud 3 l'aurait choisi. Cette séparation entre îlots résulte de la prise en charge ou non des différents protocoles de transport multidiffusion.

La présente annexe désigne collectivement les protocoles monodiffusion et multidiffusion sous le vocable *protocoles de transport*. Les nœuds adjacents dans la hiérarchie de domaine MCS effectueront de l'arbitrage en vue de convenir des ensembles de protocoles de transport pouvant être utilisés dans une conférence. Pendant l'arbitrage, les nœuds adjacents *doivent* choisir un seul protocole monodiffusion fiable (et, facultativement, un seul protocole monodiffusion fiable). Ils conviendront aussi d'un ensemble d'aucun ou de plusieurs protocoles multidiffusion qui pourront être utilisé à chaque niveau de fiabilité. Lorsqu'un métacanal est mappé pour la multidiffusion, un nœud choisira seulement des protocoles multidiffusion que certains nœuds en aval auront convenu de prendre en charge pendant l'arbitrage.

### A.5.3 Fournisseurs de groupe de multidiffusion

Dans chaque îlot multidiffusion, il y a un nœud qui est responsable de l'attribution des adresses de groupe de multidiffusion. Ce nœud est le fournisseur de groupe de multidiffusion.

Le fournisseur de groupe de multidiffusion est le nœud qui n'a pas de connexion MAP amont, ou qui a une connexion MAP amont, mais aucun protocole de multidiffusion en commun avec le nœud amont. Le fournisseur supérieur du domaine peut aussi être le fournisseur de groupe de multidiffusion pour un îlot multidiffusion, mais cela n'est pas nécessaire. Dans les Figures A.5-1 et A.5.2, le nœud 1 est le fournisseur de groupe de multidiffusion pour l'îlot multidiffusion A, alors que le nœud 3 est le fournisseur de groupe de multidiffusion pour l'îlot multidiffusion B.

#### **A.5.4 Ordonnement de données**

Le service MCS permet deux types d'ordonnement de données: non uniforme (ordonnement à la source) et uniforme (ordonnement d'ensemble).

Les données non uniformes sont habituellement envoyées au groupe de multidiffusion à son "point d'entrée" initial dans l'îlot multidiffusion. Le point d'entrée est simplement le nœud où les données pénètrent dans l'îlot multidiffusion. Ce n'est pas le nœud où l'application qui envoie les données est reliée si ce nœud se trouve à l'extérieur de l'îlot multidiffusion. Dans la Figure A.5-1, si le nœud 10 émet des données, le nœud 7 est le point d'entrée de l'îlot multidiffusion B, alors que le nœud 1 est le point d'entrée de l'îlot multidiffusion A.

Facultativement, un nœud de point d'entrée peut choisir d'utiliser le fournisseur de groupe de multidiffusion comme mandataire multidiffusion. Lorsque cette option est utilisée, les données non uniformes sont d'abord envoyées dans le sens amont par monodiffusion, puis elles sont envoyées au groupe de multidiffusion par le fournisseur de groupe de multidiffusion. Ce principe est discuté plus en détail au A.8.4.

Afin de garantir l'ordonnement d'ensemble, les données uniformes doivent être acheminées en amont vers le fournisseur supérieur par monodiffusion. Ce flux de données utilise un point d'entrée dans le "sens ascendant", mais ce principe est sans intérêt et nous n'en traiterons pas plus avant. Les données sont envoyées au groupe de multidiffusion au point d'entrée "descendant" lorsqu'elles "redescendent" (dans le cas des données uniformes, le point d'entrée descendant sera simplement nommé point d'entrée). Dans la Figure A.5-1, si le nœud 10 transmet des données uniformes, ces données seront d'abord envoyées par monodiffusion en amont vers le fournisseur supérieur. Ensuite, dans le sens aval, le nœud 1 est le point d'entrée de l'îlot multidiffusion A, alors que le nœud 3 est le point d'entrée de l'îlot multidiffusion B. Dans le cas des données uniformes, le point d'entrée sera *toujours* le fournisseur de groupe de multidiffusion de chaque îlot multidiffusion.

#### **A.5.5 Usage des principes de la multidiffusion et de la monodiffusion non fiable**

Toutes les données de commande sont envoyées par des connexions monodiffusion fiables. L'unité **MAPData** est la seule unité MAPPDU transmise au moyen d'un service de multidiffusion ou de la monodiffusion non fiable. Le Tableau A.5-1 résume la manière dont le protocole MAP utilise chacun des quatre types de protocoles de transport.

**Tableau A.5-1/T.125 – Utilisation des types de protocoles de transport**

	<b>Monodiffusion</b>	<b>Multidiffusion</b>
Fiable	Toute unité MAPDU <b>MAPData</b> peut contenir toute unité MCSPDU	Seulement <b>MAPData</b> <b>MAPData</b> peut contenir seulement une des unités MCSPDU: <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataIndication</b>
Non fiable	Seulement <b>MAPData</b> <b>MAPData</b> peut contenir seulement une des unités MCSPDU: <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataRequest</b> <b>MCSUniformSendDataIndication</b>	Seulement <b>MAPData</b> <b>MAPData</b> peut contenir seulement une des unités MCSPDU: <b>MCSSendDataRequest</b> <b>MCSSendDataIndication</b> <b>MCSUniformSendDataIndication</b>

## **A.6 Utilisation du protocole MAP**

Le présent sous-paragraphe décrit la manière dont le service MCS gère les connexions et les transferts de données lorsque le protocole d'adaptation de multidiffusion (MAP) est utilisé. Le présent sous-paragraphe n'a pas pour objet de décrire de manière exhaustive les interactions entre le service MCS et le protocole MAP. Il décrit plutôt comment le service MCS se comporte d'une manière différente que lorsqu'une pile de protocole qui permet seulement la monodiffusion est utilisée (comme celles qui sont actuellement définies dans la Recommandation T.123).

Afin de décrire complètement les différences dans la manière dont le service MCS traite le protocole MAP, il est utile de décomposer T.125 en ses quatre unités fonctionnelles: gestion de domaine, gestion de canal, transfert de données et gestion des jetons. Chacune de ces unités fait l'objet d'une brève description ci-dessous.

### **A.6.1 Gestion de domaine**

Lorsqu'une connexion est établie avec un fournisseur de service MCS distant au moyen d'une pile de protocole qui offre seulement la monodiffusion, le service MCS crée une connexion de transport (TC) pour chaque priorité de transfert de données (le nombre de priorités fait l'objet d'un arbitrage en tant que partie des paramètres du domaine). Les unités MCSPDU de commande sont toujours envoyées sur la connexion TC ayant la priorité la plus élevée. Les MCSPDU de données peuvent être envoyées sur toutes les connexions de transport (la connexion utilisée dépend de la priorité).

Lorsqu'une connexion est établie avec un fournisseur de service MCS distant utilisant le protocole MAP, le service MCS crée une seule connexion MAP (MC), quel que soit le nombre de priorités de transfert de données. Les unités MCSPDU de commande sont envoyées sur cette MC. Les MCSPDU de données ne sont en fait envoyées sur *aucune* connexion MAP, mais elles sont transmises au MAP comme un ensemble (cette question est traitée dans le sous-paragraphe transfert de données).

Les unités MCSPDU de connexion jouent le même rôle d'amorçage de nouvelle connexion MCS et d'arbitrage de certains paramètres du domaine. **Connect-Initial** est envoyé par le service MCS appelant comme première unité MCSPDU par l'intermédiaire d'une nouvelle connexion MAP. **Connect-Response** est la première unité MCSPDU envoyée par le service MCS appelé dans le sens

opposé par l'intermédiaire d'une nouvelle connexion MAP. Comme il n'y a qu'une seule connexion MAP par connexion MCS, **Connect-Additional** et **Connect-Result** ne sont pas nécessaires.

Pour que le protocole MAP puisse remplir adéquatement ses fonctions d'acheminement, le service MCS doit partager certains renseignements au sujet de toutes les connexions établies récemment: domaine auquel la connexion est liée, connexion amont ou aval et version du protocole service MCS qui sera utilisée pour cette connexion. De plus, lorsque le service MCS lie la première connexion MAP à un domaine, il doit partager avec le protocole MAP le nombre de priorités et de niveaux de fiabilité qui seront utilisés dans le domaine.

Toutes les unités MCSPDU du domaine qui traitent la gestion de domaine sont envoyées par une connexion MAP d'une manière identique à leur envoi par une connexion TC en utilisant une pile de protocole monodiffusion.

### **A.6.2 Gestion de canal**

Toutes les unités MCSPDU du domaine qui traitent la gestion de canal sont envoyées par une connexion MAP d'une manière identique à leur envoi par une connexion TC en utilisant une pile de protocole monodiffusion.

Pour que le protocole MAP puisse être en mesure de multidiffuser des données pour le compte du service MCS, il doit disposer de certains renseignements au sujet des canaux (afin de savoir où envoyer les données). Le service MCS doit partager les renseignements suivants avec le protocole MAP:

- 1) pour chaque canal actif, le protocole MAP doit connaître le canal auquel le domaine est associé, et le type du canal en question;
- 2) pour chaque canal actif, le protocole MAP doit connaître quelles connexions MAP mènent aux utilisateurs rejoints;
- 3) pour chaque canal actif, le protocole MAP doit savoir si le service MCS doit recevoir les données. Le service MCS aura besoin des données s'il doit les rediriger vers une application utilisateur reliée localement ou à une pile de protocole différente.

Les canaux statiques sont considérés comme étant actifs si au moins une application utilisateur est jointe dans le sous-arbre du nœud local. Les canaux dynamiques sont considérés comme étant actifs dès le moment où ils sont attribués.

### **A.6.3 Transfert de données**

La principale différence liée à l'utilisation du protocole MAP réside dans le traitement du transfert de données. Lorsqu'il utilise une pile de protocole qui offre seulement la monodiffusion, le service MCS prend en charge toutes les responsabilités quant au réacheminement des données à des connexions TC qui mène à des utilisateurs intéressés. Lorsque le protocole MAP est utilisé, une grande part de cette responsabilité est déléguée au protocole MAP.

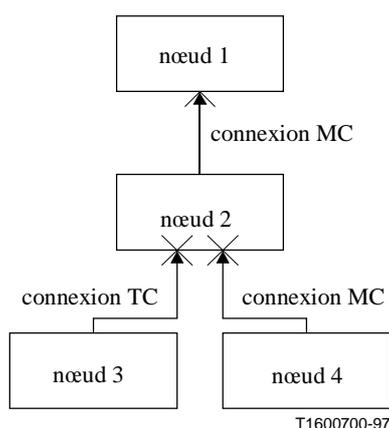
Le service MCS envoie normalement une unité MCSPDU de données au protocole MAP une seule fois, quel que soit le nombre de nœuds qui doivent les recevoir. La seule exception à cette pratique se produit lorsque le service MCS a arbitré l'utilisation d'une version différente du protocole pour certaines des connexions MC liées au domaine. Le service MCS doit envoyer l'unité MCSPDU de données codée correctement pour chaque version active du protocole. Le protocole MAP s'assurera que chaque élément reçoive les données nécessaires.

Lorsque le service MCS reçoit des données du protocole MAP, il n'est pas nécessaire que le service MCS réachemine ces données vers le protocole MAP (à moins que plusieurs versions du protocole ne soient en usage, ce qui force le service MCS à effectuer des traductions de version.

Lorsqu'il transmet des données au protocole MAP, le service MCS identifie de manière unique l'ensemble-cible de récepteurs en précisant un identificateur de domaine, une version de protocole MCS, et un identificateur de flux de données. Cela constitue les renseignements suffisants car le protocole MAP sait déjà quelles connexions MAP sont liées au domaine, et quelle version du protocole est utilisée à chaque connexion. Le protocole MAP peut alors utiliser l'identificateur de flux de données afin d'acheminer correctement les données par monodiffusion et/ou multidiffusion. Il est à noter que l'identificateur de flux de données signale aussi la priorité relative des données au protocole MAP. Cette priorité aura toujours une valeur entre 0 et  $N-1$ , 0 représentant la plus haute priorité ( $N$  est le nombre de priorités que le service MCS a indiqué être en usage dans le domaine). Aucune traduction des priorités n'est requise entre le service MCS et le protocole MAP.

Voir la Figure A.6-1, qui montre une conférence faisant appel à quatre nœuds. Deux des trois connexions constituant la conférence sont des connexions MC. La troisième connexion, qui relie le nœud 3 au nœud 2, est une connexion TC établie par l'intermédiaire de la pile de protocole RTPC définie dans la Recommandation T.123. Il est utile d'examiner la manière dont le service MCS au nœud 2 réagit aux données provenant de chacun des quatre nœuds. Pour cet exemple, supposons que des données non uniformes sont envoyées par un canal pour lequel des utilisateurs sont joints à chaque nœud. Il est à noter que cet exemple traite seulement du traitement des données au nœud 2.

- 1) Lorsque les données proviennent du nœud 1, elles sont reçues par le protocole MAP au nœud 2. Elles sont ensuite transmises au service MCS qui, à son tour, envoie les données à la connexion TC qui mène au nœud 3. Le service MCS ne renverra PAS les données au protocole MAP.
- 2) Lorsque les données proviennent du nœud 2, le service MCS envoie les données à la connexion TC qui mène au nœud 3. Il envoie aussi les données au protocole MAP (une fois).
- 3) Lorsque les données proviennent du nœud 3, elles sont transmises au service MCS sur le nœud 2 par la pile de protocole RTPC. Le service MCS transmet alors les données au protocole MAP (une fois).
- 4) Lorsque les données proviennent du nœud 4, elles sont reçues par le protocole MAP sur le nœud 2. Elles sont ensuite transmises au service MCS qui, à son tour, envoie les données à la connexion TC qui mène au nœud 3. Le service MCS ne renverra PAS les données au protocole MAP.



**Figure A.6-1/T.125 – Conférence à quatre participants représentative**

Comme le montre cet exemple, le service MCS compte sur le protocole MAP pour effectuer l'acheminement pour toutes les connexions MC groupées logiquement (par domaine et version de

protocole). Cette délégation est nécessaire car seul le protocole MAP sait quand la multidiffusion est en usage, et si elle fonctionne ou non. L'abstraction présentée au service MCS est que la multidiffusion fonctionne *toujours*.

#### **A.6.3.1 Versions de protocole différentes**

Tel que mentionné ci-dessus, il est nécessaire de différencier les PDU de données selon la version de protocole. En effet, le format des PDU de données est différent. Si la moitié des connexions MC d'un domaine utilisent une version de protocole et que l'autre moitié en utilise une autre, le service MCS doit effectuer la traduction de toute donnée transférée entre les deux groupes. Cette exigence constitue le seul cas où le service MCS peut être obligé de renvoyer des données vers la pile de protocole MAP dont elles proviennent (après traduction, évidemment).

#### **A.6.3.2 Traitement des données uniformes**

En raison de l'exigence d'ordonnancement d'ensemble, il est nécessaire de traiter les données uniformes de manière légèrement différente que les données non uniformes. Le protocole MAP peut déterminer si une PDU est uniforme ou non en examinant l'identificateur de flux de données.

Lorsque le protocole MAP reçoit une PDU de données uniformes du service MCS, il vérifie s'il y a une connexion MAP amont pour le domaine et la version de protocole spécifiés. Si c'est le cas, le protocole MAP enverra l'unité PDU en amont par monodiffusion (car ce doit être une demande). S'il n'y a pas de connexion MAP amont pour le domaine et la version de protocole spécifiés, le protocole MAP enverra l'unité PDU en aval à tous les destinataires joints (au moyen de la multidiffusion, si c'est possible).

