



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.245

(07/97)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructure des services audiovisuels – Procédures de
communication

**Protocole de commande pour
communications multimédias**

Recommandation UIT-T H.245

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE H
SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Caractéristiques des canaux de transmission pour des usages autres que téléphoniques	H.10–H.19
Emploi de circuits de type téléphonique pour la télégraphie à fréquence vocale	H.20–H.29
Circuits et câbles téléphoniques utilisés pour les divers types de transmission télégraphique et de transmissions simultanées	H.30–H.39
Circuits de type téléphonique utilisés en bélinographie	H.40–H.49
Caractéristiques des signaux de données	H.50–H.99
CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.399

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

RECOMMANDATION UIT-T H.245

PROTOCOLE DE COMMANDE POUR COMMUNICATIONS MULTIMÉDIAS

Résumé

La présente Recommandation spécifie la syntaxe et la sémantique des messages d'information relatifs aux terminaux, de même que les procédures devant être utilisées pour les négociations dans la bande au début de la communication ou pendant la communication elle-même. Les messages s'appliquent aux capacités d'émission et de réception, de même qu'à la préférence de mode à partir du terminal récepteur, la signalisation de la voie logique, les commandes et indications d'état. Les procédures de signalisation acquittées sont spécifiées pour assurer la fiabilité de la communication audiovisuelle et de la transmission de données.

Source

La Recommandation UIT-T H.245, révisée par la Commission d'études 16 de l'UIT-T (1997-2000), a été approuvée le 10 juillet 1997 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 de la technologie 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

	Page	
1	Domaine d'application	1
2	Références normatives.....	1
3	Définitions	4
4	Abréviations	5
5	Généralités.....	7
5.1	Choix du mode maître-esclave	7
5.2	Echange de capacités.....	7
5.3	Procédures de signalisation de la voie logique.....	8
5.4	Réception de la demande de fermeture de la voie logique.....	9
5.5	Modification des entrées dans le tableau de multiplexage H.223	9
5.6	Demande de mode audiovisuel et de mode données	9
5.7	Détermination du temps de propagation aller et retour	9
5.8	Boucles de maintenance	9
5.9	Commandes et indications d'état	9
6	Messages: syntaxe	10
7	Messages: définitions sémantiques.....	47
7.1	Messages de choix du mode maître ou esclave	48
7.1.1	Choix du mode maître ou esclave	48
7.1.2	Acquittement du choix de mode maître ou esclave.....	48
7.1.3	Refus du choix du mode maître ou esclave.....	48
7.1.4	Libération sur temporisation du choix de mode maître ou esclave.....	48
7.2	Messages de capacités des terminaux.....	48
7.2.1	Aperçu général	49
7.2.2	Ensemble de capacités des terminaux	49
7.2.3	Acquittement de l'ensemble de capacités des terminaux	63
7.2.4	Refus de l'ensemble de capacités des terminaux.....	63
7.2.5	Libération sur temporisation de l'ensemble de capacités des terminaux.....	63
7.3	Messages de signalisation de la voie logique	63
7.3.1	Ouverture d'une voie logique	63
7.3.2	Acquittement de l'ouverture de la voie logique.....	67
7.3.3	Refus de l'ouverture de la voie logique	69
7.3.4	Confirmation de l'ouverture de la voie logique.....	70
7.3.5	Fermeture de la voie logique	70
7.3.6	Acquittement de fermeture de la voie logique	70
7.3.7	Demande de fermeture de la voie.....	70

7.3.8	Acquittement de demande de fermeture de la voie	71
7.3.9	Refus de demande de fermeture de la voie.....	71
7.3.10	Libération sur temporisation de demande de fermeture de la voie	71
7.4	Messages de signalisation des tableaux de multiplexage	71
7.4.1	Emission d'entrée de multiplexage	71
7.4.2	Acquittement d'envoi d'entrée de multiplexage	72
7.4.3	Refus d'émission d'entrée de multiplex	72
7.4.4	Libération sur temporisation de l'entrée de multiplexage	72
7.5	Messages de demande de signalisation des tableaux de multiplexage.....	72
7.5.1	Demande d'entrée de multiplexage	72
7.5.2	Acquittement de la demande d'entrée de multiplexage	73
7.5.3	Refus de demande d'entrée de multiplexage	73
7.5.4	Libération sur temporisation de la demande d'entrée de multiplexage	73
7.6	Messages de mode demande	73
7.6.1	Mode demande	73
7.6.2	Message d'acquittement du mode demande RequestModeAcknowledge.....	76
7.6.3	Message de refus du mode demande RequestModeReject	77
7.6.4	Message de demande de libération de mode RequestModeRelease	77
7.7	Messages liés au temps de propagation aller et retour	77
7.7.1	Demande de temps de propagation aller et retour.....	77
7.7.2	Réponse sur le temps de propagation aller et retour	77
7.8	Messages relatifs à la boucle de maintenance	78
7.8.1	Demande de boucle de maintenance	78
7.8.2	Acquittement de la demande de boucle de maintenance.....	78
7.8.3	Refus de la demande de boucle de maintenance	78
7.8.4	Commande "arrêt" de la boucle de maintenance.....	78
7.9	Messages en mode communication	78
7.9.1	Commande du mode de communication.....	78
7.9.2	Demande de mode de communication	78
7.9.3	Réponse sur le mode de communication.....	78
7.10	Demande de conférence et messages de réponse	79
7.10.1	Demande de la liste des terminaux.....	79
7.10.2	Réponse sur la liste des terminaux	79
7.10.3	Demande du rôle de président de séance	79
7.10.4	Annulation de la demande du rôle de président de séance.....	79
7.10.5	Réponse à la demande du rôle de président de séance	79
7.10.6	Rejet du terminal	79
7.10.7	Refus du rejet du terminal	79
7.10.8	Demande d'identification du terminal	79
7.10.9	Réponse d'identification du terminal par l'entité de commande MC	79

	Page
7.10.10 Demande d'entrée du mot de passe selon H.243	79
7.10.11 Réponse sur le mot de passe.....	79
7.10.12 Demande d'entrée d'identification de terminal selon H.243.....	79
7.10.13 Réponse sur l'identification de terminal	79
7.10.14 Demande d'entrée d'identification de conférence H.243	80
7.10.15 Réponse sur l'identification de conférence.....	80
7.10.16 Refus de commande vidéo	80
7.10.17 Demande d'entrée de l'adresse d'extension.....	80
7.10.18 Réponse sur l'adresse d'extension.....	80
7.11 Commandes	80
7.11.1 Envoi de l'ensemble des capacités du terminal	80
7.11.2 Chiffrement	80
7.11.3 Contrôle de flux.....	81
7.11.4 Fin de session	81
7.11.5 Commandes diverses.....	82
7.11.6 Commande de conférence	83
7.12 Indications	83
7.12.1 Fonction non comprise	83
7.12.2 Indications d'état diverses.....	84
7.12.3 Indication de gigue	85
7.12.4 Indication de décalage temporel selon H.223	86
7.12.5 Nouvelle indication de voie virtuelle ATM	86
7.12.6 Messages d'entrée de l'utilisateur	87
7.12.7 Indications de conférence.....	87
7.12.8 Décalage temporel maximal sur la voie logique H2250	88
7.12.9 Indication d'emplacement MC	88
7.12.10 Indication de l'identification du fournisseur.....	88
7.12.11 Fonction non prise en compte	88
8 Procédures	89
8.1 Introduction	89
8.1.1 Méthode de spécification	89
8.1.2 Communication entre l'entité de protocole et l'utilisateur de protocole.....	89
8.1.3 Communication entre entités homologues	90
8.1.4 Diagrammes SDL	90
8.1.5 Symboles du langage SDL	90
8.2 Procédures de choix du mode maître ou esclave.....	92
8.2.1 Introduction	92
8.2.2 Communication entre l'entité MSDSE et l'utilisateur de l'entité MSDSE.....	93
8.2.3 Communication entre les entités MSDSE homologues	95
8.2.4 Procédures MSDSE.....	96

	Page
8.3	Procédures d'échange de capacités 103
8.3.1	Introduction 103
8.3.2	Communication entre l'entité CESE et l'utilisateur de l'entité CESE..... 104
8.3.3	Communication entre les entités CESE homologues 106
8.3.4	Procédures de l'entité CESE..... 107
8.4	Procédures de signalisation de la voie logique monodirectionnelle..... 111
8.4.1	Introduction 111
8.4.2	Communication entre l'entité LCSE et l'utilisateur de l'entité LCSE..... 113
8.4.3	Communications entre les entités LCSE homologues 116
8.4.4	Procédures LCSE 117
8.5	Procédures de signalisation de la voie logique bidirectionnelle..... 124
8.5.1	Introduction 124
8.5.2	Communication entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE.... 127
8.5.3	Communication entre les entités B-LCSE homologues 130
8.5.4	Procédures B-LCSE 131
8.6	Procédures de fermeture de la voie logique 140
8.6.1	Introduction 140
8.6.2	Communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE 140
8.6.3	Communication entre les entités CLCSE homologues 142
8.6.4	Procédures CLCSE..... 143
8.7	Procédures du tableau de multiplexage H.223 147
8.7.1	Introduction 147
8.7.2	Communication entre l'entité MTSE et l'utilisateur de l'entité MTSE 149
8.7.3	Communication entre les entités MTSE homologues 151
8.7.4	Procédures MTSE 152
8.8	Procédures relatives à la demande d'entrée de multiplexage..... 158
8.8.1	Introduction 158
8.8.2	Communication entre l'entité RMESE et l'utilisateur de l'entité RMESE..... 159
8.8.3	Communication entre les entités RMESE homologues 161
8.8.4	Procédures RMESE..... 162
8.9	Procédures de demande de mode 165
8.9.1	Introduction 165
8.9.2	Communication entre l'entité MRSE et l'utilisateur de l'entité MRSE 167
8.9.3	Communication entre les entités MRSE homologues..... 169
8.9.4	Procédures MRSE 170
8.10	Procédures liées au temps de propagation aller et retour 174
8.10.1	Introduction 174
8.10.2	Communication entre l'entité RTDSE et l'utilisateur de l'entité RTDSE 175
8.10.3	Communication entre les entités RTDSE homologues 176

	Page
8.10.4 Procédures RTDSE	177
8.11 Procédures relatives à la boucle de maintenance.....	180
8.11.1 Introduction	180
8.11.2 Communication entre l'entité MLSE et l'utilisateur de l'entité MLSE.....	181
8.11.3 Communication entre les entités MLSE homologues	184
8.11.4 Procédures MLSE	185
Annexe A – Affectations d'identificateur d'objet.....	190
Appendice I – Aperçu général de la syntaxe ASN.1	191
I.1 Introduction à la syntaxe ASN.1	191
I.2 Types de données de base de la syntaxe ASN.1.....	191
I.3 Types de structures de données	193
I.4 Type d'identificateur d'objet	194
Appendice II – Exemples de procédures H.245	195
II.1 Introduction	195
II.2 Entité de signalisation de choix du mode maître ou esclave	195
II.3 Entité de signalisation d'échange de capacités (CESE)	200
II.4 Entité de signalisation de la voie logique (LCSE).....	201
II.5 Entité de signalisation de fermeture de la voie logique (CLCSE).....	204
II.6 Entité de signalisation du tableau de multiplexage (MTSE)	205
II.7 Entité de signalisation de demande de mode (MRSE)	207
II.8 Entité de signalisation de temps de propagation aller et retour (RTDSE)	208
II.9 Entité de signalisation de voie logique bidirectionnelle (B-LCSE)	210
Appendice III – Récapitulatif des temporisateurs et des compteurs utilisés dans les procédures.....	212
III.1 Temporisateurs	212
III.2 Compteurs.....	213
Appendice IV – Extensions à la Recommandation H.245.....	213

Recommandation H.245

PROTOCOLE DE COMMANDE POUR COMMUNICATIONS MULTIMÉDIAS

(révisée en 1997)

1 Domaine d'application

La présente Recommandation spécifie la syntaxe et la sémantique des messages d'information des terminaux, de même que les procédures pour les utiliser dans les négociations dans la bande au début ou en cours de communication. Les messages contiennent les capacités d'émission et de réception, ainsi que la préférence de mode du côté réception, la signalisation de la voie logique et les commandes et indications d'état. Des procédures de signalisation avec acquittement sont spécifiées pour garantir le bon fonctionnement des communications audiovisuelles et des transmissions de données.

La présente Recommandation s'étend à un grand nombre d'applications, y compris le stockage/la consultation, les services de messagerie et de diffusion, de même que le mode conversationnel. Il s'applique mais n'est pas limité aux systèmes multimédias qui utilisent les multiplex définis dans les Recommandations H.222.0 et H.223 et H.225.0. Ces différents systèmes partageant la même syntaxe et la même sémantique, et sont donc binaires compatibles. Certaines des procédures s'appliquent à tous les systèmes, alors que les autres sont plus spécifiques à certains systèmes.

Les différents systèmes faisant appel à la présente Recommandation peuvent spécifier l'usage de protocoles de transport différents. Ils sont cependant destinés à être utilisés avec une couche Transport fiable, c'est-à-dire une couche qui garantisse un acheminement correct des données.

NOTE – Il convient cependant de ne pas confondre avec le système de gestion défini dans la Recommandation T.120, qui est transporté à l'intérieur du flux de données et qui a des fonctionnalités différentes de celles décrites ici – le flux de données décrit dans la Recommandation H.245 et le flux de données décrit dans la Recommandation T.120 sont complémentaires.

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.

- [1] Recommandation G.711 du CCITT (1988), *Modulation par impulsions et codage (MIC) des fréquences vocales.*
- [2] Recommandation G.722 du CCITT (1988), *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s.*
- [3] Recommandation UIT-T G.723.1 (1996), *Codeur vocal à double débit pour communications multimédias acheminées à 5,3 kbit/s et 6,3 kbit/s.*
- [4] Recommandation G.728 du CCITT (1992), *Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code.*

- [5] Recommandation UIT-T H.221 (1997), *Structure de trame pour un canal à débit de 64 à 1920 kbit/s pour les téléservices audiovisuels.*
- [6] Recommandation UIT-T H.222.0 (1995) | ISO/CEI 13818-1:1996, *Technologies de l'information – Codage générique des images animées et du son associé: Systèmes.*
- [7] Recommandation UIT-T H.222.1 (1996), *Multiplexage et synchronisation multimédias des communications audiovisuelles en environnements ATM.*
- [8] Recommandation UIT-T H.223 (1996), *Protocole de multiplexage pour communications multimédias à faible débit.*
- [9] Recommandation UIT-T H.224 (1994), *Protocole de commande en temps réel pour les applications simplex mettant en œuvre les canaux de données à faible vitesse/à grande vitesse/de protocole en multicouche définis dans la Recommandation H.221.*
- [10] Recommandation UIT-T H.230 (1997), *Signaux de commande et indicateurs synchronisés avec la trame pour les systèmes audiovisuels.*
- [11] Recommandation UIT-T H.233 (1995), *Système de confidentialité pour les services audiovisuels.*
- [12] Recommandation UIT-T H.234 (1994), *Système de gestion de clés de chiffrement et d'authentification pour les services audiovisuels.*
- [13] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s.*
- [14] Recommandation UIT-T H.262/Amd.1 (1996) | ISO/CEI 13818-2/Amd.1:1997 *Enregistrement des identificateurs de droit d'auteur.*
- [15] Recommandation UIT-T H.263 (1996), *Codage vidéo pour communications à faible débit.*
- [16] Recommandation UIT-T H.281 (1994), *Protocole de télécommande de caméra pour les visioconférences utilisant la couche H.224.*
- [17] Recommandation UIT-T H.320 (1997), *Systèmes et équipements terminaux visiophoniques à bande étroite.*
- [18] Recommandation UIT-T H.324 (1996), *Terminal pour communications multimédia à faible débit.*
- [19] Recommandation UIT-T I.363/Add.1 (1993), *Spécification de la couche d'adaptation du mode de transfert asynchrone RNIS à large bande.*
- [20] Recommandation UIT-T Q.2931 (1995), *Système de signalisation d'abonné numérique n° 2 – Spécification de la couche 3 de l'interface utilisateur-réseau pour la commande de connexion/appel de base.*
- [21] Recommandation UIT-T T.30 (1996), *Procédures pour la transmission de documents par télécopie sur le réseau téléphonique général commuté.*
- [22] Recommandation T.35 du CCITT (1991), *Procédure d'attribution des codes définis par le CCITT dans le cas de moyens non normalisés.*
- [23] Recommandation T.51 du CCITT (1992), *Jeux de caractères latins codés pour services de télématique.*
- [24] Recommandation UIT-T T.84 (1996) | ISO/CEI 10918-3 (1996): *Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: extensions.*

- [25] Recommandation UIT-T T.120 (1996), *Protocoles de données pour conférence multimédia.*
- [26] Recommandation UIT-T T.434 (1996), *Format de transfert de fichiers binaires pour les services de télématique.*
- [27] Recommandation UIT-T V.14 (1993), *Transmission de caractères arithmiques sur des voies supports synchrones.*
- [28] Recommandation UIT-T V.34 (1996), *Modem fonctionnant à des débits allant jusqu'à 33 600 bit/s pour usage sur le réseau téléphonique général commuté et sur les circuits à 2 fils de type téléphonique loués point à point.*
- [29] Recommandation UIT-T V.42 (1996), *Procédures de correction d'erreur pour les équipements de terminaison de circuits de données utilisant la conversion asynchrone/synchrone.*
- [30] Recommandation UIT-T X.680 (1994), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- [31] Recommandation UIT-T X.691 (1995), *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*
- [32] ISO/CEI 3309:1993, *Technologies de l'information – Télécommunications et échange d'informations entre les systèmes – Procédures de commande de liaisons de données à haut niveau (HDLC) – Structure de trame.*
- [33] ISO/CEI 11172-2:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 2: Vidéo.*
- [34] ISO/CEI 11172-3:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 3: Audio.*
- [35] ISO/CEI 13818-3:1995, *Technologies de l'information – Codage générique des images animées et de l'audio associé – Partie 3: Audio.*
- [36] ISO/CEI 13818-6¹, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 6: Extensions pour DSM-CC.*
- [37] ISO/CEI TR 9577:1996, *Technologies de l'information – Identification des protocoles dans la couche réseau.*
- [38] Recommandation UIT-T H.225.0 (1996), *Mise en paquets et synchronisation d'un train multimédia sur réseau local de qualité de service non garantie.*
- [39] Recommandation UIT-T H.323 (1996), *Systèmes et équipements visiophoniques pour réseau local offrant une qualité de service non garantie.*
- [40] Recommandation UIT-T H.243 (1997), *Procédures pour l'établissement de communications entre trois terminaux audiovisuels ou plus sur des canaux numériques d'un débit allant jusqu'à 1920 Mbit/s.*
- [41] Recommandation UIT-T H.230 (1997), *Signaux de commande et d'indication synchrones de la trame pour les systèmes audiovisuels.*
- [42] Recommandation UIT-T T.123 (1996), *Piles protocolaires de données propres au réseau pour conférences multimédias.*

¹ Actuellement à l'état de projet.

[43] Recommandation UIT-T E.164 (1997), *Plan de numérotage des télécommunications publiques internationales*.

3 Définitions

La présente Recommandation définit les termes suivants:

- 3.1 **voie logique bidirectionnelle**: couple de voies logiques associées entre deux terminaux, une dans chaque sens de transmission.
- 3.2 **capacité**: un terminal dispose d'une capacité donnée s'il peut coder et transmettre ou recevoir et décoder un signal particulier.
- 3.3 **voie**: liaison monodirectionnelle entre deux points terminaux.
- 3.4 **commande**: message nécessitant une action mais pas de réponse explicite.
- 3.5 **flux élémentaire**: terme générique pour un train binaire codé vidéo, audio ou autre.
- 3.6 **entrée**: référence à des éléments d'ensembles ou de tableaux, tels que des ensembles de capacités et des tableaux de multiplexage.
- 3.7 **sens direct**: sens de transmission allant du terminal émettant la demande vers l'autre terminal pour une voie logique bidirectionnelle.
- 3.8 **dans la bande**: les messages dans la bande sont ceux qui sont transportés dans la voie ou la voie logique à laquelle ils se rapportent.
- 3.9 **entrant**: une entité de signalisation entrante ne peut pas lancer une procédure, mais elle peut répondre à des messages émanant de l'entité de signalisation distante et aux primitives de son utilisateur local.
- 3.10 **indication**: message contenant des informations mais n'appelant pas d'action ou de réponse.
- 3.11 **voie logique**: trajet monodirectionnel ou bidirectionnel pour la transmission d'informations.
- 3.12 **numéro de voie logique**: numéro identifiant une voie logique unique.
- 3.13 **signalisation de voie logique**: ensemble des procédures utilisées pour ouvrir et fermer des voies logiques.
- 3.14 **terminal maître**: terminal désigné comme étant le terminal maître par la procédure de détermination maître-esclave définie dans cette Recommandation, ou par toute autre procédure.
- 3.15 **type de média**: forme simple d'information présentée à un utilisateur, ou données représentant cette information: la vidéo, l'audio et le texte sont des exemples de types de médias.
- 3.16 **mode**: ensemble de flux élémentaires qu'un terminal transmet, aimerait transmettre ou voudrait recevoir.
- 3.17 **communication multimédia**: transmission et/ou la réception de signaux correspondant à au moins deux types de médias simultanément.
- 3.18 **hors norme**: non conforme à une norme nationale ou internationale à laquelle il est fait référence dans la présente Recommandation.
- 3.19 **sortant**: l'entité de signalisation sortante est celle qui initie la procédure.
- 3.20 **multipoint**: interconnexion simultanée d'au moins trois terminaux pour permettre les communications entre plusieurs sites par l'utilisation d'équipements de commande multipoint (ponts de conférence) qui orientent de manière centralisée les flux d'information.

- 3.21 demande:** message qui entraîne une action du terminal distant et appelle une réponse immédiate de ce terminal.
- 3.22 réponse:** message apportant la réponse à une demande.
- 3.23 sens inverse:** sens de transmission allant du terminal recevant une demande vers le terminal ayant émis la demande pour une voie logique bidirectionnelle.
- 3.24 session:** séance de communication entre deux terminaux, pouvant être de type conversationnel ou non conversationnel (par exemple consultation d'une base de données).
- 3.25 terminal asservi:** terminal désigné comme étant le terminal asservi par la procédure de détermination maître-esclave définie dans la présente Recommandation, ou par toute autre procédure.
- 3.26 support:** capacité à fonctionner dans un mode donné; cependant, une prescription de prise en charge d'un mode donné ne signifie pas que ce mode doit être effectivement utilisé tout le temps; à moins d'être interdits, d'autres modes peuvent être utilisés par accord mutuel.
- 3.27 terminal:** point terminal quelconque tel qu'un terminal d'utilisateur ou tout autre système de communication comme un pont de conférence ou un serveur d'informations.
- 3.28 identificateur TSAP:** élément d'information utilisé pour multiplexer plusieurs liaisons de transport du même type sur une entité H.323 simple, toutes les liaisons de transport partageant la même adresse de réseau local (par exemple le numéro d'accès dans un environnement TCP/UDP/IP). Des identificateurs TSAP peuvent être préattribués par une instance internationale ou attribués dynamiquement pendant l'établissement d'un appel. Les identificateurs TSAP attribués dynamiquement sont de nature transitoire, c'est-à-dire que leurs valeurs ne sont valides que pendant la durée d'une seule communication.
- 3.29 voie logique monodirectionnelle:** trajet pour la transmission d'un flux élémentaire unique entre deux terminaux.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

AAL	couche d'Adaptation ATM (<i>ATM adaptation layer</i>)
AL1, 2, 3	couches d'Adaptation 1, 2 et 3 de la Recommandation H.223 (<i>H.223 adaptation layers 1,2 and 3</i>)
ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
ATM	mode de transfert asynchrone (<i>asynchronous transfer mode</i>)
B-LCSE	entité de signalisation de voie logique bidirectionnelle (<i>bi-directional logical channel signalling entity</i>)
CESE	entité de signalisation d'échange de capacités (<i>capability exchange signalling entity</i>)
CIF	format intermédiaire commun (d'une image vidéo: voir les Recommandations H.261 et H.263) (<i>common intermediate format</i>)
CLCSE	entité de signalisation de fermeture de la voie logique (<i>close logical channel signalling entity</i>)
CPCS	sous-couche de convergence de parties communes (de la couche d'Adaptation ATM 5) (<i>common part convergence sublayer</i>)
DSM-CC	moyens d'enregistrement numérique – Commandes et indications d'état (<i>digital storage media – command and control</i>)

DTMF	tonalités multifréquences à partir de deux sons (<i>dual tone multi-frequency</i>)
GOB	groupe de blocs (d'une image vidéo: voir les Recommandations H.261 et H.263) (<i>group of blocks</i>)
HDLC	commande de liaison de données à haut niveau (<i>high-level data link control</i>)
HRD	décodeur fictif de référence (se référer aux Recommandations H.261 et H.263) (<i>hypothetical reference decoder</i>)
IV	vecteur d'initialisation (utilisé pour le chiffrement: se référer aux Recommandations H.233 et H.234) (<i>initialization vector</i>)
LAPM	protocole d'accès de liaison pour modems (<i>link access protocol for modems</i>)
LCSE	entité de signalisation de voie logique (<i>logical channel signalling entity</i>)
MC	entité de commande multipoint H.323 (<i>H.323 multipoint control entity</i>)
MCU	pont de conférence ou équipement de conférence multipoint (<i>multipoint control unit</i>)
MLSE	entité de signalisation de boucle de maintenance (<i>maintenance loop signalling entity</i>)
MPI	intervalle minimal entre images (<i>minimum picture interval</i>)
MRSE	entité de signalisation de demande de mode (<i>mode request signalling entity</i>)
MSDSE	entité de signalisation pour le choix du mode maître ou esclave (<i>master slave determination signalling entity</i>)
MTSE	entité de signalisation du tableau de multiplexage (<i>multiplex table signalling entity</i>)
PCR	référence d'horloge programme (se référer à la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>program clock reference</i>)
PID	identificateur de paquet (se référer à la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>packet identifier</i>)
QCIF	format quart de CIF (<i>quarter CIF</i>)
RMESE	entité de signalisation de demande d'entrée de multiplexage (<i>request multiplex entry signalling entity</i>)
RTCP	protocole de commande de transport en temps réel (<i>real-time transport control protocol</i>)
RTDSE	entité de signalisation du temps de propagation aller et retour (<i>round trip delay signalling entity</i>)
RTGC	réseau téléphonique général commuté
RTP	protocole de transport en temps réel (<i>real-time transport protocol</i>)
SDL	langage de description et de spécification (<i>specification and description language</i>)
SDU	unité de données de service (<i>service data unit</i>)
SE	message d'échange de session (utilisé pour le chiffrement: se référer aux Recommandations H.233 et H.234), (<i>session exchange message</i>)
SQCIF	format sous-quart de CIF (<i>sub QCIF</i>)
STD	décodeur de cible système (se référer à la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>system target decoder</i>)
VC	voie virtuelle ATM (<i>ATM virtual channel</i>)

5 Généralités

La présente Recommandation fournit un grand nombre de services différents dont certains sont supposés être applicables à tous les terminaux qui y font référence, certains étant plus spécifiques à un type de terminaux donnés. Les procédures sont définies pour permettre l'échange de capacités audiovisuelles et de données; afin de demander de transmettre selon un mode audiovisuel et de données particulier; pour assurer la gestion des voies logiques utilisées dans le transport des informations audiovisuelles et de données; pour définir quel terminal sera le terminal maître et quel terminal sera le terminal esclave afin de gérer des canaux logiques; pour transporter différents signaux de commande et d'indication d'état; pour commander le débit de différentes voies logiques et de l'ensemble du multiplex; et pour mesurer le temps de propagation aller et retour, d'un terminal vers l'autre et dans l'autre sens. Les procédures sont expliquées de manière plus détaillée ci-dessous.

Après cette introduction générale, les sous-paragraphes détaillent la syntaxe et la sémantique des messages, ainsi que les procédures. La syntaxe a été définie en utilisant la notation ASN.1 [30] et la sémantique définit la signification des éléments de syntaxe tout en indiquant les contraintes syntaxiques qui ne sont pas spécifiées dans la syntaxe de la notation ASN.1. Le sous-paragraphes détaillant les procédures définit les protocoles utilisant les messages définis dans les autres sous-paragraphes.

Bien que tous les messages et toutes les procédures définis dans la présente Recommandation ne soient pas applicables à tous les terminaux, aucune indication relative à de telles restrictions n'est donnée ici. Ces restrictions relèvent des recommandations faisant référence à la présente Recommandation.

La présente Recommandation a été définie comme étant indépendante du mécanisme de transport sous-jacent, mais est supposée être utilisée avec une couche de transport fiable, c'est-à-dire une couche garantissant la transmission correcte des données.

5.1 Choix du mode maître-esclave

Des conflits peuvent survenir quand deux terminaux dans une communication déclenchent simultanément des événements similaires alors que seul un de ces événements est possible ou souhaité, par exemple quand des ressources sont disponibles pour un seul événement fortuit. Afin de résoudre de tels conflits, un terminal devra se comporter comme le terminal maître et l'autre comme le terminal esclave. Des règles spécifient comment le terminal maître et le terminal esclave doivent réagir en cas d'incompatibilité.

La procédure du choix maître-esclave permet aux terminaux dans une communication de déterminer quel est le terminal maître et quel est le terminal esclave. Une fois qu'il a été défini, le statut du terminal reste inchangé pendant la durée de l'appel.

5.2 Echange de capacités

Les procédures d'échange de capacités sont prévues pour assurer que les seuls signaux multimédias devant être transmis sont ceux qui peuvent être reçus et traités de façon appropriée par le terminal récepteur. Cela implique que les capacités de réception et de décodage d'un terminal soient connues de l'autre terminal. Il n'est pas nécessaire qu'un terminal comprenne ou enregistre toutes les capacités de réception; celles qui ne sont pas comprises ou ne peuvent pas être utilisées seront omises, ce qui ne doit pas être considéré comme une anomalie.

Toutes les capacités d'un terminal à recevoir et à décoder différents signaux sont déclarées à l'autre terminal par la transmission de l'ensemble de ses capacités.

Les capacités de réception décrivent les capacités qu'a le terminal de recevoir et de traiter les flux d'informations entrants. Les émetteurs devront limiter les représentations des informations transmises à celles que le récepteur a déclaré pouvoir utiliser. L'absence de capacités de réception indique que le terminal ne peut pas recevoir d'informations (est uniquement émetteur).

Les capacités d'émission décrivent les capacités du terminal à transmettre des flux d'information. Les capacités d'émission permettent de présenter aux récepteurs un choix de modes possibles de fonctionnement, de sorte que le récepteur puisse demander le mode dans lequel il préfère recevoir. L'absence de toute capacité d'émission indique que le terminal ne propose pas un choix de modes préférés au récepteur (mais il peut cependant transmettre tout ce qui est dans les limites des capacités du récepteur).

Ces ensembles de capacités sont conçus pour permettre l'envoi simultané de plus d'un flux pour un type de support donné. Par exemple, un terminal peut déclarer qu'il a la capacité de recevoir (ou d'envoyer) en même temps deux flux vidéo indépendants correspondant à la Recommandation H.262 et deux flux audio indépendants correspondant à la Recommandation G.722. Des messages de capacités ont été définis pour permettre à un terminal d'indiquer qu'il n'a pas de capacités fixées, mais que ses capacités dépendent des autres modes qui sont utilisés simultanément. Il est par exemple possible d'indiquer que la vidéo à plus haute résolution peut être décodée quand un algorithme audio plus simple est utilisé; ou que deux séquences vidéo à faible résolution ou une seule séquence unique à haute résolution peuvent être décodées. Il est également possible d'indiquer des compromis entre les capacités d'émission et les capacités de réception.

Des capacités non normalisées et des messages de commande peuvent être émis en utilisant la structure NonStandardParameter. Il convient de noter que même si la signification des messages non normalisés est définie par des organismes différents, les équipements construits par n'importe quel fabricant peuvent signaler tout message non normalisé, pour autant que sa signification soit connue.

Les terminaux peuvent réémettre des ensembles de capacités à tout moment.

5.3 Procédures de signalisation de la voie logique

Un protocole avec acquittement est défini pour l'ouverture et la fermeture des canaux logiques qui transportent les informations audiovisuelles et les données. L'objectif de ces procédures est de garantir qu'un terminal peut recevoir et décoder les données qui seront transmises sur une voie logique au moment où cette voie logique sera ouverte plutôt qu'au moment où les premières données seront transmises sur cette voie; et de garantir que le terminal récepteur sera prêt à recevoir et à décoder les données qui seront transmises sur la voie logique avant que la transmission ne commence. Le message d'ouverture de la voie logique OpenLogicalChannel inclut une description des données devant être transmises, par exemple H.262 MP@ML au débit de 6 Mbit/s. Les canaux logiques ne devraient être ouverts que lorsque les capacités sont suffisantes pour recevoir simultanément des données sur tous les canaux logiques ouverts.

Une partie de ce protocole concerne l'ouverture de canaux bidirectionnels. Pour éviter les incidents qui pourraient survenir au lancement simultané d'événements similaires par deux terminaux, un terminal est défini comme le terminal maître, et l'autre comme le terminal esclave. Un protocole est défini afin de désigner le terminal maître et le terminal asservi. Cependant, des systèmes appliquant la présente Recommandation peuvent spécifier d'autres moyens de choisir quel sera le terminal maître et quel sera le terminal esclave.

5.4 Réception de la demande de fermeture de la voie logique

Une voie logique est ouverte et fermée du côté de l'émetteur. Un mécanisme est défini pour permettre à un terminal récepteur de demander la fermeture d'une voie logique entrante. Le terminal émetteur peut accepter ou refuser la demande de fermeture de la voie logique. Un terminal peut, par exemple, utiliser ces procédures pour demander la fermeture d'une voie logique entrante qu'il ne peut pas décoder pour quelque raison que ce soit. Ces procédures peuvent être également utilisées pour demander la fermeture d'une voie logique bidirectionnelle par le terminal qui n'a pas ouvert la voie.

5.5 Modification des entrées dans le tableau de multiplexage H.223

Le tableau de multiplexage H.223 associe chaque octet à l'intérieur d'un message MUX H.223 à un numéro de voie logique donné. Le tableau de multiplexage H.223 peut contenir jusqu'à 15 entrées. Un mécanisme permet au terminal émetteur de spécifier et d'informer le récepteur de la présence de nouvelles entrées dans le tableau de multiplexage H.223. Un terminal récepteur peut également demander la retransmission d'une entrée du tableau de multiplexage.

5.6 Demande de mode audiovisuel et de mode données

Lorsque le protocole d'échange de capacités se termine, chacun des deux terminaux a eu connaissance des capacités de l'autre terminal à émettre et recevoir comme cela est spécifié dans les descripteurs de capacités qui ont été échangés. Un terminal ne doit pas obligatoirement déclarer toutes ses capacités; il ne doit déclarer que celles qu'il souhaite utiliser.

Un terminal peut indiquer ses capacités d'émission. Un terminal qui reçoit des capacités d'émission en provenance du terminal distant peut demander qu'un mode particulier lui soit transmis. Un terminal indique qu'il ne veut pas que son mode émission soit commandé par le terminal distant en n'envoyant pas de capacités d'émission.

5.7 Détermination du temps de propagation aller et retour

Il peut être utile dans certaines applications de connaître le temps de propagation aller et retour entre un terminal émetteur et un terminal récepteur. Un mécanisme permet de mesurer ce temps de propagation aller et retour. Ce mécanisme peut également être utilisé comme moyen de déterminer si le terminal distant fonctionne toujours.

5.8 Boucles de maintenance

Des procédures sont spécifiées en vue de mettre en service les boucles de maintenance. Il est possible de demander le bouclage d'une seule voie logique sous forme de boucle du circuit numérique ou de boucle après décodage, de même que la boucle du multiplex entier.

5.9 Commandes et indications d'état

Les commandes et indications d'état sont données pour différents besoins comme par exemple les indications d'activité ou d'inactivité des signaux audio et vidéo pour informer l'utilisateur; la demande de rafraîchissement accéléré pour la commutation de source dans les applications multipoint. Ni les commandes ni les indications d'état ne requièrent de messages de réponse à partir du terminal distant. Les commandes nécessitent une action du terminal distant alors que les indications d'état ne font qu'apporter des informations et ne nécessitent aucune action spécifique.

Une commande est définie pour permettre la commande du débit des canaux logiques et du multiplex à partir du terminal distant. Ceci répond à plusieurs objectifs: il est nécessaire d'assurer l'interfonctionnement avec des terminaux utilisant des multiplex dans lesquels les débits ne sont disponibles qu'en nombre fini, de réaliser des applications de conférence multipoint dans lesquelles

les débits provenant de différentes sources devraient être adaptés et d'assurer le contrôle de flux dans les réseaux encombrés.