Lorsque le protocole MAP reçoit une PDU de données uniformes d'une connexion MAP en aval, il vérifie si il y a une connexion MAP amont pour les mêmes domaine et version de protocole. Si c'est le cas, le protocole MAP enverra l'unité PDU en amont par monodiffusion (l'unité PDU ne sera PAS envoyée au service MCS). S'il n'y pas de connexion MAP amont pour les mêmes domaine et version de protocole, le protocole MAP enverra l'unité PDU au service MCS.

Lorsque le protocole MAP reçoit les PDU de données uniformes au moyen d'un service de multidiffusion ou d'une connexion MAP amont, l'unité PDU est envoyée au service MCS (si c'est nécessaire), ainsi que, dans le sens aval, vers les nœuds qui ne reçoivent pas de données au moyen d'un service de multidiffusion.

#### **A.6.3.3 Demande et indication**

Lors de l'envoi de données non uniformes par l'intermédiaire du protocole MAP, il est possible que les données arrivent à des nœuds au-dessus ET au-dessous du nœud en cours. C'est pour cette raison qu'il faut préciser si le service MCS doit identifier les PDU de données en tant que demandes ou en tant qu'indications.

Si la connexion amont est une connexion MAP utilisant la même version de protocole MCS, l'unité PDU envoyée au protocole MAP devrait être une demande. Sinon, elle devrait être une indication. En d'autres termes, les données qui circulent SEULEMENT dans le sens aval sont des indications, alors que les données qui circulent dans le sens amont représentent une demande (même si elles circulent aussi dans le sens aval).

Cela crée une situation dans laquelle il est possible qu'un nœud reçoive une demande **Send-Data-Request** (demande d'envoi de données) d'un nœud qui est au-dessus dans la hiérarchie. Le service MCS doit reconnaître que ce n'est pas une situation d'erreur lorsque les données proviennent du protocole MAP, et donc traiter l'unité PDU comme si elle était orientée dans le sens approprié.

Il est à noter que cette situation n'existe pas dans le cas des données uniformes, car les demandes sont unidiffusées dans le sens amont en premier lieu. Les données uniformes ne seront multidiffusées qu'à partir d'un point plus élevé, ce qui garantit que tous les récepteurs reçoivent une indication d'un nœud qui se trouve au-dessus dans la hiérarchie.

#### **A.6.3.4 Données transmises par mandataire**

Lorsqu'il envoie des données non uniformes, le protocole MAP peut soit multidiffuser les données au point d'entrée, soit décider d'utiliser un mandataire multidiffusion. La manière dont le nœud de point d'entrée prend cette décision dépasse l'objet de la présente annexe.

La multidiffusion réalisée par mandataire signifie que les données sont unidiffusées dans le sens amont jusqu'à ce qu'elles atteignent le fournisseur de groupe de multidiffusion, qui les multidiffuse ensuite dans le sens aval. Ce processus est semblable au traitement des données uniformes, sauf que les données n'ont pas à circuler jusqu'au fournisseur supérieur avant de commencer leur acheminement dans le sens aval (elles ont seulement à atteindre le fournisseur de groupe de multidiffusion). Cette technique peut être utilisée pour réduire le volume d'information d'état sur le flux de données dans les très grandes conférences. Elle peut aussi être utilisée pour permettre la multidiffusion de données lorsque le nœud de point d'entrée ne peut lui-même transmettre au moyen d'un service de multidiffusion.

L'utilisation des données obtenues par mandat est transparente au service MCS. Pour le service MCS, elles sont comme des données non uniformes.

#### **A.6.4 Gestion des jetons**

Toutes les unités MCSPDU du domaine qui traitent de la gestion des jetons sont envoyées par une connexion MAP exactement de la même manière que lorsqu'elles sont envoyées par l'intermédiaire d'une connexion TC en utilisant une pile de protocole monodiffusion.

### **A.7 Utilisation des protocoles de transport**

Le présent sous-paragraphe décrit les services que le protocole MAP prévoit recevoir des protocoles de transport.

#### **A.7.1 Protocoles de transport pour la monodiffusion**

Les services requis des protocoles monodiffusion constituent un sous-ensemble de ceux qui sont définis dans la Recommandation X.214. Le protocole MAP n'utilise pas des paramètres exactement identiques à ceux qui sont décrits dans la Recommandation X.214, et il n'utilise pas les données expédiées. Le présent sous-paragraphe traite des services *utilisés* par le protocole MAP.

Pendant l'établissement de la connexion, le protocole MAP crée une connexion monodiffusion ou plus avec le nœud adjacent. Ces connexions sont utilisées pour transmettre les données de commande et les données de l'utilisateur. Si plusieurs connexions sont établies, elles sont de priorités diverses, ce qui permet au protocole MAP d'indiquer au protocole de transport l'importance relative d'une unité MAPPDU donnée.

Dans le cas des protocoles monodiffusion fiables, le protocole MAP suppose que le service est exempt d'erreurs et ordonné. Aucun de ces deux postulats n'est posé pour les protocoles monodiffusion non fiables.

Le protocole MAP suppose aussi que tous les protocoles monodiffusion fiables sont capables de segmenter les unités MAPPDU de trop grande taille pour être transmises en tant qu'unités entières. Ces unités MAPPDU doivent être réassemblées au côté réception avant d'être envoyées au protocole MAP. Afin de prévenir la nécessité de traiter des MAPPDU de taille arbitraire excessive, il est

recommandé qu'une limite de 16 384 octets soit imposée au protocole MAP. Le protocole MAP doit s'assurer qu'aucune unité MAPPDU ne dépasse cette taille.

Dans le cas de la monodiffusion fiable, le protocole MAP suppose la présence d'un service "à orientation connexion". Cela implique que les deux côtés de la connexion sont au courant de leurs présences respectives, et qu'un protocole quelconque a participé à l'établissement de la connexion. Cela implique aussi que les deux nœuds tiennent à jour de l'information d'état au sujet de la connexion qui les relie, et qu'ils peuvent détecter automatiquement la perte de paquets et autres problèmes touchant le réseau. Lors de la création d'une connexion monodiffusion fiable, le protocole MAP participe en envoyant son propre protocole (après que le protocole de transport ait indiqué que cette action est possible).

Dans le cas de la monodiffusion non fiable, le protocole MAP suppose la présence d'un service "sans connexion". Cela signifie simplement que le protocole MAP ne fait aucune supposition au sujet du service qui est fourni. Il ne suppose pas que les données seront ordonnées. Il ne suppose pas que la perte de paquets, ou tout autre problème de réseau, seront détectés.

Les primitives de monodiffusion utilisées par le protocole MAP sont résumées au Tableau A.7-1.

**Tableau A.7-1/T.125 – Définition du service de protocole de transport monodiffusion**

Primitive	Paramètres	Fiable	Non fiable
demande T-Connect indication T-Connect réponse T-Connect confirmation T-Connect	adresse appelée, priorité adresse appelante	X X X X	
demande T-Data indication T-Data	données utilisateur données utilisateur	X X	
demande T-Unit-Data  indication T-Unit-Data	adresse de destination, priorité, données utilisateur  adresse source, données utilisateur	   X	X   X
demande T-Disconnect indication T-Disconnect	motif	X X	

#### A.7.1.1 T-Connect

Les primitives *T-Connect* sont utilisées pour créer des connexions monodiffusion fiables vers des nœuds distants.

Le nœud appelant invoquera une *demande T-Connect* pour lancer la création d'une nouvelle connexion. Si le nœud distant peut être trouvé, une *indication T-Connect* est livrée au protocole MAP sur ce nœud. Le protocole MAP acceptera normalement la connexion en invoquant une *réponse T-Connect*. Cela entraîne la présence d'une *confirmation T-Connect* au nœud appelant. Lorsque ce processus est complété, la connexion est prête et le protocole MAP poursuit sa partie de l'établissement de la connexion.

#### Définition des paramètres

*Adresse appelée*: adresse réseau du nœud qui doit être appelé.

*Adresse appelante*: adresse réseau du nœud qui place l'appel.

*Priorité*: ce champ sert à spécifier la priorité de cette connexion (par rapport à toutes les autres connexions). La gamme de valeurs admises s'étend de 0 à 15, 0 représentant la priorité la plus élevée.

#### **A.7.1.2 T-Data**

Les primitives *T-Data* sont utilisées pour transférer des unités MAPPDU au nœud distant au moyen de la monodiffusion fiable.

Lorsque le protocole MAP souhaite envoyer une unité MAPPDU au moyen de la monodiffusion fiable, il invoque une *demande T-Data*. Cela entraîne éventuellement la présence d'une *indication T-Data* au nœud distant.

#### **Définition des paramètres**

*Données utilisateur*: chaîne composée d'un nombre variable d'octets. Elle est utilisée pour transporter les unités MAPPDU.

#### **A.7.1.3 T-Unit-Data**

Les primitives *T-Unit-Data* sont utilisées pour transférer des unités MAPPDU au nœud distant au moyen de la monodiffusion non fiable.

Lorsque le protocole MAP souhaite envoyer une unité MAPPDU au moyen de la monodiffusion non fiable, il invoque une *demande T-Unit-Data*. Cela entraîne éventuellement la présence d'aucune ou de plusieurs *indications T-Unit-Data* au nœud distant. L'ordre des demandes peut ne pas correspondre à l'ordre des indications. De plus, il peut ne pas y avoir de correspondance univoque entre les demandes et les indications (certaines demandes peuvent ne jamais donner lieu à une indication, alors que certaines demandes peuvent donner lieu à plusieurs indications).

#### **Définition des paramètres**

*Adresse de destination*: adresse réseau du nœud qui recevra les données.

*Adresse source*: adresse réseau du nœud d'où proviennent les données.

*Priorité*: ce champ sert à spécifier la priorité de cette unité MAPPDU (relativement à toutes les autres unités MAPPDU). La gamme de valeurs admises s'étend de 0 à 15, 0 représentant la priorité la plus élevée.

*Données utilisateur*: chaîne composée d'un nombre variable d'octets. Elle est utilisée pour transporter les unités MAPPDU.

#### **A.7.1.4 T-Disconnect**

Les primitives *T-Disconnect* sont utilisées pour couper les connexions monodiffusion fiables existantes.

Habituellement, un côté invoquera une *demande T-Disconnect*, qui coupe la connexion et a pour effet que l'autre côté reçoit une *indication T-Disconnect*. Toutefois, si la connexion est coupée en raison de problèmes du réseau, les deux côtés reçoivent une *indication T-Disconnect*.

#### **Définition des paramètres**

*Motif*: ce paramètre contient le motif pour lequel une connexion a été coupée. Sa valeur peut être: causé par l'utilisateur; causé par le fournisseur; ou échec non spécifié.

### **A.7.2 Protocoles de transport multidiffusion**

Le présent sous-paragraphe décrit les services requis par le protocole MAP afin d'assurer la gestion d'un groupe de multidiffusion. Cela comprend les activités suivantes: joindre un groupe de

multidiffusion avant le flux de données, émettre et recevoir des données au moyen d'un service de multidiffusion, et quitter un groupe lorsque les services multidiffusion ne sont plus requis.

Tous les flux de données dans un îlot multidiffusion sont initialement unidiffusés. En réponse à la circulation de données, le protocole MAP attribue et distribue des adresses de groupe de multidiffusion qui seront utilisées afin d'optimiser le trafic sur le réseau. A chaque nœud dans l'îlot, le protocole MAP décidera s'il est nécessaire que ce nœud soit joint à un groupe assigné. Le cas échéant, il demandera aux protocoles multidiffusion de donner ce service.

Lorsque le protocole MAP décide de joindre un groupe de multidiffusion, il peut spécifier un de trois modes: émission seulement, réception seulement ou émission-réception. Certains protocoles multidiffusion peuvent être en mesure d'optimiser leur comportement s'ils savent quelle combinaison est requise, c'est pourquoi cette information leur est communiquée pendant l'opération de jonction de groupe.

Pour la multidiffusion fiable, le protocole MAP suppose que le service est exempt d'erreurs, possiblement après une perte indéterminée des données initiales provenant de chaque émetteur. Cette supposition est faite pour la multidiffusion non fiable.

Le protocole MAP ne suppose jamais que les données de multidiffusion seront ordonnées. Toutefois, pour la multidiffusion fiable, le protocole MAP suppose que tous les vides causés par l'absence d'ordre des données seront comblés. Si un nouveau nœud commence à émettre vers un groupe de multidiffusion, le protocole multidiffusion fiable de devrait *pas* commencer à transmettre ces données au protocole MAP avant que la réception en provenance de ce nœud se soit stabilisée (c'est-à-dire que tous les vides à la réception aient été comblés). Cela signifie que le flux de données reçu peut commencer après diverses pertes initiales à plusieurs récepteurs.

Il n'est pas nécessaire que les protocoles multidiffusion prennent en charge la segmentation. La raison en est que **MAPData** est la seule unité MAPPDU envoyée au moyen d'un service de multidiffusion, et qu'elle sera segmentée à la taille appropriée par le protocole MAP avant d'être envoyée au protocole multidiffusion.

Les primitives multidiffusion utilisées par le protocole MAP sont résumées au Tableau A.7-2.

**Tableau A.7-2/T.125 – Définition du service de protocole de transport multidiffusion**

Primitive	Paramètres
demande MT-Join confirmation MT-Join	adresse de groupe de multidiffusion, priorité, fanions émission/réception
demande MT-Data indication MT-Data	données utilisateur données utilisateur
demande MT-Leave indication MT-Leave	motif

### A.7.2.1 MT-Join

Les primitives *MT-Join* sont utilisées pour joindre un groupe de multidiffusion.

La *demande MT-Join* comporte des fanions qui permettent au protocole MAP de spécifier s'il prévoit émettre, recevoir ou les deux. L'état de ces fanions peut être changé par émission de *demandes MT-Join* additionnelles, sans qu'il soit nécessaire d'invoquer une *demande MT-Leave*.

La *confirmation MT-Join* indique au protocole MAP si la *demande MT-Join* en instance a réussi ou a échoué.

## Définition des paramètres

*Adresse de groupe de multidiffusion*: il s'agit de l'adresse du groupe de multidiffusion qui doit être joint. Une valeur NULLE indique que le protocole de transport devrait créer un nouveau groupe et renvoyer l'adresse au protocole MAP pour distribution.

*Priorité*: ce champ sert à spécifier la priorité de ce groupe (relativement à tous les autres groupes). La gamme de valeurs admises s'étend de 0 à 15, 0 représentant la priorité la plus élevée.

*Fanions émission/réception*: ces fanions permettent au protocole MAP de signaler au protocole multidiffusion le mode qu'il requiert parmi les trois possibles: émission seulement, réception seulement, ou émission-réception.

### A.7.2.2 MT-Data

Les primitives *MT-Data* sont utilisées pour transférer des unités MAPPDU à tous les nœuds du groupe par multidiffusion.

Ces primitives fonctionnent de manière identique pour les protocoles fiables et non fiables. Lorsque le protocole MAP souhaite envoyer une unité MAPPDU par multidiffusion, il invoque une *demande MT-Data*. Cela entraîne éventuellement la présence d'une *indication MT-Data* à tous les nœuds distants qui doivent recevoir les données. L'ordre des demandes peut ne pas correspondre à l'ordre des indications. Pour la multidiffusion non fiable, il peut ne pas y avoir de correspondance univoque entre les demandes et les indications (certaines demandes peuvent ne jamais donner lieu à des indications, alors que certaines demandes peuvent donner lieu à plusieurs indications).

## Définition des paramètres

*Données utilisateur*: chaîne composée d'un nombre variable d'octets. Elle est utilisée pour transporter les unités MAPPDU.

### A.7.2.3 MT-Leave

Les primitives *MT-Leave* sont utilisées pour quitter un groupe de multidiffusion. L'*indication MT-Leave* est utilisée pour signaler au protocole MAP que son appartenance au groupe a été interrompue prématurément.