6 Messages: syntaxe

Le présent sous-paragraphe spécifie la syntaxe des messages utilisant la notation définie en ASN.1 [30]. Les messages doivent être codés à l'émission en appliquant les règles de codage compactes spécifiées en [31] en utilisant la variante de base alignée. Le premier bit de chaque octet qui est transmis est le bit de poids le plus fort de l'octet comme cela est spécifié dans la Recommandation X.691.

MULTIMEDIA-SYSTEM-CONTROL DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

-- Export all symbols

-- =====
-- Top level Messages
-- =====

```
MultimediaSystemControlMessage ::=CHOICE
{
    request           RequestMessage,
    response        ResponseMessage,
    command         CommandMessage,
    indication     IndicationMessage,
    ...
}
```

-- A RequestMessage results in action and requires an immediate response

```
RequestMessage ::=CHOICE
{
    nonStandard           NonStandardMessage,
    masterSlaveDetermination MasterSlaveDetermination,
    terminalCapabilitySet TerminalCapabilitySet,
    openLogicalChannel    OpenLogicalChannel,
    closeLogicalChannel  CloseLogicalChannel,
    requestChannelClose  RequestChannelClose,
    multiplexEntrySend   MultiplexEntrySend,
    requestMultiplexEntry RequestMultiplexEntry,
    requestMode         RequestMode,
    roundTripDelayRequest RoundTripDelayRequest,
    maintenanceLoopRequest MaintenanceLoopRequest,
    ...,
    communicationModeRequest CommunicationModeRequest,
    conferenceRequest    ConferenceRequest
}
```

-- A *ResponseMessage* is the response to a request *Message*

```
ResponseMessage ::=CHOICE  
{  
  nonStandard                               NonStandardMessage,  
  
  masterSlaveDeterminationAck               MasterSlaveDeterminationAck,  
  masterSlaveDeterminationReject           MasterSlaveDeterminationReject,  
  
  terminalCapabilitySetAck                   TerminalCapabilitySetAck,  
  terminalCapabilitySetReject               TerminalCapabilitySetReject,  
  
  openLogicalChannelAck                     OpenLogicalChannelAck,  
  openLogicalChannelReject                 OpenLogicalChannelReject,  
  closeLogicalChannelAck                   CloseLogicalChannelAck,  
  
  requestChannelCloseAck                   RequestChannelCloseAck,  
  requestChannelCloseReject               RequestChannelCloseReject,  
  
  multiplexEntrySendAck                     MultiplexEntrySendAck,  
  multiplexEntrySendReject                 MultiplexEntrySendReject,  
  
  requestMultiplexEntryAck                 RequestMultiplexEntryAck,  
  requestMultiplexEntryReject             RequestMultiplexEntryReject,  
  
  requestModeAck                           RequestModeAck,  
  requestModeReject                       RequestModeReject,  
  
  roundTripDelayResponse                   RoundTripDelayResponse,  
  
  maintenanceLoopAck                       MaintenanceLoopAck,  
  maintenanceLoopReject                   MaintenanceLoopReject,  
  
  ...,  
  communicationModeResponse               CommunicationModeResponse,  
  
  conferenceResponse                       ConferenceResponse  
}
```

-- A *CommandMessage* requires action, but no explicit response

```
CommandMessage ::=CHOICE  
{  
  nonStandard                               NonStandardMessage,  
  
  maintenanceLoopOffCommand                 MaintenanceLoopOffCommand,  
  
  sendTerminalCapabilitySet                 SendTerminalCapabilitySet,  
  
  encryptionCommand                       EncryptionCommand,  
  
  flowControlCommand                       FlowControlCommand,  
  
  endSessionCommand                       EndSessionCommand,  
  
  miscellaneousCommand                     MiscellaneousCommand,  
  
  ...,  
  communicationModeCommand                 CommunicationModeCommand,  
}
```

```

    conferenceCommand          ConferenceCommand
}
-- An IndicationMessage is information that does not require action or response

IndicationMessage            ::=CHOICE
{
    nonStandard                NonStandardMessage,
    functionNotUnderstood      FunctionNotUnderstood,
    masterSlaveDeterminationRelease MasterSlaveDeterminationRelease,
    terminalCapabilitySetRelease TerminalCapabilitySetRelease,
    openLogicalChannelConfirm  OpenLogicalChannelConfirm,
    requestChannelCloseRelease RequestChannelCloseRelease,
    multiplexEntrySendRelease  MultiplexEntrySendRelease,
    requestMultiplexEntryRelease RequestMultiplexEntryRelease,
    requestModeRelease         RequestModeRelease,
    miscellaneousIndication     MiscellaneousIndication,
    jitterIndication            JitterIndication,
    h223SkewIndication          H223SkewIndication,
    newATMVCIndication          NewATMVCIndication,
    userInput                   UserInputIndication,
    ...,
    h2250MaximumSkewIndication H2250MaximumSkewIndication,
    mcLocationIndication        MCLocationIndication,
    conferenceIndication        ConferenceIndication,
    vendorIdentification        VendorIdentification,
    functionNotSupported        FunctionNotSupported
}

```

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

```
SequenceNumber                ::=INTEGER (0..255)
```

```
-- =====
-- Non standard Message definitions
-- =====
```

```
NonStandardMessage           ::=SEQUENCE
{
    nonStandardData           NonStandardParameter,
    ...
}

```

```
NonStandardParameter         ::=SEQUENCE
```

```

{
    nonStandardIdentifier          NonStandardIdentifier,
    data                          OCTET STRING
}

NonStandardIdentifier            ::=CHOICE
{
    object                        OBJECT IDENTIFIER,
    h221NonStandard              SEQUENCE
    {
        t35CountryCode           INTEGER (0..255),           -- country, per T.35
        t35Extension              INTEGER (0..255),           -- assigned nationally
        manufacturerCode          INTEGER (0..65535)          -- assigned nationally
    }
}

-----
-- Master-slave determination definitions
-----

MasterSlaveDetermination        ::=SEQUENCE
{
    terminalType                 INTEGER (0..255),
    statusDeterminationNumber    INTEGER (0..16777215),
    ...
}

MasterSlaveDeterminationAck     ::=SEQUENCE
{
    decision                     CHOICE
    {
        master                   NULL,
        slave                     NULL
    },
    ...
}

MasterSlaveDeterminationReject  ::=SEQUENCE
{
    cause                        CHOICE
    {
        identicalNumbers         NULL,
        ...
    },
    ...
}

MasterSlaveDeterminationRelease ::=SEQUENCE
{
    ...
}

-----
-- Capability exchange definitions
-----

TerminalCapabilitySet           ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,

```

protocolIdentifier	OBJECT IDENTIFIER, -- shall be set to the value -- {itu-t (0) recommendation (0) h (8) 245 version (0) 2}
multiplexCapability	MultiplexCapability OPTIONAL,
capabilityTable	SET SIZE (1..256) OF CapabilityTableEntry OPTIONAL,
capabilityDescriptors	SET SIZE (1..256) OF CapabilityDescriptor OPTIONAL,
...	
}	
V75Capability	::=SEQUENCE
{	
audioHeader	BOOLEAN,
...	
}	
CapabilityTableEntry	::=SEQUENCE
{	
capabilityTableEntryNumber	CapabilityTableEntryNumber,
capability	Capability OPTIONAL
}	
CapabilityDescriptor	::=SEQUENCE
{	
capabilityDescriptorNumber	CapabilityDescriptorNumber,
simultaneousCapabilities	SET SIZE (1..256) OF AlternativeCapabilitySet OPTIONAL
}	
AlternativeCapabilitySet	::=SEQUENCE SIZE (1..256) OF CapabilityTableEntryNumber
CapabilityTableEntryNumber	::=INTEGER (1..65535)
CapabilityDescriptorNumber	::=INTEGER (0..255)
TerminalCapabilitySetAck	::=SEQUENCE
{	
sequenceNumber	SequenceNumber,
...	
}	
TerminalCapabilitySetReject	::=SEQUENCE
{	
sequenceNumber	SequenceNumber,
cause	CHOICE
{	
unspecified	NULL,
undefinedTableEntryUsed	NULL,
descriptorCapacityExceeded	NULL,
tableEntryCapacityExceeded	CHOICE
{	
highestEntryNumberProcessed	CapabilityTableEntryNumber,
noneProcessed	NULL
},	
...	
},	
...	
}	
}	
}	

```

TerminalCapabilitySetRelease ::=SEQUENCE
{
    ...
}

-----
-- Capability exchange definitions: top level capability description
-----

Capability ::=CHOICE
{
    nonStandard NonStandardParameter,

    receiveVideoCapability VideoCapability,
    transmitVideoCapability VideoCapability,
    receiveAndTransmitVideoCapability VideoCapability,

    receiveAudioCapability AudioCapability,
    transmitAudioCapability AudioCapability,
    receiveAndTransmitAudioCapability AudioCapability,

    receiveDataApplicationCapability DataApplicationCapability,
    transmitDataApplicationCapability DataApplicationCapability,
    receiveAndTransmitDataApplicationCapability DataApplicationCapability,

    h233EncryptionTransmitCapability BOOLEAN,
    h233EncryptionReceiveCapability SEQUENCE
    {
        h233IVResponseTime INTEGER (0..255), -- units milliseconds
        ...
    },
    ...,
    conferenceCapability ConferenceCapability
}

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

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

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

VCCapability ::=SEQUENCE

```

```

{
  aal1                               SEQUENCE
  {
    nullClockRecovery                BOOLEAN,
    srtsClockRecovery                BOOLEAN,
    adaptiveClockRecovery            BOOLEAN,
    nullErrorCorrection              BOOLEAN,
    longInterleaver                  BOOLEAN,
    shortInterleaver                 BOOLEAN,
    errorCorrectionOnly              BOOLEAN,
    structuredDataTransfer           BOOLEAN,
    partiallyFilledCells             BOOLEAN,
    ...
  } OPTIONAL,
  aal5                               SEQUENCE
  {
    forwardMaximumSDUSize            INTEGER (0..65535), -- units octets
    backwardMaximumSDUSize          INTEGER (0..65535), -- units octets
    ...
  } OPTIONAL,
  transportStream                   BOOLEAN,
  programStream                     BOOLEAN,
  availableBitRates                 SEQUENCE
  {
    type                             CHOICE
    {
      singleBitRate                  INTEGER (1..65535), -- units 64 kbit/s
      rangeOfBitRates                SEQUENCE
      {
        lowerBitRate                 INTEGER (1..65535), -- units 64 kbit/s
        higherBitRate                INTEGER (1..65535) -- units 64 kbit/s
      }
    },
    ...
  },
  ...
}

H223Capability ::=SEQUENCE
{
  transportWithI-frames             BOOLEAN, -- I-frame transport of H.245

  videoWithAL1                     BOOLEAN,
  videoWithAL2                     BOOLEAN,
  videoWithAL3                     BOOLEAN,
  audioWithAL1                     BOOLEAN,
  audioWithAL2                     BOOLEAN,
  audioWithAL3                     BOOLEAN,
  dataWithAL1                      BOOLEAN,
  dataWithAL2                      BOOLEAN,
  dataWithAL3                      BOOLEAN,

  maximumAL2SDUSize                INTEGER (0..65535), -- units octets
  maximumAL3SDUSize                INTEGER (0..65535), -- units octets

  maximumDelayJitter               INTEGER (0..1023), -- units milliseconds

  h223MultiplexTableCapability     CHOICE
  {
    basic                           NULL,
    enhanced                         SEQUENCE
  }
}

```

```

        {
            maximumNestingDepth      INTEGER (1..15),
            maximumElementListSize   INTEGER (2..255),
            maximumSubElementListSize INTEGER (2..255),
            ...
        }
    },
    ...,
    maxMUXPDUSizeCapability          BOOLEAN
}

V76Capability                       ::=SEQUENCE
{
    suspendResumeCapabilitywAddress  BOOLEAN,
    suspendResumeCapabilitywoAddress  BOOLEAN,
    rejCapability                     BOOLEAN,
    sREJCapability                    BOOLEAN,
    mREJCapability                    BOOLEAN,
    crc8bitCapability                 BOOLEAN,
    crc16bitCapability                 BOOLEAN,
    crc32bitCapability                 BOOLEAN,
    uihCapability                     BOOLEAN,
    numOfDLCS                         INTEGER (2..8191),
    twoOctetAddressFieldCapability    BOOLEAN,
    loopBackTestCapability            BOOLEAN,
    n401Capability                    INTEGER (1..4095),
    maxWindowSizeCapability           INTEGER (1..127),
    v75Capability                     V75Capability,
    ...
}

H2250Capability                     ::=SEQUENCE
{
    maximumAudioDelayJitter           INTEGER(0..1023), -- units in milliseconds
    receiveMultipointCapability        MultipointCapability,
    transmitMultipointCapability       MultipointCapability,
    receiveAndTransmitMultipointCapability MultipointCapability,
    mcCapability                       SEQUENCE
    {
        centralizedConferenceMC        BOOLEAN,
        decentralizedConferenceMC      BOOLEAN,
        ...
    },
    rtcpVideoControlCapability         BOOLEAN, -- FIR and NACK
    mediaPacketizationCapability       MediaPacketizationCapability,
    ...
}

MediaPacketizationCapability         ::=SEQUENCE
{
    h261aVideoPacketization           BOOLEAN,
    ...
}

MultipointCapability                 ::=SEQUENCE
{
    multicastCapability                BOOLEAN,
    multiUniCastConference             BOOLEAN,
    mediaDistributionCapability         SEQUENCE OF MediaDistributionCapability,
    ...
}

```

```

MediaDistributionCapability ::=SEQUENCE
{
    centralizedControl          BOOLEAN,
    distributedControl          BOOLEAN,           -- for further study in H.323
    centralizedAudio           BOOLEAN,
    distributedAudio           BOOLEAN,
    centralizedVideo           BOOLEAN,
    distributedVideo           BOOLEAN,
    centralizedData            SEQUENCE OF DataApplicationCapability OPTIONAL,
    distributedData            SEQUENCE OF DataApplicationCapability OPTIONAL,
                                -- for further study in H.323
    ...
}

```

```

H223AnnexACapability ::=SEQUENCE
{
    transferWithI-frames       BOOLEAN,           -- I-frame transport of H.245

    videoWithAL1M             BOOLEAN,
    videoWithAL2M             BOOLEAN,
    videoWithAL3M             BOOLEAN,
    audioWithAL1M             BOOLEAN,
    audioWithAL2M             BOOLEAN,
    audioWithAL3M             BOOLEAN,
    dataWithAL1M              BOOLEAN,
    dataWithAL2M              BOOLEAN,
    dataWithAL3M              BOOLEAN,

    maximumAL2MSDUSize        INTEGER (0..65535), -- units octets
    maximumAL3MSDUSize        INTEGER (0..65535), -- units octets

    maximumDelayJitter        INTEGER (0..1023), -- units milliseconds

    reconfigurationCapability  BOOLEAN,

    h223AnnexAMultiplexTableCapability CHOICE           -- identical to H.223
    {
        basic                  NULL,
        enhanced                SEQUENCE
        {
            maximumNestingDepth  INTEGER (1..15),
            maximumElementListSize  INTEGER (2..255),
            maximumSubElementListSize  INTEGER (2..255),
            ...
        },
        ...
    },
    ...
}

```

```

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

```

```

VideoCapability ::=CHOICE
{
    nonStandard                NonStandardParameter,
    h261VideoCapability        H261VideoCapability,
    h262VideoCapability        H262VideoCapability,
    h263VideoCapability        H263VideoCapability,
}

```

```

is11172VideoCapability          IS11172VideoCapability,
...
}

H261VideoCapability             ::=SEQUENCE
{
    qcifMPI                      INTEGER (1..4) OPTIONAL,      -- units 1/29.97 Hz
    cifMPI                       INTEGER (1..4) OPTIONAL,      -- units 1/29.97 Hz
    temporalSpatialTradeOffCapability  BOOLEAN,
    maxBitRate                   INTEGER (1..19200),            -- units of 100 bit/s
    stillImageTransmission       BOOLEAN,                      -- Annex D of H.261
    ...
}

H262VideoCapability             ::=SEQUENCE
{
    profileAndLevel-SPatML       BOOLEAN,
    profileAndLevel-MPatLL       BOOLEAN,
    profileAndLevel-MPatML       BOOLEAN,
    profileAndLevel-MPatH-14     BOOLEAN,
    profileAndLevel-MPatHL       BOOLEAN,
    profileAndLevel-SNRatLL      BOOLEAN,
    profileAndLevel-SNRatML      BOOLEAN,
    profileAndLevel-SpatialatH-14  BOOLEAN,
    profileAndLevel-HPatML       BOOLEAN,
    profileAndLevel-HPatH-14     BOOLEAN,
    profileAndLevel-HPatHL       BOOLEAN,
    videoBitRate                 INTEGER (0.. 1073741823) OPTIONAL, -- units 400 bit/s
    vbvBufferSize                INTEGER (0.. 262143) OPTIONAL,  -- units 16384 bits
    samplesPerLine               INTEGER (0..16383) OPTIONAL,    -- units samples/line
    linesPerFrame                INTEGER (0..16383) OPTIONAL,    -- units lines/frame
    framesPerSecond              INTEGER (0..15) OPTIONAL,       -- frame_rate_code
    luminanceSampleRate          INTEGER (0..4294967295) OPTIONAL, -- units samples/sec
    ...
}

H263VideoCapability             ::=SEQUENCE
{
    sqcifMPI                     INTEGER (1..32) OPTIONAL,      -- units 1/29.97 Hz
    qcifMPI                      INTEGER (1..32) OPTIONAL,      -- units 1/29.97 Hz
    cifMPI                       INTEGER (1..32) OPTIONAL,      -- units 1/29.97 Hz
    cif4MPI                     INTEGER (1..32) OPTIONAL,      -- units 1/29.97 Hz
    cif16MPI                    INTEGER (1..32) OPTIONAL,      -- units 1/29.97 Hz
    maxBitRate                   INTEGER (1..192400),           -- units 100 bit/s
    unrestrictedVector           BOOLEAN,
    arithmeticCoding             BOOLEAN,
    advancedPrediction           BOOLEAN,
    pbFrames                     BOOLEAN,
    temporalSpatialTradeOffCapability  BOOLEAN,
    hrd-B                        INTEGER (0..524287) OPTIONAL,  -- units 128 bits
    bppMaxKb                    INTEGER (0..65535) OPTIONAL,    -- units 1024 bits
    ...,

    slowSqcifMPI                INTEGER (1..3600) OPTIONAL,    -- units seconds/frame
    slowQcifMPI                  INTEGER (1..3600) OPTIONAL,    -- units seconds/frame
    slowCifMPI                   INTEGER (1..3600) OPTIONAL,    -- units seconds/frame
    slowCif4MPI                  INTEGER (1..3600) OPTIONAL,    -- units seconds/frame
    slowCif16MPI                 INTEGER (1..3600) OPTIONAL,    -- units seconds/frame
    errorCompensation            BOOLEAN
}

```

```

IS11172VideoCapability ::=SEQUENCE
{
    constrainedBitstream      BOOLEAN,
    videoBitRate              INTEGER (0.. 1073741823) OPTIONAL, -- units 400 bit/s
    vbvBufferSize             INTEGER (0.. 262143) OPTIONAL,      -- units 16384 bits
    samplesPerLine            INTEGER (0..16383) OPTIONAL,        -- units samples/line
    linesPerFrame             INTEGER (0..16383) OPTIONAL,        -- units lines/frame
    pictureRate               INTEGER (0..15) OPTIONAL,
    luminanceSampleRate       INTEGER (0..4294967295) OPTIONAL,   -- units samples/sec
    ...
}

```

```

-- =====
-- Capability exchange definitions: Audio capabilities
-- =====

```

```

-- For an H.222 multiplex, the integers indicate the size of the STD buffer in units of 256 octets
-- For an H.223 multiplex, the integers indicate the maximum number of audio frames per AL-SDU
-- For an H.225.0 multiplex, the integers indicate the maximum number of audio frames per packet

```

```

AudioCapability ::=CHOICE
{
    nonStandard              NonStandardParameter,
    g711Alaw64k              INTEGER (1..256),
    g711Alaw56k              INTEGER (1..256),
    g711Ulaw64k              INTEGER (1..256),
    g711Ulaw56k              INTEGER (1..256),

    g722-64k                 INTEGER (1..256),
    g722-56k                 INTEGER (1..256),
    g722-48k                 INTEGER (1..256),

    g7231                     SEQUENCE
    {
        maxAl-sduAudioFrames  INTEGER (1..256),
        silenceSuppression    BOOLEAN
    },

    g728                      INTEGER (1..256),
    g729                      INTEGER (1..256),
    g729AnnexA                INTEGER (1..256),
    is11172AudioCapability    IS11172AudioCapability,
    is13818AudioCapability    IS13818AudioCapability,
    ...,
    g729wAnnexB               INTEGER(1..256),
    g729AnnexAwAnnexB        INTEGER(1..256),
    g7231AnnexCCapability    G7231AnnexCCapability
}

```

```

G7231AnnexCCapability ::= SEQUENCE
{
    maxAl-sduAudioFrames    INTEGER (1..256),
    silenceSuppression      BOOLEAN,
    g723AnnexCAudioMode    SEQUENCE
    {
        highRateMode0        INTEGER (27..78), -- units octets
        highRateMode1        INTEGER (27..78), -- units octets
        lowRateMode0         INTEGER (23..66), -- units octets
        lowRateMode1         INTEGER (23..66), -- units octets
        sidMode0             INTEGER (6..17),  -- units octets
        sidMode1             INTEGER (6..17),  -- units octets
    }
}

```

```

    ...
  } OPTIONAL,
  ...
}

IS11172AudioCapability ::=SEQUENCE
{
  audioLayer1          BOOLEAN,
  audioLayer2          BOOLEAN,
  audioLayer3          BOOLEAN,

  audioSampling32k     BOOLEAN,
  audioSampling44k1   BOOLEAN,
  audioSampling48k     BOOLEAN,

  singleChannel        BOOLEAN,
  twoChannels          BOOLEAN,

  bitRate              INTEGER (1..448),      -- units kbit/s
  ...
}

IS13818AudioCapability ::=SEQUENCE
{
  audioLayer1          BOOLEAN,
  audioLayer2          BOOLEAN,
  audioLayer3          BOOLEAN,

  audioSampling16k     BOOLEAN,
  audioSampling22k05  BOOLEAN,
  audioSampling24k     BOOLEAN,
  audioSampling32k     BOOLEAN,
  audioSampling44k1   BOOLEAN,
  audioSampling48k     BOOLEAN,

  singleChannel        BOOLEAN,
  twoChannels          BOOLEAN,
  threeChannels2-1     BOOLEAN,
  threeChannels3-0     BOOLEAN,
  fourChannels2-0-2-0  BOOLEAN,
  fourChannels2-2      BOOLEAN,
  fourChannels3-1      BOOLEAN,
  fiveChannels3-0-2-0  BOOLEAN,
  fiveChannels3-2      BOOLEAN,

  lowFrequencyEnhancement  BOOLEAN,

  multilingual         BOOLEAN,

  bitRate              INTEGER (1..1130),    -- units kbit/s
  ...
}

```

```

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

```

```

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

DataProtocolCapability ::=CHOICE
{
  nonStandard NonStandardParameter,
  v14buffered NULL,
  v42lapm NULL, -- may negotiate to V.42bis
  hdlcFrameTunnelling NULL,
  h310SeparateVCStack NULL,
  h310SingleVCStack NULL,
  transparent NULL,
  ...,
  segmentationAndReassembly NULL,
  hdlcFrameTunnelingwSAR NULL,
  v120 NULL, -- as in H.230
  separateLANStack NULL,

  v76wCompression CHOICE
  {
    transmitCompression CompressionType,
    receiveCompression CompressionType,
    transmitAndReceiveCompression CompressionType,
    ...
  }
}

```

```

CompressionType ::=CHOICE
{
    v42bis          V42bis,
    ...
}

V42bis ::=SEQUENCE
{
    numberOfCodewords    INTEGER (1..65536),
    maximumStringLength  INTEGER (1..256),
    ...
}

T84Profile ::=CHOICE
{
    t84Unrestricted     NULL,
    t84Restricted       SEQUENCE
    {
        qcif            BOOLEAN,
        cif             BOOLEAN,
        ccir601Seq      BOOLEAN,
        ccir601Prog     BOOLEAN,
        hdtvSeq         BOOLEAN,
        hdtvProg        BOOLEAN,

        g3FacsMH200x100  BOOLEAN,
        g3FacsMH200x200  BOOLEAN,
        g4FacsMMR200x100  BOOLEAN,
        g4FacsMMR200x200  BOOLEAN,
        jbig200x200Seq   BOOLEAN,
        jbig200x200Prog  BOOLEAN,
        jbig300x300Seq   BOOLEAN,
        jbig300x300Prog  BOOLEAN,

        digPhotoLow     BOOLEAN,
        digPhotoMedSeq  BOOLEAN,
        digPhotoMedProg  BOOLEAN,
        digPhotoHighSeq  BOOLEAN,
        digPhotoHighProg  BOOLEAN,
        ...
    }
}

=====
-- Capability Exchange Definitions: Conference
=====
ConferenceCapability ::=SEQUENCE
{
    nonStandardData     SEQUENCE OF NonStandardParameter OPTIONAL,
    chairControlCapability  BOOLEAN,
    ...
}

=====
-- Logical channel signalling definitions
=====

-- 'Forward' is used to refer to transmission in the direction from the terminal making the
-- original request for a logical channel to the other terminal, and 'reverse' is used to refer
-- to the opposite direction of transmission, in the case of a bi-directional channel request.

```

```

OpenLogicalChannel ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,

    forwardLogicalChannelParameters SEQUENCE
    {
        portNumber INTEGER (0..65535) OPTIONAL,
        dataType DataType,
        multiplexParameters CHOICE
        {
            h222LogicalChannelParameters H222LogicalChannelParameters,
            h223LogicalChannelParameters H223LogicalChannelParameters,
            v76LogicalChannelParameters V76LogicalChannelParameters,
            ....
            h2250LogicalChannelParameters H2250LogicalChannelParameters
        },
        ...
    },

    -- Used to specify the reverse channel for bi-directional open request

    reverseLogicalChannelParameters SEQUENCE
    {
        dataType DataType,
        multiplexParameters CHOICE
        {
            -- H.222 parameters are never present in reverse direction
            h223LogicalChannelParameters H223LogicalChannelParameters,
            v76LogicalChannelParameters V76LogicalChannelParameters,
            ....
            h2250LogicalChannelParameters H2250LogicalChannelParameters
        } OPTIONAL,
        ...
    } OPTIONAL,
    ...,
    separateStack NetworkAccessParameters OPTIONAL
    -- for Open responder to establish the stack
}

LogicalChannelNumber ::=INTEGER (1..65535)

NetworkAccessParameters ::=SEQUENCE
{
    distribution CHOICE
    {
        unicast NULL,
        multicast NULL,
        ...
    } OPTIONAL,

    networkAddress CHOICE
    {
        q2931Address Q2931Address,
        e164Address IA5String(SIZE(1..128)) (FROM ("0123456789#*,")),
    }
}

```

<pre> localAreaAddress ... }, associateConference externalReference ... } </pre>	<pre> TransportAddress, BOOLEAN, OCTET STRING(SIZE(1..255)) OPTIONAL, </pre>
<pre> Q2931Address { address { internationalNumber nsapAddress ... }, subaddress ... } </pre>	<pre> ::=SEQUENCE CHOICE NumericString(SIZE(1..16)), OCTET STRING (SIZE(1..20)), OCTET STRING (SIZE(1..20)) OPTIONAL, </pre>
<pre> V75Parameters { audioHeaderPresent ... } </pre>	<pre> ::= SEQUENCE BOOLEAN, </pre>
<pre> DataType { nonStandard nullData videoData audioData data encryptionData ... } </pre>	<pre> ::=CHOICE NonStandardParameter, NULL, VideoCapability, AudioCapability, DataApplicationCapability, EncryptionMode, </pre>
<pre> H222LogicalChannelParameters { resourceID subChannelID pcr-pid programDescriptors streamDescriptors ... } </pre>	<pre> ::=SEQUENCE INTEGER (0..65535), INTEGER (0..8191), INTEGER (0..8191) OPTIONAL, OCTET STRING OPTIONAL, OCTET STRING OPTIONAL, </pre>
<pre> H223LogicalChannelParameters { adaptationLayerType { nonStandard al1Framed al1NotFramed al2WithoutSequenceNumbers } } </pre>	<pre> ::=SEQUENCE CHOICE NonStandardParameter, NULL, NULL, NULL, </pre>

<pre> al2WithSequenceNumbers al3 { controlFieldOctets sendBufferSize }, ... }, segmentableFlag ... } </pre>	<pre> NULL, SEQUENCE INTEGER (0..2), INTEGER (0..16777215) </pre>	<pre> -- units octets </pre>
<pre> V76LogicalChannelParameters { hdlcParameters suspendResume { noSuspendResume suspendResumewAddress suspendResumewoAddress ... }, uIH mode { eRM { windowSize recovery { rej Srej mSREJ ... }, ... }, uNERM ... }, v75Parameters ... } </pre>	<pre> ::=SEQUENCE V76HDLParameters, CHOICE NULL, NULL, NULL, BOOLEAN, CHOICE SEQUENCE INTEGER (1..127), CHOICE NULL, NULL, NULL, NULL, V75Parameters, </pre>	
<pre> V76HDLParameters { crcLength n401 loopbackTestProcedure ... } </pre>	<pre> ::=SEQUENCE CRCLength, INTEGER (1..4095), BOOLEAN, </pre>	
<pre> CRCLength { crc8bit crc16bit crc32bit ... } </pre>	<pre> ::=CHOICE NULL, NULL, NULL, </pre>	

H2250LogicalChannelParameters	::=SEQUENCE
{	
nonStandard	SEQUENCE OF NonStandardParameter OPTIONAL,
sessionID	INTEGER(0..255),
associatedSessionID	INTEGER(1..255) OPTIONAL,
mediaChannel	TransportAddress OPTIONAL,
mediaGuaranteedDelivery	BOOLEAN OPTIONAL,
mediaControlChannel	TransportAddress OPTIONAL, -- reverse RTCP channel
mediaControlGuaranteedDelivery	BOOLEAN OPTIONAL,
silenceSuppression	BOOLEAN OPTIONAL,
destination	TerminalLabel OPTIONAL,
dynamicRTPPayloadType	INTEGER(96..127) OPTIONAL,
mediaPacketization	CHOICE
{	
h261aVideoPacketization	NULL,
...	
} OPTIONAL,	
...	
}	
TransportAddress	::=CHOICE
{	
unicastAddress	UnicastAddress,
multicastAddress	MulticastAddress,
...	
}	
UnicastAddress	::=CHOICE
{	
iPAddress	SEQUENCE
{	
network	OCTET STRING (SIZE(4)),
tsapIdentifier	INTEGER(0..65535),
...	
},	
iPXAddress	SEQUENCE
{	
node	OCTET STRING (SIZE(6)),
netnum	OCTET STRING (SIZE(4)),
tsapIdentifier	OCTET STRING (SIZE(2)),
...	
},	
iP6Address	SEQUENCE
{	
network	OCTET STRING (SIZE(16)),
tsapIdentifier	INTEGER(0..65535),
...	
},	
netBios	OCTET STRING (SIZE(16)),
iPSourceRouteAddress	SEQUENCE
{	
routing	CHOICE
{	
strict	NULL,
loose	NULL
},	
network	OCTET STRING (SIZE(4)),

```

        tsapIdentifier      INTEGER(0..65535),
        route              SEQUENCE OF OCTET STRING (SIZE(4)),
        ...
    },
    ...,
    nsap                  OCTET STRING (SIZE(1..20)),
    nonStandardAddress    NonStandardParameter
}

MulticastAddress ::= CHOICE
{
    ipAddress            SEQUENCE
    {
        network          OCTET STRING (SIZE(4)),
        tsapIdentifier    INTEGER(0..65535),
        ...
    },
    ip6Address           SEQUENCE
    {
        network          OCTET STRING (SIZE(16)),
        tsapIdentifier    INTEGER(0..65535),
        ...
    },
    ...,
    nsap                  OCTET STRING (SIZE(1..20)),
    nonStandardAddress    NonStandardParameter
}

OpenLogicalChannelAck ::= SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,

    reverseLogicalChannelParameters SEQUENCE
    {
        reverseLogicalChannelNumber LogicalChannelNumber,
        portNumber                  INTEGER (0..65535) OPTIONAL,
        multiplexParameters          CHOICE
        {
            h222LogicalChannelParameters H222LogicalChannelParameters,
            -- H.223 parameters are never present in reverse direction
            ...
            h2250LogicalChannelParameters H2250LogicalChannelParameters
        } OPTIONAL,
        -- Not present for H.223
        ...
    } OPTIONAL,
    -- Not present for uni-directional channel request
    ...,
    separateStack NetworkAccessParameters OPTIONAL,
    -- for Open requester to establish the stack
    forwardMultiplexAckParameters CHOICE
    {
        -- H.222 parameters are never present in the Ack
        -- H.223 parameters are never present in the Ack
        -- V.76 parameters are never present in the Ack
    }
}

```

```

        h2250LogicalChannelAckParameters H2250LogicalChannelAckParameters,
        ...
    } OPTIONAL
}

OpenLogicalChannelReject ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    cause CHOICE
    {
        unspecified NULL,
        unsuitableReverseParameters NULL,
        dataTypeNotSupported NULL,
        dataTypeNotAvailable NULL,
        unknownDataType NULL,
        dataTypeALCombinationNotSupported NULL,
        ...,
        multicastChannelNotAllowed NULL,
        insufficientBandwidth NULL,
        separateStackEstablishmentFailed NULL,
        invalidSessionID NULL,
        masterSlaveConflict NULL
    },
    ...
}

OpenLogicalChannelConfirm ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

H2250LogicalChannelAckParameters ::=SEQUENCE
{
    nonStandard SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID INTEGER(1..255) OPTIONAL,
    mediaChannel TransportAddress OPTIONAL,
    mediaControlChannel TransportAddress OPTIONAL, -- forward RTCP channel
    dynamicRTPPayloadType INTEGER(96..127) OPTIONAL, -- used only by the master
    ... -- or MC
}

CloseLogicalChannel ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    source CHOICE
    {
        user NULL,
        lcse NULL
    },
    ...
}

CloseLogicalChannelAck ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

```

```

RequestChannelClose ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

RequestChannelCloseAck ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

RequestChannelCloseReject ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    cause CHOICE
    {
        unspecified NULL,
        ...
    },
    ...
}

RequestChannelCloseRelease ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,
    ...
}

-- =====
-- H.223 multiplex table definitions
-- =====

MultiplexEntrySend ::=SEQUENCE
{
    sequenceNumber SequenceNumber,
    multiplexEntryDescriptors SET SIZE (1..15) OF MultiplexEntryDescriptor,
    ...
}

MultiplexEntryDescriptor ::=SEQUENCE
{
    multiplexTableEntryNumber MultiplexTableEntryNumber,
    elementList SEQUENCE SIZE (1..256) OF MultiplexElement OPTIONAL
}

MultiplexElement ::=SEQUENCE
{
    type CHOICE
    {
        logicalChannelNumber INTEGER(0..65535),
        subElementList SEQUENCE SIZE (2..255) OF MultiplexElement
    },
    repeatCount CHOICE
    {
        finite INTEGER (1..65535), -- repeats of type
        untilClosingFlag NULL -- used for last element
    }
}

```

MultiplexTableEntryNumber ::=INTEGER (1..15)

MultiplexEntrySendAck ::=SEQUENCE
{
 sequenceNumber SequenceNumber,
 multiplexTableEntryNumber SET SIZE (1..15) OF MultiplexTableEntryNumber,
 ...
}

MultiplexEntrySendReject ::=SEQUENCE
{
 sequenceNumber SequenceNumber,
 rejectionDescriptions SET SIZE (1..15) OF MultiplexEntryRejectionDescriptions,
 ...
}

MultiplexEntryRejectionDescriptions ::=SEQUENCE
{
 multiplexTableEntryNumber MultiplexTableEntryNumber,
 cause CHOICE
 {
 unspecifiedCause NULL,
 descriptorTooComplex NULL,
 ...
 },
 ...
}

MultiplexEntrySendRelease ::=SEQUENCE
{
 multiplexTableEntryNumber SET SIZE (1..15) OF MultiplexTableEntryNumber,
 ...
}

RequestMultiplexEntry ::=SEQUENCE
{
 entryNumbers SET SIZE (1..15) OF MultiplexTableEntryNumber,
 ...
}

RequestMultiplexEntryAck ::=SEQUENCE
{
 entryNumbers SET SIZE (1..15) OF MultiplexTableEntryNumber,
 ...
}

RequestMultiplexEntryReject ::=SEQUENCE
{
 entryNumbers SET SIZE (1..15) OF MultiplexTableEntryNumber,
 rejectionDescriptions SET SIZE (1..15) OF RequestMultiplexEntryRejectionDescriptions,
 ...
}