Lorsque le protocole MAP décide qu'il n'a plus besoin d'émettre *ou* de recevoir de données dans un groupe de multidiffusion donné, il invoque une *demande MT-Leave*. Le protocole multidiffusion devrait immédiatement cesser de transmettre des données du groupe au protocole MAP. Il n'y a pas de confirmation pour cette primitive.

L'*indication MT-Leave* n'a lieu que si le protocole multidiffusion doit indiquer au protocole MAP qu'il n'est plus joint (bien que le protocole MAP n'ait pas demandé de quitter). Cette indication n'est *pas* causée par le fait que d'autres nœuds quittent un groupe.

## Définition des paramètres

*Motif*: ce paramètre contient la raison pour laquelle le groupe de multidiffusion a été quitté. Il peut prendre la valeur: amorcé par le fournisseur ou défaillance non spécifiée.

### A.7.3 Information locale

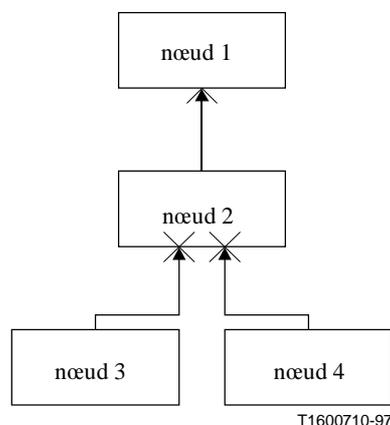
Pour que le protocole MAP puisse traiter un grand nombre de protocoles de transport simultanément, sans connaissance préalable de leurs capacités, de l'information de configuration locale doit être disponible. Chaque protocole de transport est décrit au protocole MAP de façon générique. Cette information est utilisée par le protocole MAP durant le processus d'arbitrage des protocoles de transport, et elle englobe les paramètres suivants:

- *identificateur de protocole de transport*: il est utilisé pour identifier le protocole de transport de manière univoque durant le processus d'arbitrage de protocoles;
- *type de protocole de transport*: chaque protocole de transport doit être identifié relativement à l'une des catégories suivantes: monodiffusion fiable; monodiffusion non fiable; multidiffusion fiable; ou multidiffusion non fiable;
- *pondération de préférences*: cette pondération relative permet de déterminer quels protocoles de transport sont préférés à d'autres;
- *nombre maximal de connexions*: il s'applique seulement aux protocoles monodiffusion. Il s'agit du nombre maximal de connexions que le protocole monodiffusion recommande *par connexion MAP*;
- *taille maximale de l'information utile*: il s'agit de la taille maximale de l'information utile que le protocole de transport peut traiter. A noter que, pour une monodiffusion fiable, le protocole de transport *doit* pouvoir traiter jusqu'à 16 384 octets (avec segmentation, au besoin), de sorte qu'il ne s'agit là que d'une taille maximale recommandée;
- *données de configuration*: lorsque le protocole MAP est prêt à arbitrer des protocoles avec un nœud éloigné, il obtient un bloc de données de configuration provenant de chaque protocole candidat (ce bloc peut être NUL). Cette information est transmise au nœud éloigné durant l'arbitrage de protocoles, où elle est transmise au protocole équivalent du nœud en question (s'il existe). Le protocole MAP du nœud appelé obtient ensuite les données de configuration à retourner de sa pile locale, pour l'exécution du processus d'arbitrage de protocoles. Ces données de configuration doivent être extraites séparément pour chaque connexion MAP arbitrée.

L'information contenue dans le bloc de données de configuration est définie séparément pour chaque protocole de transport. Le format de ces blocs peut être soit normalisé (avec les profils des protocoles de transport), soit propre à un constructeur. On pourrait par exemple retourner à l'appelant un numéro de port IP dynamique.

## A.8 Spécification de protocole

Le présent sous-paragraphe contient une description étape par étape de la façon dont fonctionne le protocole MAP, et elle comprend plusieurs chronogrammes des unités PDU. La Figure A.8-1 illustre la hiérarchie de domaine qui a été utilisée pour produire ces chronogrammes.



**Figure A.8-1/T.125 – Domaine échantillon référencé par exemples subséquents**

### A.8.1 Etablissement de la connexion de monodiffusion fiable initiale

Durant le traitement d'une demande de connexion MCS, il importe en premier lieu de créer une connexion de monodiffusion fiable. La connexion est réalisée selon le protocole spécifié par le service MCS ou le protocole par défaut du réseau (si aucun protocole n'est spécifié). Cette connexion est utilisée pour transmettre toutes les unités PDU de commande (à la fois pour le service MCS et le protocole MAP). La première unité MAPPDU qu'envoie le nœud appelant par la nouvelle connexion est **MAPConnectRequest**. Cette unité contient les champs suivants:

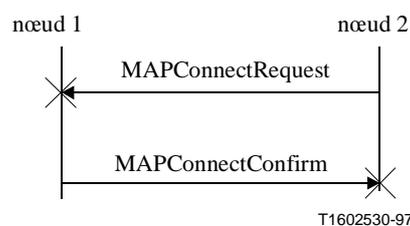
- un numéro de version de protocole identifiant la version du protocole MAP que le nœud appelant désire utiliser;
- un point d'accès au service MAP (MAPSAP) identifiant qui doit accepter l'appel au nœud appelé (spécification par le service MCS dans l'adresse réseau);
- un identificateur de référence de domaine qui lie cette connexion à un domaine particulier (pour la connexion initiale, ce champ est mis à zéro);
- valeurs de priorités basse et élevée qui indiquent la gamme de priorités à envoyer sur la connexion (pour la connexion initiale, ces champs sont tous deux mis à zéro).

Si le nœud appelé désire accepter l'appel, il retourne une unité **MAPConnectConfirm** vers le nœud appelant. Cette unité MAPPDU contient les champs suivants:

- un numéro de version de protocole identifiant la version du protocole MAP qui sera utilisée (il doit être inférieur ou égal à la valeur spécifiée par l'appelant);
- le point MAPSAP du nœud appelé qui a effectivement accepté la connexion d'arrivée.

Si le nœud appelé ne désire pas accepter l'appel d'arrivée, il retourne une unité **MAPDisconnectRequest** vers le nœud appelant. Cette unité MAPPDU contient la raison du rejet. Elle contient aussi un champ indiquant qu'aucune confirmation n'est requise, afin que la connexion soit coupée immédiatement.

La Figure A.8-2 présente un chronogramme illustrant l'établissement réussi d'un appel.



**Figure A.8-2/T.125 – Synchronisation des unités MAPPDU de type Connect**

### A.8.2 Arbitrage de protocoles de transport

Une fois que la connexion de monodiffusion initiale fiable a été établie, le service MCS procède à l'établissement de sa propre connexion. Après l'initialisation du service MCS, l'information suivante est transmise au protocole MAP: domaine auquel la connexion est liée; version de protocole utilisée par le service MCS sur la connexion; et nombre de priorités et de niveaux de fiabilité utilisés dans le domaine (information fournie au protocole MAP lorsque les paramètres de domaine sont initialement fixés). Une fois que le protocole MAP dispose de cette information, il peut procéder à l'arbitrage des protocoles de transport.

Le nœud appelant produira une unité **MAPArbitrateProtocolsRequest** contenant la liste des protocoles de transport qu'il désire utiliser (monodiffusion et multidiffusion). Cette liste doit au

moins inclure le protocole monodiffusion fiable utilisé pour créer la connexion existante. La prise en charge des autres protocoles de transport est facultative. Après avoir comparé cette liste à ses propres capacités, le nœud appelé répondra par une unité **MAPArbitrateProtocolsConfirm** indiquant quels protocoles doivent être utilisés.

Si aucun protocole non fiable n'est sélectionné, les données non fiables sont envoyées à l'aide des protocoles fiables (dans le cas de la monodiffusion, les données non fiables sont envoyées par les mêmes connexions que les données fiables). Si le service MCS a indiqué qu'il n'y aura aucune donnée non fiable dans le domaine, il est inutile d'inclure des protocoles non fiables dans l'une ou l'autre des unités MAPPDU d'arbitrage. Si aucun protocole multidiffusion n'est sélectionné, toutes les données sont envoyées par monodiffusion (entre ces deux nœuds).

La confirmation inclura toujours exactement un protocole monodiffusion fiable. Elle comprendra facultativement un protocole monodiffusion non fiable. Dans le cas de la multidiffusion, elle comprendra tous les protocoles que les deux nœuds se sont entendus pour utiliser (il peut en exister plus d'un pour chaque niveau de fiabilité).

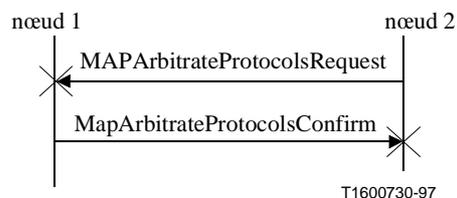
Pour ce qui est du protocole monodiffusion fiable, les nœuds arbitrent également le nombre de connexions de transport qui constituent la connexion MAP en question. S'il existe plus d'une connexion de transport, les données unidiffusées se répartissent entre elles selon leur priorité. Il est parfaitement correct que des connexions MAP différentes soient formées de connexions de transport en nombres différents, même à l'intérieur d'un îlot multidiffusion particulier. Les sous-paragraphes suivants donnent des détails sur le processus de création de connexions de transport additionnelles.

Les unités MAPPDU d'arbitrage contiennent aussi un identificateur de référence de domaine. L'émetteur recourt ensuite à cette valeur pour indiquer à quel domaine une unité MAPPDU est associée, lorsque le domaine ne peut pas se déduire implicitement. Plus précisément, cette valeur est utilisée pour créer des connexions de monodiffusion fiables additionnelles et pour transmettre des données non fiables.

Un nœud quelconque peut imposer un nouvel arbitrage des protocoles de transport avec un nœud voisin, simplement par l'émission d'une unité **MAPArbitrateProtocolsRequest**. Afin d'éviter la perte de connexions, il est toutefois nécessaire que le protocole monodiffusion fiable sélectionné précédemment soit disponible dans le menu de protocole. Si le nouvel arbitrage entraîne la modification des protocoles monodiffusion, la transition doit s'effectuer immédiatement (des détails sur ce processus se trouvent dans un sous-paragraphe subséquent). Les modifications apportées à l'ensemble des protocoles multidiffusion n'ont aucun effet direct sur les groupes multidiffusion existants. Ces modifications se reflètent seulement dans les attributions de groupe subséquentes (des détails sur ce processus se trouvent dans le sous-paragraphe sur l'attribution et la distribution des groupes multidiffusion; voir A.8.3).

Il est possible que des nœuds adjacents subissent une "collision d'arbitrage". Cela se produit lorsque les deux nœuds émettent une unité **MAPArbitrateProtocolsRequest** en même temps. Cette situation peut se détecter aux deux nœuds, car ils recevront tous deux une demande d'arbitrage alors qu'ils attendaient une confirmation. Dans ce cas, il n'est pas tenu compte de la demande provenant du nœud de niveau inférieur (et le nœud inférieur arrête d'attendre une confirmation). La demande provenant du nœud de niveau supérieur est traitée normalement.

La Figure A.8-3 illustre le processus d'arbitrage lorsque le nœud 2 appelle le nœud 1.



**Figure A.8-3/T.125 – Synchronisation des processus MAPPDU d'arbitrage**

### A.8.2.1 Création de connexions de monodiffusion fiables additionnelles

Comme il a été mentionné plus haut, les nœuds arbitrent aussi le nombre de connexions de monodiffusion fiables qui constitueront une connexion MAP particulière. S'ils décident d'en utiliser plus d'une, il est alors nécessaire de créer les connexions additionnelles. Le présent sous-paragraphe utilise le terme *connexion* pour désigner une connexion de monodiffusion fiable.

Ce processus est lancé par le nœud appelant, qui appelle le protocole monodiffusion fiable sélectionné afin de créer le nombre approprié de connexions additionnelles. A mesure que les connexions deviennent prêtes à utiliser, le nœud appelant transmet sur chacune une unité **MAPConnectRequest**. Cette unité MAPPDU contient les champs suivants:

- le numéro de version du protocole est le même que celui déjà sélectionné pour la connexion initiale;
- le point MAPSAP est le même que celui utilisé pour créer la connexion initiale;
- l'identificateur de référence de domaine est celui qui était envoyé dans l'unité **MAPArbitrateProtocolsRequest** sur la connexion initiale;
- les valeurs de priorités faible et élevée indiqueront la gamme de priorités qui sera envoyée sur cette connexion (les priorités requises, spécifiées par le service MCS, sont typiquement réparties uniformément entre les connexions disponibles).

Le nœud appelé doit accepter chacune des connexions additionnelles en émettant une unité **MAPConnectConfirm** sur chacune. Cette unité MAPPDU contiendra une copie exacte des deux premiers champs de la demande ci-dessus.

Le refus d'une des connexions additionnelles représente une erreur de protocole. Si l'un des deux nœuds ne peut pas créer le nombre de connexions adopté, *toutes* les connexions correspondant à la connexion MAP en question doivent être coupées, ce qui coupe le nœud inférieur du domaine.

Il est important que toutes les priorités requises mappent une connexion une fois ce processus terminé. A mesure que chaque connexion additionnelle est mise en service, elle est mappée avec une gamme de priorités. Toutes les priorités requises qui ne sont pas explicitement mappées avec une connexion additionnelle sont traitées par la connexion initiale. Les priorités qui demeurent *doivent* être contiguës à la priorité zéro (qui est déjà mappée avec la connexion initiale).

### A.8.2.2 Changement de connexions de monodiffusion fiables

Après un nouvel arbitrage de protocoles, il peut être nécessaire que deux nœuds passent dynamiquement à un ensemble différent de connexions de monodiffusion fiables. Cela se produit dans l'une ou l'autre des conditions suivantes: les nœuds ont choisi de passer à un protocole monodiffusion fiable différent; les nœuds ont choisi d'utiliser un nombre différent de connexions de transport; les nœuds ont choisi de changer la taille maximale de l'information utile des connexions de monodiffusion fiable (ce qui ne peut pas s'effectuer lorsque des groupes multidiffusion sont encore actifs); ou les nœuds ont échangé de nouvelles données de configuration applicables au protocole

monodiffusion fiable. A noter qu'une transition dynamique ne peut pas se produire tant que la configuration de l'ensemble précédent de connexions de monodiffusion fiable n'est pas terminée.

Afin de passer dynamiquement à un ensemble différent de connexions, il est nécessaire d'exécuter les étapes suivantes:

- 1) l'initiateur de l'arbitrage doit créer le nombre adopté de nouvelles connexions au nœud éloigné en utilisant le protocole sélectionné. Ce nœud est alors appelé "nœud appelant";
- 2) le nœud appelant émettra une unité **MAPConnectRequest** sur chaque nouvelle connexion, à mesure qu'elle devient prête. Cette unité MAPPDU contient les champs suivants:
  - le numéro de version du protocole est le même que celui déjà sélectionné pour les connexions précédentes;
  - le point MAPSAP est le même que celui utilisé pour créer les connexions précédentes;
  - l'identificateur de référence de domaine est celui qui était envoyé dans l'unité **MAPArbitrateProtocolsRequest** sur la connexion initiale précédente;
  - les valeurs de priorités faible et élevée indiqueront la gamme de priorités qui sera envoyée sur cette connexion (les priorités requises, spécifiées par le service MCS, sont typiquement réparties uniformément entre les connexions disponibles);
- 3) le nœud appelé répondra en envoyant une unité **MAPConnectConfirm** sur chaque nouvelle connexion une fois que la demande aura été reçue. Le numéro de version du protocole et le point MAPSAP sont les mêmes que dans la demande;
- 4) à la réception de chaque confirmation, le nœud appelant suit les nouvelles connexions qui sont prêtes à utiliser. En comparant la gamme de priorités des connexions précédentes avec la gamme de priorités des nouvelles connexions confirmées, le nœud appelant peut déterminer à quel moment une connexion précédente peut être coupée (étant donné que toutes ses priorités sont couvertes par les nouvelles connexions). Lorsque cela se produit, le nœud appelant envoie une unité **MAPDisconnectRequest** à la connexion précédente. A ce moment, le nœud appelant commence à envoyer tout le trafic de monodiffusion fiable dans la gamme de priorités de la connexion précédente sur les nouvelles connexions appropriées;
- 5) dès la réception d'une demande de déconnexion, le nœud appelé envoie une unité **MAPDisconnectConfirm** sur la connexion précédente. Le nœud appelé commence alors à envoyer du trafic de monodiffusion fiable dans la gamme de priorités de la connexion précédente sur les nouvelles connexions appropriées. Le nœud appelé doit également assurer le suivi des connexions précédentes qui ont été coupées. En comparant la gamme de priorités des nouvelles connexions avec la gamme de priorités des connexions précédentes, le nœud appelé peut déterminer à quel moment le traitement des données reçues d'une nouvelle connexion peut commencer sans problème (cela se produit lorsque toutes les connexions précédentes qui acheminaient des données dans la gamme de priorités de la nouvelle connexion ont été coupées);
- 6) lorsque le nœud appelant reçoit une confirmation de déconnexion, on peut supposer que toutes les données en instance de la connexion précédente ont été vidées et que la connexion peut être coupée. En comparant la gamme de priorités des nouvelles connexions avec la gamme de priorités des connexions précédentes, le nœud appelant peut déterminer à quel moment le traitement des données reçues d'une nouvelle connexion peut commencer sans problème (cela se produit lorsque toutes les connexions précédentes qui acheminaient des données dans la gamme de priorités de la nouvelle connexion ont été coupées).

### A.8.3 Attribution et distribution de groupes multidiffusion

Une fois qu'un ensemble de protocoles de transport a été arbitré, le protocole MAP peut commencer à attribuer et à distribuer des adresses de groupe multidiffusion. Afin de réduire le plus possible le nombre des groupes multidiffusion actifs, le protocole MAP ne doit pas assigner d'adresse de groupe multidiffusion à un métacanal tant que n'a pas été détecté du trafic de données qui pourrait utiliser le groupe. Aucune donnée ne se perd, car toutes les données sont initialement transmises par monodiffusion.

Lorsqu'un protocole MAP détecte un flux de données pour lequel il n'existe aucun groupe multidiffusion, une adresse de groupe multidiffusion peut être assignée. Ce processus commence à chaque nœud du domaine sans connexion MAP ascendante ou avec connexion MAP ascendante mais sans protocole multidiffusion commun avec le nœud supérieur. Ces nœuds deviennent des fournisseurs de groupe multidiffusion. Ils examineront les listes des protocoles multidiffusion pris en charge par les nœuds en dessous d'eux et tenteront de choisir les protocoles qui produiront le plus petit nombre d'îlots multidiffusion. Idéalement, il sera possible de sélectionner un seul protocole pris en charge par *tous* les nœuds en aval (permettant à tous les nœuds de niveau inférieur de se retrouver dans le même îlot multidiffusion).

Après qu'un protocole multidiffusion a été sélectionné pour chaque nœud descendant, le protocole MAP obtient une adresse de groupe multidiffusion pour chaque protocole différent à utiliser. Selon un ensemble local de politiques, l'adresse de groupe à assigner peut déjà être utilisée pour un autre métacanal ou elle peut être nouvellement attribuée. La façon dont s'effectue cette attribution dépasse le cadre de ce protocole. Une fois qu'une adresse de groupe multidiffusion a été déterminée, elle est propagée en aval au moyen de l'unité **MAPAddGroupRequest**.

L'unité **MAPAddGroupRequest** indique au destinataire à quels canal, niveau de fiabilité et gamme de priorités correspond l'assignation. Elle indique aussi au destinataire quels protocole multidiffusion et adresse de groupe doivent être utilisés. Tel que décrit dans un sous-paragraphe précédent, la sélection de protocoles multidiffusion pour les nœuds descendants entraîne la définition de limites d'îlot multidiffusion. Tous les nœuds invités à utiliser les mêmes protocole multidiffusion et adresse de groupe se trouvent, par définition, dans le même îlot multidiffusion.

Lorsqu'un nœud reçoit une unité **MAPAddGroupRequest**, il ajoute le groupe à sa liste de groupes multidiffusion actifs. S'il existe des connexions MAP descendantes, il doit sélectionner un protocole multidiffusion pour chaque nœud aval. Dans la mesure du possible, il doit utiliser le protocole même qui a été sélectionné pour lui par le nœud ci-dessus. Si cela est impossible pour quelques nœuds aval, il doit tenter de sélectionner les protocoles qui produiront le plus petit nombre des îlots multidiffusion les plus gros. Une fois les sélections effectuées, le nœud obtiendra une adresse de groupe multidiffusion pour chaque protocole différent à utiliser, et il propagera l'information en aval à l'aide d'une unité **MAPAddGroupRequest**.

La Figure A.8-4 illustre le processus de propagation en aval d'information de groupe multidiffusion.

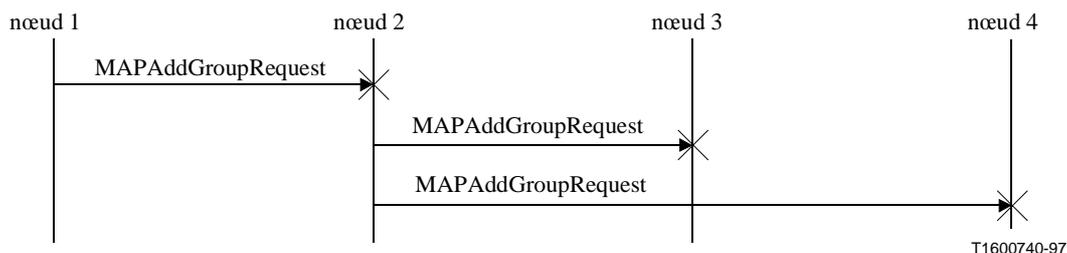


Figure A.8-4/T.125 – Synchronisation de la procédure d'ajout de groupe

#### A.8.4 Gestion du trafic de données multidiffusion (réglage automatique)

Une fois qu'une adresse de groupe multidiffusion a été distribuée, les nœuds de l'îlot multidiffusion peuvent commencer à l'utiliser. Il n'est toutefois pas acceptable que les nœuds présupposent le fonctionnement immédiat de la multidiffusion. Bon nombre de facteurs peuvent empêcher le bon fonctionnement de la multidiffusion. Premièrement, ce ne sont pas tous les réseaux qui prennent en charge la multidiffusion. Cela peut faire en sorte que des données multidiffusion n'atteignent pas un ou plusieurs des nœuds de l'îlot multidiffusion. Même si le réseau ne prend pas en charge la multidiffusion, d'autres facteurs, comme la durée restante (TTL, *time-to-live*), peuvent empêcher des données multidiffusées d'atteindre leur destination.

Pour ces raisons, entre autres, le protocole MAP emploie une stratégie de réglage automatique qui lui permet de dépendre seulement de la multidiffusion lorsque celle-ci est en fonctionnement. Cette stratégie repose sur l'envoi de toutes les données par monodiffusion ET par multidiffusion, puis par la coupure dynamique de la monodiffusion sur les connexions menant à des nœuds qui reçoivent correctement les données par l'intermédiaire de la multidiffusion.

Les données unidiffusées à l'intérieur d'un îlot multidiffusion sont commandées indépendamment pour chaque "flux de données". Un flux de données est identifié de manière univoque par les cinq champs suivants: identificateur de l'émetteur, identificateur du canal, niveau de fiabilité, priorité, et type de données. Par défaut, toutes les données sont transmises par monodiffusion à tous les nœuds de l'îlot multidiffusion qui ont besoin de les recevoir. Les données sont également envoyées au groupe multidiffusion par un nœud particulier de l'îlot multidiffusion (l'émetteur multidiffusion). Lorsqu'un nœud détecte un flux de données qu'il reçoit par multidiffusion, il envoie une unité **MAPDisableUnicastRequest** à son nœud voisin dans la direction de l'émetteur multidiffusion, de telle sorte que le nœud voisin cesse d'envoyer le flux de données par monodiffusion.

L'identité de l'émetteur multidiffusion correspondant à un flux de données particulier est déterminée comme suit par le type de données:

- pour les données non uniformes, l'émetteur multidiffusion est simplement le nœud correspondant au point d'entrée original du flux de données;
- pour les données uniformes, toutes les données sont unidiffusées en amont vers le fournisseur supérieur, avant d'être admissibles à la multidiffusion. Les données sont envoyées vers le groupe multidiffusion par le fournisseur de groupe multidiffusion, sur sa voie de retour (l'émetteur multidiffusion est *toujours* le fournisseur de groupe multidiffusion). Lors de l'émission d'une unité **MAPDisableUnicastRequest** visant des données uniformes, il est toujours envoyé en amont. De plus, il ne doit pas être tenu compte de l'identificateur de l'émetteur pour identifier de manière univoque des flux de données uniformes;
- pour les données transmises par mandataire, toutes les données sont unidiffusées vers le fournisseur de groupe multidiffusion. Le fournisseur de groupe multidiffusion achemine les données en amont par monodiffusion (s'il y a lieu), tout en retournant simultanément les données en aval (à la fois par monodiffusion et multidiffusion). Comme pour les données uniformes, l'émetteur multidiffusion est *toujours* le fournisseur de groupe multidiffusion. Lorsqu'il émet une unité **MAPDisableUnicastRequest** pour les données transmises par mandataire, cette unité est toujours envoyée en amont. De plus, il ne doit pas être tenu compte de l'identificateur de l'émetteur pour identifier de manière univoque les flux de données transmises par mandataire.

Le type des données transmises par mandataire offre un autre moyen d'envoyer des données non uniformes. Les données transmises par mandataire **ne peuvent pas** servir à transmettre des données uniformes. L'utilisation de données transmises par mandataire réduit la quantité de l'information d'état du flux de données, étant donné que tous les émetteurs de données transmises par mandataire

pour un ensemble donné identificateur de canal/niveau de fiabilité/priorité sont fusionnés en un seul flux de données. Les données transmises par mandataire sont également utiles dans les réseaux qui ne permettent pas aux nœuds de produire de l'information multidiffusion. La connaissance du moment où des données transmises par mandataire doivent être utilisées constitue une question de mise en œuvre locale. Cependant, une fois qu'un nœud commence à envoyer des données transmises par mandataire pour un ensemble donné identificateur d'émetteur/identificateur de canal/niveau de fiabilité/priorité, il ne peut pas décider par la suite d'utiliser le type de données non uniformes (l'inverse est également vrai). La décision d'utiliser des données transmises par mandataire est strictement locale, et différents nœuds peuvent traiter des émetteurs locaux différemment.

Une autre considération importante s'applique à l'utilisation de données transmises par mandataire. Chaque nœud du trajet entre l'émetteur et le fournisseur de groupe multidiffusion voit passer les données deux fois (une fois sur leur trajet ascendant et une fois sur leur trajet descendant). Les nœuds ne doivent envoyer les données au service MCS qu'une seule fois. Il n'importe pas que cela s'effectue sur le trajet ascendant ou sur le trajet descendant (mais le processus doit s'effectuer de façon constante). De plus, le nœud d'origine ne doit *jamais* retourner les données vers l'utilisateur qui les a envoyées.

Avant qu'un nœud indique à son voisin d'invalider la monodiffusion pour un flux de données, il doit être certain de pouvoir passer à la multidiffusion sans perte de données. Pour ce faire, il doit recevoir au moins un paquet de données par monodiffusion, afin de savoir quel doit être le prochain numéro de la séquence (à noter que les numéros séquentiels sont propres à chaque flux de données et assignés par l'émetteur du flux de données). Avant ou après la réception de ce paquet de monodiffusion, le nœud doit recevoir un paquet de données par multidiffusion, au moyen d'un numéro séquentiel inférieur ou égal au prochain numéro séquentiel prévu. Cela indique la superposition des deux flux de données et permet d'invalider la monodiffusion.

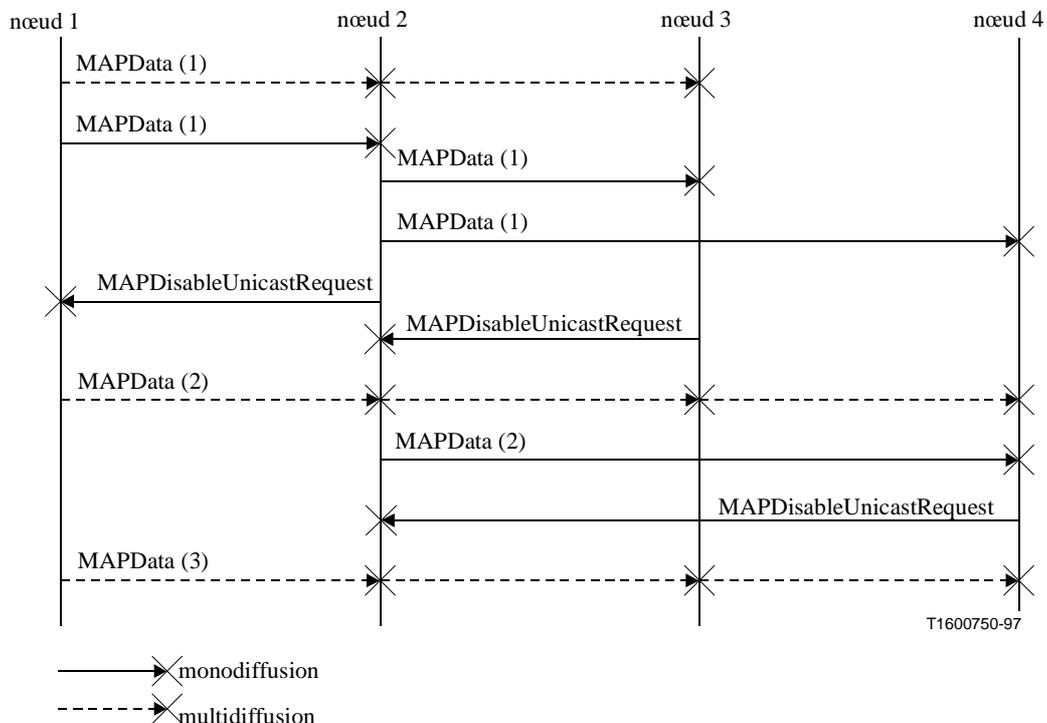
Lorsque des données sont d'abord transmises dans un nouveau flux de données, elles sont envoyées par monodiffusion ET multidiffusion par l'intermédiaire de l'îlot. Avec le temps, la monodiffusion est invalidée dans la mesure du possible, ce qui crée un trafic de réseau optimal.