RequestMultiplexEntryRejectionDescriptions ::=SEQUENCE
{
 multiplexTableEntryNumber MultiplexTableEntryNumber,
 cause CHOICE

```

    {
        unspecifiedCause      NULL,
        ...
    },
    ...
}

RequestMultiplexEntryRelease ::=SEQUENCE
{
    entryNumbers             SET SIZE (1..15) OF MultiplexTableEntryNumber,
    ...
}

-- =====
-- Request mode definitions
-- =====

-- RequestMode is a list, in order or preference, of modes that a terminal would like
-- to have transmitted to it.

RequestMode                  ::=SEQUENCE
{
    sequenceNumber           SequenceNumber,
    requestedModes           SEQUENCE SIZE (1..256) OF ModeDescription,
    ...
}

RequestModeAck               ::=SEQUENCE
{
    sequenceNumber           SequenceNumber,
    response                 CHOICE
    {
        willTransmitMostPreferredMode  NULL,
        willTransmitLessPreferredMode  NULL,
        ...
    },
    ...
}

RequestModeReject            ::=SEQUENCE
{
    sequenceNumber           SequenceNumber,
    cause                   CHOICE
    {
        modeUnavailable       NULL,
        multipointConstraint  NULL,
        requestDenied         NULL,
        ...
    },
    ...
}

RequestModeRelease           ::=SEQUENCE
{
    ...
}

```

```

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

```

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

```

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

```

```

H223ModeParameters ::=SEQUENCE
{
    adaptationLayerType CHOICE
    {
        nonStandard NonStandardParameter,
        al1Framed NULL,
        al1NotFramed NULL,
        al2WithoutSequenceNumbers NULL,
        al2WithSequenceNumbers NULL,
        al3 SEQUENCE
        {
            controlFieldOctets INTEGER(0..2),
            sendBufferSize INTEGER(0..16777215) -- units octets
        },
        ...
    },
    segmentableFlag BOOLEAN,
    ...
}

```

```

V76ModeParameters ::=CHOICE
{
    suspendResumewAddress NULL,
    suspendResumewoAddress NULL,
    ...
}

```

```

=====
-- Request mode definitions: Video modes
=====

```

```

VideoMode ::=CHOICE
{
    nonStandard NonStandardParameter,
    h261VideoMode H261VideoMode,
    h262VideoMode H262VideoMode,
}

```

```

    h263VideoMode
    is11172VideoMode
    ...
}

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

H262VideoMode ::=SEQUENCE
{
    profileAndLevel CHOICE
    {
        profileAndLevel-SPatML NULL,
        profileAndLevel-MPatLL NULL,
        profileAndLevel-MPatMLNULL,
        profileAndLevel-MPatH-14 NULL,
        profileAndLevel-MPatHL NULL,
        profileAndLevel-SNRatLL NULL,
        profileAndLevel-SNRatML NULL,
        profileAndLevel-SpatialatH-14 NULL,
        profileAndLevel-HPatML NULL,
        profileAndLevel-HPatH-14 NULL,
        profileAndLevel-HPatHL NULL,
        ...
    },
    videoBitRate INTEGER(0..1073741823) OPTIONAL, -- units 400bit/s
    vbvBufferSize INTEGER(0..262143) OPTIONAL, -- units 16384bits
    samplesPerLine INTEGER(0..16383) OPTIONAL, -- units samples/line
    linesPerFrame INTEGER(0..16383) OPTIONAL, -- units lines/frame
    framesPerSecond INTEGER(0..15) OPTIONAL, -- frame_rate_code
    luminanceSampleRate INTEGER(0..4294967295) OPTIONAL, -- units samples/sec
    ...
}

H263VideoMode ::=SEQUENCE
{
    resolution CHOICE
    {
        sqcif NULL,
        qcif NULL,
        cif NULL,
        cif4 NULL,
        cif16 NULL,
        ...
    },
    bitRate INTEGER (1..19200), -- units 100 bit/s
    unrestrictedVector BOOLEAN,
    arithmeticCoding BOOLEAN,
    advancedPrediction BOOLEAN,
    pbFrames BOOLEAN,
    ...,
}

```

```

        errorCompensation          BOOLEAN
    }

IS11172VideoMode                ::=SEQUENCE
{
    constrainedBitstream          BOOLEAN,
    videoBitRate                  INTEGER(0..1073741823) OPTIONAL,      -- units 400bit/s
    vbvBufferSize                 INTEGER(0..262143) OPTIONAL,          -- units 16384bits
    samplesPerLine                 INTEGER(0..16383) OPTIONAL,          -- units samples/line
    linesPerFrame                  INTEGER(0..16383) OPTIONAL,          -- units lines/frame
    pictureRate                     INTEGER(0..15) OPTIONAL,
    luminanceSampleRate            INTEGER(0..4294967295) OPTIONAL,      -- units samples/sec
    ...
}

=====
-- Request mode definitions: Audio modes
=====

AudioMode                        ::=CHOICE
{
    nonStandard                   NonStandardParameter,
    g711Alaw64k                   NULL,
    g711Alaw56k                   NULL,
    g711Ulaw64k                   NULL,
    g711Ulaw56k                   NULL,

    g722-64k                      NULL,
    g722-56k                      NULL,
    g722-48k                      NULL,

    g728                          NULL,
    g729                          NULL,
    g729AnnexA                    NULL,

    g7231                          CHOICE
    {
        noSilenceSuppressionLowRate  NULL,
        noSilenceSuppressionHighRate NULL,
        silenceSuppressionLowRate    NULL,
        silenceSuppressionHighRate   NULL
    },

    is11172AudioMode              IS11172AudioMode,
    is13818AudioMode              IS13818AudioMode,

    ...,

    g729wAnnexB                   INTEGER(1..256),
    g729AnnexAwAnnexB             INTEGER(1..256),
    g7231AnnexCMode                {G7231AnnexCMode}
}

IS11172AudioMode                ::=SEQUENCE
{
    audioLayer                     CHOICE
    {
        audioLayer1                NULL,

```

```

        audioLayer2          NULL,
        audioLayer3          NULL
    },
    audioSampling             CHOICE
    {
        audioSampling32k     NULL,
        audioSampling44k1    NULL,
        audioSampling48k     NULL
    },
    multichannelType          CHOICE
    {
        singleChannel         NULL,
        twoChannelStereo      NULL,
        twoChannelDual        NULL
    },
    bitRate                   INTEGER (1..448),    --units kbit/s
    ...
}
IS13818AudioMode           ::=SEQUENCE
{
    audioLayer               CHOICE
    {
        audioLayer1          NULL,
        audioLayer2          NULL,
        audioLayer3          NULL
    },
    audioSampling             CHOICE
    {
        audioSampling16k     NULL,
        audioSampling22k05   NULL,
        audioSampling24k     NULL,
        audioSampling32k     NULL,
        audioSampling44k1    NULL,
        audioSampling48k     NULL
    },
    multichannelType          CHOICE
    {
        singleChannel         NULL,
        twoChannelStereo      NULL,
        twoChannelDual        NULL,
        threeChannels2-1      NULL,
        threeChannels3-0      NULL,
        fourChannels2-0-2-0   NULL,
        fourChannels2-2       NULL,
        fourChannels3-1       NULL,
        fiveChannels3-0-2-0   NULL,
        fiveChannels3-2       NULL
    },
    lowFrequencyEnhancement  BOOLEAN,
    multilingual              BOOLEAN,
    bitRate                   INTEGER (1..1130),  --units kbit/s
    ...
}

```

```

G7231AnnexCMode ::= SEQUENCE
{
    maxAl-sduAudioFrames      INTEGER (1..256),
    silenceSuppression        BOOLEAN,
    g723AnnexCAudioMode      SEQUENCE
    {
        highRateMode0        INTEGER (27..78),      -- units octets
        highRateMode1        INTEGER (27..78),      -- units octets
        lowRateMode0         INTEGER (23..66),      -- units octets
        lowRateMode1         INTEGER (23..66),      -- units octets
        sidMode0             INTEGER (6..17),       -- units octets
        sidMode1             INTEGER (6..17),       -- units octets
        ...
    },
    ...
}

-----
-- Request mode definitions: Data modes
-----

```

```

DataMode ::= SEQUENCE
{
    application                CHOICE
    {
        nonStandard            NonStandardParameter,
        t120                   DataProtocolCapability,
        dsm-cc                 DataProtocolCapability,
        userData               DataProtocolCapability,
        t84                    DataProtocolCapability,
        t434                   DataProtocolCapability,
        h224                   DataProtocolCapability,
        nlpid                  SEQUENCE
        {
            nlpidProtocol      DataProtocolCapability,
            nlpidData          OCTET STRING
        },
        dsvdControl            NULL,
        h222DataPartitioning  DataProtocolCapability,
        ...
    },
    bitRate                   INTEGER (0..4294967295),  -- units 100 bit/s
    ...
}

-----
-- Request mode definitions: Encryption modes
-----

```

```

EncryptionMode ::= CHOICE
{
    nonStandard                NonStandardParameter,
    h233Encryption            NULL,
    ...
}

-----
-- Round Trip Delay definitions
-----

```

```

RoundTripDelayRequest ::=SEQUENCE
{
    sequenceNumber      SequenceNumber,
    ...
}

RoundTripDelayResponse ::=SEQUENCE
{
    sequenceNumber      SequenceNumber,
    ...
}

-----
-- Maintenance Loop definitions
-----

MaintenanceLoopRequest ::=SEQUENCE
{
    type                CHOICE
    {
        systemLoop      NULL,
        mediaLoop        LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    ...
}

MaintenanceLoopAck ::=SEQUENCE
{
    type                CHOICE
    {
        systemLoop      NULL,
        mediaLoop        LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    ...
}

MaintenanceLoopReject ::=SEQUENCE
{
    type                CHOICE
    {
        systemLoop      NULL,
        mediaLoop        LogicalChannelNumber,
        logicalChannelLoop LogicalChannelNumber,
        ...
    },
    cause                CHOICE
    {
        canNotPerformLoop NULL,
        ...
    },
    ...
}

MaintenanceLoopOffCommand ::=SEQUENCE
{
    ...
}

```

```

=====
-- Communication Mode definitions
=====

```

```

CommunicationModeCommand ::=SEQUENCE
{
    communicationModeTable SET SIZE(1..256) OF CommunicationModeTableEntry,
    ...
}

```

```

CommunicationModeRequest ::=SEQUENCE
{
    ...
}

```

```

CommunicationModeResponse ::=CHOICE
{
    communicationModeTable SET SIZE(1..256) OF CommunicationModeTableEntry,
    ...
}

```

```

CommunicationModeTableEntry ::=SEQUENCE
{
    nonStandard SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID INTEGER(1..255),
    associatedSessionID INTEGER(1..255) OPTIONAL,

    terminalLabel TerminalLabel OPTIONAL, -- if not present,
                                                -- it refers to all participants
                                                -- in the conference

    sessionDescription BMPString (SIZE(1..128)), -- Basic ISO/IEC 10646-1 (Unicode)
    dataType CHOICE
    {
        videoData VideoCapability,
        audioData AudioCapability,
        data DataApplicationCapability,
        ...
    },
    mediaChannel TransportAddress OPTIONAL,
    mediaGuaranteedDelivery BOOLEAN OPTIONAL,
    mediaControlChannel TransportAddress OPTIONAL, -- reverse RTCP channel
    mediaControlGuaranteedDelivery BOOLEAN OPTIONAL,
    ...
}

```

```

=====
-- Conference Request definitions
=====

```

```

ConferenceRequest ::=CHOICE
{
    terminalListRequest NULL, -- same as H.230 TCU (term->MC)

    makeMeChair NULL, -- same as H.230 CCA (term->MC)
    cancelMakeMeChair NULL, -- same as H.230 CIS (term->MC)

    dropTerminal TerminalLabel, -- same as H.230 CCD(term->MC)

    requestTerminalID TerminalLabel, -- sames as TCP (term->MC)
}

```

```

enterH243Password          NULL,          -- same as H.230 TCS1 (MC->term)
enterH243TerminalID        NULL,          -- same as H.230 TCS2/TCI
                           -- (MC->term)
enterH243ConferenceID      NULL,          -- same as H.230 TCS3 (MC->term)
...,
enterExtensionAddress      NULL,          -- same as H.230 TCS4 (GW->term)
}

```

```

TerminalLabel              ::=SEQUENCE
{
    mcuNumber               McuNumber,
    terminalNumber           TerminalNumber,
    ...
}

```

```

McuNumber                  ::=INTEGER(0..192)
TerminalNumber              ::=INTEGER(0..192)

```

```

-- =====
-- Conference Response definitions
-- =====

```

```

ConferenceResponse         ::=CHOICE
{
    mCTerminalIDResponse   SEQUENCE          -- response to TCP(same as TIP)
    {                      -- sent by MC only
        terminalLabel       TerminalLabel,
        terminalID          TerminalID,
        ...
    },

    terminalIDResponse      SEQUENCE          -- response to TCS2 or TCI
    {                      -- same as IIS
        terminalLabel       TerminalLabel,  -- (term->MC)
        terminalID          TerminalID,
        ...
    },

    conferenceIDResponse    SEQUENCE          -- response to TCS3
    {                      -- same as IIS
        terminalLabel       TerminalLabel,  -- (term->MC)
        conferenceID       ConferenceID,
        ...
    },

    passwordResponse        SEQUENCE          -- response to TCS1
    {                      -- same as IIS
        terminalLabel       TerminalLabel,  -- (term->MC)
        password           Password,
        ...
    },

    terminalListResponse    SET SIZE (1..256) OF TerminalLabel,

    videoCommandReject      NULL,          -- same as H.230 VCR
    terminalDropReject       NULL,          -- same as H.230 CIR

    makeMeChairResponse     CHOICE          -- same as H.230 CCR
    {
        grantedChairToken   NULL,          -- same as H.230 CIT
    }
}

```

```

        deniedChairToken      NULL,          -- same as H.230 CCR
        ...
    },
    ...,
    extensionAddressResponse  SEQUENCE -- response to TCS4
    {
        extensionAddress      TerminalID, -- same as IIS (term->GW)
        ...
    }
}
TerminalID ::=OCTET STRING (SIZE(1..128)) -- as per H.230
ConferenceID ::=OCTET STRING (SIZE(1..32))
Password ::=OCTET STRING (SIZE(1..32))

```

```

-- =====
-- Command Message definitions
-- =====

```

```

-- =====
-- Command Message: Send Terminal Capability Set
-- =====

```

```

SendTerminalCapabilitySet ::=CHOICE
{
    specificRequest          SEQUENCE
    {
        multiplexCapability  BOOLEAN,

        capabilityTableEntryNumbers SET SIZE (1..65535) OF CapabilityTableEntryNumber
        OPTIONAL,

        capabilityDescriptorNumbers SET SIZE (1..256) OF CapabilityDescriptorNumber OPTIONAL,
        ...
    },
    genericRequest           NULL,
    ...
}

```

```

-- =====
-- Command Message: Encryption
-- =====

```

```

EncryptionCommand ::=CHOICE
{
    encryptionSE             OCTET STRING, -- per H.233, but no error protection
    encryptionIVRequest     NULL,          -- requests new IV
    encryptionAlgorithmID   SEQUENCE
    {
        h233AlgorithmIdentifier SequenceNumber,
        associatedAlgorithm     NonStandardParameter
    },
    ...
}

```

```

=====
-- Command Message: Flow Control
=====

```

```

FlowControlCommand ::=SEQUENCE
{
  scope CHOICE
  {
    logicalChannelNumber LogicalChannelNumber,
    resourceID INTEGER (0..65535),
    wholeMultiplex NULL
  },
  restriction CHOICE
  {
    maximumBitRate INTEGER (0..16777215), -- units 100 bit/s
    noRestriction NULL
  },
  ...
}

```

```

=====
-- Command Message: Change or End Session
=====

```

```

EndSessionCommand ::=CHOICE
{
  nonStandard NonStandardParameter,
  disconnect NULL,
  gstnOptions CHOICE
  {
    telephonyMode NULL,
    v8bis NULL,
    v34DSVD NULL,
    v34DuplexFAX NULL,
    v34H324 NULL,
    ...
  },
  ...
}

```

```

=====
-- Command Message: Conference Commands
=====

```

```

ConferenceCommand ::=CHOICE
{
  broadcastMyLogicalChannel LogicalChannelNumber, -- similar to H.230 MCV
  cancelBroadcastMyLogicalChannel LogicalChannelNumber, -- similar to H.230 Cancel-MCV

  makeTerminalBroadcaster TerminalLabel, -- same as H.230 VCB
  cancelMakeTerminalBroadcaster NULL, -- same as H.230 Cancel-VCB

  sendThisSource TerminalLabel, -- same as H.230 VCS
  cancelSendThisSource NULL, -- same as H.230 cancel VCS

  dropConference NULL, -- same as H.230 CCK
  ...
}

```

```

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

```

```

MiscellaneousCommand ::=SEQUENCE
{
  logicalChannelNumber LogicalChannelNumber,
  type CHOICE
  {
    equalizeDelay NULL, -- same as H.230 ACE
    zeroDelay NULL, -- same as H.230 ACZ
    multipointModeCommand NULL,
    cancelMultipointModeCommand NULL,
    videoFreezePicture NULL,
    videoFastUpdatePicture NULL,

    videoFastUpdateGOB SEQUENCE
    {
      firstGOB INTEGER (0..17),
      numberOfGOBs INTEGER (1..18)
    },

    videoTemporalSpatialTradeOff INTEGER (0..31), -- commands a trade-off value

    videoSendSyncEveryGOB NULL,
    videoSendSyncEveryGOBCancelNULL,

    ...,
    videoFastUpdateMB SEQUENCE
    {
      firstGOB INTEGER (0..255) OPTIONAL,
      firstMB INTEGER (1..8192) OPTIONAL,
      numberOfMBs INTEGER (1..8192),
      ...
    },
    maxH223MUXPDUsize INTEGER(1..65535) -- units octets
  },
  ...
}

```

```

=====
-- Indication Message definitions
=====

```

```

-----
-- Indication Message: Function not understood
-----

```

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

```

FunctionNotUnderstood ::=CHOICE
{
  request RequestMessage,
  response ResponseMessage,
  command CommandMessage
}

```

```

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

```

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

```

FunctionNotSupported ::=SEQUENCE
{
    cause CHOICE
    {
        syntaxError NULL,
        semanticError NULL,
        unknownFunction NULL,
        ...
    },
    returnedFunction OCTET STRING OPTIONAL,
    ...
}

```

```

=====
-- Indication Message: Conference
=====

```

```

ConferenceIndication ::=CHOICE
{
    sbeNumber INTEGER (0..9), -- same as H.230 SBE Number
    terminalNumberAssign TerminalLabel, -- same as H.230 TIA
    terminalJoinedConference TerminalLabel, -- same as H.230 TIN
    terminalLeftConference TerminalLabel, -- same as H.230 TID
    seenByAtLeastOneOther NULL, -- same as H.230 MIV
    cancelSeenByAtLeastOneOther NULL, -- same as H.230 cancel MIV
    seenByAll NULL, -- like H.230 MIV
    cancelSeenByAll NULL, -- like H.230 MIV
    terminalYouAreSeeing TerminalLabel, -- same as H.230 TIN
    requestForFloor NULL, -- same as H.230 TIF
    ...
}

```

```

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

```

```

MiscellaneousIndication ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    type CHOICE
    {
        logicalChannelActive NULL, -- same as H.230 AIA and VIA
        logicalChannelInactive NULL, -- same as H.230 AIM and VIS
        multipointConference NULL,
        cancelMultipointConference NULL,
    }
}

```

multipointZeroComm	NULL,	-- same as H.230 MIZ
cancelMultipointZeroComm	NULL,	-- same as H.230 cancel MIZ
multipointSecondaryStatus	NULL,	-- same as H.230 MIS
cancelMultipointSecondaryStatus	NULL,	-- same as H.230 cancel MIS
videoIndicateReadyToActivate	NULL,	-- same as H.230 VIR
videoTemporalSpatialTradeOff	INTEGER (0..31),	-- indicates current trade-off
...,		
videoNotDecodedMBs	SEQUENCE	
{		
firstMB	INTEGER (1..8192),	
numberOfMBs	INTEGER (1..8192),	
temporalReference	INTEGER (0..255),	
...		
}		
},		
...		
}		

=====
-- Indication Message: Jitter Indication
=====

JitterIndication	::=SEQUENCE
{	
scope	CHOICE
{	
logicalChannelNumber	LogicalChannelNumber,
resourceID	INTEGER (0..65535),
wholeMultiplex	NULL
},	
estimatedReceivedJitterMantissa	INTEGER (0..3),
estimatedReceivedJitterExponent	INTEGER (0..7),
skippedFrameCount	INTEGER (0..15) OPTIONAL,
additionalDecoderBuffer	INTEGER (0..262143) OPTIONAL, -- 262143 is 2 ¹⁸ - 1
...	
}	

=====
-- Indication Message: H.223 logical channel skew
=====

H223SkewIndication	::=SEQUENCE
{	
logicalChannelNumber1	LogicalChannelNumber,
logicalChannelNumber2	LogicalChannelNumber,
skew	INTEGER (0..4095), -- units milliseconds
...	
}	

=====
-- Indication Message: H.225.0 maximum logical channel skew
=====

H2250MaximumSkewIndication	::=SEQUENCE
{	
logicalChannelNumber1	LogicalChannelNumber,

```

    logicalChannelNumber2          LogicalChannelNumber,
    maximumSkew                    INTEGER (0..4095),      -- units milliseconds
    ...
}

=====
-- Indication Message: MC Location Indication
=====

MCLocationIndication              ::=SEQUENCE
{
    signalAddress                  TransportAddress, -- this is the H.323 Call Signalling
                                     -- address of the entity which
                                     -- contains the MC
    ...
}

=====
-- Indication Message: Vendor Identification
=====

VendorIdentification              ::=SEQUENCE
{
    vendor                         NonStandardIdentifier,
    productNumber                  OCTET STRING (SIZE(1..256)) OPTIONAL, -- per vendor
    versionNumber                  OCTET STRING (SIZE(1..256)) OPTIONAL, -- per productNumber
    ...
}

=====
-- Indication Message: New ATM virtual channel indication
=====

NewATMVCIndication                ::=SEQUENCE
{
    resourceID                     INTEGER(0..65535),
    bitRate                        INTEGER(1..65535), -- units 64 kbit/s
    bitRateLockedToPCRClock        BOOLEAN,
    bitRateLockedToNetworkClock    BOOLEAN,
    aal                             CHOICE
    {
        aal1                        SEQUENCE
        {
            clockRecovery            CHOICE
            {
                nullClockRecovery    NULL,
                srtsClockRecovery     NULL,
                adaptiveClockRecovery NULL,
                ...
            },
            errorCorrection           CHOICE
            {
                nullErrorCorrection   NULL,
                longInterleaver       NULL,
                shortInterleaver      NULL,
                errorCorrectionOnly    NULL,
                ...
            },
            structuredDataTransfer    BOOLEAN,
        }
    }
}

```

```

        partiallyFilledCells    BOOLEAN,
        ...
    },
    aal5                        SEQUENCE
    {
        forwardMaximumSDUSize  INTEGER (0..65535), -- units octets
        backwardMaximumSDUSize INTEGER (0..65535), -- units octets
        ...
    },
    ...
},
multiplex                      CHOICE
{
    noMultiplex                NULL,
    transportStream           NULL,
    programStream             NULL,
    ...
},
...
}

```

```

-- =====
-- Indication Message: user input
-- =====

```

```

UserInputIndication           ::=CHOICE
{
    nonStandard                NonStandardParameter,
    alphanumeric              GeneralString,
    ...,
    userInputSupportIndication CHOICE
    {
        nonStandard           NonStandardParameter,
        basicString           NULL,
        iA5String             NULL,
        generalString         NULL,
        ...
    }
}

```

END

7 Messages: définitions sémantiques

Le présent sous-paragraphe donne les définitions sémantiques et les contraintes applicables aux éléments de syntaxe définis dans la section précédente.

MultimediaSystemControlMessage: choix de types de messages. Les messages définis dans la présente Recommandation sont classés comme messages de demande, de réponse, de commande et d'indication d'état.

RequestMessage: un message de demande induit à une action du terminal distant et requiert une réponse immédiate de celui-ci. Le message nonStandard peut être utilisé pour envoyer des demandes non normalisées.

ResponseMessage: un message de réponse est émis en réponse à un message de demande. Le message nonStandard peut être utilisé pour transmettre des réponses non normalisées.

CommandMessage: un message de commande nécessite d'entreprendre une action mais n'implique pas de réponse explicite. Le message nonStandard peut être utilisé pour envoyer des commandes non normalisées.

IndicationMessage: une indication contient des informations qui ne nécessitent pas d'action ou de réponse. Le message nonStandard peut être utilisé pour envoyer des indications non normalisées.

NonStandardParameter: cela peut être utilisé pour indiquer un paramètre non normalisé et comprend un code d'identification et les paramètres réels qui sont codés comme une chaîne d'octets.

NonStandardIdentifieur: utilisé pour identifier un type de paramètre non normalisé. Il est utilisé soit comme un identificateur d'objet, soit comme un identificateur conforme à la Recommandation H.221 représenté par une chaîne d'octets comprenant exactement quatre octets qui sont l'indicatif de pays (octet 1 selon la Recommandation T.35 [22]; octet 2*), le code de fabricant (les deux octets suivants*) *= affectation nationale. Les codes de fabricant sont les mêmes que ceux qui sont attribués pour les besoins de la Recommandation H.320 [17]. Les identificateurs non normalisés H.245 peuvent être du type "objet" ou du type "h.221nonStandard" à la discrétion du fabricant définissant le message non normalisé, étant donné que les IDENTIFICATEURS D'OBJET et les messages h221NonStandard proviennent d'espaces disjoints qui ne peuvent pas être confondus. Cependant, étant donné que les messages h221Non Standard sont également utilisés par H.320, de tels messages proviennent du même espace que les messages H.320 et devront avoir la même signification.

7.1 Messages de choix du mode maître ou esclave

Cet ensemble de messages est utilisé par un protocole visant à choisir quel terminal est le terminal maître et quel autre terminal est le terminal esclave.

7.1.1 Choix du mode maître ou esclave

Ce message est envoyé d'une entité MSDSE vers une entité MSDSE homologue.

TerminalType est un numéro qui identifie différents types de terminaux tels que les terminaux, les ponts de conférence et les passerelles. L'attribution de valeurs vers les types de terminaux est hors du domaine d'application de la présente Recommandation.

Le nombre statusDeterminationNumber est un nombre aléatoire dans le domaine $0 \dots 2^{24} - 1$.

7.1.2 Acquiescement du choix de mode maître ou esclave

Ceci est utilisé pour confirmer quel est le terminal maître ou quel est le terminal esclave, comme cela est indiqué par la décision. Quand la décision est du type maître, le terminal recevant ce message est le terminal maître et quand la décision est du type esclave, c'est le terminal esclave.

7.1.3 Refus du choix du mode maître ou esclave

Ceci est utilisé pour refuser le message MasterSlaveDetermination. Quand la raison est du type identicalNumbers, le refus était dû à des nombres aléatoires qui sont équivalents et aux types de terminaux qui sont les mêmes.

7.1.4 Libération sur temporisation du choix de mode maître ou esclave

Ceci est envoyé dans le cas d'une temporisation.

7.2 Messages de capacités des terminaux

Cet ensemble de messages permet d'assurer l'échange des capacités entre les deux terminaux.

7.2.1 Aperçu général

Le terminal émetteur attribue un numéro dans un tableau capabilityTable pour chaque mode individuel dans lequel le terminal peut fonctionner. Par exemple, l'audio selon la Recommandation G.723.1, l'audio selon la Recommandation G.728 et le format CIF de la vidéo selon H.263 se verraient attribuer des numéros distincts.

Ces numéros de capacités sont groupés dans des structures AlternativeCapabilitySet. Chaque AlternativeCapabilitySet indique que le terminal peut fonctionner dans exactement un mode énuméré dans l'ensemble. Par exemple une liste AlternativeCapabilitySet {G.711 G.723.1, G.728} signifie que le terminal peut fonctionner dans un et un seul mode audio, quel qu'il soit.

Ces structures AlternativeCapabilitySet sont groupées en structures simultaneousCapabilities. Chaque structure simultaneousCapabilities indique un ensemble de mode que le terminal peut utiliser simultanément. Par exemple, une structure simultaneousCapability contenant les deux structures AlternativeCapabilitySet {H.261, H.263} et {G.711, G.723.1, G.728} signifie que le terminal peut faire fonctionner simultanément l'un ou l'autre des codecs vidéo avec n'importe lequel des codecs audio. L'ensemble simultaneousCapabilities {{H.261}, {H.261, H.263}, {G.711, G.723.1, G.728}} signifie que le terminal peut faire fonctionner simultanément deux canaux vidéo et un canal audio: un canal vidéo selon la Recommandation H.261, un autre canal vidéo selon les Recommandations H.261 ou H.263, et un canal audio selon une des Recommandations G.711, G.723.1 ou G.728.

NOTE – Les capacités réelles enregistrées dans le tableau de capacités capabilityTable sont souvent plus complexes que celles présentées ici. Par exemple, chaque capacité H.263 indique des détails incluant la possibilité d'utiliser différents formats d'image à des intervalles d'image minimaux donnés, de même que la possibilité d'utiliser des modes de codage facultatifs.

Les capacités totales du terminal sont décrites par un ensemble de structures CapabilityDescriptor, chacune d'elle étant une structure simultaneousCapabilities unique et un numéro capabilityDescriptorNumber. En envoyant plus d'un descripteur CapabilityDescriptor, le terminal peut signaler les dépendances entre les modes de fonctionnement en décrivant différents ensembles de modes qu'il peut simultanément utiliser. Par exemple, un terminal disposant de deux structures CapabilityDescriptors, l'une {{H.261, H.263}, {G.711, G.723.1, G.728}} comme dans l'exemple antérieur et l'autre {{H.262}, {G.711}} signifie que le terminal peut également faire fonctionner le codec vidéo H.262, mais uniquement avec un codec audio G.711 de faible complexité.

Les terminaux peuvent ajouter dynamiquement des capacités durant une session de communication en ajoutant des structures supplémentaires CapabilityDescriptor, ou en retirant des capacités en envoyant des structures révisées CapabilityDescriptor. Tous les terminaux devront transmettre au moins une structure CapabilityDescriptor.

7.2.2 Ensemble de capacités des terminaux

Ce message contient des informations sur les capacités du terminal à fonctionner en émission et réception. Il indique également la version de la présente Recommandation qui est utilisée. Il est envoyé d'une entité CESE sortante vers une entité CESE homologue entrante.

Le paramètre sequenceNumber est utilisé pour désigner des instances d'ensembles TerminalCapabilitySet de sorte que la réponse correspondante puisse être identifiée.

Le paramètre protocolIdentifier est utilisé pour indiquer la version de la présente Recommandation qui est utilisée. L'Annexe A énumère les identificateurs d'objet définis en vue de leur utilisation dans la présente Recommandation.

Le paramètre `multiplexCapability` indique les capacités relatives au multiplexage et à la couche d'adaptation au réseau. Un terminal devra inclure le paramètre `multiplexCapability` dans le premier ensemble `TerminalCapabilitySet` envoyé.

Le paramètre `V75Capability` indique les capacités de l'entité de commande selon V.75. Le paramètre `audioHeader` indique la capacité de l'en-tête audio selon V.75.

7.2.2.1 Tableau de capacités

Un tableau de capacités est une liste numérotée de capacités. Un terminal devra pouvoir effectuer tout ce qui est énuméré dans son tableau de capacités, mais ne devra pas nécessairement pouvoir réaliser plus d'une de ces capacités.

Un ensemble `TerminalCapabilitySet` peut contenir zéro entrée de tableau `CapabilityTableEntry` ou davantage. Au début, aucune entrée de tableau n'est définie. Quand une entrée `CapabilityTableEntry` est reçue, elle remplace l'entrée `CapabilityTableEntry` préalablement reçue avec le même numéro `CapabilityTableEntryNumber`. Une entrée `CapabilityTableEntry` sans capacité peut être utilisée pour éliminer l'entrée `CapabilityTableEntry` préalablement reçue avec le même numéro `CapabilityTableEntryNumber`.

7.2.2.2 Descripteurs de capacité

Les descripteurs `CapabilityDescriptors` sont utilisés pour indiquer les capacités de fonctionnement du terminal en émission et réception. Chaque descripteur de capacité `CapabilityDescriptor` est une déclaration indépendante sur les capacités du terminal.

Le numéro `capabilityDescriptorNumber` est utilisé pour numéroter les descripteurs de capacités `CapabilityDescriptors`. Si un terminal a une préférence pour le mode dans lequel il voudrait émettre ou recevoir, et s'il veut l'exprimer lorsqu'il transmet ses capacités, il peut le faire en donnant des valeurs faibles aux numéros `capabilityDescriptorNumber` des descripteurs `CapabilityDescriptors` qui correspondent au mode ou aux modes préférentiels.

L'ensemble `simultaneousCapabilities` est un ensemble d'ensembles `AlternativeCapabilitySet`. Cet ensemble est utilisé pour énumérer les capacités simultanées du terminal.

Un ensemble `AlternativeCapabilitySet` est une séquence de numéros `CapabilityTableEntryNumbers`. Seules les entrées `CapabilityTableEntry`s qui ont été définies devront être présentes dans un ensemble `AlternativeCapabilitySet`, bien qu'il soit possible de définir des entrées `CapabilityTableEntry`s et de s'y référer dans le même ensemble `TerminalCapabilitySet`. Si un terminal a une préférence pour le mode dans lequel il aimerait émettre ou recevoir, et s'il veut l'exprimer lorsqu'il émet ses capacités, il peut le faire en énumérant les éléments dans `AlternativeCapabilitySets` par ordre de préférence décroissante.

Un terminal doit être à même de mettre en œuvre simultanément n'importe quelle capacité provenant de chaque ensemble `AlternativeCapabilitySet` énuméré dans `simultaneousCapabilities`.

Au moins un descripteur de capacités devra avoir la structure suivante: Il devra y avoir au moins un ensemble `AlternativeCapabilitySet` contenant seulement les possibilités d'un type de support unique pour chaque type de support de communication que le terminal peut utiliser. Ceci est destiné à assurer que le terminal distant peut choisir un mode de transmission incluant au moins une instance de chaque type de support de communication que le récepteur peut utiliser.

NOTE 1 – La répétition d'une capacité dans un ensemble `AlternativeCapabilitySet` est superflue et ne donne pas d'autres informations, alors que la répétition d'une capacité dans différents ensembles `AlternativeCapabilitySets` pour le même `CapabilityDescriptor` indique la possibilité d'une instance supplémentaire, simultanée de la capacité donnée.

NOTE 2 – Les terminaux qui ne peuvent pas modifier l'attribution de ressources peuvent indiquer complètement leurs capacités en utilisant un descripteur CapabilityDescriptor unique.

7.2.2.3 Capacités

Les choix receiveVideoCapability, receiveAudioCapability et receiveDataApplicationCapability indiquent la capacité de recevoir selon les capacités qui sont respectivement VideoCapability, AudioCapability et DataApplicationCapability.

Les choix transmitVideoCapability, transmitAudioCapability et transmitDataApplicationCapability indiquent la capacité d'émettre selon les capacités qui sont respectivement VideoCapability, AudioCapability et DataApplicationCapability.

Les choix receiveAndTransmitVideoCapability, receiveAndTransmitAudioCapability et receiveAndTransmitDataApplicationCapability indiquent la capacité de recevoir et d'émettre selon les capacités qui sont respectivement VideoCapability, AudioCapability et DataApplicationCapability. Ces codages peuvent être utilisés pour indiquer que les capacités d'émission et de réception ne sont pas indépendantes.

Le paramètre booléen h233EncryptionTransmitCapability, quand il est mis à la valeur "vrai", indique que le terminal assure le chiffrement selon les Recommandations H.233 [11] et H.234 [12].

Le temps de réponse h233IVResponseTime se mesure en millisecondes et il indique le temps minimal que le récepteur demande à l'émetteur d'attendre après que la transmission d'un message IV soit achevée avant de commencer à utiliser le nouveau vecteur IV. Le moyen de transmettre le vecteur IV n'est pas défini dans la présente Recommandation.

ConferenceCapability indique des capacités de conférence telles que la possibilité d'utiliser le mode président de séance tel qu'il est décrit dans la Recommandation H.243.

7.2.2.4 Capacités de multiplexage

La structure MultiplexCapability contient des capacités qui se rapportent au multiplexage et à l'interface réseau. Un terminal doit envoyer le paramètre MultiplexCapability dans le premier TerminalCapabilitySet envoyé. Sauf indication contraire, ces capacités sont des capacités de réception.

H222Capability: indique les capacités de multiplexage et d'interface réseau qui sont spécifiques au multiplex défini dans la Recommandation H.222.1 [7].