La Figure A.8-5 donne un exemple de cette stratégie de réglage automatique (voir la hiérarchie de domaine sur la Figure A.8-1). Cet exemple suit la transmission de trois paquets de données provenant d'un précédent flux de données inconnu, de la façon décrite ci-dessous:

- 1) une application d'utilisateur raccordée au nœud 1 transmet un paquet de données. Le nœud 1 transmet les données vers le groupe multidiffusion. Il transmet aussi les données vers le nœud 2 par monodiffusion, car le nœud 2 n'a pas encore invalidé la monodiffusion pour le flux de données en question. Le nœud 4 n'a pas encore terminé d'adhérer au groupe, de sorte que le paquet de données multidiffusion arrive seulement aux nœuds 2 et 3. Les nœuds 2 et 3 gardent les données étant donné qu'ils ne savent pas encore quel numéro séquentiel est valide. Lorsque le nœud 2 reçoit les données par monodiffusion, il les redirige vers les nœuds 3 et 4 (par monodiffusion). Les nœuds 2 et 3 reconnaissent maintenant que le paquet multidiffusé est identique au paquet unidiffusé, de sorte qu'ils émettent chacun une unité **MAPDisableUnicastRequest** dans la direction de l'émetteur;
- 2) l'application de l'utilisateur au nœud 1 transmet un autre paquet de données dans le même flux. Comme auparavant, le nœud 1 transmet les données vers le groupe multidiffusion. Cette fois, le nœud 1 *ne transmet pas* le paquet vers le nœud 2 par monodiffusion, étant donné que le nœud 2 a déjà invalidé la monodiffusion pour le flux de données en question. Cette fois, la transmission multidiffusée arrive à l'ensemble des trois nœuds en aval (y compris le nœud 4). Le nœud 2 transmet les données par monodiffusion vers le nœud 4, mais pas vers le nœud 3 (étant donné que le nœud 3 a invalidé la monodiffusion pour ce flux

de données). Le nœud 4 sait maintenant qu'il est synchronisé par rapport au flux de données multidiffusées, de sorte qu'il émet une unité **MAPDisableUnicastRequest** en amont;

- 3) l'application de l'utilisateur au nœud 1 transmet un troisième paquet de données dans le même flux. Comme auparavant, le nœud 1 transmet les données au groupe multidiffusion. Cette fois, cependant, les données ne sont pas unidiffusées, car tous les nœuds de l'îlot ont invalidé la monodiffusion pour le flux en question.



**Figure A.8-5/T.125 – Exemple de synchronisation de la procédure d'invalidation de monodiffusion pour un flux de données particulier**

### A.8.5 Abandon de la multidiffusion

Le sous-paragraphe précédent a mis l'accent sur ce qui arrive lorsque la multidiffusion est en service. Le présent sous-paragraphe explique ce que peut faire un nœud pour optimiser ses transmissions dans le cas où la multidiffusion ne semble pas fonctionner.

Lorsqu'un nœud commence à transmettre des données pour un flux non uniforme, il utilise à la fois la monodiffusion et la multidiffusion. Si les nœuds adjacents reçoivent les données multidiffusées, ils invalideront les transmissions de la monodiffusion de la façon décrite dans le sous-paragraphe précédent. Si les nœuds adjacents ne reçoivent pas les données par multidiffusion, cette invalidation de la monodiffusion n'aura jamais lieu. Le nœud émetteur pourrait alors se retrouver forcé d'émettre chaque unité PDU de données un plus grand nombre de fois que nécessaire (une fois pour chaque nœud adjacent, plus une fois pour la multidiffusion). Ce problème est plus marqué pour un nœud feuille, qui pourrait en venir à utiliser deux fois la largeur de bande requise. Cela peut s'avérer inacceptable (surtout si le nœud est raccordé par une connexion à basse vitesse). Malheureusement, le nœud ne peut pas simplement arrêter d'émettre des données par multidiffusion. Il est possible qu'un nœud (non adjacent) situé quelque part dans l'îlot reçoive les données multidiffusées et ait déjà invalidé la monodiffusion.

Chaque unité PDU **MAPData** contient un fanion qui peut être utilisé pour résoudre ce problème. Ce fanion est appelé fanion Unicast Forward (acheminement de monodiffusion). Ce fanion force tous les nœuds intermédiaires de l'îlot multidiffusion à acheminer encore une fois ce flux de données par monodiffusion vers l'ensemble des récepteurs (même s'ils ont auparavant invalidé la monodiffusion). Lorsqu'un nœud décide qu'il veut arrêter d'émettre un flux de données particulier par multidiffusion, il émet simplement toutes les futures unités **MAPData** (pour le flux de données en question) par monodiffusion seulement, le fanion Unicast Forward étant remis à 1. Le nœud peut ainsi, en toute sécurité, arrêter la transmission par multidiffusion.

#### A.8.6 Suppression de groupes multidiffusion

Lorsqu'un fournisseur de groupe multidiffusion s'aperçoit qu'un groupe multidiffusion n'est plus nécessaire, le groupe en question peut être supprimé de l'îlot. Pour des données fiables, il est très important que le groupe soit supprimé d'une façon qui garantisse l'absence de pertes de données.

Pour obtenir ce résultat, le fournisseur de groupe multidiffusion invalidera d'abord la réception multidiffusion pour le groupe. Une unité **MAPDisableMulticastRequest** sera envoyée à cette fin à tous les nœuds de l'îlot (cette demande est automatiquement acheminée par chaque nœud récepteur vers les nœuds qui se trouvent à un niveau inférieur). Cette unité MAPPDU fait en sorte que chaque nœud entreprenne le processus de nouvelle validation de la monodiffusion relativement à tous les flux de données du groupe pour lesquels la monodiffusion a été invalidée précédemment. Cela s'effectue par l'envoi d'une unité **MAPEnableUnicastRequest** en direction de l'émetteur pour chaque flux de données en question. L'unité MAPPDU marque aussi le groupe comme invalidé, ce qui empêche toute invalidation ultérieure de la monodiffusion pour le groupe. A noter que cela N'A PAS pour effet d'arrêter l'émission de données vers le groupe multidiffusion. De plus, les nœuds doivent rester à l'état d'adhésion en tant que récepteurs du groupe multidiffusion, au moins jusqu'à ce que la monodiffusion ait été correctement validée pour tous les flux de données en question.

Lorsque l'unité **MAPDisableMulticastRequest** atteint les nœuds inférieurs de l'îlot multidiffusion, ces nœuds peuvent répondre par une unité **MAPDisableMulticastConfirm**. Auparavant, ils doivent s'assurer que tous les flux de données pour lesquels la monodiffusion a été invalidée précédemment dans le groupe sont complètement validés de nouveau. Chaque nœud doit à cette fin avoir reçu une unité **MAPEnableUnicastConfirm** pour chaque demande en instance. De plus, chaque nœud doit examiner le numéro séquentiel de la confirmation pour veiller à ce que toutes les données en cours de multidiffusion aient bien été reçues (si tel n'est pas le cas, il doit attendre que cela se produise). Cela fait, l'unité **MAPDisableMulticastConfirm** est envoyée au nœud de niveau supérieur.

Les nœuds intermédiaires de l'îlot doivent attendre qu'ils aient reçu des unités PDU **MAPDisableMulticastConfirm** provenant de tous les nœuds situés à un niveau inférieur dans l'îlot. Ils doivent également attendre que tous les flux de données unidiffusés qui ont été invalidés précédemment soient validés de nouveau, de la façon décrite ci-dessus. Ils peuvent ensuite émettre une unité **MAPDisableMulticastConfirm** vers le nœud de niveau supérieur.

Une fois que le fournisseur de groupe multidiffusion a reçu des unités PDU **MAPDisableMulticastConfirm** provenant de tous les nœuds en aval et qu'il s'est assuré que ses flux de données de monodiffusion sont validés de nouveau, il est connu que le groupe n'est plus en utilisation. A ce moment, le fournisseur de groupe multidiffusion peut envoyer une unité **MAPRemoveGroupRequest** en aval. Cette unité se propagera vers tous les nœuds de l'îlot, amenant chaque nœud à quitter le groupe (s'il y avait adhéré), et elle sera éliminée de leur liste de groupes actifs.

La Figure A.8-6 illustre le processus d'élimination de groupe pour le domaine représentatif de la Figure A.8-1. Il s'agit de la suite de l'exemple de la Figure A.8-5, dans lequel la monodiffusion a été invalidée pour un seul émetteur raccordé au nœud 1.

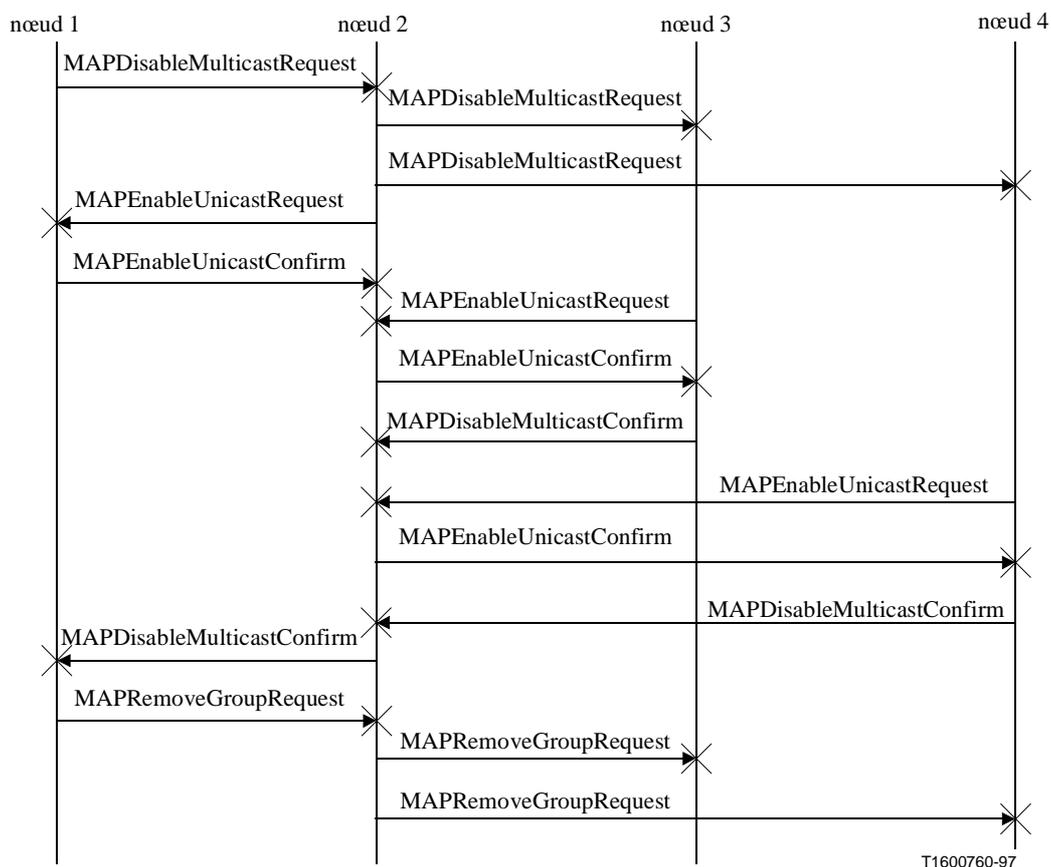


Figure A.8-6/T.125 – Synchronisation de la procédure d'élimination de groupe

## A.9 Description des unités MAPPDU

Le présent sous-paragraphe décrit les unités PDU qui sont utilisées par le protocole MAP. Une brève description décrit d'abord la fonction de chaque unité MAPPDU. Vient ensuite un tableau indiquant les champs qui sont codés dans l'unité MAPPDU. Enfin, chaque champ de l'unité MAPPDU est décrit.

A noter que le prochain sous-paragraphe contient la définition ASN.1 de chaque unité MAPPDU.

### A.9.1 MAPConnectRequest

Lors de la création d'une nouvelle connexion entre deux nœuds pour un domaine donné, la première étape consiste à ouvrir une connexion de monodiffusion fiable. L'unité **MAPConnectRequest** est la première unité MAPPDU transmise par l'intermédiaire de cette connexion. Le nœud appelant envoie cette unité MAPPDU pour indiquer au nœud appelé la version de protocole qu'il désire utiliser. Il s'agit seulement d'une valeur suggérée. Il revient au nœud appelé de prendre la décision finale et de la retourner au nœud appelant en envoyant une unité **MAPConnectConfirm**.

Cette unité MAPPDU comprend un champ de point d'accès au service MAP (MAPSAP), qui permet au nœud appelant de spécifier qui, au nœud appelé, doit recevoir l'appel d'arrivée.

Cette unité MAPPDU contient aussi un identificateur de référence de domaine, qui permet au nœud appelant d'associer cette connexion à un domaine particulier. Cela se produit lorsqu'un arbitrage de protocoles de transport entraîne la création de connexions additionnelles ou la transition vers un ensemble différent de connexions.

Le tableau suivant résume le contenu de l'unité **MAPConnectRequest**.

Numéro de version
MAPSAP de connexion
Identificateur de référence de domaine
Gamme de priorités

*Numéro de version*: ce champ indique la version de ce protocole que le nœud appelant désire utiliser. Elle comprend à la fois des numéros de version majeure et de version mineure. La valeur de ce champ pour le protocole défini dans la présente annexe est 1.0 (version majeure 1, version mineure 0). A noter qu'il s'agit de la valeur suggérée. Le nœud appelé déterminera la version de protocole réelle à utiliser.

*MAPSAP de connexion*: ce champ indique au récepteur quel point d'accès au service MAP doit traiter la connexion d'arrivée. La valeur zéro (0) indique que l'appelant désire accéder au point MAPSAP par défaut. Toutes les autres valeurs sont assignées et obtenues hors bande par rapport à ce protocole.

*Identificateur de référence de domaine*: ce champ permet au nœud appelant d'associer cette connexion à un domaine particulier du nœud appelé. Lors de la création de la connexion initiale à un nœud éloigné, ce champ doit être remis à zéro (0). Lors de la création de connexions additionnelles ou du passage à un nouvel ensemble de connexions, ce champ doit être remis à la valeur qui était envoyée dans l'unité **MAPArbitrateProtocolsRequest** sur la connexion initiale. Avec l'adresse d'appel, cette valeur permet au nœud appelé de savoir à quel domaine la connexion d'arrivée doit être liée.

*Gamme de priorités*: il s'agit de la gamme de priorités des données qui seront acheminées par la connexion. La connexion initiale à un nœud éloigné permet toujours de traiter *seulement* la priorité zéro (zéro correspond à la priorité supérieure).

### A.9.2 MAPConnectConfirm

Cette unité MAPPDU est utilisée pour accepter une nouvelle connexion à partir d'un autre nœud. Elle informe le nœud appelant de la version de protocole qui sera utilisée pour la connexion. Elle informe aussi le nœud appelant du point MAPSAP éloigné qui a accepté l'appel.

Le tableau ci-dessous résume le contenu de l'unité **MAPConnectConfirm**.

Numéro de version
MAPSAP de connexion

*Numéro de version*: ce champ indique la version de ce protocole qui sera utilisée pour la connexion. Elle comprend à la fois des numéros de version majeure et de version mineure. La valeur de ce champ pour le protocole défini dans la présente annexe est 1.0 (version majeure 1, version mineure 0). Si le nœud appelant ne peut pas traiter (ou ne veut pas traiter) la version spécifiée par le nœud appelé de l'unité MAPPDU en question, il doit se déconnecter.

*MAPSAP de connexion*: il s'agit du point MAPSAP au nœud appelé qui a effectivement accepté l'appel. Si le point MAPSAP spécifié dans l'unité **MapConnectRequest** était à une valeur autre que zéro, ce champ doit être remis à cette même valeur. Si le point MAPSAP spécifié dans l'unité **MapConnectRequest** était à zéro, ce champ est remis à la valeur valide du point MAPSAP qui a accepté l'appel (et qui peut également être zéro).

### A.9.3 MAPDisconnectRequest

Cette unité MAPPDU est utilisée pour couper une connexion de transport existante entre deux nœuds ou pour rejeter une connexion d'arrivée si le nœud appelé ne veut pas prendre part à la conférence. Une fois que cette unité MAPPDU a été envoyée, la connexion ne peut pas être utilisée pour la transmission d'autres données.

Si l'unité MAPPDU en question est transmise dans le cadre d'une transition à un nouvel ensemble de connexions de monodiffusion fiable, l'émetteur doit remettre à 1 le fanion Confirm Required (confirmation requise). Le récepteur est ainsi informé du fait qu'il doit répondre en émettant une unité **MAPDisconnectConfirm** sur la même connexion.

Lorsque le fanion Confirm Required est au niveau 1, le nœud émetteur doit attendre l'unité **MAPDisconnectConfirm** avant de couper la connexion. Cette mesure est nécessaire pour que toutes les données soient éliminées de la connexion avant la coupure de celle-ci.

Le tableau ci-dessous résume le contenu de l'unité **MAPDisconnectRequest**.

Raison
Confirmation requise

*Raison*: ce champ indique au destinataire pourquoi la connexion est coupée ou rejetée.

*Confirmation requise*: il s'agit d'un fanion booléen qui indique si le récepteur doit ou non répondre par une unité **MAPDisconnectConfirm**. Ce fanion est remis à 1 si la déconnexion se produit dans une transition vers un ensemble différent de connexions de monodiffusion fiable. L'attente de la confirmation permet aux deux extrémités de savoir que toutes les unités MAPPDU transmises sur la connexion ont été traitées.

### A.9.4 MAPDisconnectConfirm

Cette unité MAPPDU est transmise en réponse à une unité **MAPDisconnectRequest** dans laquelle le fanion Confirm Required est remis à 1. En envoyant cette confirmation, les deux extrémités peuvent être assurées que toutes les données ont été éliminées de la connexion avant de couper celle-ci et de commencer à en utiliser une nouvelle.

Dès la réception de cette unité MAPPDU, le nœud peut, sans problème, couper la connexion sur laquelle l'unité est arrivée. Selon l'état des connexions de remplacement, il peut être possible que le nœud commence à traiter des données sur une ou plusieurs de ces connexions.

Aucun champ n'est associé à cette unité MAPPDU.

### A.9.5 MAPArbitrateProtocolsRequest

Une fois qu'une connexion a été établie pour un domaine donné entre deux nœuds, cette unité MAPPDU peut servir à arbitrer le choix des protocoles de transport qui seront utilisés pour l'acheminement des données. Cette unité MAPPDU *doit* être envoyée par le nœud appelant lors de l'établissement de la connexion initiale. Elle peut également être envoyée par tout nœud s'il devient nécessaire, à un moment quelconque, de procéder à un nouvel arbitrage du choix des protocoles de transport.

Cette unité MAPPDU indique au récepteur quels sont les protocoles de transport pris en charge par l'émetteur. Cela comprend les protocoles monodiffusion et multidiffusion, pour les deux niveaux de fiabilité. Il faut noter qu'il est *obligatoire* d'indiquer dans cette unité MAPPDU le protocole monodiffusion fiable utilisé pour la connexion existante (cela permet au récepteur de rejeter une

transition de protocole monodiffusion fiable en sélectionnant simplement le protocole déjà en cours d'utilisation).

L'émetteur peut assigner à chaque protocole de transport énuméré dans cette unité MAPPDU une valeur de pondération des préférences. Dans le cas des protocoles multidiffusion, il y a également un champ utilisé par un nœud aval pour indiquer combien de nœuds de son sous-arbre peuvent prendre en charge le protocole. Le nœud amont peut utiliser cette information pour mieux sélectionner les protocoles multidiffusion d'un métacanal donné.

Cette unité MAPPDU sert également à communiquer au récepteur un identificateur de référence de domaine. Cette valeur, conjuguée à l'adresse réseau de l'émetteur, peut ensuite servir au récepteur pour identifier de façon univoque un domaine donné. Cela est nécessaire dans le cas où le domaine ne peut pas être déterminé de façon implicite. En particulier, cette valeur est utilisée lors de la création de connexions monodiffusion fiables supplémentaires, ou en cas d'émission d'unités PDU de données non fiables.

Si deux nœuds adjacents tentent de lancer simultanément un nouvel arbitrage, la demande du nœud aval est ignorée. Cette situation peut être détectée aux deux nœuds, étant donné qu'ils recevront tous deux une demande d'arbitrage alors qu'ils s'attendent à recevoir une confirmation. Le nœud amont répond en rejetant la demande (et en continuant d'attendre la confirmation). Le nœud aval répond en traitant la demande (et en cessant d'attendre la confirmation). Si le nœud aval désire toujours procéder à un nouvel arbitrage, il peut émettre une autre demande (après avoir envoyé la confirmation).

Le tableau ci-dessous résume le contenu des unités **MAPArbitrateProtocolsRequest**.

Identificateur de référence de domaine
Fanion de données à suivre
Menu de protocoles de transport

*Identificateur de référence de domaine:* il s'agit d'une valeur assignée localement, qui est utilisée par l'émetteur pour désigner un domaine particulier. A la fin du processus d'arbitrage, l'émetteur peut utiliser cette valeur pour indiquer à quel domaine une unité PDU **MAPConnectRequest** est associée. Elle sert également à indiquer à quel domaine une unité PDU **MAPData** non fiable est associée.

*Fanion de données à suivre:* quand il est mis à un, ce fanion indique qu'il y a trop de protocoles de transport pour une seule unité MAPPDU et que le récepteur doit attendre l'arrivée d'unités PDU **MAPArbitrateProtocolsRequest** subséquentes avant d'entreprendre le traitement. La dernière unité PDU de la séquence devrait réinitialiser ce fanion.

*Menu de protocoles de transport:* ce champ contient une série d'entrées de demande de protocoles de transport (*Transport Protocol Request Entries*). Il s'agit d'une liste des protocoles de transport que l'émetteur peut utiliser aux fins de la conférence. Il devrait s'agir de la liste complète des protocoles pris en charge par l'émetteur (les protocoles non fiables sont facultatifs). Le protocole monodiffusion fiable utilisé pour acheminer cette MAPPDU doit figurer dans la liste. Chaque entrée contient un champ permettant à l'appelant de préciser quels sont les protocoles qu'il préfère utiliser. Pour un type de protocole donné, l'émetteur devrait également énumérer les entrées par ordre de préférence. Il revient au récepteur de sélectionner les protocoles pouvant effectivement être utilisés pour la connexion. Dans le cas des protocoles monodiffusion, le récepteur devrait tenter de sélectionner, si possible, les protocoles préférés. Dans le cas des protocoles multidiffusion, le récepteur devrait sélectionner tous les protocoles qui sont communs à l'émetteur et au récepteur. Il peut ne pas y avoir de protocole multidiffusion commun pour un niveau de fiabilité donné (toutes les données seront alors unidiffusées à ce niveau de fiabilité). Il peut également ne pas y avoir de protocole

monodiffusion non fiable commun (les données non fiables seront alors acheminées par les connexions fiables).

#### A.9.5.1 Entrée de demande de protocoles de transport

Le menu est simplement une série d'entrées de demande de protocole de transport (*Transport Protocol Request Entries*), dont le format est défini dans le tableau ci-dessous.

Identificateur de protocole de transport
Type de protocole de transport
Adresse réseau
Fanion de taille maximale fixe d'information utile
Taille maximale d'information utile
Pondération de préférences (facultatif)
Compte de nœuds (facultatif)
Nombre de connexions (facultatif)
Données de configuration (facultatif)

*Identificateur de protocole de transport*: ce champ identifie de façon univoque le protocole de transport. Les valeurs de ce champ devront être normalisées, mais il sera également possible de préciser des valeurs exclusives.

*Type de protocole de transport*: ce champ est mis à l'une de ces quatre valeurs: monodiffusion fiable, monodiffusion non fiable, multidiffusion fiable ou multidiffusion non fiable.

*Adresse réseau*: il s'agit de l'adresse à laquelle on peut joindre l'expéditeur au moyen du protocole.

*Fanion de taille maximale fixe d'information utile*: il s'agit d'un fanion booléen qui indique si la taille maximale d'information utile est négociable ou non pour cette entrée de protocole de transport. De plus amples renseignements sont fournis ci-dessous, à l'entrée "Taille maximale d'information utile".

*Taille maximale d'information utile*: ce champ précise la taille maximale d'information utile prise en charge par le protocole. Si le fanion de taille maximale fixe d'information utile est réinitialisé, il s'agit de la taille maximale prise en charge par la mise en œuvre locale (mais une taille inférieure peut être négociée). Si le fanion de taille maximale fixe d'information utile est mis à un, il s'agit de la taille exacte à utiliser pour accepter cette entrée de protocole de transport. Le protocole MAP utilisera les valeurs négociées pour établir la taille maximale des trames de données (de plus amples renseignements concernant les trames de données sont présentés à la section **MAPData**). La plus petite taille maximale d'information utile prise en charge par le protocole MAP est de 128.

*Pondération de préférences*: ce champ facultatif sert à signaler au récepteur quels sont les protocoles préférés. Une valeur plus élevée signale qu'un protocole est préféré à un protocole ayant une valeur moins élevée. En l'absence de ce champ, on suppose que la valeur de pondération de préférences est de zéro. Ce champ n'est pas utilisé par les protocoles multidiffusion quand l'unité MAPPDU est acheminée en aval.

*Compte de nœuds*: ce champ facultatif n'est utilisé que pour les protocoles multidiffusion (fiables et non fiables), et seulement quand l'unité MAPPDU est acheminée en amont. Il indique combien de nœuds, au niveau de l'émetteur ou à un niveau inférieur, prennent en charge cette entrée de protocole multidiffusion. Ce compte peut être utilisé par le nœud amont pour mieux choisir le protocole multidiffusion à utiliser lors du mappage d'un nouveau métacanal et de nœuds multidiffusion. Pour que le compte demeure exact, il faut procéder à un nouvel arbitrage quand un nouveau nœud MAP

adhère à la conférence ou quand un nœud quitte la conférence. Ce nouvel arbitrage ne vise que la branche de l'arbre de domaine qui mène vers le point où le nœud a adhéré à la conférence ou l'a quittée. Pour réduire le trafic réseau, il ne devrait pas y avoir de nouvel arbitrage *chaque* fois qu'un nœud adhère ou quitte, quand la conférence dépasse une certaine taille. Ce champ peut également servir à prévenir l'utilisation de la multidiffusion quand il y a moins de  $N$  nœuds en jeu ( $N = 3?$ ).

*Nombre de connexions*: ce champ facultatif n'est présent que pour les protocoles monodiffusion fiables. Il indique le nombre de connexions de transport que souhaite créer l'émetteur pour une connexion MAP donnée (si ce protocole est choisi). Si le récepteur choisit ce protocole, il devrait choisir une valeur égale ou inférieure à celle précisée par l'émetteur.

*Données de configuration*: ce champ facultatif est composé d'une chaîne d'octets de longueur variable et est utilisé par les piles de protocoles de transport pour acheminer l'information de configuration. Ce champ ne sera pas présent si les piles de transport ne veulent pas échanger d'information de configuration, ni s'il s'agit d'un nouvel arbitrage et que les piles de transport désirent conserver la valeur déjà convenue. Il faut noter que si ce champ *est* présent pendant un nouvel arbitrage, et que ce protocole est choisi comme protocole monodiffusion, il faut faire une transition vers une nouvelle série de connexions monodiffusion (étant donné que les données de configuration peuvent avoir été modifiées). Le format de ce champ est propre à chaque protocole de transport. Dans le cas des protocoles de transport normalisés, ce champ doit être défini dans une norme. Dans le cas des protocoles de transport exclusifs, le fabricant peut choisir ce qui lui semble approprié. Ce champ peut servir à communiquer des capacités, et il peut être modifié dans la confirmation **MAPArbitrateProtocolsConfirm** pour fins d'arbitrage de ces capacités. Ce champ ne doit pas être volumineux au point où la taille de l'unité MAPPDU excède 16 384 octets.

#### A.9.6 MAPArbitrateProtocolsConfirm

Cette unité MAPPDU sert à arrêter définitivement l'arbitrage des protocoles de transport d'une connexion donnée. Après réception d'une demande **MAPArbitrateProtocolsRequest**, le récepteur doit répondre en émettant cette MAPPDU.

La demande MAPPDU contient une liste de tous les protocoles de transport pris en charge par l'émetteur (y compris les préférences). Le récepteur doit choisir dans cette liste les protocoles pouvant servir à la connexion. Cela comprend *exactement* un protocole monodiffusion fiable (et, facultativement, un protocole monodiffusion non fiable). Si l'émetteur précise des préférences relativement aux protocoles monodiffusion, il faudrait en tenir compte. Le récepteur devrait également établir l'ensemble complet de protocoles multidiffusion qui sont communs aux deux nœuds. Le retour de cet ensemble commun dans l'unité de confirmation aide à garantir que les meilleurs protocoles multidiffusion sont sélectionnés lors de l'attribution de groupes.

Cette unité MAPPDU sert également à communiquer au récepteur un identificateur de référence de domaine. Cette valeur, conjuguée à l'adresse réseau de l'émetteur, peut ensuite servir au récepteur pour identifier de façon univoque un domaine donné. Cela est nécessaire dans le cas où le domaine ne peut pas être déterminé de façon implicite. En particulier, cette valeur est utilisée en cas d'émission d'unités PDU de données non fiables.

Le tableau ci-dessous résume le contenu des unités **MAPArbitrateProtocolsConfirm**.

Identificateur de référence de domaine
Fanion de données à suivre
Menu de protocoles de transport

*Identificateur de référence de domaine*: il s'agit d'une valeur assignée localement, qui est utilisée par l'émetteur pour désigner un domaine particulier. A la fin du processus d'arbitrage, l'émetteur peut utiliser cette valeur pour indiquer à quel domaine une unité PDU **MAPData** non fiable est associée.

*Fanion de données à suivre*: quand il est mis à un, ce fanion indique qu'il y a trop de protocoles de transport pour une seule unité MAPPDU et que le récepteur doit attendre l'arrivée d'unités PDU **MAPArbitrateProtocolsConfirm** subséquentes avant d'entreprendre le traitement. La dernière unité PDU de la séquence devrait réinitialiser ce fanion.

*Menu de protocoles de transport*: ce champ contient une série d'entrées de confirmation de protocole de transport (*Transport Protocol Confirm Entries*). Il s'agit d'une liste des protocoles de transport que le nœud appelé a décidé d'utiliser aux fins de la connexion. Le nœud appelé doit au moins indiquer un protocole monodiffusion fiable. Il peut également indiquer un protocole monodiffusion non fiable et des protocoles associés aux niveaux de fiabilité multidiffusion. En l'absence d'un protocole monodiffusion non fiable, les données non fiables seront unidiffusées au moyen des connexions fiables. En l'absence d'un niveau de fiabilité multidiffusion particulier, toutes les données associées à ce niveau de fiabilité seront unidiffusées entre les deux nœuds (les îlots multidiffusion seront séparés pour ce niveau de fiabilité). Si les protocoles monodiffusion arbitrés sont différents des protocoles déjà utilisés, il doit y avoir une transition entre les protocoles monodiffusion.

#### **A.9.6.1 Entrée de confirmation de protocole de transport**

Le menu est simplement une série d'entrées de confirmation de protocole de transport (*Transport Protocol Confirm Entries*), dont le format est défini dans le tableau ci-dessous.

Identificateur de protocole de transport
Type de protocole de transport
Adresse réseau
Taille maximale d'information utile
Pondération de préférences (facultatif)
Compte de nœuds (facultatif)
Nombre de connexions (facultatif)
Données de configuration (facultatif)

*Identificateur de protocole de transport*: ce champ identifie de façon univoque le protocole de transport. Les valeurs de ce champ devront être normalisées, mais il sera également possible de préciser des valeurs exclusives.