Le paramètre numberOfVCs indique combien de voies virtuelles (VC, *virtual channel*) ATM simultanées peuvent être utilisées par le terminal. Cela comprend toutes les voies virtuelles qui véhiculent les données conformément à H.245, T.120, DSM-CC ou tout autre type de données, et tous les circuits virtuels qui contiennent les informations audiovisuelles, mais ne comprend pas la voie virtuelle (VC) utilisée pour la signalisation selon la Recommandation Q.2931 [20].

Le paramètre vcCapability est un ensemble dont la taille est égale à la valeur numberOfVCs, qui indique les capacités présentes pour chaque voie virtuelle (VC) disponible.

La séquence aal1 indique, le cas échéant, quelle est la capacité et quelles sont les options acceptées pour la couche d'Adaptation 1 de l'ATM, comme cela est spécifié dans la Recommandation I. 363 [19]. Les codages sont définis dans le Tableau 1.

Tableau 1/H.245 – Codage 1 de la couche d'Adaptation de l'ATM

Codage ASN.1	Signification sémantique du codage
nullClockRecovery	pas de mécanisme de récupération de la fréquence d'horloge de la source: transport sur circuit synchrone
srtsClockRecovery	mécanisme de récupération de la fréquence d'horloge de la source par horodatage résiduel synchrone
adaptativeClockRecovery	mécanisme adaptatif de récupération de la fréquence d'horloge de la source
nullErrorCorrection	aucune correction d'erreurs n'est assurée
longInterleaver	la méthode de correction d'erreurs sans voie de retour pour le transport de signaux sensibles aux pertes est disponible
shortInterleaver	la méthode de correction d'erreurs sans voie de retour pour le transport de signaux sensibles aux retards est disponible
errorCorrectionOnly	la méthode de correction d'erreurs sans voie de retour sans entrelacement des cellules est disponible
structuredDataTransfer	le transfert des données structurées est pris en charge
partiallyFilledCells	les cellules remplies partiellement sont prises en charge

La séquence aal5 indique, le cas échéant, quelles capacités de la couche d'Adaptation 5 de l'ATM et quelles options sont assurées, comme cela est spécifié dans la Recommandation I.363 [19]. Les tailles forwardMaximumSDUSize et backwardMaximumSDUSize indiquent la taille maximale de paquets CPCS-SDU dans les sens avant et arrière, mesurée en octets. L'une des couches Adaptation aal1 ou aal5, voire les deux couches, doivent être présentes.

Les variables booléennes transportStream et programStream, quand elles sont mises à "vrai", indiquent la possibilité de prendre en compte respectivement les multiplex du flux de transport et du flux de programme [6].

La variable availableBitRates indique les capacités en débit pour la voie virtuelle. C'est une séquence de différents débits qui peut être utilisée, mesurée en unités de 64 kbit/s. Les débits sont énumérés par ordre décroissant, à partir du débit utilisable le plus élevé. Les débits utilisés peuvent être indiqués comme des valeurs distinctes en utilisant le paramètre singleBitRate de champ, ou comme une gamme de débits rangeOfBitRates allant du débit lowerBitRate au débit HigherBitRate, ce qui indique que toutes les valeurs comprises entre la limite inférieure et la limite supérieure, y compris les limites elles-mêmes, sont utilisables. Les débits indiqués sont mesurés au niveau de la couche AAL-SAP.

H223Capability: indique les capacités spécifiques au multiplex selon H.223 [8].

La variable booléenne transportWithI-frames, quand elle est mise à "vrai", indique que le terminal peut envoyer et recevoir des messages sur les canaux de commande en utilisant des trames I du protocole LAPM comme cela est défini dans la Recommandation V.42 [29].

Les variables booléennes videoWithAL1, videoWithAL2, videoWithAL3, audioWithAL1, audioWithAL2, audioWithAL3, dataWithAL1, dataWithAL2 et dataWithAL3, quand elles sont mises à "vrai", indiquent la capacité de recevoir le type de support de communication donné (vidéo, audio ou données) en utilisant la couche d'Adaptation donnée (AL1, AL2, ou AL3).

Les nombres entiers maximumA12SDUSize et maximumA13SDUSize indiquent le nombre maximal d'octets dans chaque SDU que le terminal peut recevoir en utilisant la couche d'Adaptation de type 2 et de type 3 respectivement.

Le nombre maximumDelayJitter indique la gigue maximale de multiplexage crête à crête que l'émetteur va engendrer. Elle est mesurée en millisecondes. La gigue de multiplexage est définie comme la différence de temps entre le moment où le premier octet d'une trame audio est délivré dans le flux multiplexé et le moment où il serait délivré sur un canal à débit constant ne comportant pas de multiplex.

h223MultiplexTableCapability: indique les capacités du terminal à recevoir et traiter des entrées de tableau de multiplexage

Le paramètre "de base" indique que le multiplex ne peut recevoir que les descripteurs MultiplexEntryDescriptors de base tels qu'ils sont définis dans la Recommandation H.223 [8].

Le paramètre "amélioré" indique que le multiplex peut recevoir des descripteurs MultiplexEntryDescriptors améliorés avec les paramètres additionnels définis ci-dessous.

Le paramètre maximumNestingDepth indique la profondeur d'imbrication maximale sur les champs subElementList invoqués récursivement. Les descripteurs multiplexEntryDescriptors qui n'utilisent pas le champ subElementList seront considérés comme ayant un ordre d'imbrication nul.

Le paramètre maximumElementListSize indique le nombre maximal des champs dans la SÉQUENCE ASN.1.

Le paramètre maximumSubElementListSize indique le nombre maximal de sous-élément dans la liste subElementList.

Le paramètre booléen maxMUXPDUSizeCapability, quand il est à "vrai" indique que l'émetteur est capable de limiter la taille transmise des paquets MUX-PDU selon H.223. Il n'a pas de signification lorsqu'il fait partie d'une capacité de réception.

V76Capability: indique des capacités propres au multiplex défini dans la Recommandation V.76.

Le paramètre suspendResumeCapabilitywAddress indique la capacité de mettre en œuvre la procédure d'interruption/reprise avec un champ d'adresse. Le paramètre suspendResumeCapabilitywoAddress indique la capacité de mettre en œuvre la procédure d'interruption/reprise V76 sans champ d'adresse.

Le paramètre rejCapability indique la capacité de la fonction de correction d'erreurs du multiplex V.76 à mettre en œuvre des refus.

Le paramètre sREJCapability indique la capacité de la fonction de correction d'erreurs du multiplex V.76 à mettre en œuvre des refus sélectifs.

Le paramètre mREJCapability indique la capacité de la fonction de correction d'erreurs du multiplex V.76 à mettre en œuvre plusieurs refus sélectifs.

Le paramètre crc8bitCapability correspond à la capacité du multiplex à utiliser des CRC de 8 bits.

Le paramètre crc16bitCapability correspond à la capacité du multiplex à utiliser des CRC de 16 bits.

Le paramètre crc32bitCapability correspond à la capacité du multiplex à utiliser des CRC de 32 bits.

Le paramètre uihCapability indique la possibilité de mise en œuvre de trames UIH selon V.76.

Le paramètre numOfDLCs indique le nombre de voies DLC que le multiplex peut mettre en œuvre.

Le paramètre twoOctetAddressFieldCapability indique la possibilité d'un multiplex V.76 à mettre en œuvre un champ d'adresse de deux octets.

Le paramètre loopBackTestCapability indique la mise en œuvre de boucle selon la Recommandation V.76. Le paramètre n401Capability indique la valeur maximale du nombre N401 décrit dans la

Recommandation V.76. Le paramètre `maxWindowSizeCapability` indique la taille maximale de la fenêtre que le multiplex V.76 peut mettre en œuvre.

H2250Capability: indique des capacités propres à la couche de mise en paquets des supports d'information selon H.225.0.

Le paramètre `maximumAudioDelayJitter` indique la valeur de remise crête à crête maximale des paquets du signal audio vers la couche Transport que l'émetteur produira. Il se mesure en millisecondes.

Le paramètre `receiveMultipointCapability` indique les capacités de réception d'un terminal dans une conférence multipoint.

Le paramètre `transmitMultipointCapability` indique les capacités d'émission d'un terminal dans une conférence multipoint.

Le paramètre `receiveAndTransmitMultipointCapability` indique les capacités de réception et d'émission d'un terminal dans une conférence multipoint.

Le paramètre `mcCapability` indique la possibilité d'un terminal à se comporter en entité de commande multipoint (MC, *multipoint control entity*) dans une conférence à architecture centralisée ou répartie.

Le paramètre `rtcpVideoControlCapability` indique la possibilité de traitement des deux messages acquittement négatif (NACK, *negative acknowledgement*) et demande de rafraîchissement intra complet (FIR, *full intra request*) du protocole RTCP à la fois dans un terminal.

La paramètre `MediaPacketizationCapability` indique quel algorithme de mise en paquets facultative du support de communications est employé, le cas échéant.

Le paramètre `h261aVideoPacketization` indique que le format de remplacement faisant appel au protocole RTP pour les données vidéo H.261, décrit dans la Recommandation H.225.0 est utilisé.

Le paramètre `MultipointCapability` indique les capacités d'un terminal inhérentes aux conférences multipoint.

Le paramètre `multicastCapability` indique la capacité d'un terminal de diffuser les trafics audio et vidéo à plusieurs destinataires.

Le paramètre `multiUniCastConference` indique la capacité d'un terminal de participer à une conférence faisant appel à des transmissions dédiées vers plusieurs destinataires.

Le paramètre `MediaDistributionCapability` indique les capacités en émission et en réception d'un support de transmission d'un terminal dans une conférence multipoint. Les paramètres "centralizedAudio" et "centralizedVidéo" seront mis à "VRAI" pour les terminaux H.323. Si la vidéo est mise en œuvre, le paramètre "centralizedVidéo" sera mis à "VRAI". Si les protocoles T.120 sont mis en œuvre, la capacité "Centralized Data T.120 Data Application" sera présente.

Les paramètres audio, commande et vidéo, de type centralisé ou réparti, indiquent la capacité d'un terminal à participer à une conférence pour laquelle ce ou ces types est ou sont utilisés par les supports de communication. Les paramètres "centralizedData" et "distributedData" indiquent la capacité d'un terminal à participer à une conférence avec ces types d'architecture de transmission des supports d'information par leurs protocoles "Data Application Protocol" dédiés. La capacité `MediaDistributionCapability` est une séquence pour permettre la déclaration de capacités simultanées (c'est-à-dire l'audio de type centralisé avec la vidéo de type réparti ou la vidéo de type centralisé avec l'audio de type réparti, ou des capacités données propres selon le protocole "Data Application Protocol").

7.2.2.5 Capacités vidéo

Ceci indique des capacités à établir des communications vidéo. L'indication de plus d'une capacité unique à l'intérieur d'une structure VideoCapability unique ne fournit aucune indication sur la possibilité de traitements simultanés. La possibilité de traitements simultanés peut être indiquée par des instances de structures VideoCapability dans des ensembles AlternativeCapabilitySets avec un descripteur CapabilityDescriptor unique.

H.261VideoCapability: indique des capacités selon la Recommandation H.261 [13].

Le nombre qcifMPI indique, le cas échéant, l'intervalle minimal entre images au pas de 1/29,97 pour le codage et le décodage des images de format QCIF, et s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format QCIF.

Le nombre cifMPI indique, le cas échéant, l'intervalle minimal entre images au pas de 1/29,97 pour le codage et le décodage des images de format CIF, et s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format CIF.

La variable booléenne temporalSpatialTradeOffCapability, quand elle est mise à "vrai", indique que le codeur peut modifier son compromis entre les résolutions temporelle et spatiale comme l'exige le terminal distant. Cela n'a pas de signification quand cela fait partie d'une capacité de réception.

Le paramètre maxBitRate indique le débit maximal auquel l'émetteur peut émettre ou le récepteur peut recevoir de la vidéo par pas de 100 bit/s.

Le paramètre stillImageTransmission indique la capacité de mise en œuvre d'images fixes telles qu'elles sont spécifiées à l'Annexe D de la Recommandation H.261.

H.262VideoCapability: indique les capacités H.262 [14].

La liste des variables booléennes indique la capacité de traitement de certains profils et de certains niveaux: une valeur mise à "vrai" indique qu'une telle opération est possible, alors qu'une valeur mise à "faux" indique qu'une telle opération n'est pas possible. Un codeur doit produire des trains de bits conformes aux spécifications d'un profil et d'un niveau pour lesquels il a indiqué une capacité, mais également à l'intérieur des limites imposées par les champs optionnels (voir ci-dessous). Un décodeur doit pouvoir accepter tous les trains de bits conformes au profil et au niveau pour lesquels il a indiqué une capacité, pour autant qu'ils soient à l'intérieur des limites indiquées par les champs optionnels. Les champs optionnels sont des nombres entiers dont les unités sont définies dans le Tableau 2.

Tableau 2/H.245 – Unités pour les codages selon H.262

Codage ASN.1	Unités pour le paramètre référencé
videobitRate	400 bits par seconde
vbvBufferSize	16 384 bits
samplesPerLine	échantillons par ligne
linesPerFrame	lignes par image
framesPerSecond	Index, frame_rate_code, dans le Tableau 6-4/H.262
luminanceSampleRate	échantillons par seconde

H.263VideoCapability: indique les capacités H.263 [15].

Le nombre `sqcifMPI` indique, le cas échéant, l'intervalle minimal d'images au pas de 1/29,97 pour le codage et le décodage des images de format SQCIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format SQCIF.

Le nombre `qcifMPI` indique, le cas échéant, l'intervalle minimal d'images au pas de 1/29,97 pour le codage et le décodage des images de format QCIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format QCIF.

Le nombre `cifMPI` indique, le cas échéant, l'intervalle minimal d'images au pas de 1/29,97 pour le codage et le décodage des images de format CIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format CIF.

Le nombre `cif4MPI` indique, le cas échéant, l'intervalle minimal d'images au pas de 1/29,97 pour le codage et le décodage des images de format 4CIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format 4CIF.

Le format `cif16MPI` indique, le cas échéant, l'intervalle minimal d'images au pas de 1/29,97 pour le codage et le décodage des images de format 16CIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format 16CIF.

Le paramètre `maxBitRate` indique le débit maximal auquel l'émetteur peut transmettre la vidéo ou auquel le récepteur peut recevoir de la vidéo par multiples de 100 bit/s.

Les paramètres booléens `unrestrictedVector`, `arithmeticCoding`, `advancedPrediction` et `pbFrames`, quand ils sont mis à "vrai", indiquent la capacité de transmettre et de recevoir ces modes optionnels définis dans les annexes de la Recommandation H.263.

Le paramètre booléen `temporalSpatialTradeOffCapability`, quand il est mis à "vrai", indique que le codeur peut faire varier son compromis entre les résolutions temporelle et spatiale comme l'exige le terminal distant. Ce paramètre n'a pas de signification quand il fait partie des capacités de réception.

Le nombre entier `hrd-B` indique, le cas échéant, le paramètre B du décodeur HRD et se mesure par pas de 128 bits. Quand il n'est pas présent, la valeur par défaut définie dans la Recommandation H.263 est applicable. C'est une capacité de réception et cette capacité n'a pas de signification dans des ensembles de capacités de transmission.

Le nombre entier `bppMaxKb`, indique le cas échéant, le nombre maximal de bit pour une image codée que le récepteur prend correctement en réception et en traitement, et se mesure par pas de 1024 bits. Quand il n'est pas présent, la valeur par défaut définie dans la Recommandation H.263 est applicable. C'est une capacité de réception et cette capacité n'a pas de signification dans des ensembles de capacités de transmission.

Les capacités suivantes sont prévues pour l'utilisation dans certaines applications à très faible fréquence image telles que les applications de vidéosurveillance:

Le nombre `slowSqcifMPI` indique, le cas échéant, l'intervalle minimal d'images en multiples de secondes par trame pour le codage et le décodage des images de format SQCIF. S'il n'est pas présent et si le nombre `sqcifMPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format SQCIF. Si le nombre `sqcifMPI` est présent, le nombre `slowSqcifMPI` ne devra pas être présent.

Le nombre `slowQcifMPI` indique, le cas échéant, l'intervalle minimal d'images en multiples de secondes par trame pour le codage et le décodage des images de format QCIF. S'il n'est pas présent et si le nombre `qcifMPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format QCIF. Si le nombre `qcifMPI` est présent, le nombre `slowQcifMPI` ne devra pas être présent.

Le nombre `slowCifMPI` indique, le cas échéant, l'intervalle minimal d'images en multiples de secondes par trame pour le codage et le décodage des images de format CIF. S'il n'est pas présent et

si le nombre cifMPI n'est pas présent, aucune capacité n'est indiquée pour les images de format CIF. Si le nombre cifMPI est présent, le nombre slowCifMPI ne devra pas être présent.

Le nombre slowCif4MPI indique, le cas échéant, l'intervalle minimal d'images en multiples de secondes par trame pour le codage et le décodage des images de format 4CIF. S'il n'est pas présent et si le nombre cif4MPI n'est pas présent, aucune capacité n'est indiquée pour les images de format 4CIF. Si le nombre cif4MPI est présent, le nombre slowcif4MPI ne devra pas être présent.

Le format slowCif16MPI indique, le cas échéant, l'intervalle minimal d'images en multiples de secondes par trame pour le codage et le décodage des images de format 16CIF. S'il n'est pas présent et si le nombre cif16MPI n'est pas présent, aucune capacité n'est indiquée pour les images de format 16CIF. Si le nombre cif16MPI est présent, le nombre slowCif16MPI ne devra pas être présent.

Le paramètre booléen errorCompensation, quand il est mis à "vrai", indique la capacité de transmettre et de recevoir une information sous forme de contre-réaction pour la compensation d'erreurs telle que l'illustre l'Appendice I/H.263. Quand il appartient à une capacité de transmission, il indique la capacité du codeur à effectuer le traitement des indications videoNotDecodedMBs et de compensation des erreurs. Quand il appartient à une capacité de réception, il indique la capacité du décodeur à effectuer la détection des macroblocs (MB) erronées, les traite comme s'ils n'étaient pas codés et envoie les indications videoNotDecodedMB appropriées.

Les valeurs des intervalles MPI sont applicables quand l'ensemble de tous les modes facultatifs, pour lesquels une capacité est indiquée, est utilisé, de même que lorsque n'importe quelle combinaison de ces modes est utilisée. Un terminal peut signaler la capacité MPI correspondant à un intervalle plus petit quand certaines options ne sont pas utilisées, en transmettant une autre structure VideoCapability comprenant cet intervalle MPI plus petit et indiquant l'ensemble réduit d'options.

IS11172 VideoCapability: indique les capacités IS11172 [33]

Le paramètre constrainedBitstream indique la capacité de tenir compte des trains de bits dans lesquels le fanion constrained_parameters est mis à "1": une valeur mise à vrai indique qu'une telle opération est possible, alors que la valeur mise à "faux" indique qu'une telle opération n'est pas possible. Un codeur devra produire des trains de bits respectant les limites imposées par les champs optionnels (voir ci-dessous). Un décodeur devra pouvoir accepter tous les trains binaires respectant les limites indiquées par les champs optionnels. Les champs optionnels sont des nombres entiers dont les unités sont définies dans le Tableau 3.

Tableau 3/H.245 – Unités pour le codage IS11172-2

Codage ASN.1	Unités pour le paramètre référencé
videoBitRate	400 bit/s
vbvBufferSize	16 384 bits
samplesPerLine	échantillons par ligne
linesPerFrame	lignes par image
pictureRate	référence au 2.4.3.2 ISO/CEI 11172-2
luminanceSampleRate	échantillons par seconde

7.2.2.6 Capacités audio

Cela indique les capacités pour l'audio. L'indication de plus d'une capacité unique dans une structure AudioCapability unique n'indique pas une capacité de traitements simultanés. Une capacité de traitements simultanés peut être indiquée par des instances de capacité AudioCapability dans

différents ensembles AlternativeCapabilitySets d'un descripteur unique de capacité CapabilityDescriptor.

La capacité d'émettre et de recevoir l'audio de la série G est indiquée par un choix de nombres entiers. Quand un multiplex H.222.1 est utilisé, ces nombres se réfèrent à la taille disponible de la mémoire tampon STD, comptée en multiples de 256 octets. Quand un multiplex H.223 est utilisé, ces nombres se réfèrent au nombre maximal de trames audio par paquet AL-SDU. Quand un multiplex H.225.0 est utilisé, ces nombres se réfèrent au nombre maximal de trames audio par paquet. La signification exacte des codages est indiquée dans le Tableau 4.

Tableau 4/H.245 – Codages de l'audio de la série G

Codage ASN.1	Signification sémantique du codage
g711Alaw64k	audio G.711 à 64 kbit/s, loi A
g711Alaw56k	audio G.711 à 56 kbit/s, loi A tronquée jusqu'à 7 bits
g711Ulaw64k	audio G.711 à 64 kbit/s, loi μ
g711Ulaw56k	audio G.711 à 56 kbit/s, loi μ tronquée à 7 bits
g722-64k	audio G.722, 7 kHz à 64 kbit/s
g722-56k	audio G.722, 7 kHz à 56 kbit/s
g722-48k	audio G.722, 7 kHz à 48 kbit/s
g7231	G.723.1 soit à 5,3 soit à 6,3 kbit/s
g728	audio G.728 à 16 kbit/s
g729	audio G.729 à 8 kbit/s
g729AnnexA	audio selon l'Annexe A de G.729 à 8 kbit/s
g729wAnnexB	audio G.729 à 8 kbit/s avec codage des périodes de silences selon l'Annexe B
g729AnnexAwAnnexB	audio selon l'Annexe A de G.729 à 8 kbit/s avec suppression des silences selon l'Annexe B
g7231AnnexCCapability	G.723.1 avec l'Annexe C

G7231: indique la capacité de traitement selon G.723.1 par le codec audio. Le paramètre maxAI-sduAudioFrames indique le nombre maximal de trames audio par paquet AL-SDU. Le paramètre booléen silenceSupression, quand il est mis à vrai, indique la capacité d'utiliser le codage des silences défini dans l'Annexe A/G.723.1.

G.7231AnnexCCapability: indique la capacité de traitement selon G.723.1 avec son Annexe C par le codec audio . Le paramètre maxAI-sduAudioFrames indique le nombre maximal de trames audio par paquet AL-SDU. Le paramètre booléen silenceSupression, quand il est mis à vrai, indique la capacité d'utiliser le codage des silences défini dans l'Annexe A/G.723.1. Le paramètre g723AnnexCAudioMode ne devra pas être présent quand le paramètre G7231AnnexCCapability est inclus dans un message TerminalCapabilitySet, mais devra être présent quand le paramètre G7231AnnexCCapability est inclus dans le message OpenLogicalChannel. Les champs highRateMode0, highRateMode1, lowRateMode0, lowRateMode1, sidMode0, et sidMode1, indiquent le nombre d'octets par trame pour chacun des modes audio et de protection d'erreurs selon la Recommandations G.723.1 et son Annexe C qui seront utilisés sur le canal logique.

IS11172AudioCapability: indique la capacité de traitement du signal audio codé selon ISO/CEI 11172-3 [34].

Les paramètres booléens qui sont mis à la valeur "vrai" indiquent que le mode particulier de fonctionnement est possible, alors que la mise à la valeur "faux" indique que ce mode n'est pas possible. Les paramètres booléens audioLayer1, audioLayer2 et audioLayer3 indiquent quelles couches de codage audio peuvent être traitées. Les paramètres booléens audioSampling32k, audioSampling44k1 et audioSampling48k indiquent quelles sont les fréquences qui peuvent être utilisées parmi les fréquences d'échantillonnage de l'audio de 32 kHz, 44,1 kHz et 48 kHz respectivement. Les paramètres booléens singleChannel et twoChannels indiquent la capacité de fonctionnement respectivement sur un canal unique et sur un canal stéréo/double. Le nombre entier bitRate indique la capacité maximale en débit pour l'audio et se mesure en multiples de kbit/s par seconde.

IS13818AudioCapability: indique la capacité de traitement du signal audio codé selon ISO/CEI 13818-3 [35].

Les paramètres booléens qui sont mis à la valeur "vrai" indiquent que le mode particulier de fonctionnement est possible, alors que la mise à la valeur "faux" indique que ce mode n'est pas possible. Les variables booléennes audioLayer1, audioLayer2 et audioLayer3 indiquent quelles couches de codage audio peuvent être traitées. Les variables booléennes audioSampling16k, audioSampling22k05, audioSampling24k, audioSampling32k, audioSampling44k1 et audioSampling48k indiquent quelles sont les fréquences qui peuvent être utilisées parmi les fréquences d'échantillonnage de l'audio de 16 kHz, 22,05 kHz, 24 kHz, 32 kHz, 44,1 kHz et 48 kHz respectivement.

Les paramètres booléens relatifs au fonctionnement à plusieurs canaux indiquent la capacité de fonctionner dans les modes particuliers, comme cela est spécifié dans le Tableau 5.

Tableau 5/H.245 – Codages à canaux multiples selon ISO/CEI 13818-3

Codage ASN.1	Signification sémantique du codage
singleChannel	mode monocanal utilisant la configuration 1/0. Mode canal unique (comme dans ISO/CEI 11172-3)
twoChannels	mode à deux canaux utilisant la configuration 2/0. Mode canal stéréo ou à deux canaux (comme dans ISO/CEI 11172-3)
threeChannels2-1	mode à trois canaux utilisant la configuration 2/1. Canaux gauche, droit et canal d'ambiance unique.
threeChannels3-0	mode à trois canaux utilisant la configuration 3/0. Canaux gauche, centre et droit, sans canal d'ambiance.
fourChannels 2-0-2-0	mode à quatre canaux, utilisant la configuration 2/0 + 2/0. Canaux gauche et droit du premier programme, et gauche et droit du second programme.
fourChannels2-2	mode à quatre canaux utilisant la configuration 2/2. Canaux gauche et droit et canaux d'ambiance gauche et droit.
fourChannels3-1	mode à quatre canaux utilisant la configuration 3/1. Canaux gauche, centre et droit, et canal d'ambiance unique.
fiveChannels3-0-2-0	mode à cinq canaux utilisant la configuration 3/0 + 2/0. Canaux gauche, centre et droit du premier programme et gauche et droit du second programme.
fiveChannels3-2	mode à cinq canaux utilisant la configuration 3/2. Canaux gauche, centre et droit, canaux d'ambiance gauche et droit.

Le paramètre booléen `lowFrequencyEnhancement` indique la capacité d'un canal de renforcement des basses fréquences.

Le paramètre booléen `multilingue`, quand il est mis à "vrai", indique la capacité de disposer d'un nombre de canaux multilingues allant jusqu'à sept, et quand il est mis à "faux", aucun canal multilingue n'est disponible.

Le nombre entier `bitRate` indique la capacité maximale du débit pour l'audio et se mesure en multiples de kbit/s par seconde.

7.2.2.7 Capacités d'application de données

Cela indique les capacités de mise en œuvre de transmission de données. L'indication de plus d'une capacité unique dans une structure unique `DataApplicationCapability` n'indique pas une capacité de traitements simultanés. La capacité de traitements simultanés peut être indiquée par des instances de structure `DataApplicationCapability` dans différents ensembles `AlternativeCapabilitySets` d'un descripteur `CapabilityDescriptor` unique.

Les recommandations faisant référence à la présente Recommandation peuvent imposer des restrictions sur certains des modes pouvant être signalés.

Certaines des capacités de données nécessitent des canaux logiques bidirectionnels, par exemple pour mettre en œuvre un protocole de retransmission. Cette exigence est implicitement incluse dans les codages de capacités appropriés.

DataApplicationCapability: une énumération d'applications de données et de débits. Chaque application de données indiquée doit être accompagnée par une ou plusieurs capacités de protocoles `DataProtocolCapability`.

Le paramètre `maxBitRate` indique en unités de 100 bit/s le débit maximal auquel un émetteur peut envoyer des données vidéo ou auquel un récepteur peut recevoir les données d'application en cause.

Le paramètre `t120` indique la capacité d'utiliser le protocole T.120 [25].

Le paramètre `dsm-cc` indique la capacité d'utiliser le protocole DSM-CC [36].

Le paramètre `userData` indique la capacité d'utiliser des données de l'utilisateur non spécifiées à partir d'accès de données externes.

Le paramètre `t84` indique la capacité d'utiliser le transfert d'images (JPEG, JBIG, télécopie Gr.3/4) de type conforme à la Recommandation T.84 [24].

Le paramètre `t434` indique la capacité d'utiliser le transfert des fichiers binaires télématiques selon T.434 [26].

Le paramètre `h224` indique la capacité d'utiliser le protocole de commande de dispositif simplex en temps réel selon H.224 [9].

Le paramètre `nlpid` indique la capacité d'utiliser le protocole de la couche Réseau comme cela est spécifié par `nlpidData` selon la définition de l'ISO/CEI TR 9577 [37]. Ces protocoles incluent notamment le protocole Internet (IP, *internet protocol*) et le protocole point à point (PPP, *point-to-point protocol*) de l'IETF.

NOTE – L'usage de NLPID est largement décrit dans IETC RFC1490, "Interconnexion multiprotocole sur relais de trames".

Le paramètre `dsvdControl` indique la capacité du terminal DSVD à utiliser un canal de commande hors bande.

Le paramètre `H222DataPartitioning` indique la capacité à utiliser un usage modifié et limité de la subdivision des données conformément à la Recommandation H.262, comme cela est spécifié dans la

Recommandation H.222.1, dans laquelle les données d'amélioration sont transmises par un canal de données figurant dans la liste des capacités DataProtocolCapability.

t30 fax: cette séquence codée indique l'utilisation de mode analogique G3V de l'Annexe C/T30, comme spécifié dans la Recommandation T.39 pour les modes DSVF/MSVF.

DataProtocolCapability: contient une énumération de protocoles de données.

v14buffered indique la possibilité d'utiliser une application spécifiée de données en utilisant la conversion V.14 [27] avec une mémoire tampon.

v42lapm indique la capacité d'utiliser une application spécifiée de données en utilisant le protocole LAPM défini dans la Recommandation V.42 [29].

hdlcFrameTunneling indique la capacité d'utiliser une application de données spécifiée en utilisant une trame HDLC à octet distinctif. Il convient de se référer à l'ISO/CEI 3309, 4.5.2 [32].

h310SeparateVCStack indique la capacité d'utiliser une application de données spécifiée en utilisant la pile de protocole définie dans la Recommandation H.310 pour le transport des messages H.245 dans une voie virtuelle ATM différent de celui utilisé pour les communications audiovisuelles.

h310SingleVCStack indique la capacité d'utiliser une application de données spécifiée en utilisant la pile de protocole définie dans la Recommandation H.310 pour le transport des messages H.245 dans la même voie virtuelle ATM que celui utilisé pour les communications audiovisuelles.

Transparent indique la capacité d'utiliser une application de données spécifiée en utilisant un transfert de données transparent.

V120: l'utilisation de v120 fait l'objet d'un complément d'étude dans la Recommandation H.323.

Le paramètre separateLANStack indique qu'une pile de transport séparée sera utilisée dans le transport des données. L'intention d'une liaison de réseau distincte pour les données est indiquée par le type de données dataType dans OpenLogicalChannel qui est réduite à la valeur h310SeparateVCStack ou separateLANStack de l'énumération DataProtocolCapability. Quand la capacité choisie DataApplicationCapability est égale à t120, ces choix impliquent l'utilisation du profil de base T.123 pour le réseau RNIS-LB et les réseaux LAN respectivement. D'autres profils de réseaux LAN peuvent être choisis par un paramètre nonStandardDataProtocolCapability.

Si le paramètre separateLANStack est choisi et si le paramètre separateStack est présent dans la demande OpenLogicalChannel, le récepteur devrait tenter l'établissement de la pile indiquée. Il doit répondre par le message OpenLogicalChannelAck en cas de succès, sinon il répondra par le message OpenLogicalChannelReject avec une raison appropriée.

Si le paramètre separateLANStack est choisi et si le paramètre separateStack est absent dans la demande OpenLogicalChannel, le récepteur devrait fournir un paramètre approprié separateStack dans sa réponse OpenLogicalChannelAck. Le récepteur de cette réponse (le demandeur initial) devrait alors tenter l'établissement de la pile indiquée. Il doit émettre un message CloseLogicalChannel en cas d'échec.

Si le paramètre separateLANStack est choisi et si le paramètre separateStack est présent dans la demande OpenLogicalChannel, on peut y substituer le paramètre separateStack dans la réponse OpenLogicalChannelAck. Si le demandeur initial ne tolère pas de substitution, il doit émettre le message CloseLogicalChannel.

Si le paramètre separateLANStack est choisi et si le paramètre separateStack est absent dans la demande OpenLogicalChannel, de même que dans la réponse OpenLogicalChannelAck, le demandeur initial peut déduire que le répondeur ne comprend pas ces extensions en langage ASN.1 et devrait émettre un message CloseLogicalChannel pour se remettre dans un état normal.

La paramètre v76wCompression indique la capacité d'utiliser la compression de données sur un canal de données V.76.

T84Profile: indique les types de profil d'images fixes que le terminal peut utiliser.

t84Unrestricted ne fournit pas d'indication sur le type d'images fixes conforme à T.84 que le terminal peut utiliser: les informations de la couche T.84 devraient être utilisées pour déterminer si une image particulière peut être reçue.

t84Restricted indique le type d'images fixes conforme à T.84 que le terminal peut utiliser.

qcif indique que l'image couleur de type séquentiel YCrCb peut être utilisée avec une résolution QCIF.

cif indique que l'image couleur de type séquentiel YCrCb peut être utilisée avec une résolution CIF.

ccir601Seq indique que l'image couleur de type séquentiel YCrCb peut être utilisée avec une résolution CCIR601.

ccir601Prog indique que l'image couleur de type progressif YCrCb peut être utilisé avec une résolution CCIR601.

hdtvSeq indique que l'image couleur de type séquentiel YCrCb peut être utilisée avec une résolution HDTV.

hdtvProg indique que l'image couleur de type progressif YCrCb peut être utilisée avec une résolution HDTV.

g3FacsMH200x100 indique la prise en charge de la télécopie du Groupe 3 pour les images deux tons à codage séquentiel MH (Huffman modifié) à résolution normale (200 × 100 ppi).

g3FacsMH200x200 indique la prise en charge de la télécopie du Groupe 3 pour les images deux tons à codage séquentiel MH (Huffman modifié) à résolution fine (200 × 200 ppi).

g4FacsMMR200x100 indique la prise en charge de la télécopie du Groupe 4 pour les images deux tons à codage séquentiel MMR (modification de Reed modifié) à résolution normale (200 × 100 ppi).

g4FacsMMR200x200 indique la prise en charge de la télécopie du Groupe 4 pour les images deux tons à codage séquentiel MMR (modification de Reed modifié) à résolution fine (200 × 200 ppi).

jbig200x200Seq indique la prise en charge des images deux tons à codage JBIG séquentiel sur deux niveaux à la résolution 200 × 200 ppi.

jbig200x200Prog indique la prise en charge des images deux tons à codage JBIG séquentiel sur deux niveaux à la résolution 200 × 200 ppi.

jbig300x300Seq indique la prise en charge des images deux tons à codage JBIG séquentiel sur deux niveaux à la résolution 300 × 300 ppi.

jbig300x300Prog indique la prise en charge des images deux tons à codage JBIG séquentiel sur deux niveaux à la résolution 300 × 300 ppi.

digPhotoLow indique la prise en charge des images couleurs à codage selon JPEG séquentiel jusqu'à une dimension de 720 × 576.

digPhotoMedSeq indique la prise en charge des images couleurs à codage selon JPEG séquentiel jusqu'à une dimension de 1140 × 1152.

digPhotoMedProg indique la prise en charge des images couleurs à codage selon JPEG progressif jusqu'à une dimension de 1440 × 1152.

digPhotoHighProg indique la prise en charge des images couleur à codage selon JPEG progressif jusqu'à une dimension de 2880 × 2304.

7.2.3 Acquittement de l'ensemble de capacités des terminaux

Ce message est utilisé pour confirmer la réception d'un ensemble TerminalCapabilitySet provenant de l'entité CESE homologue.

Le numéro sequenceNumber doit être le même que le numéro sequenceNumber de l'ensemble de capacités TerminalCapabilitySet pour lequel ceci est la confirmation.

7.2.4 Refus de l'ensemble de capacités des terminaux

Ceci est utilisé pour refuser un ensemble TerminalCapabilitySet provenant de l'entité CESE homologue.

Le numéro sequenceNumber doit être le même que le numéro sequenceNumber de l'ensemble TerminalCapabilitySet pour lequel ceci est l'acquittement négatif.

Les raisons de l'envoi de ce message sont indiquées dans le Tableau 6.