*Type de protocole de transport*: ce champ est mis à l'une de ces quatre valeurs: monodiffusion fiable, monodiffusion non fiable, multidiffusion fiable ou multidiffusion non fiable.

*Adresse réseau*: il s'agit de l'adresse à laquelle on peut joindre l'expéditeur au moyen du protocole.

*Taille maximale d'information utile*: ce champ précise la taille maximale d'information utile utilisée par le protocole. Il faut noter que la taille maximale d'information utile d'un protocole multidiffusion donné *doit* être la même que la taille maximale d'information utile du protocole monodiffusion ayant le même niveau de fiabilité, étant donné que les trames de données pourraient devoir être unidiffusées à l'intérieur de l'îlot multidiffusion. La plus petite taille maximale d'information utile prise en charge par le protocole MAP est de 128.

*Pondération de préférences*: ce champ facultatif sert à signaler au récepteur quels sont les protocoles préférés. Une valeur plus élevée signale qu'un protocole est préféré à un protocole ayant une valeur

moins élevée. En l'absence de ce champ, on suppose que la valeur de pondération des préférences est de zéro. Ce champ ne sert que pour les protocoles multidiffusion, quand l'unité MAPPDU est acheminée en amont.

*Compte de nœuds*: ce champ facultatif n'est utilisé que pour les protocoles multidiffusion (fiabiles et non fiabiles), et seulement quand l'unité MAPPDU est acheminée en amont. Il indique combien de nœuds, au niveau de l'émetteur ou à un niveau inférieur, prennent en charge cette entrée de protocole multidiffusion. Ce compte peut être utilisé par le nœud amont pour mieux choisir le protocole multidiffusion à utiliser lors du mappage d'un nouveau métacanal et de nœuds multidiffusion. Pour que le compte demeure exact, il faut procéder à un nouvel arbitrage quand un nouveau nœud MAP adhère à la conférence ou quand un nœud quitte la conférence. Ce nouvel arbitrage ne vise que la branche de l'arbre de domaine qui mène vers le point où le nœud a adhéré à la conférence ou l'a quittée. Pour réduire le trafic réseau, il ne devrait pas y avoir de nouvel arbitrage *chaque* fois qu'un nœud adhère ou quitte, quand la conférence dépasse une certaine taille. Ce champ peut également servir à prévenir l'utilisation de la multidiffusion quand il y a moins de  $N$  nœuds en jeu ( $N = 3?$ ).

*Nombre de connexions*: ce champ facultatif n'est présent que pour les protocoles monodiffusion. Il indique le nombre de connexions de transport qui seront créées pour une connexion MAP donnée (pour ce niveau de fiabilité). Cette valeur devrait être la moindre d'entre la valeur demandée et la valeur prise en charge localement.

*Données de configuration*: ce champ facultatif est composé d'une chaîne d'octets de longueur variable et est utilisé par les piles de protocoles de transport pour acheminer l'information de configuration. Ce champ ne sera pas présent si les piles de transport ne veulent pas échanger d'information de configuration, ni s'il s'agit d'un nouvel arbitrage et que les piles de transport désirent conserver la valeur déjà convenue. Il faut noter que si ce champ *est* présent pendant un nouvel arbitrage, et que ce protocole est choisi comme protocole monodiffusion, il faut faire une transition vers une nouvelle série de connexions monodiffusion (étant donné que les données de configuration peuvent avoir été modifiées). Le format de ce champ est propre à chaque protocole de transport. Dans le cas des protocoles de transport normalisés, ce champ doit être défini dans la norme. Dans le cas des piles exclusives, le fabricant peut choisir ce qui lui semble approprié. Ce champ peut servir à communiquer des capacités, et il peut être modifié pour fins d'arbitrage de ces capacités. Ce champ ne doit pas être volumineux au point où la taille de l'unité MAPPDU excède 16 384 octets.

### A.9.7 MAPData

Cette unité MAPPDU est le pilier du protocole. Elle sert à l'acheminement de toutes les données utilisateur vers leur destination ultime. Contrairement à la pile de protocoles TCP/IP définie dans la Recommandation T.123, le protocole MAP permet la concaténation d'unités MCSPDU multiples en une seule unité PDU **MAPData** (pour favoriser l'utilisation efficace du réseau). Le protocole MAP permet également la segmentation d'une unité MCSPDU unique en unités PDU **MAPData** multiples (quand la longueur de l'unité MCSPDU excède la taille maximale d'information utile).

Chaque élément de données d'une unité PDU **MAPData** se nomme *trame de données*. Une même unité PDU **MAPData** peut contenir une ou plusieurs trames de données. La taille maximale d'une trame de données est fixée pendant le processus d'arbitrage du protocole de transport. Il s'agit de la taille maximale qui pourra être contenue dans l'information utile maximale à un niveau de fiabilité donné (en tenant compte des bits de service). La taille maximale des trames de données peut différer d'un niveau de fiabilité à un autre.

Les unités MCSPDU sont divisées en trames de données au nœud faisant fonction de point d'entrée initial du flux de données. Le cas échéant, elles sont réassemblées à tous les nœuds de réception.

Les unités **MAPData** prennent en charge le concept de compression d'en-tête. Il y a plusieurs champs qui sont inclus à la fois dans l'en-tête de l'unité **MAPData** et dans l'en-tête de l'unité PDU de

données MCS. Si ces champs peuvent être extraits de l'en-tête MCS de la première trame de données de l'unité MAPPDU, ils peuvent être omis dans l'en-tête **MAPData** (il s'agit de champs facultatifs). Si la première trame de données de l'unité MAPPDU est une unité PDU de données MCS segmentée, l'émetteur décidera si suffisamment de données d'en-tête MCS sont présentes pour permettre l'extraction de ces champs. Le récepteur doit être prêt à extraire ces champs s'ils ne sont pas explicitement inclus dans l'en-tête **MAPData**.

Le contenu de l'unité **MAPData** est résumé dans le tableau ci-dessous.

Descripteur de données (facultatif)
Série de trames de données

*Descripteur de données*: ce champ facultatif est uniquement présent lors de l'envoi d'unités PDU de données MCS. Quand le descripteur est présent, les champs qu'il contient décrivent les données contenues dans l'unité MAPPDU. S'il est absent, cela signifie que l'unité MAPPDU contient des unités PDU de commande MCS, qui doivent être acheminées directement au MCS au nœud de réception.

*Série de trames de données*: il s'agit d'une séquence d'entrées de trames de données (telle que définie ci-dessous).

#### A.9.7.1 Descripteur de données

Un descripteur de données n'est présent dans une unité PDU **MAPData** que si elle achemine des unités PDU de données MCS. Le tableau ci-dessous résume le contenu du descripteur de données.

Fanion d'acheminement amont monodiffusion
Numéro séquentiel de départ
Identificateur de flux de données (facultatif)
Identificateur de référence de domaine (facultatif)

*Fanion d'acheminement amont monodiffusion*: ce fanion est utilisé pour obliger le destinataire à unidiffuser l'unité MAPPDU en amont. Quand ce fanion est mis à un, le destinataire doit commencer à transmettre, en mode monodiffusion, toutes les données du flux de données (même vers les nœuds voisins qui ont précédemment invalidé la monodiffusion). La transmission monodiffusion obligatoire commence avec cette unité MAPPDU. Le fanion doit demeurer à un quand l'unité PDU est transmise en amont. Quand un émetteur décide de mettre ce fanion à un pour un flux de données particulier, le fanion doit être mis à un pour toutes les unités PDU **MAPData** subséquentes de ce flux de données (pour éviter les basculements en série). Si ce fanion est réinitialisé, l'unité MAPPDU doit être traitée normalement.

*Numéro séquentiel de départ*: ce champ contient le numéro séquentiel unique assigné à la première trame de données de l'unité MAPPDU. Le numéro des trames de données subséquentes de l'unité MAPPDU est simplement incrémenté à partir de ce numéro séquentiel de départ. Les numéros séquentiels sont assignés indépendamment à chaque trame de données par l'expéditeur du flux de données.

*Identificateur de flux de données*: ce champ facultatif doit être présent quand l'information ne peut pas être extraite de la première trame de données. Les champs qui composent l'identificateur de flux de données sont les champs nécessaires pour bien acheminer les données et pour signaler à quel flux

de données unique les trames de données sont associées (et par conséquent établir l'identité du compteur séquentiel).

*Identificateur de référence de domaine*: ce champ facultatif est présent seulement dans le cas de données non fiables. Il s'agit de l'identificateur de référence de domaine qui a été transmis par l'émetteur lors du processus d'arbitrage de protocole. Ce champ peut être présent dans le cas de transmissions multidiffusion (pour des raisons de performance), mais il devrait être ignoré. Cette valeur, conjointement avec l'adresse réseau de l'expéditeur, suffit à déterminer de façon univoque à quel domaine les données sont associées.

#### A.9.7.2 Identificateur de flux de données

Un identificateur de flux de données sert à indiquer à quel flux de données est associé un ensemble de trames de données. Cette information doit être incluse explicitement quand elle ne peut pas être extraite de la première trame de données. Le tableau ci-dessous résume le contenu de l'identificateur de flux de données.

Identificateur de l'émetteur (facultatif)
Identificateur de métacanal (facultatif)
Identificateur de canal
Niveau de fiabilité
Priorité
Type de données (facultatif)

*Identificateur de l'émetteur*: ce champ facultatif n'est présent que dans le cas de données non uniformes, ou dans le cas de données uniformes ou de données transmises par mandataire unidiffusées en amont. Ce champ peut également être omis si l'information peut être extraite de l'en-tête MCS de la première trame de données. Ce champ précise l'identificateur d'utilisateur MCS de l'application utilisateur qui a expédié les données associées au flux de données.

*Identificateur de métacanal*: ce champ facultatif peut être omis si l'information peut être extraite de l'en-tête MCS de la première trame de données. Il indique quel métacanal est référencé et comprend trois sous-champs: identificateur de canal, niveau de fiabilité et priorité.

*Type de données*: ce champ facultatif peut être omis si l'information peut être extraite de l'en-tête MCS de la première trame de données. Il indique le type de données, soit: données non uniformes, données uniformes ou données transmises par mandataire.

#### A.9.7.3 Entrée de trame de données

Les unités PDU **MAPData** contiennent une série de trames de données. Le tableau ci-dessous résume le contenu de l'entrée de trame de données.

Fanion de premier segment
Fanion de dernier segment
Données d'utilisateur

*Fanion de premier segment*: ce fanion indique si la trame de données est ou n'est pas le premier segment d'une unité MCSPDU. Si le fanion est mis à 1, il s'agit de la première trame de données de l'unité MCSPDU. S'il est réinitialisé, il ne s'agit pas de la première trame de données d'une unité MCSPDU.

*Fanion de dernier segment*: ce fanion indique si la trame de données est ou n'est pas le dernier segment d'une unité MCSPDU. Si le fanion est mis à 1, il s'agit de la dernière trame de données d'une unité MCSPDU. S'il est réinitialisé, il ne s'agit pas de la dernière trame de données d'une unité MCSPDU.

*Données d'utilisateur*: il s'agit des données utilisateur de l'unité MCSPDU.

### A.9.8 MAPAddGroupRequest

Cette unité MAPPDU est utilisée pour propager en aval les assignations de groupe de multidiffusion dans un domaine. Cela se fait dès que le protocole MAP détecte un flux de données pour lequel il n'y a pas d'assignation de groupe. Ce processus est lancé par chaque nœud du domaine qui n'a pas de connexion ascendante MAP, ou qui a une connexion ascendante MAP, mais qui n'a pas de protocole multidiffusion en commun avec le nœud amont. Ce processus est maintenu par chaque nœud qui reçoit une demande **MAPAddGroupRequest** du nœud amont.

Chaque nœud devrait examiner les listes des protocoles multidiffusion qui sont pris en charge par les nœuds aval (ces listes sont établies pendant le processus d'arbitrage de protocole de transport). En fonction de cette information, chaque nœud doit indiquer aux nœuds aval quel protocole multidiffusion et quelle adresse de groupe ils doivent utiliser. Il devrait choisir le ou les protocoles qui mènent à la création d'îlots multidiffusion les moins nombreux et les plus grands possible.

Lors de la sélection d'un protocole pour chaque nœud aval, il faut tenir compte de plusieurs facteurs:

- 1) *quelles versions de protocole sont utilisées par les nœuds aval?* Lors de l'établissement de la connexion, le service MCS et le protocole MAP arbitrent les versions de protocole avec chaque nœud adjacent. Lors de la sélection de protocoles multidiffusion pour les nœuds aval, il ne faut pas oublier que les nœuds utilisant différentes versions d'un protocole *ne peuvent pas* se trouver dans le même îlot multidiffusion (même s'ils prennent en charge le même protocole multidiffusion), étant donné que les formats des unités PDU sont différents;
- 2) *quels sont les protocoles pris en charge par les nœuds aval?* Il est évident qu'un nœud ne peut choisir que parmi les protocoles que les nœuds aval ont convenu de prendre en charge. Il doit chercher à choisir les protocoles les plus universellement pris en charge;
- 3) *quelles sont les tailles maximales d'information utile qui ont été arbitrées avec les nœuds aval?* Pendant le processus d'arbitrage de protocole, chaque nœud doit tenter de sélectionner les mêmes tailles maximales d'information utile que celles qui ont déjà été arbitrées avec d'autres connexions du même domaine. Si des nœuds différents insistent pour utiliser des tailles maximales d'information utile différentes, ils ne peuvent pas se trouver dans le même îlot multidiffusion (tous les nœuds d'un îlot doivent convenir d'adopter une même taille maximale d'information utile, pour la monodiffusion et la multidiffusion);
- 4) *quel protocole a été assigné en amont?* Si un nœud effectue ce choix en réponse à une demande **MAPAddGroupRequest** amont, il doit tenter d'utiliser le même protocole que celui qui lui a été assigné (si possible). Cela se fonde sur le fait que le nœud amont a une meilleure vue d'ensemble, et peut donc prendre de meilleures décisions;
- 5) *combien y a-t-il de nœuds dans chaque sous-arbre qui prennent en charge chaque protocole?* Pendant l'arbitrage de protocole, les nœuds aval incluent un compte de nœuds avec chaque protocole multidiffusion qu'ils conviennent d'utiliser. Ce nombre indique au nœud amont combien il y a de nœuds dans chacun de ses sous-arbres qui prennent en charge ce protocole. En additionnant ces nombres pour toutes les connexions descendantes, un nœud peut choisir le protocole multidiffusion le plus largement pris en charge. Cela donne une certaine vue au-delà d'un niveau de la hiérarchie;

- 6) *quels protocoles sont préférés?* Pendant l'arbitrage de protocole, les nœuds aval peuvent également communiquer leurs préférences en matière de protocoles. Cette information pourrait servir à la sélection ultime des protocoles si les autres facteurs ne permettent pas de dégager clairement un meilleur choix.

Une fois qu'un protocole multidiffusion a été choisi pour chacun des nœuds aval, cette information leur est communiquée au moyen d'une demande **MAPAddGroupRequest** transmise à chacun de ces nœuds. Chaque nœud destinataire ajoutera alors ce groupe à sa liste de groupes actifs. Il propagera également cette assignation de groupe en aval en répétant les étapes décrites ci-dessus pour chacune de ses connexions descendantes.

Etant donné que la demande **MAPAddGroupRequest** précise la gamme de priorités qui doivent utiliser le groupe de multidiffusion, différentes priorités peuvent utiliser différents groupes (même pour le même canal). Lors de la sélection du groupe de multidiffusion auquel envoyer des données, le protocole MAP utilise l'identificateur de canal, le niveau de fiabilité *et* la priorité. Si une priorité donnée n'est pas configurée pour la multidiffusion, les données de cette priorité sont unidiffusées.

Il faut noter que ce n'est pas tous les nœuds d'un îlot multidiffusion qui devront adhérer à un groupe de multidiffusion immédiatement. Un nœud adhérera au groupe en présence de l'une ou l'autre des conditions suivantes: il y a des rattachements d'utilisateurs locaux qui doivent recevoir les données; il y a des connexions avec d'autres piles de transport qui doivent recevoir les données; ou le nœud doit réacheminer les données par monodiffusion vers un ou plusieurs nœuds voisins. Certains protocoles de transport peuvent également exiger qu'un nœud adhère à un groupe avant de lui transmettre des données (même si ce nœud n'est pas intéressé à recevoir des données du groupe).

Le tableau ci-dessous résume le contenu de la demande **MAPAddGroupRequest**.

Métacanal
Identificateur de protocole de transport
Adresse de groupe de multidiffusion

*Métacanal*: ce champ indique quel métacanal est mappé avec un groupe de multidiffusion.

*Identificateur de protocole de transport*: ce champ indique de façon univoque le protocole de transport à utiliser pour le métacanal.

*Adresse de groupe de multidiffusion*: il s'agit de l'adresse de groupe de multidiffusion sur laquelle les données pour le métacanal seront transmises. Ce champ est formaté comme une adresse NSAP.

#### **A.9.8.1 Métacanal**

Le tableau ci-dessous définit les champs qui composent un métacanal. Ce type sert à désigner l'abstraction à laquelle le protocole MAP assigne des groupes de multidiffusion.

Identificateur de canal
Niveau de fiabilité
Gamme de priorités

*Identificateur de canal*: ce champ indique quel canal MCS est associé au métacanal.

*Niveau de fiabilité*: ce champ indique quel niveau de fiabilité est associé au métacanal.

*Gamme de priorités*: il s'agit de la gamme de priorités qui est associée au métacanal (zéro est la priorité la plus élevée).

### A.9.9 MAPRemoveGroupRequest

Cette unité MAPPDU est utilisée pour éliminer des groupes de multidiffusion dans les nœuds aval de l'îlot multidiffusion. Ce processus est lancé par le fournisseur de groupes de multidiffusion quand il devient nécessaire de mettre hors service un groupe de multidiffusion. Quand un nœud reçoit une demande **MAPRemoveGroupRequest**, il doit la retransmettre à tous les nœuds en aval de l'îlot multidiffusion.

Généralement, une demande **MAPRemoveGroupRequest** ne sera pas transmise en aval tant qu'il n'aura pas déjà été mis fin à l'utilisation du groupe de multidiffusion. On peut s'assurer qu'un groupe de multidiffusion n'est pas activement utilisé en émettant en aval une demande **MAPDisableMulticastRequest** et en attendant la réception d'une confirmation **MAPDisableMulticastConfirm** (de tous les nœuds aval de l'îlot multidiffusion). Après la réception d'une telle confirmation, le fournisseur de groupes de multidiffusion est certain que personne n'utilise le groupe, et que celui-ci peut être retiré sans risque de perte de données.

Quand un nœud reçoit une demande **MAPRemoveGroupRequest**, il devrait éliminer immédiatement le groupe et réacheminer la demande en aval à tous les nœuds aval de l'îlot.

Le tableau ci-dessous résume le contenu de la demande **MAPRemoveGroupRequest**.

Métacanal
-----------

*Métacanal*: ce champ indique de quel métacanal le groupe est retiré.

### A.9.10 MAPDisableUnicastRequest

Cette unité MAPPDU est utilisée pour invalider la monodiffusion de données d'un nœud voisin pour un flux de données précis.

Un nœud ne devrait pas transmettre cette unité MAPPDU tant qu'il ne s'est pas assuré qu'il peut passer à la réception multidiffusion sans perte de données. A cette fin, il doit avoir reçu au moins une trame de données monodiffusion, pour connaître le numéro de la prochaine séquence attendue. En outre, il doit recevoir une trame de données multidiffusion dont le numéro séquentiel est égal ou inférieur au numéro séquentiel attendu (pour contrôler la synchronisation). Ce n'est qu'alors que le nœud sait qu'il peut couper le flux de données monodiffusion (pour ce flux de données).

Le tableau ci-dessous résume le contenu de la demande **MAPDisableUnicastRequest**.

Identificateur de flux de données
-----------------------------------

*Identificateur de flux de données*: ce champ de données contient l'information permettant de déterminer pour quel flux de données la monodiffusion est invalidée. L'identificateur de flux de données est défini de façon détaillée dans le sous-paragraphe traitant des unités PDU **MAPData**. L'identificateur de l'émetteur sera omis pour les flux descendants de données uniformes et de données transmises par mandataire, mais il sera présent dans les autres cas. Les champs Identificateur de métacanal et Type de données seront toujours présents.

### A.9.11 MAPEnableUnicastRequest

Cette unité MAPPDU sert à revalider la monodiffusion d'un nœud voisin pour un champ de données précis.

Cette unité MAPPDU serait transmise quand un nœud détermine qu'il ne reçoit plus de données multidiffusion, ou que le taux d'erreurs de la transmission multidiffusion est tellement élevé que son

utilisation n'est plus efficace. Il doit y avoir une forme de communication locale entre le protocole MAP et le protocole de transport pour décider de la nécessité de rétablir la monodiffusion. Cela pourrait également se produire suite à la réception d'unités PDU **MAPSequenceNumber** indiquant un arrêt du flux de données multidiffusion.

Le tableau ci-dessous résume le contenu de la demande **MAPEnableUnicastRequest**.

Identificateur de flux de données
-----------------------------------

*Identificateur de flux de données*: ce champ contient l'information permettant de déterminer pour quel flux de données la monodiffusion est validée. L'identificateur de flux de données est défini de façon détaillée dans le sous-paragraphe traitant des unités PDU **MAPData**. L'identificateur de l'émetteur sera omis pour les flux descendants de données uniformes et de données transmises par mandataire, mais il sera présent dans les autres cas. Les champs Identificateur de métacanal et Type de données seront toujours présents.

### A.9.12 **MAPEnableUnicastConfirm**

Cette unité MAPPDU confirme le rétablissement de la monodiffusion pour le flux de données spécifié. Quand un nœud demande à un nœud voisin de rétablir la monodiffusion, il peut devoir recevoir plus de données multidiffusion. Cela se produirait s'il était déjà trop tard pour que le nœud voisin transmette certaines trames de données en monodiffusion parce qu'elles auraient déjà été transférées. Cette unité MAPPDU permet au nœud voisin d'indiquer à l'expéditeur de la demande **MAPEnableUnicastRequest** quel sera le numéro séquentiel de la prochaine trame de données monodiffusion. Ainsi, le nœud peut continuer d'utiliser le mode multidiffusion (si possible) pour recevoir toutes les trames de données précédant celle portant le numéro séquentiel indiqué. Cela est nécessaire pour éviter la perte de données pendant la transition de la multidiffusion à la monodiffusion.

Par exemple, un nœud a reçu les numéros séquentiels jusqu'au numéro 20 pour un flux de données particulier (mais le taux d'erreurs est très élevé). Il peut alors décider de passer à la monodiffusion pour réduire le taux d'erreurs. Cependant, le nœud voisin a déjà traité la trame portant le numéro séquentiel 23 de ce flux de données. Si le nœud passe immédiatement à la monodiffusion, il perdra les trames ayant les numéros séquentiels 21, 22 et 23. Cette confirmation indique à l'expéditeur que le prochain numéro séquentiel monodiffusion sera le numéro 24, de sorte que toutes les séquences jusqu'à la séquence 23 doivent être reçues en multidiffusion, après quoi le nœud pourra passer à la monodiffusion.

Le tableau ci-dessous résume le contenu de la confirmation **MAPEnableUnicastConfirm**.

Identificateur de flux de données
Numéro séquentiel

*Identificateur de flux de données*: ce champ contient l'information permettant de déterminer pour quel flux de données la monodiffusion est validée. L'identificateur de flux de données est défini de façon détaillée dans le sous-paragraphe traitant des unités PDU **MAPData**. L'identificateur de l'émetteur sera omis pour les flux descendants de données uniformes et de données transmises par mandataire, mais il sera présent dans les autres cas. Les champs Identificateur de métacanal et Type de données seront toujours présents.

*Numéro séquentiel*: il s'agit du prochain numéro séquentiel du flux de données qui sera acheminé vers le nœud en mode monodiffusion. Toutes les trames ayant un numéro séquentiel inférieur doivent être reçues en mode multidiffusion, avant que le nœud ne puisse passer à un flux de données monodiffusion.

### A.9.13 **MAPDisableMulticastRequest**

Cette unité MAPPDU est acheminée en aval dans l'îlot multidiffusion pour invalider la multidiffusion avant le retrait d'un groupe.

Quand un nœud reçoit une demande **MAPDisableMulticastRequest**, il doit lancer le processus de revalidation de la monodiffusion pour tout flux de données pour lequel la monodiffusion avait été invalidée (pour ce groupe). A cette fin, il transmet une demande **MAPEnableUnicastRequest** à l'émetteur de chacun des flux de données visés. Le nœud devrait également marquer le groupe pour indiquer qu'il est à l'état invalidé, ce qui empêche le nœud d'invalider la monodiffusion pour tous les flux de données du groupe. Il ne doit cependant PAS cesser de transmettre au groupe.

Le récepteur devrait retransmettre la demande **MAPDisableMulticastRequest** à tous les nœuds aval. Un nœud ne peut pas émettre une confirmation **MAPDisableMulticastConfirm** tant qu'il n'a pas reçu une confirmation similaire de tous les nœuds aval ET qu'il n'a pas rétabli avec succès la monodiffusion de tous les flux de données pour lesquels elle avait été invalidée (pour ce groupe). Il ne peut pas supposer que la monodiffusion a été rétablie pour un flux de données tant qu'il n'a pas reçu une confirmation **MAPEnableUnicastConfirm** pour chaque demande en cours (et il doit avoir reçu toutes les données multidiffusées ayant un numéro séquentiel inférieur à celui indiqué dans la confirmation).

Le tableau ci-dessous résume le contenu de la demande **MAPDisableMulticastRequest**.

Métacanal
-----------

*Métacanal*: il s'agit du métacanal associé au groupe pour lequel la multidiffusion est non validée. Le type métacanal est décrit de façon détaillée dans la section traitant de la demande **MAPRemoveGroupRequest**.

### A.9.14 **MAPDisableMulticastConfirm**

Cette unité MAPPDU est utilisée pour confirmer que tous les nœuds en aval du récepteur ont cessé d'utiliser un groupe de multidiffusion.

Quand un nœud reçoit une demande **MAPDisableMulticastRequest** du nœud amont, il devrait émettre une confirmation **MAPDisableMulticastConfirm** quand les deux conditions ci-dessous sont satisfaites: il doit avoir reçu des unités PDU **MAPDisableMulticastConfirm** de tous les nœuds aval dans l'îlot multidiffusion (le cas échéant), et il doit être certain que la monodiffusion est validée pour tous les flux de données du groupe.

La première condition est satisfaite suite à l'acheminement de la demande **MAPDisableMulticastRequest** aux nœuds aval. La seconde condition est satisfaite suite à la revalidation de la monodiffusion de tous les flux de données pour lesquels elle avait été invalidée (cela se fait au moyen de l'unité PDU **MAPEnableUnicastRequest**).

Le tableau ci-dessous résume le contenu de la confirmation **MAPDisableMulticastConfirm**.

Métacanal
-----------

*Métacanal*: il s'agit du métacanal associé au groupe pour lequel la multidiffusion a été invalidée. Le type métacanal est décrit de façon détaillée dans le sous-paragraphe traitant de la demande **MAPRemoveGroupRequest**.

#### **A.9.15 MAPEnableMulticastRequest**

Cette unité MAPPDU est utilisée pour revalider un groupe de multidiffusion qui a été précédemment invalidé.

Quand un nœud reçoit une demande **MAPEnableMulticastRequest** à l'égard d'un groupe marqué invalidé (mais toujours valide), il revalide l'utilisation de ce groupe. Cela signifie que les unités PDU **MAPDisableUnicastRequest** peuvent de nouveau être traitées pour couper le flux de données monodiffusion. Un groupe revalidé reprend l'état qu'il avait à sa création.

Cette demande devrait être retransmise à tous les nœuds en aval du récepteur.

Le tableau ci-dessous résume le contenu de la demande **MAPEnableMulticastRequest**.

Métacanal
-----------

*Métacanal*: il s'agit du métacanal associé au groupe pour lequel la multidiffusion est revalidée. Le type métacanal est décrit de façon détaillée dans le sous-paragraphe traitant de la demande **MAPRemoveGroupRequest**.

#### **A.9.16 MAPSequenceNumber**

Cette unité MAPPDU est utilisée pour indiquer aux récepteurs aval quels sont les numéros séquentiels qui auraient dû avoir été reçus jusqu'au moment présent. Elle est propagée par monodiffusion à partir de l'expéditeur du flux de données, et elle est retransmise par tous les nœuds de réception aux autres nœuds qui sont davantage éloignés de l'émetteur (dans le sens de la connexion ascendante et de toute connexion descendante qui mène à des utilisateurs raccordés au canal précisé). Ainsi, tous les nœuds sont informés des numéros séquentiels du flux de données qui auraient dû être reçus.

Cette unité MAPPDU pourrait servir à déterminer quand un nœud cesse simplement de recevoir des données dans un groupe de multidiffusion. Cela pourrait être utile dans le cas de protocoles non fiables où il n'y a pas de tentatives de correction des erreurs, ainsi que dans le cas des protocoles de transport qui ne détectent pas clairement la perte d'un trajet multidiffusion entre un émetteur et un récepteur. L'utilisation exacte de l'unité MAPPDU doit faire l'objet d'études supplémentaires.

Le tableau ci-dessous résume le contenu de l'unité **MAPSequenceNumber**.

Identificateur de flux de données
-----------------------------------

Numéro séquentiel
-------------------

*Identificateur de flux de données*: ce champ contient l'information permettant de déterminer à quel flux de données le numéro séquentiel est associé. L'identificateur de flux de données est défini de façon détaillée dans la section traitant des unités PDU **MAPData**. L'identificateur de l'émetteur sera omis pour les flux descendants de données uniformes et de données transmises par mandataire, mais il sera présent dans les autres cas. Les champs Identificateur de métacanal et Type de données seront toujours présents.

*Numéro séquentiel*: il s'agit du dernier numéro séquentiel transmis pour le flux de données précisé.

## A.10 Définition de la notation ASN.1 des unités MAPPDU

Le présent sous-paragraphe présente le format réel des unités MAPPDU, fondé sur la notation ASN.1. Son contenu sera également disponible sous forme de fichier texte compilable distinct.

```
--*****
--*
--*      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
    nonStandardParameters
    ...
}

MAPSequenceNumberPDU ::= SEQUENCE
{
    dataFlowIdentifier
    sequenceNumber
    nonStandardParameters
    ...
}

MAP PDU ::= CHOICE
{
    mapConnectRequest
    mapConnectConfirm
    mapDisconnectRequest
    mapDisconnectConfirm
    mapArbitrateProtocolsRequest
    mapArbitrateProtocolsConfirm
    mapData
    mapAddGroupRequest
    mapRemoveGroupRequest
    mapDisableUnicastRequest
    mapEnableUnicastRequest
    mapEnableUnicastConfirm
    mapDisableMulticastRequest
    mapDisableMulticastConfirm
    mapEnableMulticastRequest
    mapSequenceNumber
    nonStandardPDU
    ...
}

```

END

```

--*****
--*
--*      End of ASN.1 Definition for MAP PDUs
--*
--*****

```



## SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
<b>Série T</b>	<b>Terminaux des services télématiques</b>
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information
Série Z	Langages de programmation