Tableau 6/H.245 – Raisons de refus d'un ensemble TerminalCapabilitySet

Codage ASN.1	Raison
non spécifié	aucune raison de refus n'est spécifiée
undefinedTableEntryUsed	un descripteur de capacités fait référence à un élément du tableau capabilityTable qui n'est pas défini
descriptorCapacityExceeded	le terminal n'a pas pu stocker toutes les informations de l'ensemble TerminalCapabilitySet
tableEntryCapacityExceeded	le terminal n'a pas pu stocker davantage d'entrées que celles qui étaient indiquées dans highestEntryNumberProcessed ou sinon n'a pu enregistrer aucune d'elles

7.2.5 Libération sur temporisation de l'ensemble de capacités des terminaux

Ce message est envoyé en cas de fin de temporisation.

7.3 Messages de signalisation de la voie logique

Cet ensemble de messages est destiné à la signalisation de la voie logique. Le même ensemble de messages est utilisé pour la signalisation de la voie logique monodirectionnelle et bidirectionnelle: Certains paramètres ne sont cependant présents que dans le cas de la signalisation de la voie logique bidirectionnelle.

Le sens "direct" est utilisé pour faire référence à la transmission à partir du terminal qui présente la demande initiale pour une voie logique vers l'autre terminal et le sens "inverse" est utilisé pour faire référence au sens opposé de transmission, dans le cas d'une demande de voie logique bidirectionnelle.

7.3.1 Ouverture d'une voie logique

Ce message est utilisé pour les tentatives d'ouverture d'une voie logique monodirectionnelle entre une entité LCSE sortante et une entité LCSE homologue entrante, et pour ouvrir une voie logique bidirectionnelle entre une entité B-LCSE sortante et une entité B-LCSE entrante homologue.

forwardLogicalChannelNumber: indique le numéro de voie logique de la voie logique directe qui doit être ouverte.

forwardLogicalChannelParameters: inclut les paramètres associés à la voie logique dans le cas où l'on essaie d'ouvrir une voie logique monodirectionnelle, de même que les paramètres associés à la voie logique logique dans le sens direct en cas de tentative d'ouverture d'une voie logique bidirectionnelle.

La structure **reverseLogicalChannelParameters** contient les paramètres associés à la voie logique inverse dans le cas où l'on tente d'ouvrir une voie logique bidirectionnelle. Sa présence indique que la demande est une voie logique bidirectionnelle avec les paramètres indiqués, et son absence indique que la demande concerne une voie logique monodirectionnelle.

NOTE – Les paramètres H.222 ne sont pas inclus dans la structure reverseLogicalChannelParameters étant donné que leurs valeurs ne sont pas connues du terminal qui présente la demande.

Le paramètre portNumber est un paramètre de bout en bout, permettant à un utilisateur d'associer un accès d'entrée ou un accès de sortie, ou un numéro de voie logique de couche supérieure, à la voie logique.

dataType indique les données qui doivent être transmises sur la voie logique.

Si le paramètre est de type nullData, la voie logique ne sera pas utilisée pour la transmission de flux de données élémentaires, mais uniquement pour les informations de la couche d'Adaptation. Au cas où la vidéo doit être transmise dans un sens uniquement mais où un protocole de retransmission doit être utilisé, tel que la couche AL3 définie dans la Recommandation H.223, une voie de retour est nécessaire pour transmettre les demandes de retransmission. Il peut également être utilisé pour décrire une voie logique contenant uniquement des références d'horloges (PCR, *program clock reference*) dans le cas des flux de transport TS de H.222.1 [7].

Des terminaux pouvant uniquement fonctionner en mode monodirectionnel (émission ou réception) sur des types de supports d'information utilisant des voies logiques bidirectionnelles devront émettre des capacités uniquement dans le sens de fonctionnement pouvant être utilisé. La direction inverse devra utiliser le paramètre nullData type, pour lequel aucune capacité n'est nécessaire. Les terminaux en mode émission uniquement devraient envoyer des capacités d'émission, mais les terminaux ne devraient pas supposer que l'absence de capacités d'émission implique que le fonctionnement en mode émission seulement ne soit pas possible.

Le paramètre separateStack indique qu'une pile de transport distincte sera utilisée pour transporter les données, et fournit une adresse pour l'établissement de la pile distincte de protocole qui peut être conforme à Q.2931, à E.164 ou à une adresse de transport d'un réseau local d'entreprise.

La structure networkAccessParameters définit la répartition, l'adresse réseau, de même que les informations de création et d'association devant être utilisées pour le paramètre separateStack.

Le paramètre répartition devra exister quand l'adresse networkAddress sera fixée à localAreaNetwork et devra indiquer si l'adresse networkAddress est une adresse de transport pour une transmission à un seul destinataire ou à plusieurs destinataires.

Le paramètre networkAddress indique l'adresse de la pile réelle à utiliser, Q.2931, E.164 ou l'adresse de transport du réseau local d'entreprise.

AssociateConference indique si une conférence en mode données est nouvelle ou non (associateConference = FALSE) ou si elle constitue une conférence de données existante qui devrait être associée à la communication audio/vidéo (associateConference = TRUE).

Le paramètre externalReference indique les informations pouvant être utilisées pour fournir des associations ou fournir plus d'informations sur la pile separateStack.

Si le paramètre est du type VideoCapability ou AudioCapability, la voie logique peut être utilisée pour n'importe laquelle des variantes indiquées par chaque capacité distincte; il devra être possible de passer d'une variante à l'autre en utilisant uniquement la signalisation qui est dans la bande propre à la voie logique. Par exemple, dans le cas de la vidéo codée selon H.261, si les deux formats QCIF et CIF sont indiqués, il devra être possible de passer d'une variante à l'autre en respectant la structure image par image. Dans le cas de l'ensemble DataApplicationCapability, une seule instance de capacité peut être indiquée, étant donné qu'il n'y a pas de signalisation dans la bande qui permette d'intervertir de telles variantes.

Si le paramètre est encryptionData, la voie logique sera utilisée pour le transport des informations de chiffrement comme cela est spécifié.

La structure **H222LogicalChannelParameters**: utilisée pour indiquer les paramètres spécifiques à l'utilisation de la Recommandation H.222.1 [7]. Elle doit être présente dans la structure forwardLogicalChannelParameters et ne doit pas être présente dans la structure reverseLogicalChannelParameters.

Le numéro resourceID indique dans quelle voie virtuelle ATM doit être transportée la voie logique. Les moyens par lesquels ce paramètre est associé à une voie virtuelle (VC) de l'ATM ne sont pas spécifiés dans la présente Recommandation.

Le numéro subChannelID indique quel sous-canal H.222.1 est utilisé pour la voie logique. Il doit être égal à l'identificateur de paquet PID dans un flux de transport (TS, *transport stream*) et à l'identificateur de flux stream_id dans un flux de programme (PS, *program stream*).

Le numéro pcr-pid indique l'identificateur de paquet PID utilisé pour le transport des références d'horloge de programme (PCR) quand le flux de transport est utilisé. Il doit être présent quand la voie virtuelle ATM transporte un flux de transport (TS) et ne doit pas être présent quand la voie virtuelle ATM transporte un flux de programme (PS).

Le champ programDescriptors est une chaîne d'octets facultative qui, si elle est présente, contient un ou plusieurs descripteurs, comme cela est spécifié dans les Recommandations H.222.0 et H.222.1, qui décrit le programme dont les informations à transporter dans la voie logique font partie.

Le champ streamDescriptors est une chaîne d'octets facultative qui, si elle est présente, contient un ou plusieurs descripteurs, comme cela est spécifié dans les Recommandations H.222.0 et H.222.1, qui décrit les informations à transporter dans la voie logique.

La structure **H223LogicalChannelParameters**: utilisée pour indiquer des paramètres spécifiques à l'utilisation de la Recommandation H.223 [8]. Elle doit être présente dans les structures forwardLogicalChannelParameters et reverseLogicalChannelParameters.

Le paramètre adaptationLayerType indique le type de couche d'Adaptation et les options à utiliser sur la voie logique. Les codages sont les suivants: nonStandard, al1Framed (mode tramé AL1), al1NotFramed (mode non tramé AL1), al2WithoutSequenceNumbers (AL2 sans numéro de séquence présent), al2WithSequenceNumbers (AL2 avec numéros de séquence présents) et al3 (AL3, indiquant le nombre d'octets du champ de commande qui seront présents et la taille de la mémoire tampon d'émission, B_s , qui sera utilisée, la taille étant mesurée en octets).

Le paramètre segmentableFlag, quand il est égal à "vrai", indique que la voie est déclarée comme pouvant être segmentée et, quand il est égal à "faux", indique que la voie est déclarée comme ne pouvant pas être segmentée.

V76LogicalChannelParameters: utilisé pour indiquer les paramètres spécifiques à l'utilisation de V.76.

Le paramètre `audioHeader` est utilisé pour indiquer l'utilisation d'un en-tête audio sur la voie logique. Ceci est un paramètre valable pour les canaux dont le type `DataType` est `audio`.

Le paramètre `suspendResume` est utilisé pour indiquer que la voie peut utiliser les procédures d'interruption-reprise pour suspendre d'autres voies logiques. Trois options de voie peuvent être choisies; pas d'interruption-reprise sur la voie, interruption-reprise utilisant une adresse ou interruption-reprise sans adresse comme cela est défini dans la Recommandation V.76. Le paramètre `suspendResumewAddress` indique que la voie avec interruption-reprise doit utiliser le champ d'adresse tel qu'il est défini dans la Recommandation V.76. Le paramètre `suspendResumewoAddress` indique que la voie avec interruption-reprise ne doit pas utiliser le champ d'adresse.

Le paramètre `eRM` indique que la voie logique doit effectuer des procédures de reprise sur erreur comme cela est défini dans la Recommandation V.76.

Le paramètre `uNERM` indique que la voie logique doit fonctionner dans le mode sans reprise sur erreur comme cela est défini dans la Recommandation V.76.

Pour la description des paramètres `n401`, `windowSize` et `loopbackTestProcedure` voir 12.2.1 de la Recommandation V.42 et ses subdivisions. Pour les besoins de la Recommandation V.70, le paramètre `n401` devra être codé en octets.

Le paramètre `crcLength` est un paramètre facultatif indiquant la longueur du contrôle de redondance cyclique (CRC) utilisé dans le mode de reprise sur erreur. En l'absence de ce paramètre, la longueur du contrôle de redondance cyclique (CRC) par défaut devra être utilisée. Le paramètre `crc8bit` indique l'utilisation du contrôle de redondance cyclique (CRC) sur 8 bits, le paramètre `crc16bit` indique l'utilisation du contrôle de redondance cyclique (CRC) sur 16 bits et le paramètre `crc32bit` indique l'utilisation du contrôle de redondance cyclique (CRC) sur 32 bits, comme cela est défini dans la Recommandation V.76.

Le paramètre `recovery` est un paramètre facultatif indiquant les procédures de reprise sur erreur définies dans la Recommandation V.76. En l'absence de ce paramètre, la procédure de reprise sur erreur par défaut devra être utilisée. Le paramètre `sREJ` indique l'utilisation de la procédure de refus sélectif de trame et le paramètre `mSREJ` indique l'utilisation de la procédure de refus sélectif multiple comme cela est défini dans la Recommandation V.76.

Le paramètre d'en-tête `uIH` indique l'utilisation de trames UIH selon la Recommandation V.76.

Le paramètre `rej` indique l'utilisation de la procédure de refus selon la Recommandation V.76.

Le paramètre `V75Parameters` est utilisé pour indiquer un paramètre spécifique à l'utilisation de la Recommandation V.75. Le paramètre `audioHeaderPresent` indique la présence de l'en-tête audio selon V.75.

H2250LogicalChannelParameters: Cette structure permet d'indiquer les paramètres spécifiques à l'utilisation de la Recommandation H.225.0. Cette structure devra être présente dans les structures `forwardLogicalChannelParameters` et `reverseLogicalChannelParameters`.

L'identificateur `sessionId` est un identificateur unique de session du protocole RTP dans une conférence. Il est utilisé par le terminal émetteur pour faire référence à la session à laquelle la voie logique se rapporte. Seule l'entité maîtresse peut créer l'identificateur de session. Il y a par convention deux sessions primaires. La première de ces sessions avec un identificateur de session égal à 1 est la session audio et la seconde session primaire avec un identificateur de session égal à 2 est la session vidéo. Une entité esclave peut ouvrir une session additionnelle en fournissant un identificateur de session égal à 0 dans le message `openLogicalChannel`. L'entité maîtresse créera un identificateur de session unique et le placera dans le message `openLogicalChannelAck`.

L'identificateur `associatedSessionID` est utilisé pour associer une session à une autre session. En général, on associera une session audio à une session vidéo pour indiquer quelles sessions doivent faire l'objet de traitements pour synchroniser la parole avec le mouvement des lèvres.

Le paramètre `mediaChannel` indique une adresse `transportAddress` à utiliser pour la voie logique. Il n'est pas présent dans le message `OpenLogicalChannel` quand la transmission s'effectue vers un seul destinataire. Si l'adresse `transportAddress` a des destinataires multiples, l'entité maîtresse sera responsable de la création des adresses de transport vers plusieurs destinataires et devra inclure les adresses dans le message `OpenLogicalChannel`. Une entité esclave souhaitant ouvrir une nouvelle voie logique vers plusieurs destinataires devra mettre des zéros dans le champ `transportAddress` vers plusieurs destinataires. L'entité maîtresse créera et fournira l'adresse `transportAddress` vers plusieurs destinataires dans le message `OpenLogicalChannelAck` pour l'entité esclave. Il convient de noter que l'entité de commande multipoint (MC) utilisera la commande `communicationModeCommand` pour spécifier les détails concernant toutes les sessions de protocole RTP dans la conférence.

La voie `mediaChannel` est utilisée pour décrire l'adresse transport de la voie logique. Les adresses IPv4 et IPv6 devront être codées avec l'octet de poids fort de l'adresse en position de premier octet dans le type correspondant "OCTET STRING" (chaîne d'octet), par exemple l'adresse 130.1.2.97 de classe B du protocole IPv4 devra avoir le nombre "130" codé dans le premier octet du type "OCTET STRING" qui sera suivi du nombre "1" et ainsi de suite. L'adresse a148:2:3:4:a:b:c:d du protocole IPv6 devra avoir le nombre "a1" codé dans le premier octet, "48" dans le second, "00" dans le troisième, "02" dans le quatrième et ainsi de suite. Les adresses IPX, le nœud, le numéro de réseau (netnum) et l'accès devront être codés avec l'octet de poids fort de chaque champ placé en position de premier octet du type "OCTET STRING" (chaîne d'octet) concerné.

Le paramètre `mediaGuaranteedDelivery` indique si le transport sous-jacent du support d'information doit être ou non choisi pour garantir la transmission des données.

Le paramètre `mediaControlChannel` indique le canal de commande de support d'information sur lequel l'émetteur de la voie logique ouverte écoutera les messages de commande du support d'information pour cette session. Ce champ est présent uniquement quand un canal de commande du support d'information est nécessaire.

Le paramètre `mediaGuaranteeDelivery` indique si le transport sous-jacent du support d'information doit être ou non choisi pour garantir la transmission des données. Ce champ est présent uniquement quand un canal de commande du support d'information est nécessaire.

Le paramètre `silenceSuppression` est utilisé pour indiquer si l'émetteur cesse d'envoyer des paquets durant les intervalles de silence. Il devra être inclus dans le message `openLogicalChannel` pour un canal audio et omis pour tout autre type de canal.

Le paramètre `destination` indique l'étiquette `terminalLabel` de la destination pour autant qu'une étiquette ait été attribuée.

Le paramètre `dynamicRTPPayloadType` indique la valeur dynamique du débit qui est utilisée dans la Recommandation H.323 pour l'autre algorithme de mise en paquets de la vidéo H.261 selon la Recommandation H.225.0. Ce champ est présent uniquement quand une valeur dynamique de débit du protocole RTP est utilisée.

Le paramètre `mediaPacketization` indique quel algorithme optionnel de mise en paquets est utilisé.

7.3.2 Acquittement de l'ouverture de la voie logique

On utilise ce message pour confirmer l'acceptation de la demande de connexion de la voie logique en provenance de l'entité homologue LCSE ou B-LCSE. Dans le cas d'une demande de voie logique monodirectionnelle, ce message indique l'acceptation de cette voie logique monodirectionnelle. Dans

le cas d'une demande de voie logique bidirectionnelle, ce message indique l'acceptation de cette voie logique bidirectionnelle, de même que les paramètres appropriés de la voie logique inverse.

Le numéro `forwardLogicalChannelNumber` indique le numéro de la voie logique de la voie directe qui est en cours d'ouverture.

Le paramètre `reverseLogicalChannelParameters` est présent si et seulement s'il est mis en réponse à une demande de voie logique bidirectionnelle.

Le numéro `reverseLogicalChannelNumber` indique le numéro de la voie logique de la voie inverse.

Le numéro `portNumber` est un paramètre de bout en bout, permettant à un utilisateur d'associer un accès d'entrée ou un accès de sortie, ou un numéro de voie logique de couche supérieure, à la voie logique inverse.

Le paramètre `multiplexParameters` indique les paramètres spécifiques au multiplex, H.222, H.223 ou H.225.0, qui est utilisé pour transporter la voie logique inverse.

Le paramètre `separateStack` indique qu'une pile de transport séparée sera utilisée pour transporter les données et il fournit une adresse, qui peut être une adresse Q.2931, E.164 ou une adresse de réseau local d'entreprise, à utiliser pour l'établissement de la pile.

La structure `forwardMultiplexAckParameters` indique quels sont les paramètres spécifiques au multiplex, H.222, H.223, ou H.225.0, qui sont utilisés pour transporter la voie logique directe.

La structure `H2250LogicalChannelAckParameters` est utilisé pour indiquer les paramètres spécifiques à l'utilisation de la Recommandation H.225.0.

L'identificateur `sessionID` est un identificateur de session RTP unique dans la conférence, pouvant être créé uniquement par l'entité maîtresse. Il est créé et fourni par l'entité maîtresse pour autant que l'entité esclave souhaite créer une nouvelle session en spécifiant un identificateur de session non valide égal à 0 dans le message `openLogicalChannelAck`.

Le paramètre `mediaChannel` indique une adresse de transport `TransportAddress` devant être utilisée pour la voie logique. Il doit être présent dans le message `OpenLogicalChannelAck` quand le transport transmet vers un destinataire. Si le paramètre `transportAddress` correspond à une transmission vers plusieurs destinataires, l'entité maîtresse est responsable de la création de l'adresse de transport pour plusieurs destinataires et devra inclure l'adresse dans le message `OpenLogicalChannel`. Une entité esclave qui souhaite ouvrir un nouveau canal de transmission vers plusieurs destinataires devra mettre des zéros dans le champ `transportAddress` à l'usage de plusieurs destinataires. L'entité maîtresse créera et fournira l'adresse `transportAddress` de transmission vers plusieurs destinataires dans le message `OpenLogicalChannelAck` pour l'entité esclave. Il convient de noter que l'entité de commande multipoint (MC) utilisera l'instruction `communicationModeCommand` pour spécifier les détails concernant toutes les sessions de protocole RTP dans la conférence.

Le paramètre `mediaChannel` est utilisé pour décrire l'adresse de transport pour la voie logique. Les adresses IPv4 et IPv6 doivent être codées avec l'octet de poids fort de l'adresse en position de premier octet dans le type correspondant "OCTET STRING" (chaîne d'octet), par exemple 130.1.2.97 de classe B du protocole IPv4 devra avoir le nombre "130" codé dans le premier octet du type "OCTET STRING" qui sera suivi du nombre "1" et ainsi de suite. L'adresse a148:2:3:4:a:b:c:d du protocole IPv6 devra avoir le nombre "a1" codé dans le premier octet, "48" dans le second, "00" dans le troisième, "02" dans le quatrième et ainsi de suite. Les adresses IPX, le nœud, le numéro de réseau (netnum) et l'accès devront être codés avec l'octet de poids fort de chaque champ en position de premier octet du type dans la chaîne "OCTET STRING" correspondante.

Le paramètre `mediaControlChannel` indique le canal de commande de support d'information sur lequel l'émetteur de la structure `openLogicalChannelAck` écoutera les messages de commande du

support d'information pour cette session. Ce champ est présent uniquement quand un canal de commande du support d'information est nécessaire.

Le paramètre `dynamicRTPPayloadType` indique la valeur dynamique du débit qui est utilisée dans H.323 pour l'autre algorithme de mise en paquets vidéo H.261 selon la Recommandation H.225.0. Ce champ est présent uniquement quand une valeur dynamique de débit du protocole RTP est utilisée.

NOTE – Les paramètres H.223 ne sont pas inclus dans la structure `reverseLogicalChannelParameters` étant donné que leurs valeurs ont été spécifiées dans le message de demande `OpenLogicalChannel`.

7.3.3 Refus de l'ouverture de la voie logique

Ce message est utilisé pour refuser la demande de connexion de la voie logique en provenance de l'entité homologue LCSE ou B-LCSE.

NOTE – Dans le cas d'une demande de voie logique bidirectionnelle, le refus s'applique à la fois aux voies logiques directe et inverse. Il n'est pas possible d'accepter l'une et de refuser l'autre.

Le numéro `forwardLogicalChannelNumber` indique le numéro de la voie logique de la voie logique directe spécifiée dans la demande faisant l'objet d'un refus.

Le champ des raisons indique le motif du refus de l'établissement de la voie logique. Les valeurs de raison sont indiquées dans le Tableau 7.

Tableau 7/H.245 – Motifs de refus du message d'ouverture d'une voie `OpenLogicalChannel`

Codage ASN.1	Motif
<code>unspecified</code>	aucun motif n'est spécifié pour le refus
<code>unsuitableReverseParameters</code>	ce paramètre sera uniquement utilisé pour refuser une demande de la voie logique bidirectionnelle quand le seul motif du refus est que la structure demandée <code>reverseLogicalChannelParameters</code> est inappropriée. Un tel refus doit immédiatement être suivi de la procédure d'initialisation pour ouvrir une voie logique bidirectionnelle similaire mais pouvant être acceptée.
<code>dataTypeNotSupported</code>	le terminal n'a pas été capable d'utiliser le paramètre type de données <code>dataType</code> indiqué dans <code>OpenLogicalChannel</code> .
<code>dataTypeNotAvailable</code>	le terminal n'a pas été capable d'utiliser simultanément le paramètre type de données <code>dataType</code> indiqué dans <code>OpenLogicalChannel</code> avec le paramètre <code>dataTypes</code> correspondant aux voies logiques qui sont déjà ouvertes.
<code>unknownDataType</code>	le terminal n'a pas compris le type de données <code>dataType</code> indiqué dans <code>OpenLogicalChannel</code> .
<code>dataTypeALCombinationNotSupported</code>	le terminal n'a pas été capable d'utiliser simultanément le paramètre type de données <code>dataType</code> indiqué dans <code>OpenLogicalChannel</code> avec le type de couche d'Adaptation indiqué dans la structure <code>H.223logicalChannelParameters</code> .
<code>multicastChannelNotAllowed</code>	le canal de transmission vers plusieurs destinataires n'a pas pu être ouvert.
<code>insufficientBandwith</code>	le canal n'a pas pu être ouvert parce que la permission d'utiliser la largeur de bande demandée pour la voie logique a été refusée.

**Tableau 7/H.245 – Motifs de refus du message d'ouverture d'une voie
OpenLogicalChannel (fin)**

Codage ASN.1	Motif
separateStackEstablishmentFailed	une demande de lancement du module de données d'une communication sur une pile distincte a échoué
invalidSessionID	tentative par l'entité esclave de fixer l'identificateur SessionID lors de l'ouverture d'une voie logique vers l'entité maîtresse
masterSlaveConflict	tentative par l'entité esclave d'ouvrir la voie logique pour laquelle l'entité maîtresse a déterminé qu'une incompatibilité pouvait survenir (voir 8.4.1.3 et 8.5.1.3)

7.3.4 Confirmation de l'ouverture de la voie logique

Ce message est utilisé dans la signalisation bidirectionnelle pour indiquer vers l'entité entrante B-LCSE que la voie inverse est ouverte et peut être utilisée pour la transmission.

Le numéro logicalChannelNumber indique le numéro de la voie logique de la voie directe qui a été ouverte.

7.3.5 Fermeture de la voie logique

Ce message est utilisé par l'entité LCSE ou B-LCSE sortante pour fermer une connexion de voie logique entre deux entités LCSE ou B-LCSE homologues.

NOTE – Dans le cas d'une voie logique bidirectionnelle, ce message ferme à la fois les voies directe et inverse. Il n'est pas possible de fermer une voie à l'exclusion de l'autre.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe qui doit être fermé.

L'origine de la libération de la voie logique est indiquée dans le Tableau 8.

Tableau 8/H.245 – Origines de la libération de la voie logique

Codage ASN.1	Origine
utilisateur	l'utilisateur de l'entité LCSE ou B-LCSE est à l'origine de la libération
lcse	l'entité LCSE ou B-LCSE est à l'origine de la libération. Cela peut survenir à la suite d'une erreur de protocole

7.3.6 Acquiescement de fermeture de la voie logique

Ce message est utilisé pour confirmer la fermeture d'une connexion de la voie logique.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe que l'on ferme.

7.3.7 Demande de fermeture de la voie

Ceci est utilisé par l'entité CLCSE sortante pour demander la fermeture d'une connexion de la voie directe entre deux entités LCSE homologues.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe dont la fermeture est demandée.

7.3.8 Acquittement de demande de fermeture de la voie

Ceci est utilisé par l'entité CLCSE entrante pour indiquer que la connexion de la voie logique sera fermée.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe dont la fermeture est demandée.

7.3.9 Refus de demande de fermeture de la voie

Ce message est utilisé par l'entité CLCSE entrante pour indiquer que la connexion de la voie logique ne sera pas fermée.

La structure forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe dont la fermeture est demandée.

Le champ des raisons indique le motif du refus de la demande de fermeture de la voie logique. La seule valeur de raison valable est "non spécifié".

7.3.10 Libération sur temporisation de demande de fermeture de la voie

Ce message est émis par l'entité CLCSE sortante dans le cas d'une fin de temporisation.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique de la voie directe dont la fermeture est demandée.

7.4 Messages de signalisation des tableaux de multiplexage

Cet ensemble de messages est destiné à assurer la transmission des entrées du Tableau de multiplexage H.223 entre l'émetteur et le récepteur.

7.4.1 Emission d'entrée de multiplexage

Ce paramètre est utilisé pour envoyer les entrées du Tableau de multiplexage H.223 entre l'émetteur et le récepteur. Il est envoyé à partir d'une entité MTSE sortante vers une entité MTSE homologue entrante.

Le paramètre sequenceNumber est utilisé pour désigner des instances de MultiplexEntrySend de sorte que la réponse correspondante puisse être identifiée.

La structure MultiplexEntryDescriptors est un ensemble de descripteurs MultiplexEntryDescriptors allant de 1 à 15.

MultiplexEntryDescriptor: décrit une entrée unique de tableau de multiplex. Il inclut le numéro MultiplexTableEntryNumber, de même qu'une liste d'éléments MultiplexElements. Une liste incomplète d'éléments indique que l'entrée est désactivée.

MultiplexElement: structure récurrente qui décrit un élément unique et un compteur de répétitions. S'il est du type logicalChannelNumber, l'élément indique une portion unique à partir d'un canal logique donné et le nombre itératif indique la longueur de la portion en octets. S'il est du type subElementList, l'élément indique une séquence d'éléments de multiplexage imbriqués et le compteur de répétitions indique le nombre de fois où la séquence est répétée. Dans chacun des cas, si le champ repeatCount est untilClosingFlag, cela signifie que l'élément doit être répété indéfiniment jusqu'au fanion de fermeture de l'unité de protocole MUX-PDU.

Dans chaque descripteur MultiplexEntryDescriptor, le compteur de répétitions de l'élément final MultiplexElement dans la liste elementList devra être mis à la valeur "untilClosingFlag" et le compteur de répétitions de tous les autres MultiplexElements dans la liste elementList devra être mis à la valeur "finie". Ceci garantit que toutes les entrées du tableau de multiplex définissent un schéma

de séquence de multiplex de longueur indéfinie, qui se répète jusqu'au fanion de fermeture de l'unité de protocole MUX-PDU. Un descripteur MultiplexEntryDescriptor avec un champ elementList incomplet doit indiquer une entrée désactivée.

Chaque demande MultiplexEntrySend peut contenir jusqu'à 15 descripteurs MultiplexEntryDescriptors, chacun décrivant une entrée de tableau de multiplexage unique. Les entrées de multiplexage peuvent être envoyées dans n'importe quel ordre.

7.4.2 Acquittement d'envoi d'entrée de multiplexage

Ce paramètre est utilisé pour confirmer la réception d'un ou plusieurs descripteurs multiplexEntryDescriptors à partir d'un paramètre MultiplexEntrySend émis par l'entité MTSE homologue.

Le numéro sequenceNumber devra être le même que le numéro sequenceNumber dans l'envoi MultiplexEntrySend pour lequel ceci est la confirmation.

La structure multiplexTableEntryNumber indique quelles entrées du tableau de multiplexage sont confirmées.

7.4.3 Refus d'émission d'entrée de multiplex

Ce paramètre est utilisé pour refuser un ou plusieurs descripteurs multiplexEntryDescriptors à partir d'une structure MultiplexEntrySend envoyée à partir de l'entité MTSE homologue.

Le numéro sequenceNumber doit être le même que le numéro sequenceNumber dans la structure MultiplexEntrySend pour lequel ceci est le refus.

La structure MultiplexEntryRejectionDescriptions spécifie quelles entrées du tableau sont refusées et quelle est la raison du refus. Les raisons du refus sont indiquées dans le Tableau 9.

Tableau 9/H.245 – Raisons du refus du message MultiplexEntrySend

Codage ASN.1	Raison
unspecified	aucune raison de refus n'est spécifiée
descriptorTooComplex	le descripteur MultiplexEntryDescriptor a dépassé la capacité du terminal récepteur

7.4.4 Libération sur temporisation de l'entrée de multiplexage

Ce message est envoyé par l'entité MTSE sortante dans le cas d'une fin de temporisation.

Le numéro multiplexTableEntryNumber indique quelles entrées de tableau de multiplexage viennent en fin de temporisation.

7.5 Messages de demande de signalisation des tableaux de multiplexage

Cet ensemble de messages est utilisé pour demander la retransmission d'un ou plusieurs descripteurs MultiplexEntryDescriptors entre l'émetteur et le récepteur.

7.5.1 Demande d'entrée de multiplexage

Ce message est utilisé pour demander la retransmission d'un ou plusieurs descripteurs MultiplexEntryDescriptors.

Le numéro entryNumbers est la liste des numéros MultiplexTableEntryNumbers des descripteurs MultiplexEntryDescriptors pour lesquels la retransmission est demandée.

7.5.2 Acquitement de la demande d'entrée de multiplexage

Ce message est utilisé par l'entité RMESE entrante pour indiquer que l'entrée de multiplexage sera transmise.

Le numéro entryNumbers est la liste des numéros MultiplexTableEntryNumbers des descripteurs MultiplexEntryDescriptors qui seront transmis.

7.5.3 Refus de demande d'entrée de multiplexage

Ce message est utilisé par l'entité RMESE entrante pour indiquer que l'entrée de multiplexage ne sera pas transmise.

Le numéro entryNumbers est la liste des numéros MultiplexTableEntryNumbers des descripteurs MultiplexEntryDescriptors qui ne seront pas transmis. Les valeurs du numéro MultiplexTableEntryNumber dans le paramètre entryNumbers devraient correspondre aux valeurs du numéro MultiplexTableEntryNumber dans le paramètre rejectionDescriptions, car sinon des erreurs pourraient survenir pendant le fonctionnement.

La structure RequestMultiplexEntryRejectionDescriptions spécifie quelles sont les entrées du tableau qui sont refusées et quelle est la raison de ce refus. Les raisons du refus sont indiquées dans le Tableau 10.

Tableau 10/H.245 – Raisons du refus d'un message MultiplexEntrySend

Codage ASN.1	Raison
non spécifié	aucune raison de refus n'est spécifiée

7.5.4 Libération sur temporisation de la demande d'entrée de multiplexage

Ce message est envoyé par l'entité RMESE sortante dans le cas d'une fin de temporisation.

Le numéro entryNumbers est une liste des numéros MultiplexTableEntryNumbers des descripteurs MultiplexEntryDescriptors pour lesquels la fin de temporisation s'est produite.

7.6 Messages de mode demande

Cet ensemble de messages est utilisé par un terminal récepteur pour demander des modes particuliers de transmission à partir du terminal émetteur.

7.6.1 Mode demande

Ce paramètre est utilisé pour demander des modes particuliers de transmission à partir du terminal émetteur. Il s'agit d'une liste, par ordre de préférence (les modes les plus demandés étant les premiers), des modes que le terminal souhaite recevoir. Chaque mode est décrit en utilisant un paramètre ModeDescription.

Le paramètre sequenceNumber est utilisé pour désigner des instances de mode demande RequestMode de sorte que la réponse correspondante puisse être identifiée.

ModeDescription: un ensemble constitué d'un ou plusieurs éléments ModeElements.

ModeElement: utilisé pour décrire un élément de mode, c'est-à-dire une des parties constituantes d'une description complète de mode. Cet élément indique le type de train de bits élémentaires qui est demandé et, optionnellement, comment on demande que cet élément de mode soit multiplexé.

Le paramètre type est utilisé pour indiquer le type de train de bits élémentaires qui est demandé. C'est un choix entre les modes VideoMode, AudioMode, DataMode et EncryptionMode.

La structure **h223ModeParameters**: utilisée pour déclarer les paramètres spécifiques à l'utilisation de la Recommandation H.223 [8].

La structure **adaptationLayerType** indique quelle couche d'Adaptation et quelles options sont demandées pour le type demandé. Les codages sont les suivants: **nonStandard**, **al1Framed** (mode tramé AL1), **al1NotFramed** (mode non tramé AL1), **al2WithoutSequenceNumbers** (AL2 sans numéros de séquence existants), **al2WithSequenceNumbers** (AL2 avec numéros de séquence existants), et **al3** (AL3 en indiquant le nombre d'octets du champ de commande qui sera présent, ainsi que la taille de la mémoire tampon d'émission, B_s , qui sera utilisée, la taille étant mesurée en octets).

Le paramètre **segmentableFlag**, quand il est égal à "vrai", indique que le multiplexage avec segmentation est demandé, et quand il est égal à "faux", il indique que le multiplexage sans segmentation est demandé.

7.6.1.1 Mode vidéo

Ceci est un choix de modes vidéo **VideoModes**.

H261VideoMode: indique la résolution d'image demandée (soit le format QCIF ou CIF), le débit, en unités de 100 bits/s, de même que la transmission d'images fixes.

H262VideoMode: indique le profil et le niveau demandés, et les champs optionnels indiquent, le cas échéant, les valeurs demandées pour les paramètres indiqués. Les champs optionnels sont des nombres entiers dont les unités sont définies dans le Tableau 2.

H263VideoMode: indique la résolution d'image demandée (format SQCIF, QCIF, CIF, 4CIF et 16CIF) de même que le débit, en unités de 100 bit/s.

Les paramètres booléens **unrestrictedVector**, **arithmeticCoding**, **advancedPrediction**, et **pbFrames**, quand ils sont mis à la valeur "vrai", indiquent qu'il est demandé d'utiliser ces modes optionnels qui ont été définis dans les annexes de la Recommandation H.263.

Le paramètre booléen **errorCompensation**, quand il est mis à vrai, indique que le codeur est en mesure de traiter les indications **videoNotDecodedMBs**, de même que les erreurs de compensation comme cela est illustré à l'Appendice I/H.263. Il n'est pas nécessaire que le codeur réponde aux indications **videoNotDecoded**. Dans un équipement de commande multipoint (MCU, *multipoint control unit*), l'équipement de commande peut très bien ne pas répondre à toutes les indications.

IS11172VideoMode: indique une demande pour la structure **constrainedBitstream** et les champs optionnels indiquent, le cas échéant, les valeurs demandées pour les paramètres indiqués. Les champs optionnels sont des nombres entiers dont les unités sont définies dans le Tableau 3.

7.6.1.2 Mode audio

Ceci est un choix de modes audio **AudioModes**.

La signification exacte du codage audio de la série G est indiquée dans le Tableau 4. Il y a quatre options pour l'audio selon G.723.1, pour permettre qu'un des débits (le faible débit de 5,3 kbit/s ou le débit élevé de 6,3 kbit/s) soit demandé avec ou sans codage des périodes de silence.

G7231AnnexCMode: utilisé pour demander le codage audio selon l'Annexe C/G.723.1. La structure **maxA1-sduAudioFrames** indique le nombre maximal demandé de trames audio selon AL-SDU. La variable booléenne **silenceSupression**, quand elle est mise à vrai, demande l'utilisation du codage des périodes de silence définie dans l'Annexe A/G.723.1. Les champs **g723AnnexCAudioMode**, **highRateMode0**, **highRateMode1**, **lowRateMode0**, **lowRateMode1**, **sidMode0** et **sidMode1** indiquent le nombre d'octets par trame demandé pour chacun des modes de protection audio et de protection contre les erreurs selon la Recommandation G.723.1 et l'Annexe C/G.723.1.

IS11172AudioMode: utilisé pour demander le codage audio selon l'ISO/CEI 11172-3 [34].

Le paramètre audioLayer indique quelle couche de codage est demandée, audioLayer1, audioLayer2 ou audioLayer3.

Le paramètre audioSampling indique quelle fréquence d'échantillonnage est demandée: les paramètres audioSampling32k, audioSampling44k1 et audioSampling48k indiquent des fréquences d'échantillonnage audio qui sont égales à 32 kHz, 44,1 kHz et 48 kHz respectivement.

Le paramètre multi-channelType indique quel mode multivoie est demandé: singleChannel, twoChannelStereo et twoChannelDual demandent respectivement un fonctionnement à canal unique, à canal stéréo et à deux canaux.

Le paramètre bitRate indique le débit audio demandé qui est mesuré en unités de kbit/s.

IS13818AudioMode: utilisé pour demander le codage audio selon l'ISO/CEI 13818-3 [35].

Le paramètre audioLayer indique quelle couche de codage est demandée: audioLayer 1, audioLayer 2 ou audioLayer 3.

La structure audioSampling indique quelle fréquence d'échantillonnage est demandée: audioSampling16k, audioSampling22k05, audioSampling24k, audioSampling32k, audioSampling44k1 et audioSampling48k indiquent les fréquences d'échantillonnage audio à 16 kHz, 22,05 kHz, 24k Hz, 32 kHz, 44,1 kHz et 48 kHz respectivement.

La structure multi-channelType indique quel mode multivoie est demandé comme cela est spécifié dans le Tableau 11.

Tableau 11/H.245 – Codages ISO/CEI 13818-3 pour plusieurs canaux

Codage ASN.1	Signification sémantique du codage
singleChannel	mode monocanal utilisant la configuration 1/0. Mode canal unique (comme dans ISO/CEI 11172-3)
twoChannelStereo	mode à deux canaux utilisant la configuration 2/0. Mode canal stéréo (comme dans ISO/CEI 11172-3)
twoChannelDual	deux canaux, utilisant la configuration 2/0. Mode à deux canaux (comme dans ISO/CEI 11172-3)
threeChannels2-1	mode à trois canaux utilisant la configuration 2/1. Canaux gauche, droit et canal d'ambiance unique
threeChannels3-0	mode à trois canaux utilisant la configuration 3/0. Canaux gauche, centre et droit, sans canal d'ambiance
fourChannels 2-0-2-0	mode à quatre canaux, utilisant la configuration 2/0 + 2/0. Canaux gauche et droit du premier programme et canaux gauche et droit du second programme
fourChannels2-2	mode à quatre canaux utilisant la configuration 2/2. Canaux gauche et droit et canaux d'ambiance gauche et droit
fourChannels3-1	mode à quatre canaux utilisant la configuration 3/1. Canaux gauche, centre et droit, et canal d'ambiance unique
fiveChannels3-0-2-0	mode à cinq canaux utilisant la configuration 3/0 + 2/0. Canaux gauche, centre et droit du premier programme et canaux gauche et droit du second programme
fiveChannels3-2	mode à cinq canaux utilisant la configuration 3/2. Canaux gauche, centre et droit, canaux d'ambiance gauche et droit

Le paramètre booléen `lowFrequencyEnhancement`, quand il est mis à la valeur "vrai", demande un canal de renforcement des basses fréquences.

Le paramètre booléen `multilingue`, quand il est mis à la valeur "vrai", demande jusqu'à sept canaux multilingues.

Le paramètre `bitRate` indique le débit audio demandé et se mesure en multiples de kbit/s par seconde.

7.6.1.3 Mode données

Ceci est un choix d'applications et de débits de transmission de données.

Le paramètre `bitRate` indique le débit demandé en unités de 100 bit/s.

Le paramètre `t120` demande l'utilisation du protocole T.120 [25].

Le paramètre `dsm-cc` demande l'utilisation du protocole DSM-CC [36].

Le paramètre `userData` demande l'utilisation de données d'utilisateur non spécifiées provenant de ports de données externes.

Le paramètre `t84` demande l'utilisation de T.84 [24] pour le transfert de ces images (JPEG, JBIG, Télécopie Gr.3/4).

Le paramètre `t434` demande l'utilisation de T.434 [26] pour le transfert de fichiers binaires télématiques.

Le paramètre `h224` demande l'utilisation du protocole H.224 [9] de commande en temps réel pour les applications simplex.

Le paramètre `nlpid` demande l'utilisation de l'application de données spécifiées entre les couches Liaison et réseau.

Le paramètre `V.76Control` demande l'utilisation du terminal V.76 afin d'utiliser un canal de commande hors bande.

La structure `h222DataPartitioning` demande l'application de l'usage modifié et restreint des subdivisions de données de H.262, comme cela est spécifié dans la Recommandation H.222.1, dans laquelle les données d'amélioration sont transmises par un canal de données figurant dans la liste des capacités `DataProtocolCapability`.

7.6.1.4 Mode de chiffrement

Ceci est un choix de modes de chiffrement.

Le paramètre `h233Encryption` demande l'utilisation du chiffrement selon les Recommandations H.233 [11] et H.234 [12].

7.6.2 Message d'acquiescement du mode demande `RequestModeAcknowledge`

Ce message est envoyé pour confirmer que le terminal émetteur a l'intention de transmettre dans un des modes demandés par le terminal de récepteur.

Le numéro `sequenceNumber` doit être identique au numéro `sequenceNumber` dans le mode `RequestMode` pour lequel ceci est la confirmation.

Le champ des réponses indique une action à partir du terminal distant. Les valeurs possibles de réponse sont indiquées dans le Tableau 12.

Tableau 12/H.245 – Réponses de confirmation du mode demande

Codage ASN.1	Réponse
willTransmitMostPreferredMode	le terminal émetteur sera basculé sur le mode préférentiel du récepteur
willTransmitLessPreferredMode	le terminal émetteur sera basculé sur l'un des modes préférentiels du récepteur, mais pas le mode de premier choix

7.6.3 Message de refus du mode demande RequestModeReject

Ce message est envoyé pour refuser la demande présentée par le terminal récepteur.

Le numéro sequenceNumber devra être identique au numéro sequenceNumber dans le paramètre de mode RequestMode pour lequel ceci est la réponse.

Le champ des raisons indique la raison du refus du mode demandé. Les valeurs de raison sont indiquées dans le Tableau 13.

Tableau 13/H.245 – Réponses de refus du mode demande

Codage ASN.1	Réponse
modeUnavailable	le terminal émetteur ne modifiera pas son mode d'émission étant donné que les modes demandés ne sont pas disponibles
multipointConstraint	le terminal émetteur ne modifiera pas son mode d'émission en raison d'une contrainte multipoint
requestDenied	le terminal émetteur ne modifiera pas son mode d'émission

7.6.4 Message de demande de libération de mode RequestModeRelease

Ce paramètre est utilisé par l'entité sortante MRSE dans le cas d'une fin de temporisation.

7.7 Messages liés au temps de propagation aller et retour

Cet ensemble de messages est utilisé par un terminal pour déterminer le temps de propagation aller et retour entre deux terminaux en communication. Il permet également à un utilisateur de la Recommandation H.245 de déterminer si l'entité de protocole homologue H.245 est active.

7.7.1 Demande de temps de propagation aller et retour

Ce message est envoyé de l'entité RTDSE sortante vers l'entité RTDSE entrante.

Le numéro sequenceNumber est utilisé pour désigner les instances de demande de temps de propagation aller et retour RoundTripDelayRequest de sorte que la réponse correspondante puisse être identifiée.

7.7.2 Réponse sur le temps de propagation aller et retour

Ce message est envoyé de l'entité RTDSE entrante vers l'entité RTDSE sortante.

Le numéro sequenceNumber devra être identique au numéro sequenceNumber dans la demande RoundTripDelayRequest pour laquelle ceci est la réponse.

7.8 Messages relatifs à la boucle de maintenance

Cet ensemble de messages est utilisé par un terminal pour effectuer les fonctions de boucle de maintenance.

7.8.1 Demande de boucle de maintenance

Ce message est envoyé pour demander un type particulier de boucle distante. Les paramètres de type `mediaLoop` et `logicalChannelLoop` demandent la boucle distante d'un canal logique uniquement comme cela est indiqué par le numéro `LogicalChannelNumber`, alors que le paramètre de type `systemeLoop` s'applique à tous les canaux logiques. La définition exacte de ces types est spécifique au système et se trouve hors du domaine de la présente Recommandation.

7.8.2 Acquittement de la demande de boucle de maintenance

Ce message est utilisé pour confirmer que le terminal exécutera la boucle comme cela a été demandé.

7.8.3 Refus de la demande de boucle de maintenance

Ce message est utilisé pour indiquer que le terminal n'exécutera pas la boucle comme cela a été demandé.

Un terminal peut utiliser la raison `canNotPerformLoop` pour indiquer qu'il ne dispose pas de la capacité pour exécuter la boucle demandée.

7.8.4 Commande "arrêt" de la boucle de maintenance

A la réception de cette commande, le terminal doit déconnecter toutes les boucles et rétablir les circuits audio, vidéo et de données à l'état normal.

7.9 Messages en mode communication

Cet ensemble de messages est utilisé par une entité de commande multipoint H.323 pour transmettre le mode communication d'une conférence H.323.

7.9.1 Commande du mode de communication

Cette commande est envoyée par l'entité de commande H.323 pour spécifier le mode de transmission pour chaque type de données, avec un destinataire unique ou des destinataires multiples. Cette commande peut entraîner le passage de la conférence centralisée à la conférence décentralisée. Un tel passage peut entraîner la fermeture de tous les canaux logiques existants et l'ouverture de nouveaux canaux.

La structure `CommunicationModeTable` spécifie toutes les sessions dans la conférence. Pour chaque session, elle spécifie l'identificateur de session de protocole RTP, un identificateur associé de session de protocole RTP, un numéro de terminal, une description de session, un mode pour chaque type de données, une adresse à un destinataire ou plusieurs destinataires pour le canal d'informations et une demande associée de remise garantie, une adresse de commande d'informations pour le canal RTCP de retour et une demande associée de remise garantie.

7.9.2 Demande de mode de communication

Ceci est envoyé vers l'entité de commande multipoint (MC) pour demander le mode de communication de la conférence en cours.

7.9.3 Réponse sur le mode de communication

Ceci est envoyé par l'entité de commande multipoint (MC), en réponse à une demande `CommunicationModeRequest` pour spécifier le mode de communication d'une conférence.

7.10 Demande de conférence et messages de réponse

L'identificateur TerminalID, qui est utilisé dans les messages de conférence et les messages de réponse a une longueur de 128 octets. Lorsqu'un terminal H.323 et un terminal H.320 communiquent par une passerelle H.323, ce champ est tronqué à 32 octets.

7.10.1 Demande de la liste des terminaux

Cette demande équivaut à la commande TCU de H.230 telle qu'elle est décrite dans la Recommandation H.243.

7.10.2 Réponse sur la liste des terminaux

Cette demande équivaut à la séquence de numéro terminalNumbers telle qu'elle est décrite dans la Recommandation H.230.

7.10.3 Demande du rôle de président de séance

Cette demande équivaut au CCA tel qu'il est décrit dans la Recommandation H.230.

7.10.4 Annulation de la demande du rôle de président de séance

Cette demande équivaut au CIS tel qu'il est décrit dans la Recommandation H.230.

7.10.5 Réponse à la demande du rôle de président de séance

Cette demande équivaut à n'importe quel CIT selon la Recommandation H.230 si le jeton du rôle de président est accepté ou à n'importe quel CCR selon la Recommandation H.230 si le jeton du rôle de président est refusé.

7.10.6 Rejet du terminal

Cette demande équivaut au CCD tel qu'il est décrit dans la Recommandation H.230.

7.10.7 Refus du rejet du terminal

Cette demande équivaut au CIR tel qu'il est décrit dans la Recommandation H.230.

7.10.8 Demande d'identification du terminal

Ce numéro d'identification équivaut au TCP tel qu'il est décrit dans la Recommandation H.230.

7.10.9 Réponse d'identification du terminal par l'entité de commande MC

Cette réponse équivaut au TIP tel qu'il est décrit dans la Recommandation H.230.

7.10.10 Demande d'entrée du mot de passe selon H.243

Cette demande équivaut au TCS1 tel qu'il est décrit dans la Recommandation H.230.

7.10.11 Réponse sur le mot de passe

Cette réponse équivaut à IIS tel qu'il est décrit dans la Recommandation H.230.

7.10.12 Demande d'entrée d'identification de terminal selon H.243

Cette réponse équivaut à TCS2/TCI tel qu'il est décrit dans la Recommandation H.230.

7.10.13 Réponse sur l'identification de terminal

Cette réponse équivaut à IIS tel qu'il est décrit dans la Recommandation H.230.

7.10.14 Demande d'entrée d'identification de conférence H.243

Cette réponse équivaut à TCS3 tel qu'il est décrit dans la Recommandation H.230.

7.10.15 Réponse sur l'identification de conférence

Cette réponse équivaut à IIS tel qu'il est décrit dans la Recommandation H.230.

7.10.16 Refus de commande vidéo

Cette demande équivaut à VCR tel qu'il est décrit dans la Recommandation H.230.

7.10.17 Demande d'entrée de l'adresse d'extension

Cette demande équivaut à TCS4 tel qu'il est décrit dans la Recommandation H.230.

7.10.18 Réponse sur l'adresse d'extension

Cette réponse équivaut à IIS tel qu'il est décrit dans la Recommandation H.230.

7.11 Commandes

Un message de commande nécessite qu'une action particulière soit prise, mais n'exige pas de réponse explicite.

7.11.1 Envoi de l'ensemble des capacités du terminal

Le message `specificRequest` donne l'ordre au terminal distant d'indiquer ses capacités d'émission et de réception en envoyant au moins un ensemble `TerminalCapabilitySets` contenant les informations demandées, comme cela est spécifié ci-dessous. Ce message peut être envoyé à n'importe quel moment pour obtenir les capacités du terminal distant, par exemple, par suite d'une interruption ou de tout autre motif de défaillance; cependant, de tels messages ne devraient pas être envoyés répétitivement sans raison majeure.

Un terminal devra uniquement demander la transmission des numéros `capabilityTableEntryNumbers` et `capabilityDescriptorNumbers` qu'il a préalablement reçus. Un terminal devra ignorer toutes les demandes de transmission de ces paramètres qu'il n'a pas préalablement transmises, auquel cas et il faudra considérer qu'aucune défaillance ne s'est produite.

Le paramètre booléen `multiplexCapability`, quand il est mis à "vrai", demande la transmission du paramètre `MultiplexCapability`.

La structure `capabilityTableEntryNumbers` est un ensemble de numéros `capabilityTableEntryNumbers` indiquant les éléments de la structure `CapabilityTableEntry` dont le terminal demande la transmission.

La structure `capabilityDescriptorNumbers` est un ensemble de numéros `capabilityDescriptorNumbers` indiquant les descripteurs de capacités `CapabilityDescriptors` dont le terminal demande la transmission.

Le paramètre `genericRequest` exige que le terminal distant envoie l'ensemble de la capacité du terminal.

7.11.2 Chiffrement

Ce message est utilisé pour échanger des capacités de chiffrement et pour donner l'instruction de transmettre un vecteur d'initialisation (IV, *initialization vector*), voir les Recommandations H.233 [11] et H.234 [12].

encryptionSE est un message d'échange de sessions (SE, *session exchange*) selon H.233, sauf que les bits de protection contre les erreurs décrits dans la Recommandation H.233 ne doivent pas être appliqués.

encryptionIVRequest donne l'instruction au module de chiffrement distant de transmettre un nouveau vecteur IV dans une voie logique ouverte pour les données encryptionData.

L'identificateur encryptionAlgorithmID indique au récepteur que le terminal émetteur associera la valeur d'identificateur de l'algorithme h233AlgorithmIdentifier à l'algorithme de chiffrement associatedAlgorithm non normalisé.

7.11.3 Contrôle de flux

Cette commande est utilisée pour spécifier la limite supérieure du débit d'une seule voie logique ou de l'ensemble du multiplex. Un terminal peut envoyer cette instruction pour restreindre le débit envoyé par le terminal distant. Un terminal recevant cet ordre doit l'exécuter.

Quand l'objet est du type logicalChannelNumber, la limite s'applique à la voie logique donnée; quand l'objet est du type resourceID, la limite s'applique à la voie virtuelle ATM donné; et quand l'objet est du type wholeMultiplex, la limite s'applique à l'ensemble du multiplex.

maximumBitRate se mesure en multiples de 100 bits/s en moyenne, sur des périodes consécutives sans chevauchement d'une seconde. Quand ce paramètre est présent, la limite spécifiée remplace n'importe quelle limite antérieure, que celle-ci soit supérieure ou inférieure. Quand ce paramètre est absent, toute limite antérieure relative au débit pour la voie n'est plus applicable.

Le point auquel s'applique la limite du débit et la spécification relative aux bits spécifiques qui sont pris en compte dans le calcul du débit ne sont pas indiqués dans la présente Recommandation, mais devraient être spécifiés dans des Recommandations utilisant la présente Recommandation.

Chaque transmission de ce message concerne une voie logique spécifique ou l'ensemble du multiplex. Plus d'une de ces commandes peut être valable en même temps, jusqu'à atteindre le nombre de voies logiques plus une, pour donner les limites totales du multiplex.

NOTE – Quand le débit pouvant être transmis sur une voie logique est limité à des valeurs particulières, par exemple celles de l'audio de G.723.1 et quand la transmission est demandée à un débit inférieur au débit le plus faible pour un fonctionnement normal, le terminal répondra en arrêtant la transmission sur la voie logique.

7.11.4 Fin de session

Cette instruction indique la fin de la session H.245. Après avoir transmis EndSessionCommand, le terminal ne devra plus envoyer aucun des messages définis dans la présente Recommandation.

disconnect indique que la connexion sera abandonnée.

gstnOptions: Parmi les options possibles, ce paramètre représente le choix du mode qui sera pris en compte dès la fin de la session H.245, quand un modem de la série V est utilisé sur le réseau RTGC.

Les options possibles sont indiquées dans le Tableau 14.

Tableau 14/H.245 – Options après l'instruction EndSessionCommand quand un modem de la série V est utilisé sur le réseau RTGC

Codage ASN.1	Option
telephonyMode	le terminal doit lancer les procédures de libération définies dans la Recommandation de la série V relative au modem, sauf qu'il ne doit pas libérer physiquement la connexion RTGC
v8bis	le terminal doit déclencher les procédures de libération définies dans la Recommandation de la série V relative au modem et commencer une session V.8 <i>bis</i>
v34v76	le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour assurer le fonctionnement de V.76
v34DuplexFAX	le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour assurer le fonctionnement en mode T.30 FAX [21]
v34H324	le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour assurer le fonctionnement en mode H.324 [18]

7.11.5 Commandes diverses

Ces commandes sont utilisées pour toute une série de commandes, dont certaines sont décrites dans les Recommandations H.221 [5] et H.230 [10].

Le numéro de voie logique `logicalChannelNumber` indique le numéro de la voie logique à laquelle l'instruction s'applique. Ce numéro doit indiquer une voie logique ouverte pour les données vidéo quand les commandes sont du type `videoFreezePicture`, `videoFastUpdatePicture`, `videoFastUpdateGOB`, `videoTemporalSpatialTradeOff`, `videoSendSyncEveryGOB`, `videoFastUpdateMB` et `videoSendSyncEveryGOBCancel`. Quand les commandes sont du type `equaliseDelay`, `zeroDelay`, `multipointModeCommand` ou `cancelMultipointModeCommand` impliquant plusieurs voies logiques, le numéro de voie logique `logicalChannelNumber` devra être arbitraire, mais devra être un numéro `logicalChannelNumber` valable (c'est-à-dire dans la gamme 1-65535) et le récepteur devra omettre la valeur.

Les commandes `equaliseDelay` et `zeroDelay` devront avoir la même signification que les commandes ACE et ACZ définies dans la Recommandation H.230 [10].

La commande `multipointModeCommand` donne l'ordre à un terminal récepteur de répondre à toutes les demandes `requestMode` présentées par l'équipement de commande multipoint (MCU). Un exemple de changement de mode est un changement de mode audio de G.711 à G.728.

La commande `cancelMultipointModeCommand` annule une commande `multipointModeCommand` qui a été préalablement émise.

Le paramètre `videoFreezePicture` donne l'ordre au décodeur vidéo d'effectuer un rafraîchissement accéléré de l'image vidéo réelle et de visualiser ultérieurement l'image gelée jusqu'à la réception du signal approprié de commande de libération d'image gelée.

Le paramètre `videoFastUpdatePicture` donne l'ordre au décodeur vidéo de passer au mode de rafraîchissement accéléré à la première opportunité.

Le paramètre `videoFastUpdateGOB` donne l'ordre au codeur vidéo distant d'effectuer un rafraîchissement accéléré d'au moins un groupe de blocs (GOB). Le premier groupe de blocs (GOB) indique le numéro du premier groupe de blocs (GOB) devant être rafraîchi et le numéro `numberOfGOBs` indique le nombre de groupes de blocs (GOB) devant être rafraîchis. Cet élément ne doit être utilisé que pour des algorithmes de compression de données vidéo définissant des groupes de blocs (GOB), par exemple l'élément selon les Recommandations H.261 et H.263.

Le paramètre `videoTemporalSpatialTradeOff` donne l'ordre au codeur vidéo distant de modifier son compromis entre la définition temporelle et spatiale. Une valeur de 0 régit une définition spatiale élevée et une valeur de 31 régit une valeur de fréquence de trame élevée. Les valeurs de 0 à 31 indiquent de façon monotone qu'une fréquence de trame plus élevée est souhaitée. Les valeurs réelles ne correspondent pas à des valeurs précises de définition spatiale ou de fréquence de trame.

Le paramètre `videoSendSyncEveryGOB` donne l'ordre au codeur vidéo distant d'utiliser la synchronisation pour chaque groupe de blocs (GOB) comme cela est défini dans la Recommandation H.263 [15], jusqu'à ce que la commande `videoSendSyncEveryGOBCancel` soit reçue, moment à partir duquel le codeur vidéo distant peut décider de la fréquence de synchronisation des groupes de blocs (GOB). Ces commandes ne doivent être utilisées qu'avec un codage vidéo conforme à la Recommandation H.263.

Le paramètre `videoFastUpdateMB` donne l'ordre au codeur vidéo distant d'effectuer un rafraîchissement accéléré d'un ou de plusieurs macroblocs. Le paramètre `firstGOB` indique le numéro du premier groupe de blocs (GOB) devant être mis à jour et ne concerne que H.263, le paramètre `firstMB` n'indiquant que le numéro du premier macrobloc devant être mis à jour et ne concernant que la Recommandation H.261. Le paramètre `numberOfMBs` indique le nombre de macroblocs devant être mis à jour. Il ne sera utilisé qu'avec les algorithmes de codage d'image qui définissent les macroblocs, par exemple conformément aux Recommandations H.261 et H.263. Les terminaux peuvent répondre à cette commande avec une mise à jour des groupes de blocs (GOB) comprenant les macroblocs demandés.

La structure `maxH223MUXPDUsize` donne l'ordre à l'émetteur de réduire la taille des paquets MUX-PDU H223 qu'il envoie vers un nombre maximal d'octets qui a été spécifié.

7.11.6 Commande de conférence

La structure `BroadcastMyLogicalChannel` devra être identifiée à MCV H.230 mais devra utiliser exclusivement un canal logique unique.

La structure `CancelBroadcastMyLogicalChannel` devra être identifiée à Cancel-MCV selon H.230 mais devra utiliser exclusivement un canal logique unique.

La structure `MakeTerminalBroadcaster` devra être définie comme VCB selon H.230.

La structure `CancelMakeTerminalBroadcaster` devra être définie comme Cancel-VCB.

Le paramètre `SendThisSource` devra être défini comme VCS selon H.230.

Le paramètre `CancelSendThisSource` devra être défini comme Cancel-VCS selon H.230.

Le paramètre `DropConference` devra être défini comme CCK selon H.230.

7.12 Indications

Une indication contient des informations ne nécessitant ni action ni réponse.

7.12.1 Fonction non comprise

Ce message est utilisé pour renvoyer des demandes, des réponses et des instructions qui n'ont pas été comprises vers l'émetteur.

Si un terminal reçoit une demande, une réponse ou une commande qu'il ne comprend pas, que ce soit parce que celle-ci n'est pas normalisée ou qu'elle ait été définie dans une révision ultérieure de la présente Recommandation, il doit répondre en envoyant `FunctionNotSupported` ou `FunctionNotUnderstood`.

NOTE – Le paramètre `FunctionNotUnderstood` a été désigné par `FunctionNotSupported` dans la version 1996 de la présente Recommandation. Le nom de cette fonction a été modifié pour permettre l'ajout d'une commande `FunctionNotSupported` plus performante sans interrompre la compatibilité ascendante avec la syntaxe de la version 1996.

7.12.2 Indications d'état diverses

Ceci est utilisé pour diverses indications d'état dont certaines sont présentes dans les Recommandations H.221 [5] et H.230 [10].

Le paramètre `logicalChannelNumber` indique le numéro de voie logique auquel s'applique l'indication d'état. Ce numéro doit indiquer un numéro de voie logique ouverte pour les données dans les applications vidéo quand le type est `videoIndicateReadyToActivate` et `videoTemporalSpatialTradeOff`. Quand les paramètres sont du type `multipointConference`, `cancelMultipointConference`, `multipointZeroComm`, `cancelMultipointZeroComm`, `multipointSecondaryStatus` ou `cancelMultipointSecondaryStatus` impliquant l'utilisation de plusieurs voies logiques, le paramètre `logicalChannelNumber` devra être choisi arbitrairement et devra être mis à une valeur `LogicalChannelNumber` correcte (c'est-à-dire dans la gamme 1 à 65535) et le récepteur devra omettre cette valeur.

La structure `logicalChannelInactive` est utilisée pour indiquer que le contenu du canal logique ne représente pas un signal normal. Elle est analogue à AIM ou à VIS définis dans H.230.

La structure `logicalChannelActive` est complémentaire de la structure précédente. Elle est analogue à AIA ou à VIA définis dans H.230. `MultipointZeroComm`, `cancelMultipointZeroComm`, `multipointSecondaryStatus` et `cancelMultipointSecondaryStatus` devront avoir la même signification que MIZ, `cancelMIS`, MIS et `cancelMIS` respectivement, comme cela a été défini dans la Recommandation H.230.

Le paramètre `multipointConference` indique que le terminal est associé à une conférence multipoint H.243 et qu'il est prévu que le terminal obéisse à une symétrisation du débit. La symétrisation du débit doit cependant être mise en application par les messages de commande de contrôle de flux `FlowControlCommand`. Il convient de noter que le paramètre `multipointConference` a exactement la même signification que le paramètre MCC dans la Recommandation H.230. Il convient de noter également que le paramètre `multipointConference`, comme le paramètre MCC, n'exige pas de symétrie de mode.

Le paramètre `videoIndicateReadyToActivate` devra avoir la même signification que le VIR défini dans la Recommandation H.230, c'est-à-dire qu'il est transmis par un terminal pour lequel l'utilisateur a décidé de ne pas envoyer la vidéo avant d'avoir également reçu la vidéo à partir de l'autre terminal.

La structure `videoTemporalSpatialTradeOff` indique au décodeur vidéo distant quel est le compromis actuel entre la résolution temporelle et la résolution spatiale. Une valeur de 0 indique une résolution spatiale élevée et une valeur de 31 indique une fréquence image élevée. Les valeurs comprises entre 0 et 31 indiquent de façon monotone un taux d'image élevé. Les valeurs réelles ne correspondent pas à des valeurs précises de résolution spatiale de taux d'image. Un terminal qui a déclaré une capacité `temporalSpatialTradeOffCapability` devra transmettre cette indication à chaque fois que le compromis change et quand un canal logique vidéo est initialement ouvert.

L'élément `videoNotDecodedMBs` indique au codeur vidéo distant qu'un ensemble de macroblocs a été reçu de façon erronée et que n'importe quel macrobloc dans l'ensemble spécifié a été traité comme non codé. Le codeur peut utiliser ces informations pour compenser les erreurs de transmission, comme cela est illustré dans l'Appendice I/H.263. Le paramètre `firstMB` indique le numéro du premier macrobloc traité comme étant non codé, et le paramètre `numberOfMBs` indique le nombre de macroblocs traités comme étant non codés. La numérotation des macroblocs est effectuée selon la Recommandation H.263. La référence temporelle de l'image contenant des

macroblocs non décodés est indiquée dans le paramètre `temporalReference`. Cette indication devra être uniquement utilisée avec l'algorithme de codage d'image selon la Recommandation H.263.

7.12.3 Indication de gigue

On l'utilise pour indiquer la quantité de gigue d'une voie logique telle qu'elle a été évaluée par le terminal récepteur. Cela peut être utile pour choisir le débit et la gestion de la mémoire tampon des canaux vidéo, ou pour définir la fréquence de transmission appropriée des informations de rythme, etc. Le codeur vidéo aura alors le choix entre utiliser ces informations pour restreindre le débit vidéo, le surremplissage ou le sous-remplissage de la mémoire tampon du décodeur, compte tenu de la gigue existante. Si le codeur accepte cette option, cela permettra le fonctionnement des modèles existants de mémoires tampons de décodeurs vidéo, quelle que soit l'amplitude de la gigue reçue, de même que le bon fonctionnement avec un retard minimal.

Quand le domaine d'application est du type `logicalChannelNumber`, les informations s'appliquent à une voie logique donnée; quand le domaine d'application est du type `virtualChannelID`, les informations s'appliquent à la voie virtuelle (VC) de l'ATM donné; et quand le domaine d'application est du type `wholeMultiplex`, les informations s'appliquent à l'ensemble du multiplex.

Les éléments `estimatedReceivedJitterMantissa` et `estimatedReceivedJitterExponent` permettent d'évaluer la gigue reçue par le terminal qui a envoyé le message.

L'élément `estimatedReceivedJitterMantissa` indique la mantisse de l'évaluation de gigue telle qu'elle est indiquée dans le Tableau 15.

Tableau 15/H.245 – Mantisse de `estimatedReceivedJitterMantissa` dans l'indication de gigue `JitterIndication`

<code>estimatedReceivedJitterMantissa</code>	Mantisse
0	1
1	2,5
2	5
3	7,5

`estimatedReceivedJitterExponent` indique l'exposant de l'évaluation de gigue tel qu'il est indiqué dans la Tableau 16.

Tableau 16/H.245 – Exposant de estimatedReceivedJitterExponent dans l'indication de gigue JitterIndication

estimatedReceivedJitterExponent	Exposant
0	hors gamme
1	1 μ s
2	10 μ s
3	100 μ s
4	1 ms
5	10 ms
6	100 ms
7	1 s

L'estimation de gigue est obtenue en multipliant la mantisse par l'exposant, à moins que l'exposant estimatedReceivedJitterExponent soit égal à zéro, auquel cas on estime que la gigue est supérieure à 7,5 secondes.

Le compteur skippedFrameCount indique le nombre de trames qui a été omis par le décodeur depuis que le dernier message JitterIndication a été reçu. Etant donné que la valeur maximale de codage est égale à 15, si cette option est mise en œuvre, cette information doit être transmise avant qu'un nombre de trames supérieur à 15 ait été omis.

NOTE – Etant donné que des trames sont omises quand il y a sous-remplissage de la mémoire tampon du décodeur, une gigue supplémentaire peut provoquer un sous-remplissage de cette mémoire tampon plus ou moins fréquemment par rapport aux omissions de trames prévues par le codeur.

L'élément additionalDecoderBuffer indique la taille supplémentaire de la mémoire tampon du décodeur vidéo au-dessus ou en dessous de celle exigée par le profil et le niveau indiqués. Ceci est défini de la même façon que vbv_buffer_size H.262 [14].

7.12.4 Indication de décalage temporel selon H.223

Ce message est utilisé pour indiquer au terminal distant la quantité moyenne de retard temporel entre deux voies logiques.

Les numéros logicalChannelNumber1 et logicalChannelNumber2 sont des numéros de voies logiques ouvertes.

Le décalage temporel se mesure en millisecondes et indique le retard devant être appliqué aux données appartenant à la voie logique logicalChannelNumber2, observée à la sortie du multiplex, pour réaliser la synchronisation avec la voie logique logicalChannelNumber1, observée à la sortie du multiplex. Le décalage temporel inclut les différences des instants d'échantillonnage, le retard du codeur, de même que le retard de la mémoire tampon de l'émetteur, et est mesuré par rapport au temps de transmission du premier bit de données représentant un échantillon donné. Le retard réel nécessaire à la synchronisation dépend de la réalisation du décodeur et doit être considéré localement lors de la conception du récepteur.

7.12.5 Nouvelle indication de voie virtuelle ATM

Ce message est utilisé pour indiquer les paramètres de voie virtuelle ATM que le terminal a l'intention d'ouvrir.

L'identificateur resourceID est utilisé pour identifier la voie virtuelle ATM. Le moyen par lequel ce paramètre est associé à une voie virtuelle ATM n'est pas spécifié dans la présente Recommandation.

Le paramètre `bitRate` indique le débit, mesuré au niveau de la couche AAL-SAP, de la voie virtuelle, et est mesuré en unités de 64 kbit/s.

La structure `bitRateLockedToPCRClock` indique que le débit de la voie virtuelle est rythmé sur l'horloge utilisée pour produire les valeurs de référence d'horloge H.222.0 (référence d'horloge de programme ou référence d'horloge de système).

La structure `bitRateLockedToNetworkClock` indique que le débit de la voie virtuelle est rythmé sur l'horloge de réseau local. Cela ne garantit pas que le rythme de débit sera aligné sur le réseau local du récepteur, étant donné que des horloges de réseau peuvent ne pas être disponibles.

Le paramètre `aal` indique que la couche d'Adaptation de l'ATM sera utilisée, de même que ses paramètres.

La séquence `aal1` de séquence indique quelles options de la couche d'Adaptation 1 de l'ATM, comme cela est spécifié dans la Recommandation I.363 [19], sont utilisées. Les codages sont définis dans le Tableau 1.

La séquence `aal5` indique quelles options de la couche d'Adaptation 5 de l'ATM, comme cela est spécifié dans la Recommandation I.363 [19], sont utilisées. Les paramètres `forwardMaximumSDUSize` et `backwardMaximumSDUSize` indiquent la taille CPCS-SDU maximale dans les sens direct et inverse, mesurée en octets.

Le multiplex indique le type de multiplex qui sera utilisé sur la voie virtuelle ATM. Les options sont `noMultiplex` (pas de multiplex H.222.0), le flux de transport H.222.0 et le flux de programme H.222.0.

7.12.6 Messages d'entrée de l'utilisateur

Ce message est utilisé pour les messages d'entrée de l'utilisateur.

Le mode alphanumérique correspond à une chaîne de caractères codés selon la Recommandation T.51 [23]. Ceci pourrait être utilisé pour les messages émis à partir du clavier de l'utilisateur, équivalant à l'ensemble des codes multifréquence DTMF.

UserInputSupportIndication: cette structure indique au terminal distant quels types de chaînes GENERALSTRING sont acceptées par le terminal.

NOTE – Il est prévu que la plupart des réalisations de décodeurs PER ne pourront pas décoder d'autres chaînes que celles de l'alphabet IA5. Cette indication devrait être utilisée pour "prévenir" le terminal distant de ne pas essayer d'élaborer des algorithmes de codage en longueur variable.

Le paramètre `nonStandard` est un paramètre `NonStandardParameter` indiquant l'usage non normalisé du message d'indication `UserInput`.

Le paramètre booléen `basicString`, quand il est égal à la valeur "vrai", indique que les paramètres 0-9,* et # sont utilisés.

Le paramètre booléen `ia5String`, quand il est mis à la valeur vrai, indique que le jeu complet des caractères IA5String est utilisé.

Le paramètre booléen `generalString`, quand il est mis à la valeur vrai, indique que l'ensemble complet des caractères GeneralString est utilisé.

NOTE – Toutes les données qui sont transmises dans la présente Recommandation, y compris les messages d'entrée de l'utilisateur, ne seront pas chiffrées.

7.12.7 Indications de conférence

Le paramètre `sbeNumber` devra être défini comme numéro SBE selon la Recommandation H.230.

Le paramètre `terminalNumberAssign` devra être défini comme l'indication TIA selon la Recommandation H.230.

Le paramètre `terminalJoinedConference` devra être défini comme l'indication TIN selon la Recommandation H.230.

Le paramètre `terminalLeftConference` devra être défini comme l'indication TID selon la Recommandation H.230.

Le paramètre `seenByAtLeastOneOther` devra être défini comme l'indication MIV selon la Recommandation H.230.

Le paramètre `cancelSeenByAtLeastOneOther` devra être défini comme l'indication `cancel-MIV` selon la Recommandation H.230.

Le paramètre `seenByAll` devra être défini comme l'indication MIV selon la Recommandation H.230.

Le paramètre `cancelSeenByAll` devra être défini comme l'indication MIV selon la Recommandation H.230.

Le paramètre `terminalYouAreSeeing` devra être défini comme l'indication TIN selon la Recommandation H.230.

Le paramètre `requestForFloor` devra être défini comme l'indication TIF selon la Recommandation H.230.

7.12.8 Décalage temporel maximal sur la voie logique H2250

Le paramètre `H2250MaximumSkewIndication` indique le décalage temporel maximal entre les voies logiques.

Le décalage temporel se mesure en millisecondes et indique le nombre maximal de millisecondes pendant lequel les données sur la voie `logicalChannelNumber2` sont retardées par rapport aux données sur la voie `logicalChannelNumber1` telles qu'elles sont transmises vers le réseau de transport. Le décalage temporel se mesure par rapport à l'instant de transmission au réseau de transport du premier bit de données représentant un échantillon donné. La synchronisation de l'audio par rapport au mouvement des lèvres, si elle est souhaitée, dépend du récepteur et devra être obtenue par l'utilisation d'horodateurs.

7.12.9 Indication d'emplacement MC

Cette indication est envoyée par l'entité de commande multipoint (MC) pour indiquer à d'autres terminaux l'adresse de signalisation qui devrait être utilisée pour atteindre l'entité de commande multipoint (MC).

7.12.10 Indication de l'identification du fournisseur

L'indication `vendorIdentification` devrait être envoyée au début de chaque appel pour identifier le fabricant, le produit et le numéro de version du produit.

7.12.11 Fonction non prise en compte

Ce message est utilisé pour renvoyer à l'émetteur les demandes, les réponses et les commandes qui n'ont pas été comprises.

L'ensemble du message `RequestMessage`, `ResponseMessage` ou `CommandMessage` est renvoyé.

Si un terminal reçoit une demande, une réponse ou une commande qu'il ne comprend pas, soit parce qu'il s'agit d'un message non normalisé ou parce que ce message a été défini comme une révision

ultérieure de la présente Recommandation, il devra répondre en envoyant le message `FunctionNotSupported`.

Si un terminal reçoit une demande, une réponse ou une commande ayant un codage incorrect, il mettra la valeur `syntaxError` dans le paramètre de raison. Si le terminal a un codage correct, mais si les valeurs codées sont sémantiquement incorrectes, le terminal mettra la valeur `semanticError` dans le paramètre de raison. Si le message est une extension non reconnue vers l'un des messages `MultimediaSystemControlMessage`, `RequestMessage`, `ResponseMessage` ou `CommandMessage`, le terminal devra mettre la valeur `unknownFunction` dans le paramètre de raison.

Dans chacun des cas, l'ensemble du message `MultimediaSystemControlMessage` devrait être renvoyé comme chaîne d'octets dans le paramètre `returnedFunction`.

Le message `FunctionNotSupported` ne devra pas être utilisée à un autre moment. En particulier, quand une extension non reconnue est présente à d'autres points de la syntaxe, le message `FunctionNotSupported` ne devra pas être utilisé; le terminal devra répondre au message de façon normale, comme si aucune extension n'était présente. La structure `FunctionNotSupported` ne devra jamais être envoyée en réponse à une indication reçue.

8 Procédures

8.1 Introduction

Le présent sous-paragraphe définit les procédures génériques de commande de système multimédia utilisant les messages définis dans la présente Recommandation. Les Recommandations utilisant la présente Recommandation doivent indiquer quelles sont les procédures applicables, et définir également quelles sont toutes les spécifications qui s'y rattachent.

Les procédures destinées à effectuer les fonctions suivantes sont décrites dans le présent sous-paragraphe:

- choix du mode maître ou esclave;
- échange de capacités du terminal;
- signalisation de la voie logique monodirectionnelle;
- signalisation de la voie logique bidirectionnelle;
- demande de fermeture de la voie logique du terminal de réception;
- modification d'entrée de tableau de multiplexage selon H.223;
- demande d'entrée de multiplexage;
- demande de mode d'émission de récepteur à émetteur;
- définition du temps de propagation aller et retour;
- boucle de maintenance.

8.1.1 Méthode de spécification

Les procédures sont généralement spécifiées dans le présent sous-paragraphe en utilisant le langage SDL. Le langage SDL apporte une spécification des procédures sous forme de diagrammes et inclut une spécification des mesures à prendre en cas d'interruption de déroulement.

8.1.2 Communication entre l'entité de protocole et l'utilisateur de protocole

L'interaction avec l'utilisateur d'une fonction particulière est spécifiée en termes de primitives transférées à l'interface entre l'entité de protocole et l'utilisateur du protocole. Les primitives ont pour

objectif de définir des procédures de protocole et non pas de spécifier ou de rendre obligatoire une réalisation particulière. Il peut y avoir un nombre de paramètres associés à chaque primitive.

Afin d'apporter une contribution à cette spécification, des états sont définis dans les protocoles. Ils sont conceptuels et reflètent l'état général de l'entité de protocole dans les séquences de primitives échangées entre l'entité de protocole et l'utilisateur, de même que dans l'échange de messages entre l'entité de protocole et l'entité homologue.

Pour chacune des entités de protocole, la séquence autorisée de primitives entre l'utilisateur et l'entité de protocole est définie en utilisant un diagramme de changement d'état. La séquence autorisée apporte des contraintes aux actions de l'utilisateur et définit les réponses possibles de l'entité de protocole.

Un paramètre de primitive décrit comme étant nul équivaut à l'absence de ce paramètre.

8.1.3 Communication entre entités homologues

Les informations de protocole sont transférées vers l'entité de protocole homologue par l'intermédiaire des messages pertinents définis dans le paragraphe 6. Certaines entités de protocole décrites ont des variables d'état qui leur sont associées. Un certain nombre d'entités de protocoles décrites comportent également des temporisateurs qui leur sont associés.

Un temporisateur est décrit par la notation T_n , n étant un numéro. Dans les diagrammes SDL, l'armement du temporisateur signifie qu'un temporisateur est chargé à une valeur spécifiée et la temporisation démarre. La suspension de temporisation signifie qu'un temporisateur est arrêté et que sa valeur au moment de la suspension est retenue. La fin de temporisation signifie qu'un temporisateur a fonctionné pendant la durée spécifiée et qu'il a atteint la valeur de zéro.

Une entité de protocole peut également avoir des paramètres associés. Un paramètre est identifié par la notation N_n , n désignant un nombre.

Ces temporisateurs et compteurs sont énumérés dans l'Appendice III.

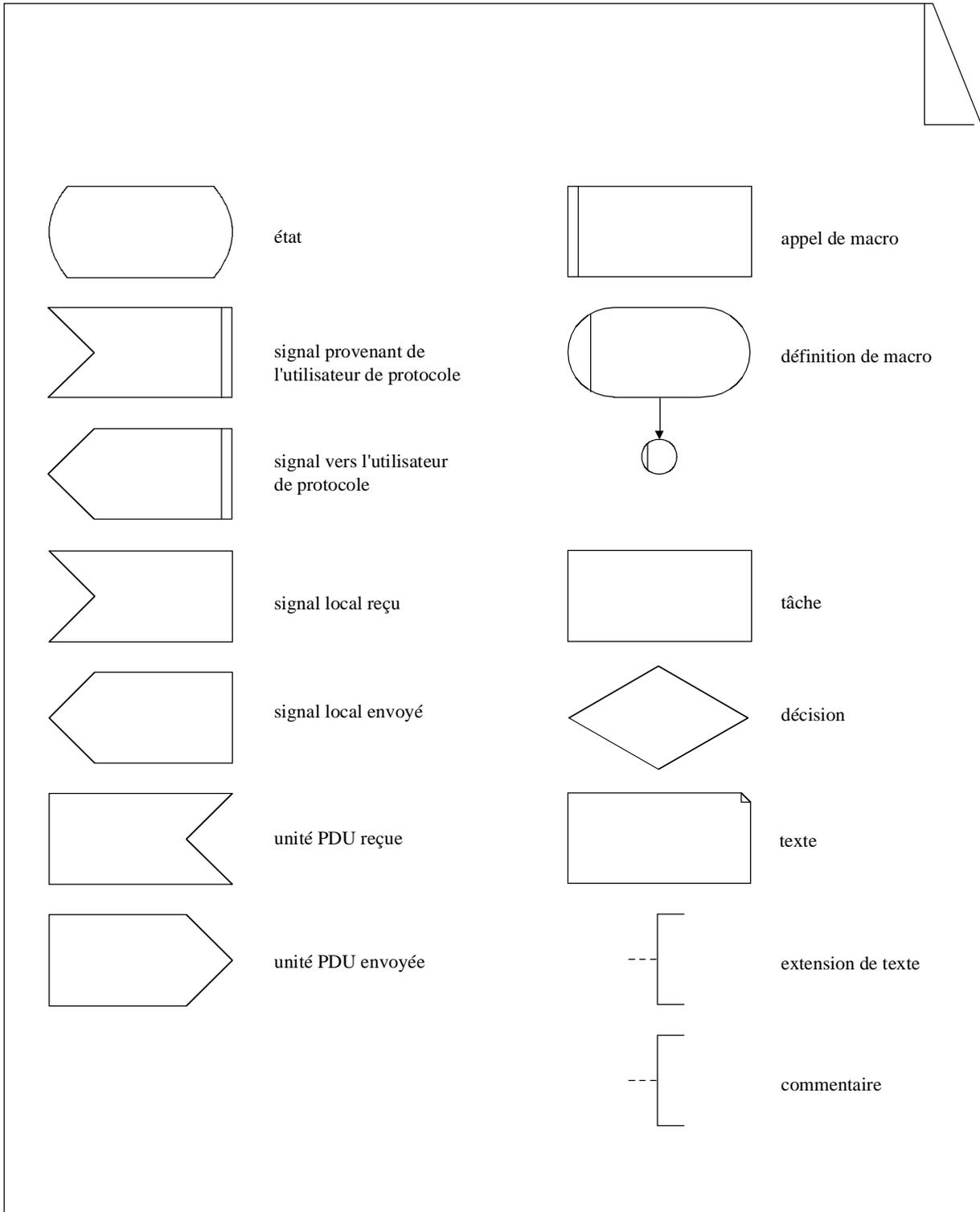
Certaines entités de protocole définissent une primitive d'erreur pour indiquer les états d'erreur de protocole à une entité de gestion.

8.1.4 Diagrammes SDL

Les diagrammes SDL indiquent les actions à prendre dans les interactions autorisées avec l'utilisateur du protocole, de même qu'à la réception de messages provenant de l'entité de protocole homologue. Les primitives qui ne sont pas autorisées pour un état donné ne sont pas indiquées dans les diagrammes SDL. Les réponses à la réception de messages non pertinents sont cependant décrites dans les diagrammes SDL.

8.1.5 Symboles du langage SDL

Les symboles du langage SDL sont décrits dans la Figure 1.



T1519120-95

Figure 1/H.245 – Symboles du langage SDL

8.2 Procédures de choix du mode maître ou esclave

8.2.1 Introduction

Des difficultés peuvent survenir quand au moins deux des terminaux mis en communication déclenchent des événements similaires simultanément, pour lesquels des ressources sont disponibles pour une seule manifestation de l'événement, comme par exemple l'ouverture de voies logiques. Pour résoudre de telles difficultés, un des terminaux peut réagir comme le terminal maître et l'autre (les autres terminaux) peuvent réagir comme le terminal esclave ou les terminaux esclaves. Les procédures décrites ci-dessous permettent aux terminaux en communication de choisir quel sera le terminal maître et quel sera le terminal esclave.

Le protocole décrit ici est désigné comme l'entité de signalisation permettant le choix du mode maître ou esclave (MSDSE, *master slave determination signalling entity*). Il y a une instance d'entité MSDSE dans chaque terminal mis en communication.

L'un des terminaux peut déclencher la procédure de choix du mode maître ou esclave en présentant la primitive de demande DETERMINE à son entité MSDSE. Le résultat de la procédure est renvoyé par la primitive d'indication DETERMINE et les primitives de confirmation DETERMINE. Alors que la primitive d'indication DETERMINE indique le résultat, elle n'indique pas que le résultat est connu du terminal distant. La primitive de confirmation DETERMINE indique le résultat et confirme que ce résultat est également connu du terminal distant.

Un terminal devra répondre à des procédures faisant référence à un résultat connu et sont déclenchées à tout moment par le terminal distant après que le résultat du choix du mode soit connu du terminal local. Ceci peut avoir lieu avant que le terminal local ait reçu confirmation que le terminal distant a également connaissance du résultat. Un terminal ne devra pas déclencher des procédures qui dépendent de la connaissance du résultat avant d'avoir reçu confirmation que le terminal distant a également eu connaissance du résultat.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

8.2.1.1 Aperçu général du protocole – Initialisation par l'utilisateur local

Une procédure de choix du mode maître ou esclave est déclenchée quand la primitive de demande DETERMINE est présentée par l'utilisateur de l'entité MSDSE. Un message MasterSlaveDetermination est envoyé à l'entité MSDSE homologue et le temporisateur T106 démarre. Si un message MasterSlaveDeterminationAck est reçu en réponse au message MasterSlaveDetermination, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive de confirmation DETERMINE que la procédure de choix du mode maître ou esclave a réussi et un message MasterSlaveDeterminationAck est envoyé à l'entité MSDSE homologue. Si, cependant, un message MasterSlaveDeterminationReject est reçu en réponse à un message MasterSlaveDetermination, alors un nouveau numéro de choix de mode est créé, la temporisation T106 redémarre et un autre message MasterSlaveDetermination est envoyé. Si après l'envoi d'un message MasterSlaveDetermination un nombre de fois égal à N100 fois, un accusé de réception MasterSlaveDeterminationAck n'a toujours pas été reçu, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que la procédure de choix du mode maître ou esclave n'a pas eu le résultat escompté.

Si le temporisateur T106 vient à expiration, alors l'utilisateur de l'entité MSDSE est informé par la primitive d'indication REJECT et un message MasterSlaveDeterminationRelease est envoyé à l'entité MSDSE homologue.

8.2.1.2 Aperçu général du protocole – Initialisation par l'utilisateur distant

Quand un message `MasterSlaveDetermination` est reçu par l'entité MSDSE, une procédure de choix de mode est déclenchée. Si la procédure de choix de mode renvoie un résultat déterminé, alors l'utilisateur est informé du résultat du choix de mode maître ou esclave par la primitive d'indication `DETERMINE`; un message `MasterSlaveDeterminationAck` est envoyé à l'entité MSDSE homologue et le temporisateur T106 démarre. Si un message `MasterSlaveDeterminationAck` est reçu en réponse au message `MasterSlaveDeterminationAck`, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive de confirmation `DETERMINE` que la procédure de choix du mode maître ou esclave a réussi.

Si le temporisateur T106 vient à expiration, l'utilisateur MSDSE est informé par la primitive d'indication `REJECT`.

Si, cependant, la procédure de choix de mode revient à un résultat indéterminé, alors le message `MasterSlaveDeterminationReject` est envoyé à l'entité MSDSE homologue.

8.2.1.3 Aperçu général du protocole – Initialisation simultanée

Quand un message `MasterSlaveDetermination` est reçu par l'entité MSDSE qui a elle-même déclenché une procédure de choix de mode et attend un message d'acquiescement `MasterSlaveDeterminationAck` ou de refus `MasterSlaveDeterminationReject`, alors une procédure de choix de statut est déclenchée. Si la procédure de choix de mode renvoie un résultat de choix, l'entité MSDSE répond comme si la procédure était déclenchée par l'utilisateur distant et les procédures décrites ci-dessus pour cet état sont applicables.

Si, cependant, la procédure de choix du mode renvoie un résultat indéterminé, alors un nouveau numéro de choix de mode est créé et l'entité MSDSE répond comme si la procédure était de nouveau déclenchée par l'utilisateur de l'entité MSDSE locale comme cela est décrit ci-dessus.

8.2.1.4 Procédure de choix du mode

La procédure suivante est utilisée pour déterminer quel terminal est le terminal maître à partir des valeurs des paramètres `terminalType` et `statusDeterminationNumber`. En premier lieu, les valeurs `terminalType` sont comparées et le terminal ayant le numéro de type de terminal le plus grand est défini comme étant le terminal maître. Si les numéros de type de terminal sont les mêmes, les numéros `statusDeterminationNumbers` sont comparés en utilisant l'arithmétique modulo pour choisir quel sera le terminal maître.

Si les deux terminaux ont des valeurs de champ `terminalType` égales et si la différence entre les valeurs de champ `statusDeterminationNumber` modulo 2^{24} est 0 ou 2^{23} , un résultat indéterminé est obtenu.

8.2.2 Communication entre l'entité MSDSE et l'utilisateur de l'entité MSDSE

8.2.2.1 Primitives entre l'entité MSDSE et l'utilisateur de l'entité MSDSE

Les communications entre l'entité MSDSE et l'utilisateur de l'entité MSDSE sont effectuées en utilisant les primitives indiquées dans le Tableau 17.

Tableau 17/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
DETERMINE (choisir)	– (Note 1)	TYPE	non défini (Note 2)	TYPE
REJECT	non défini	–	non défini	non défini
ERROR (erreur)	non défini	ERRCODE	non défini	non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

8.2.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) la primitive DETERMINE (choisir) est utilisée pour lancer la procédure de choix du mode maître ou esclave et pour renvoyer le résultat obtenu à partir de cette même procédure.

La primitive de demande DETERMINE est utilisée pour lancer la procédure de choix du mode maître ou esclave.

La primitive d'indication DETERMINE.indication est utilisée pour indiquer le résultat de la procédure de choix du mode maître ou esclave. Etant donné que le résultat de la procédure peut ne pas être connu du terminal distant, le terminal ne devra pas lancer des procédures qui dépendent de la connaissance du résultat, bien qu'il doive répondre à toutes les procédures dépendant de la connaissance du résultat.

La primitive de confirmation DETERMINE est utilisée pour indiquer le résultat de la procédure de choix du mode maître ou esclave et pour indiquer que la procédure est connue des deux terminaux. Le terminal peut déclencher et répondre à toutes les procédures qui dépendent de la connaissance du résultat;

- b) la primitive REJECT indique que la procédure de choix du mode maître ou esclave a échoué;
- c) la primitive ERROR signale les erreurs de l'entité MSDSE à une entité de gestion.

8.2.2.3 Définition des paramètres

La définition des paramètres de primitives indiqués dans le Tableau 17 est la suivante:

- a) le paramètre TYPE indique l'état du terminal. Il a la valeur de "MASTER" (maître) ou "SLAVE" (esclave);
- b) la valeur ERRCODE indique le type d'erreur de l'entité MSDSE. Le Tableau 21 indique les valeurs que le paramètre ERRCODE peut prendre.

8.2.2.4 Etats de l'entité MSDSE

Les états suivants sont utilisés pour spécifier la séquence autorisée des primitives entre l'entité MSDSE et l'utilisateur de l'entité MSDSE.

Etat 0: IDLE (repos).

Aucune procédure de choix du mode maître ou esclave n'a été déclenchée.

Etat 1: OUTGOING AWAITING RESPONSE (entité sortante en attente de réponse)

L'utilisateur de l'entité MSDSE locale a demandé une procédure de choix du mode maître ou esclave. Une réponse est attendue de l'entité MSDSE distante.

Etat 2: INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)

L'utilisateur de l'entité MSDSE distante a lancé la procédure de choix du mode maître ou esclave dans l'entité MSDSE locale. Un accusé de réception a été envoyé à l'entité MSDSE distante et une réponse est attendue en provenance de l'entité MSDSE distante.

8.2.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité MSDSE et l'utilisateur de l'entité MSDSE est définie ici. Les séquences autorisées sont indiquées à la Figure 2.

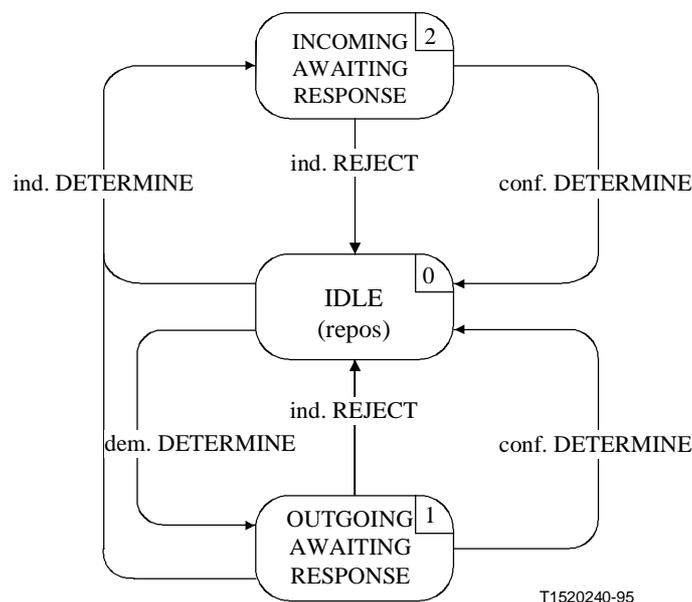


Figure 2/H.245 – Diagramme de changement d'états pour la séquence de primitives dans l'entité MSDSE

8.2.3 Communication entre les entités MSDSE homologues

8.2.3.1 Messages MSDSE

Le Tableau 18 indique les messages et les champs des entités MSDSE tels qu'ils ont été définis dans le paragraphe 6 et qui relèvent du protocole spécifique aux entités MSDSE.

Tableau 18/H.245 – Noms et champs des messages des entités MSDSE

Fonction	Message	Champ
determination (choix)	MasterSlaveDetermination	terminalType statusDeterminationNumber
	MasterSlaveDeterminationAck	decision
	MasterSlaveDeterminationReject	cause
reprise sur erreur	MasterSlaveDeterminationRelease	–

8.2.3.2 Variables d'état d'une entité MSDSE

Les variables d'état suivantes de l'entité MSDSE sont ainsi définies:

sv_TT

Cette variable d'état contient le numéro de type pour ce terminal.

sv_SDNUM

Cette variable d'état contient le numéro de choix du mode pour ce terminal.

Sv_STATUS

Cette variable d'état est utilisée pour enregistrer le résultat de la dernière procédure de choix du mode maître ou esclave. Elle a les valeurs de "maître", "esclave" et "indéterminé".

Sv_NCOUNT

Cette variable d'état est utilisée pour compter le nombre de messages MasterSlaveDetermination qui ont été envoyés pendant l'état OUTGOING AWAITING RESPONSE (ATTENTE DE LA RÉPONSE SORTANTE).

8.2.3.3 Temporisateurs MSDSE

Le temporisateur suivant est spécifié pour l'entité MSDSE sortante:

T106

Ce temporisateur est utilisé pendant l'état OUTGOING AWAITING RESPONSE et durant l'état INCOMING AWAITING RESPONSE. Il spécifie le temps maximal autorisé pendant lequel aucun message d'acquiescement ne peut être reçu.

8.2.3.4 Compteurs MSDSE

Le paramètre suivant est spécifié pour l'entité MSDSE:

N100

Ce paramètre spécifie la valeur maximale de sv_NCOUNT.

8.2.4 Procédures MSDSE

8.2.4.1 Introduction

La Figure 3 récapitule les primitives MSDSE et leurs paramètres, ainsi que les messages.

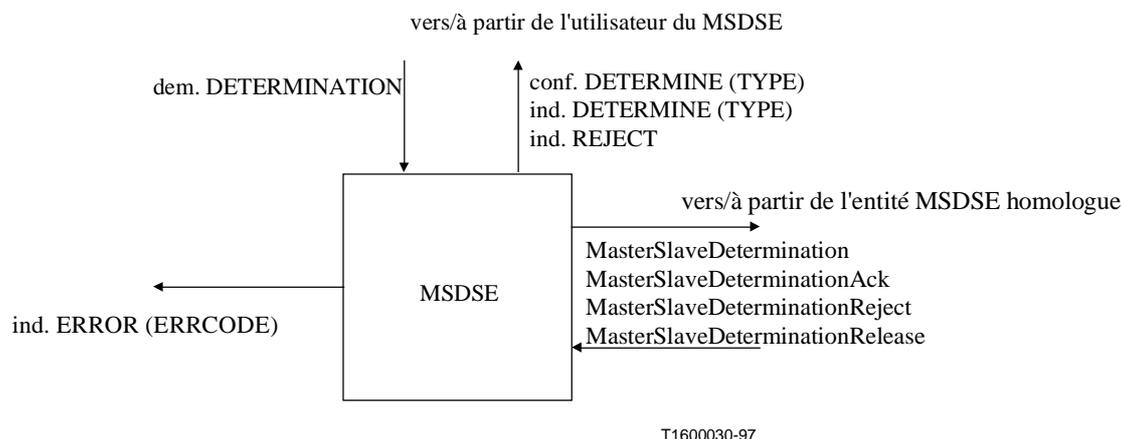


Figure 3/H.245 – Primitives et messages dans l'entité MSDSE

8.2.4.2 Valeurs par défaut des paramètres de primitives

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent les valeurs indiquées dans le Tableau 19.

Tableau 19/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut
conf. DETERMINE	TYPE	MasterSlaveDeterminationAck.decision
ind. DETERMINE	TYPE	sv_STATUS

8.2.4.3 Valeurs par défaut des champs de message

Quand cela n'est pas explicitement indiqué dans les messages SDL, les champs de messages prennent les valeurs indiquées dans le Tableau 20.

Tableau 20/H.245 – Valeurs par défaut des champs de messages

Message	Champ	Valeur par défaut
MasterSlaveDetermination	terminalType statusDetermination Number	sv_TT sv_SDNUM
MasterSlaveDeterminationAck	decision	opposé de sv_STATUS, c'est-à-dire si (sv_STATUS == master) choix = esclave si (sv_STATUS == slave) choix = maître
MasterSlaveDeterminationReject	cause	identicalNumbers

8.2.4.4 Valeurs des paramètres ERRCODE

Le Tableau 21 indique les valeurs que le paramètre ERRCODE de la primitive d'indication ERROR peut prendre pour l'entité MSDSE.

Tableau 21/H.245 – Valeurs des paramètres ERRCODE dans l'entité MSDSE

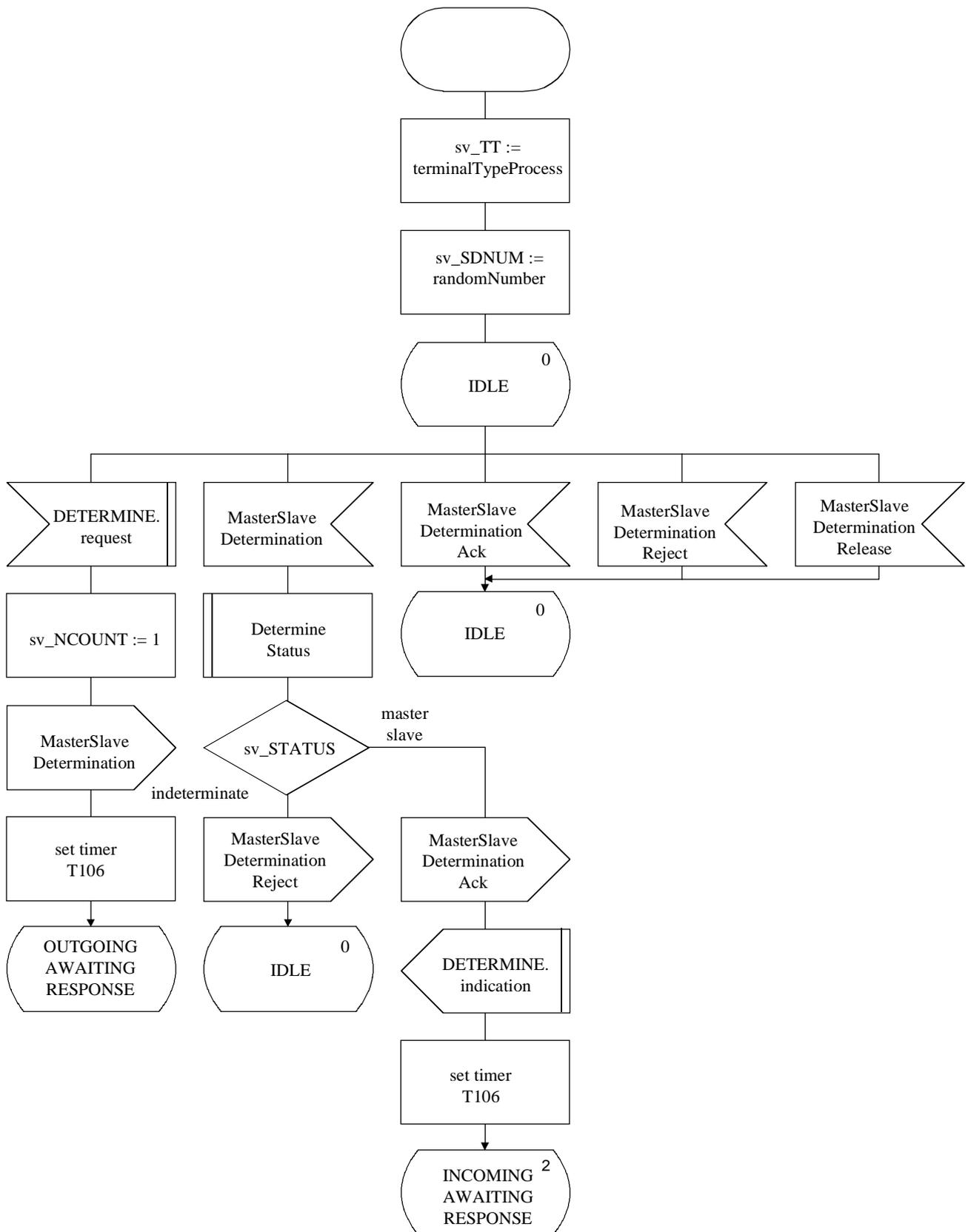
Type d'erreur	Code d'erreur	Condition d'erreur	Etat
pas de réponse de l'entité MSDSE distante	A	fin de temporisation locale T106	OUTGOING AWAITING RESPONSE (entité sortante en attente de réponse) INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)
le terminal distant ne voit pas la réponse de l'entité MSDSE locale	B	fin de temporisation distante T106	OUTGOING AWAITING RESPONSE (entité sortante en attente de réponse) INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)
message inapproprié	C	MasterSlaveDetermination	INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)
	D	MasterSlaveDetermination Reject	INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)
valeur de champ incohérente	E	MasterSlaveDetermination Ack.decision != sv_STATUS	INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)
nombre maximal de tentatives	F	sv_NCOUNT == N100	OUTGOING AWAITING RESPONSE (entité sortante en attente de réponse)

8.2.4.5 Description SDL

Les procédures de l'entité MSDSE sont exprimées sous forme de langage SDL dans la Figure 4.

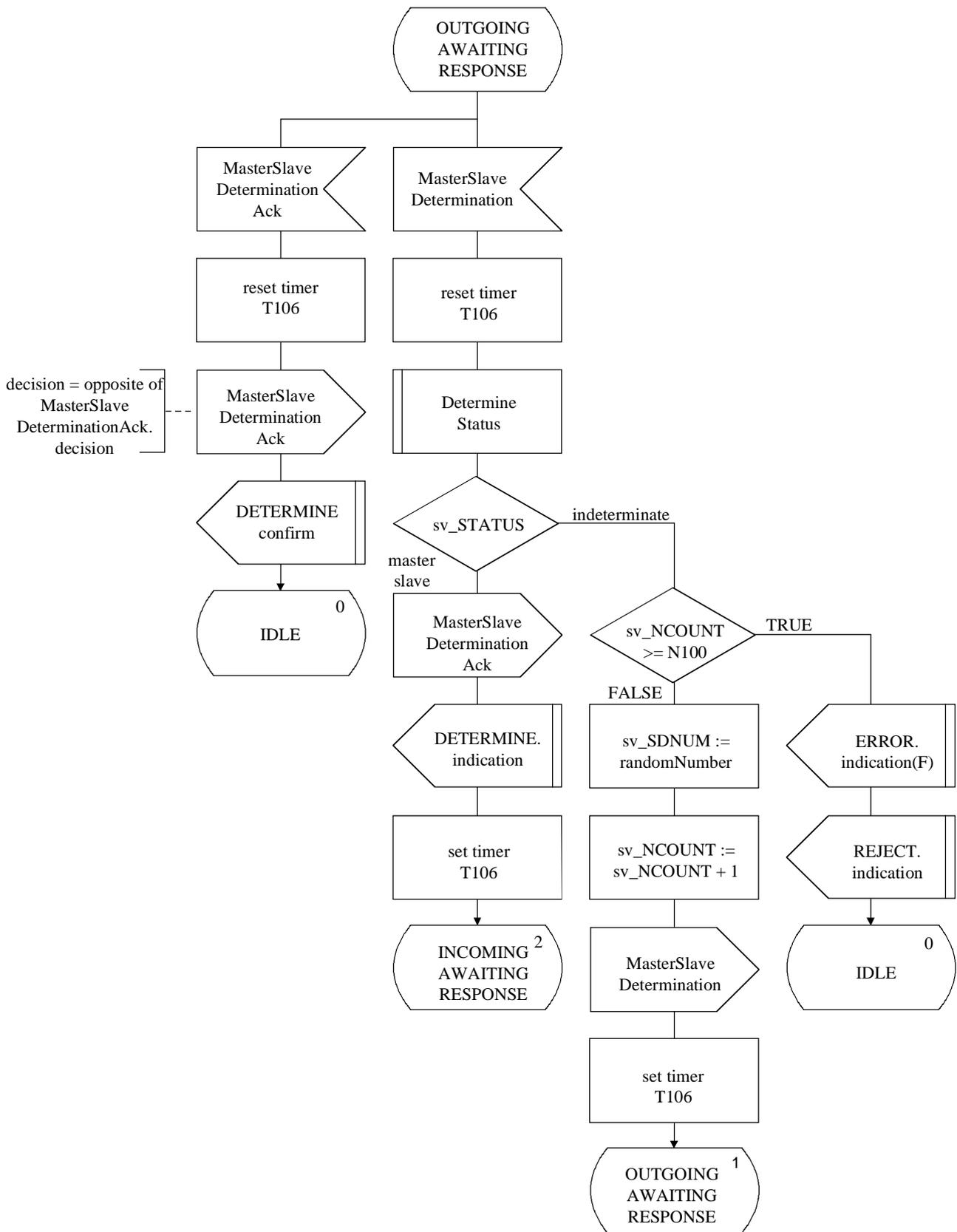
La structure terminalTypeProcess est une procédure qui renvoie un nombre identifiant différents types de terminaux tels que les terminaux, les équipements de commande multipoint et les passerelles.

La structure randomNumber est une procédure qui renvoie un nombre aléatoire dans la gamme $0..2^{24}-1$.



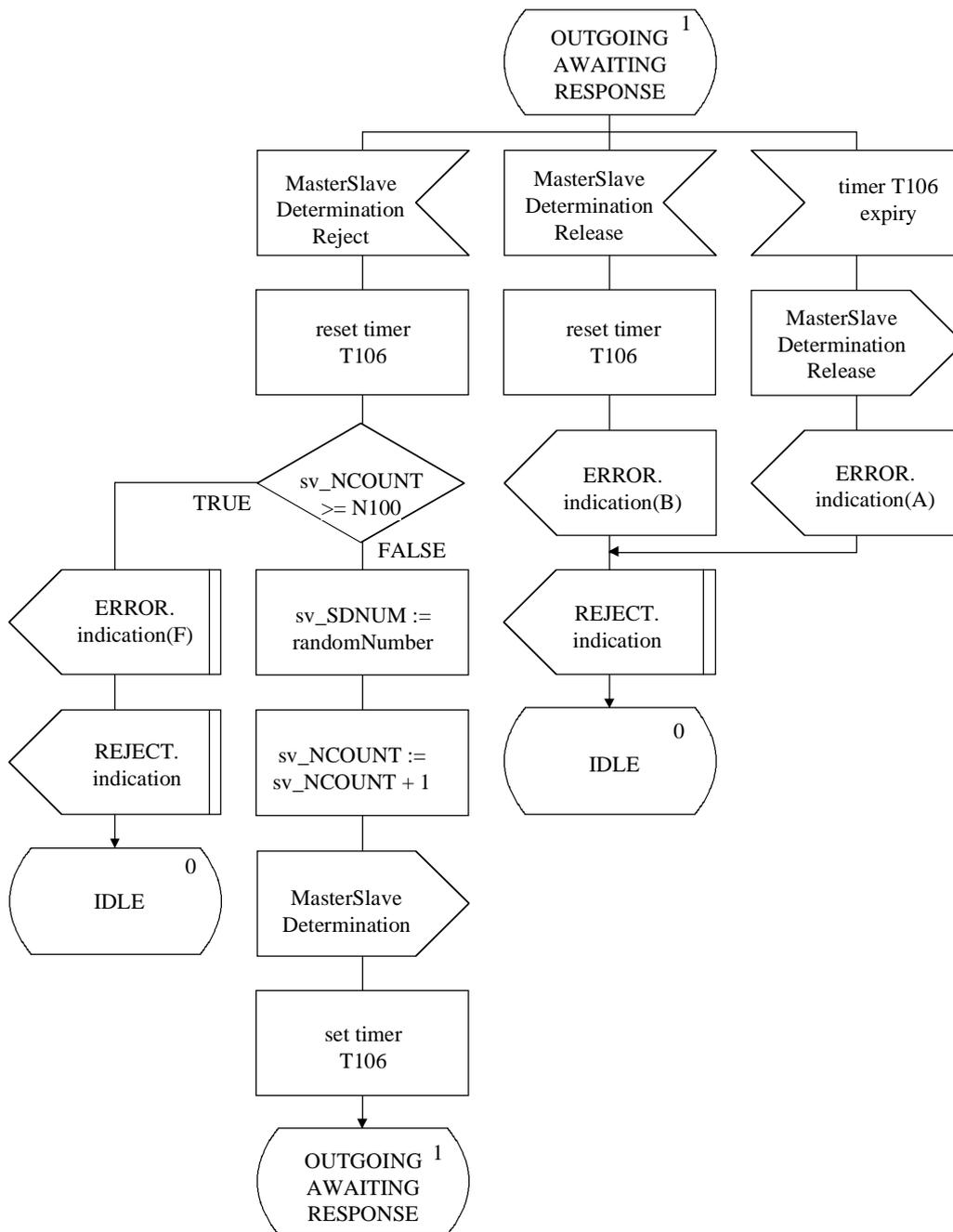
T1600040-97

Figure 4 i)/H.245 – Diagramme SDL de l'entité MSDSE



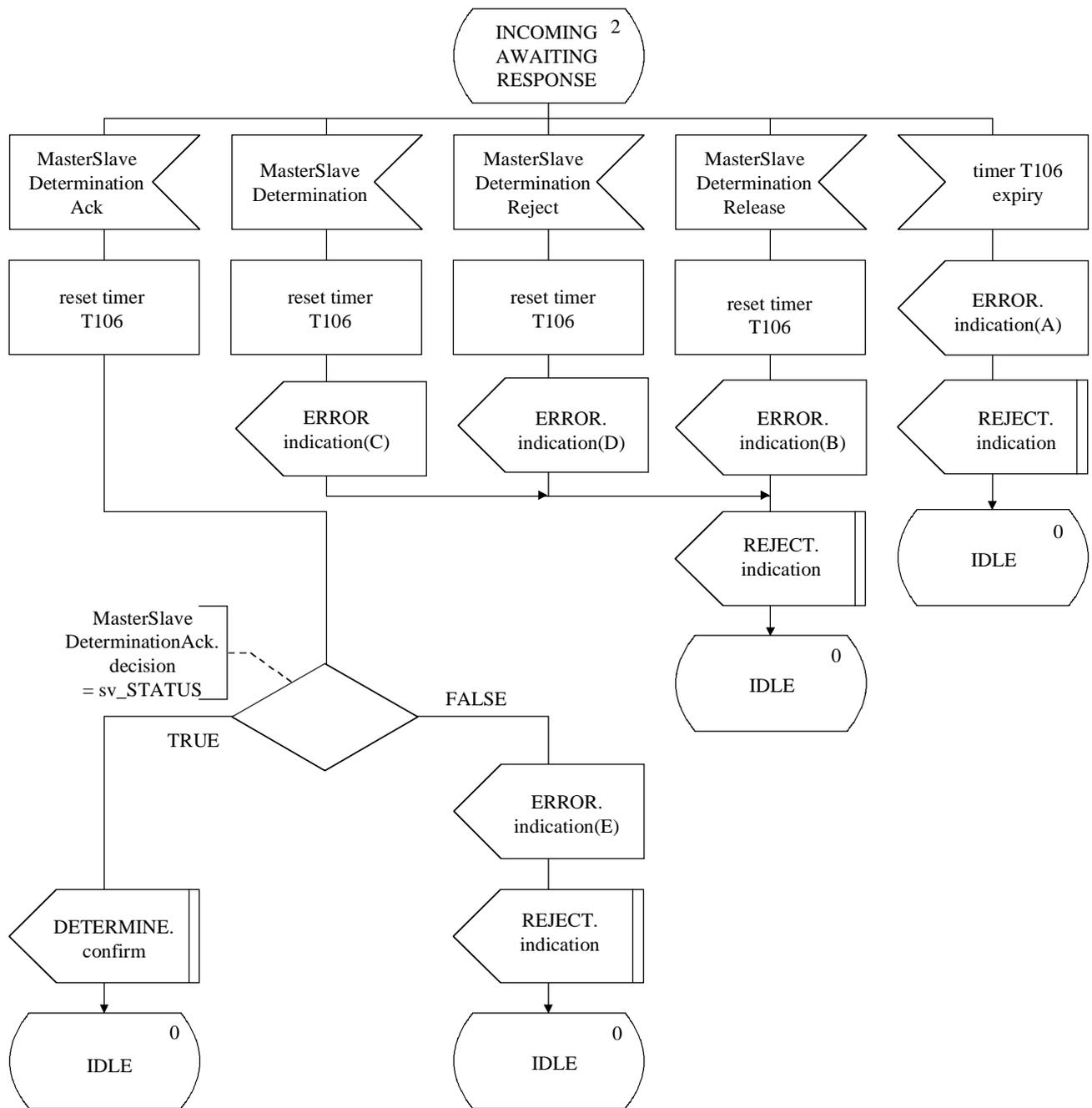
T1520260-95

Figure 4 ii)/H.245 – Diagramme SDL de l'entité MSDSE



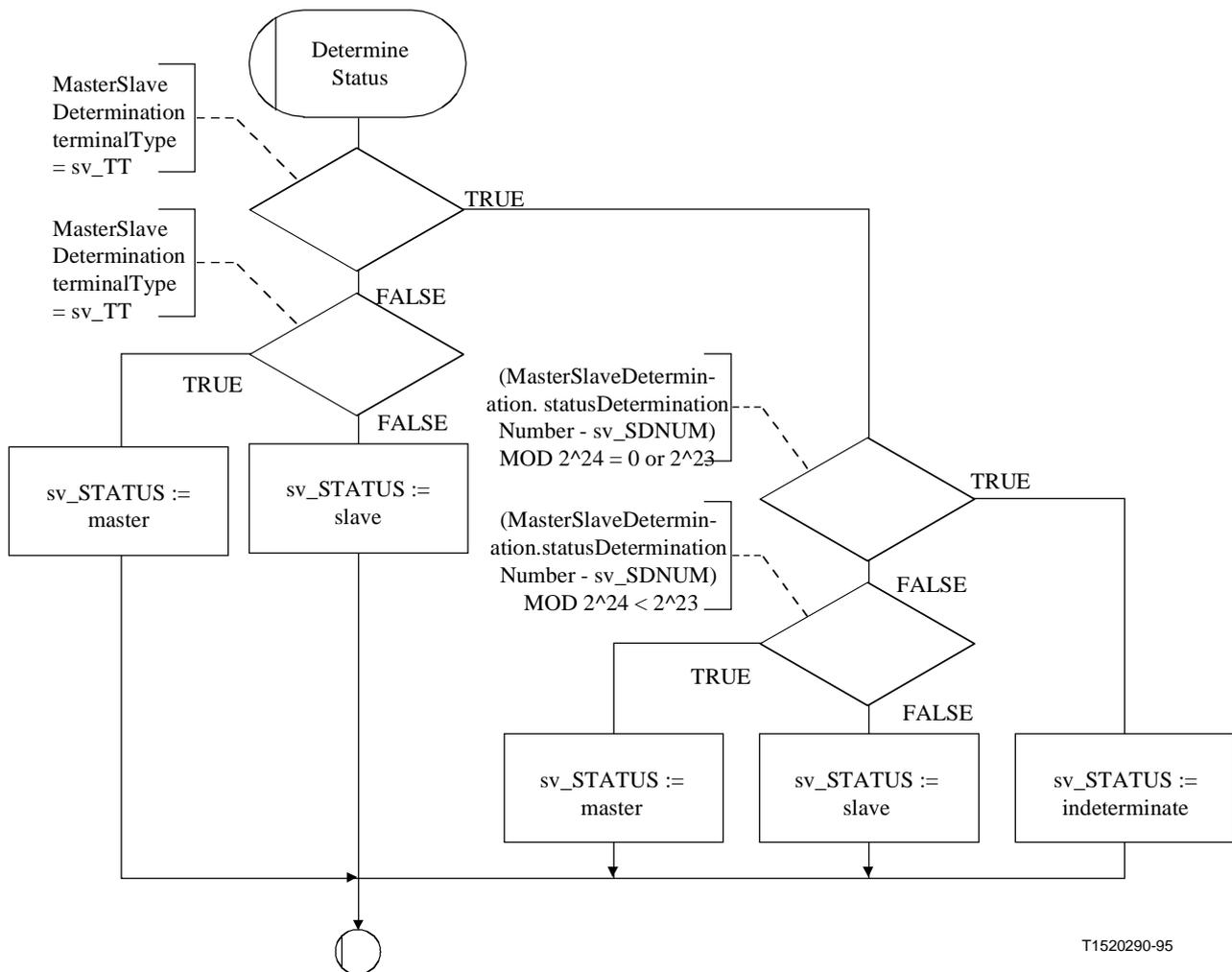
T1520270-95

Figure 4 iii)/H.245 – Diagramme SDL de l'entité MSDSE



T1520280-95

Figure 4 iv)/H.245 – Diagramme SDL de l'entité MSDSE



T1520290-95

Figure 4 v)/H.245 – Diagramme SDL de l'entité MSDSE

8.3 Procédures d'échange de capacités

8.3.1 Introduction

Ces procédures sont utilisées par les terminaux pour faire connaître leurs capacités et sont appelées "entité de signalisation de l'échange de capacités" (CESE, *capability exchange signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité CESE et l'utilisateur de l'entité CESE. Les informations de protocole sont transférées vers l'entité CESE homologue par l'intermédiaire des messages pertinents définis dans le paragraphe 6. Il y a une entité CESE entrante et une entité CESE sortante. A chacune des extrémités entrante et sortante, il y a une instance d'entité CESE pour chaque appel.

Tous les terminaux dont l'utilisation est prévue dans les applications point à point ou ceux connectés à un équipement de commande multipoint (MCU) devront pouvoir identifier un ensemble TerminalCapabilitySet et sa structure, et de telles valeurs de capacité qui sont obligatoires pour ces applications; toutes les valeurs de capacité non reconnues devront être omises et aucune défaillance ne devra être impliquée.

L'échange de capacités peut être effectué à n'importe quel moment. Cet échange de capacités peut signaler les capacités à la fois changées et inchangées. Les capacités inchangées ne devraient pas être envoyées de façon répétitive sans raison majeure.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

8.3.1.1 Aperçu général du protocole – Entité CESE sortante

Un échange de capacités est déclenché quand la primitive de demande TRANSFER est présentée par l'utilisateur dans l'entité CESE sortante. Un message TerminalCapabilitySet est envoyé vers l'entité CESE entrante homologue et le temporisateur T101 démarre. Si un message TerminalCapabilitySetAck est reçu en réponse au message TerminalCapabilitySet, alors le temporisateur T101 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER que l'échange de capacités a réussi. Si, cependant, un message TerminalCapabilitySetReject est reçu en réponse au message TerminalCapabilitySet, alors le temporisateur T101 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité CESE homologue a refusé l'échange de capacités.

Si le temporisateur T101 arrive en fin de temporisation, alors l'utilisateur de l'entité CESE est informé par la primitive d'indication REJECT et un message TerminalCapabilitySetRelease est envoyé.

8.3.1.2 Aperçu général du protocole – Entité CESE entrante

Quand un message TerminalCapabilitySet est reçu par l'entité CESE entrante, l'utilisateur est informé de la demande d'échange de capacités par la primitive d'indication TRANSFER. L'utilisateur de l'entité CESE entrante signale l'acceptation de la demande d'échange de capacités en présentant la primitive de réponse TRANSFER et un message TerminalCapabilitySetAck est envoyé à l'entité CESE sortante homologue. L'utilisateur de l'entité CESE entrante signale le refus de la demande d'échange de capacités en présentant la primitive de demande REJECT et un message TerminalCapabilitySetReject est envoyé à l'entité CESE sortante homologue.

8.3.2 Communication entre l'entité CESE et l'utilisateur de l'entité CESE

8.3.2.1 Primitives entre l'entité CESE et l'utilisateur de l'entité CESE

Les communications entre l'entité CESE et l'utilisateur de l'entité CESE s'effectuent en utilisant les primitives indiquées dans le Tableau 22.

Tableau 22/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	– (Note 1)	–
REJECT	CAUSE	SOURCE CAUSE	non défini (Note 2)	non défini
NOTE 1 – "–" signifie absence de paramètres.				
NOTE 2 – "non définie" signifie que cette primitive n'est pas définie.				

8.3.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives TRANSFER sont utilisées pour le transfert des échanges de capacités;
- b) les primitives REJECT sont utilisées pour refuser une entrée de descripteur de capacités et mettre fin à un transfert de capacités en cours.

8.3.2.3 Définition des paramètres

La définition des paramètres des primitives indiquées au Tableau 22 est la suivante:

- a) le paramètre PROTOID est le paramètre d'identification du protocole. Il est mappé au champ protocolIdentifier du message TerminalCapabilitySet et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est obligatoire.
- b) le paramètre MUXCAP est le paramètre des capacités du multiplex. Il est mappé au champ multiplexCapability du message TerminalCapabilitySet et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- c) le paramètre CAPTABLE est le paramètre du tableau de capacités. Ce paramètre peut décrire une ou plusieurs entrées de tableau de capacités. Ce paramètre est mappé au champ capabilityTable du message TerminalCapabilitySet et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- d) le paramètre CAPDESCRIPTORS est le paramètre des descripteurs de capacités. Ce paramètre peut décrire un ou plusieurs descripteurs de capacités. Ce paramètre est mappé au champ capabilityDescriptors du message TerminalCapabilitySet et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- e) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur "USER" ou "PROTOCOL". Ce dernier cas peut provenir d'une fin de temporisation;
- f) le paramètre CAUSE indique le motif du refus du paramètre CAPTABLE ou CAPDESCRIPTORS. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

8.3.2.4 Etats de l'entité CESE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité CESE et l'utilisateur de l'entité CESE.

Les états correspondant à une entité CESE sortante sont:

Etat 0: IDLE (repos)

L'entité CESE est à l'état "idle".

Etat 1: AWAITING RESPONSE (attente de la réponse)

L'entité CESE attend une réponse de l'entité CESE distante.

Les états correspondant à une entité CESE entrante sont:

Etat 0: IDLE

L'entité CESE est à l'état "idle".

Etat 1: AWAITING RESPONSE (attente de la réponse)

L'entité CESE attend une réponse de la part de l'utilisateur de l'entité CESE.

8.3.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité CESE et l'utilisateur de l'entité CESE est définie ci-après. La séquence autorisée de primitives fait référence à des états de l'entité CESE tels qu'ils

sont perçus par l'utilisateur de l'entité CESE. Les séquences autorisées sont spécifiées distinctement pour une entité CESE sortante et une entité CESE entrante, comme l'indiquent respectivement les Figures 5 et 6.

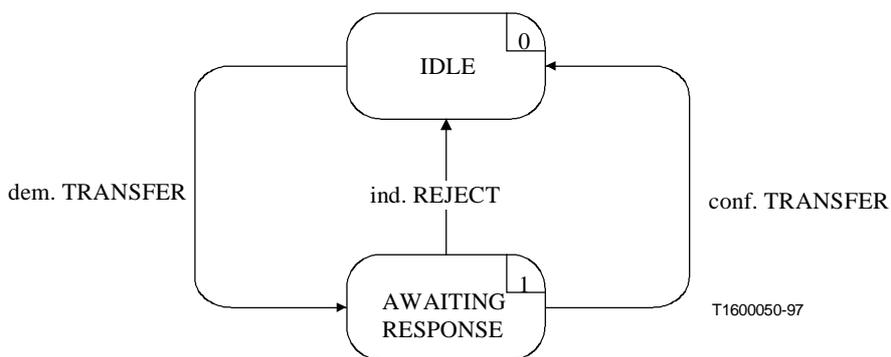


Figure 5/H.245 – Diagramme de changement d'états pour la séquence de primitives d'une entité CESE sortante

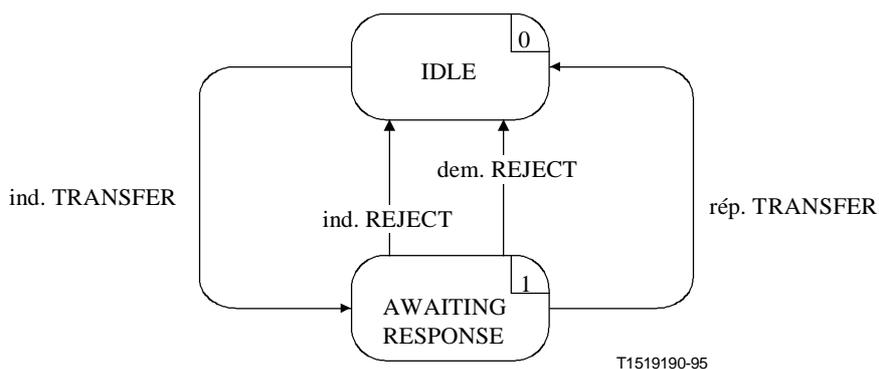


Figure 6/H.245 – Diagramme de changement d'états pour la séquence de primitives d'une entité CESE entrante

8.3.3 Communication entre les entités CESE homologues

8.3.3.1 Messages

Le Tableau 23 montre les messages et les champs des entités CESE, tels qu'ils sont définis dans le paragraphe 6 et se rapportent au protocole des entités CESE.

Tableau 23/H.245 – Noms et champs de messages de l'entité CESE

Fonction	Message	Sens	Champ
transfer	TerminalCapacitySet	S → I	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors
	TerminalCapabilitySetAck	S ← I	sequenceNumber
reject	TerminalCapabilitySetReject	S ← E	sequenceNumber cause
reset	TerminalCapabilitySetRelease	S → E	–
S sortant E entrant			

8.3.3.2 Variables d'état de l'entité CESE

Les variables d'état suivantes sont définies pour l'entité CESE sortante:

out_SQ

Cette variable d'état est utilisée pour indiquer le message TerminalCapabilitySet le plus récent. Elle est augmentée de un et mappée au message TerminalCapabilitySet du champ sequenceNumber avant la transmission du message TerminalCapabilitySet. L'opération arithmétique exécutée sur la variable out_SQ est modulo 256.

Les variables d'état suivantes sont définies dans l'entité CESE entrante:

in_SQ

Cette variable d'état est utilisée pour enregistrer la valeur du champ sequenceNumber du message TerminalCapabilitySet reçu le plus récemment. Les messages TerminalCapabilitySetAck et TerminalCapabilitySetReject ont leurs champs sequenceNumber mis à la valeur de in_SQ, avant d'être envoyés vers l'entité CESE homologue.

8.3.3.3 Temporisateurs CESE

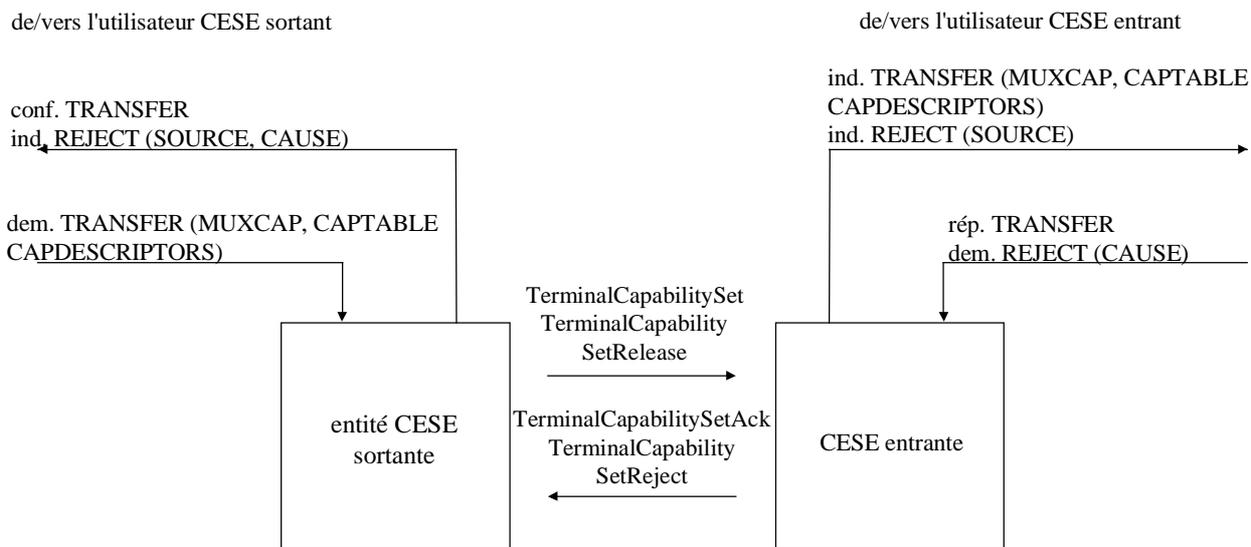
Le temporisateur suivant est spécifié pour l'entité CESE sortante:

T101

Ce temporisateur est utilisé à l'état Awaiting Response (attente de la réponse). Il spécifie le temps maximal écoulé avant qu'un message TerminalCapabilitySetAck ou TerminalCapabilitySetReject puisse être reçu.

8.3.4 Procédures de l'entité CESE

La Figure 7 récapitule les primitives de l'entité CESE et leurs paramètres, de même que les messages, pour chacune des entités CESE entrante et sortante.



T1519200-95

Figure 7/H.245 – Primitives et messages dans l'entité de signalisation de l'échange de capacités (CESE)

8.3.4.1 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs qui sont indiquées dans le Tableau 24, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau 24/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
ind. TRANSFER	PROTOID	TerminalCapabilitySet.protocolIdentifier
	MUXCAP	TerminalCapabilitySet.multiplexCapability
	CAPTABLE	TerminalCapabilitySet.capabilityTable
	CAPDESCRIPTORS	TerminalCapabilitySet.capabilityDescriptors
ind. REJECT	SOURCE	USER
	CAUSE	null

8.3.4.2 Valeurs par défaut des champs de message

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs de message prennent des valeurs identiques à celles indiquées dans le Tableau 25.

Tableau 25/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note)
TerminalCapabilitySet	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors	out_SQ dem. TRANSFER (PROTOID) dem. TRANSFER (MUXCAP) dem. TRANSFER (CAPTABLE) dem. TRANSFER (CAPDESCRIPTORS)
TerminalCapabilitySetAck	sequenceNumber	in_SQ
TerminalCapabilitySetReject	sequenceNumber cause	in_SQ dem. REJECT (CAUSE)
TerminalCapabilitySetRelease	–	–

NOTE – Un champ de message ne sera pas codé si le paramètre correspondant de la primitive est nul, c'est-à-dire est absent.

8.3.4.3 Description SDL

L'entité CESE sortante et les procédures de l'entité CESE entrante sont décrites en langage SDL dans les Figures 8 et 9 respectivement.

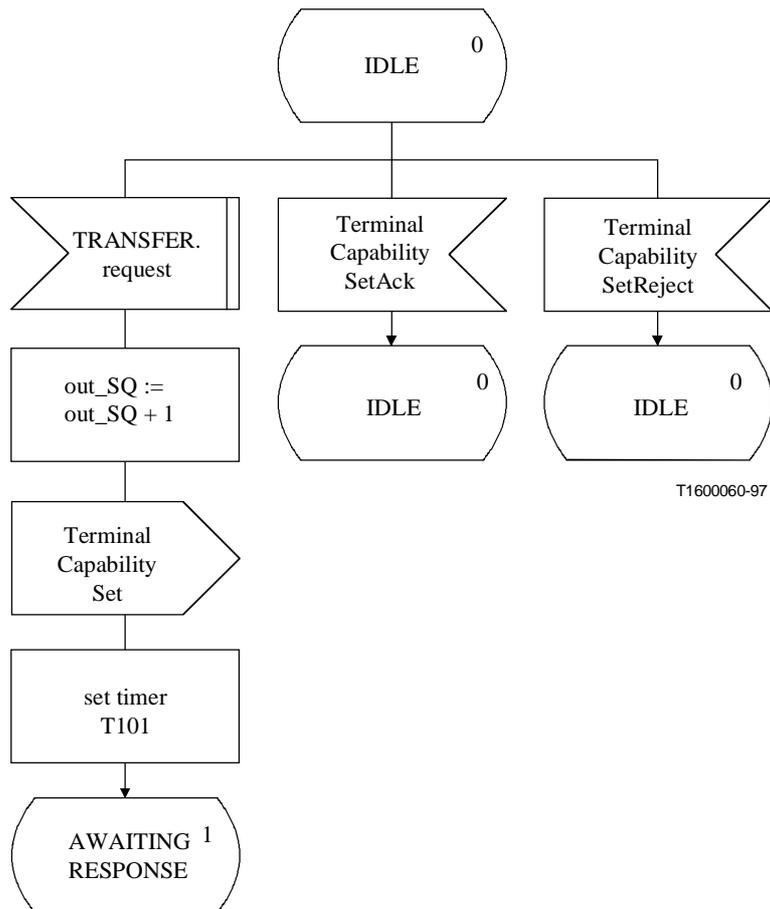
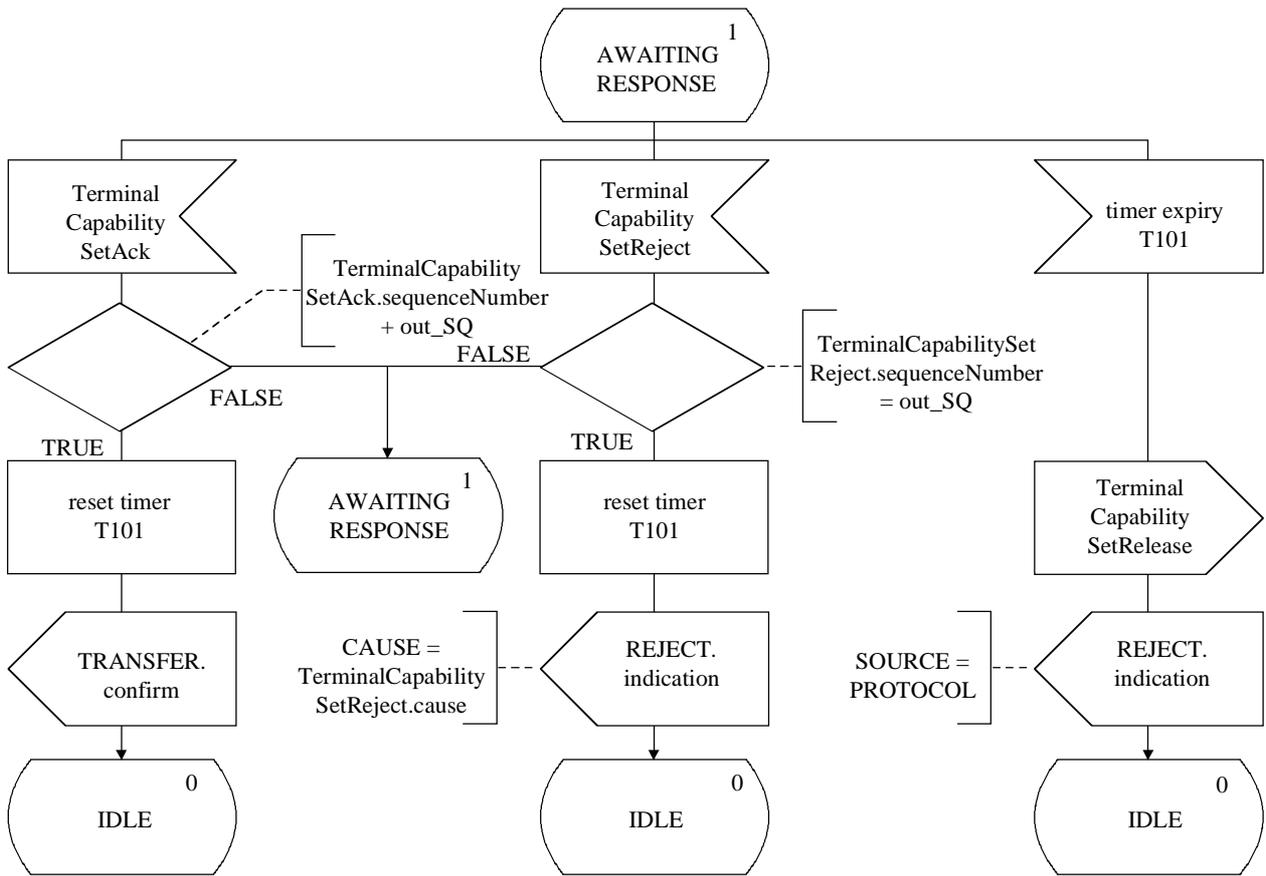
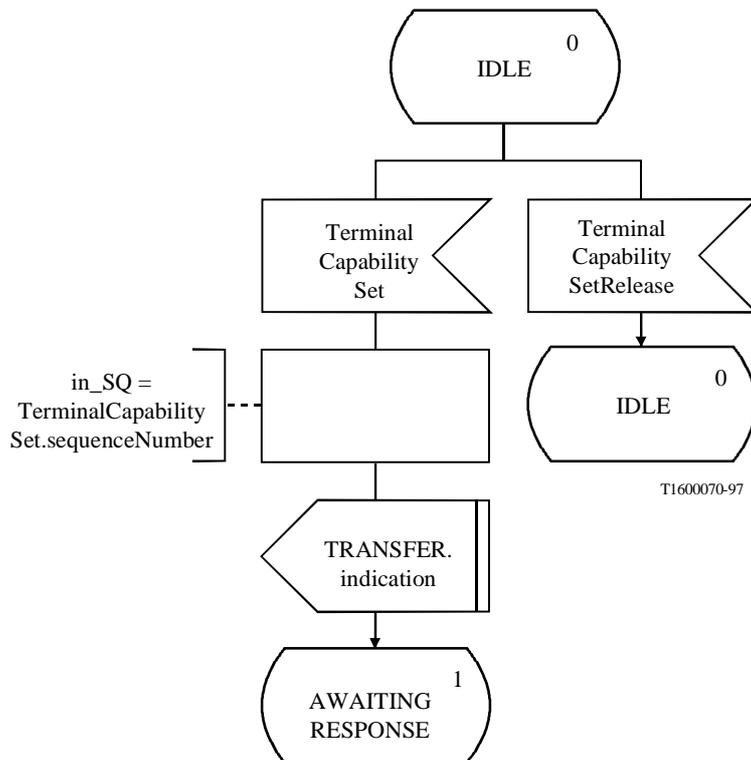


Figure 8 i)/H.245 – Description SDL de l'entité CESE sortante



T1520300-95

Figure 8 ii)/H.245 – Description SDL de l'entité CESE sortante



T1600070-97

Figure 9 i)/H.245 – Description SDL de l'entité CESE entrante

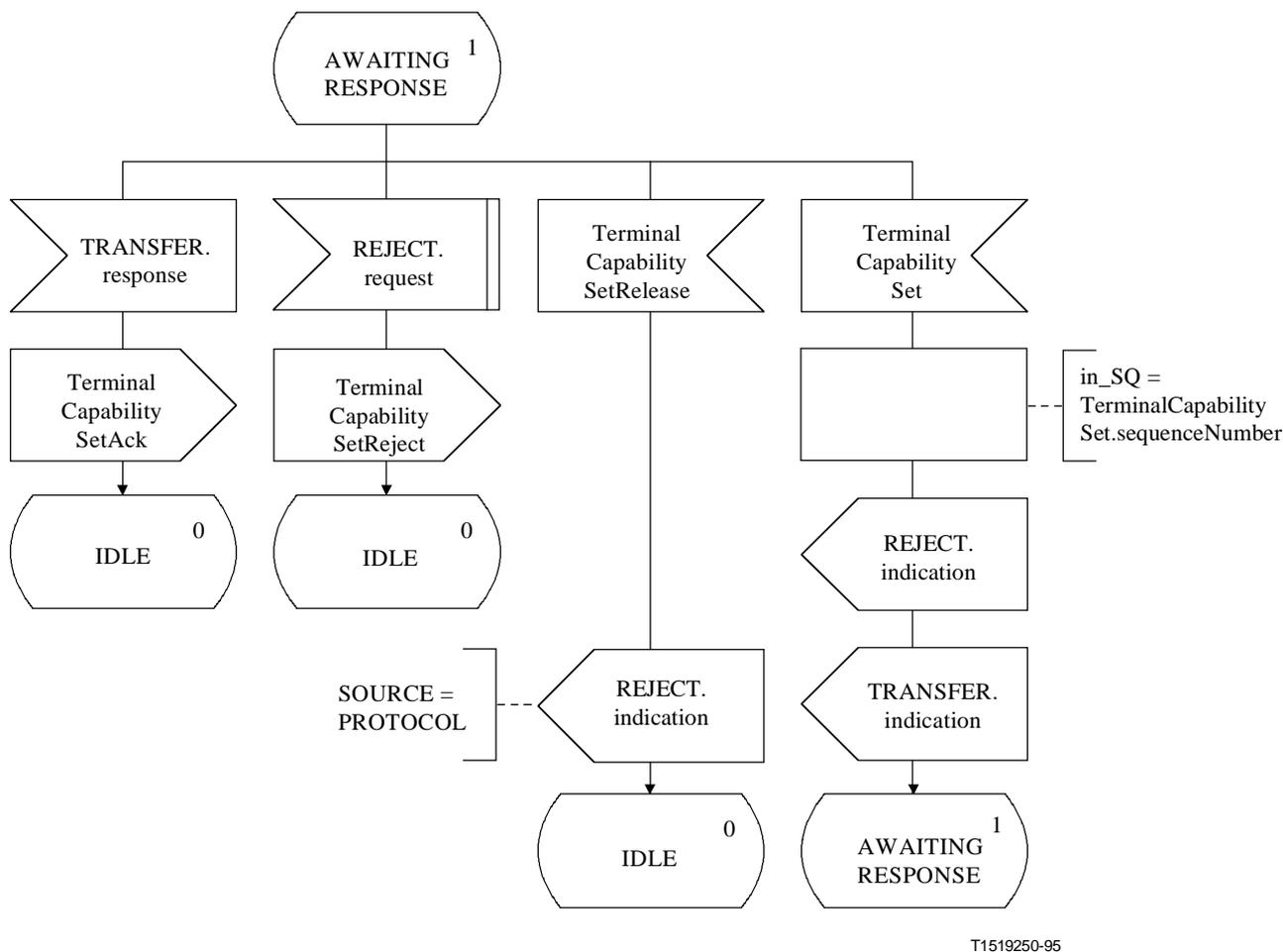


Figure 9 ii)/H.245 – Description SDL de l'entité CESE entrante

8.4 Procédures de signalisation de la voie logique monodirectionnelle

8.4.1 Introduction

Le protocole spécifié ici permet l'ouverture et la fermeture fiable des voies logiques monodirectionnelles en utilisant des procédures avec acquittement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de la voie logique (LCSE, *logical channel signalling entity*). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité LCSE et l'utilisateur de l'entité LCSE, et entre les états LCSE. Les informations de protocole sont transmises vers l'entité LCSE homologue au moyen des messages pertinents définis dans le paragraphe 6.

Il y a une entité LCSE sortante et une entité LCSE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité LCSE correspondant à chaque voie logique monodirectionnelle. Il n'y a pas de relation entre une entité LCSE entrante et une entité LCSE sortante d'un côté, autre que celle obtenue au moyen des primitives provenant de l'utilisateur LCSE et allant vers ce même utilisateur. Les conditions d'erreur de l'entité LCSE sont signalées.

Les données devront être envoyées uniquement vers une voie logique dans l'état ESTABLISHED. Si les données sont reçues sur une voie logique qui n'est pas dans l'état ESTABLISHED, les données devront être ignorées et il faudra considérer qu'aucune défaillance ne s'est produite.

Le changement de mode devrait s'effectuer par l'ouverture et la fermeture des voies logiques existantes, ou par l'ouverture de nouvelles voies logiques.

NOTE – Certaines Recommandations qui utilisent la présente Recommandation peuvent définir certaines voies logiques par défaut. Celles-ci devront être considérées comme étant ETABLIES dès le début de la communication et ne devront pas être ouvertes en utilisant ces procédures. Cependant, ces voies peuvent être fermées par ces procédures et elles peuvent ultérieurement être réouvertes pour répondre au même objectif ou à un objectif différent.

Un terminal n'ayant plus la capacité de traiter les signaux sur une voie logique devrait prendre des mesures appropriées: celles-ci devraient comprendre la fermeture de la voie logique et la transmission des informations pertinentes relatives aux capacités (modifiées) vers le terminal distant.

Le texte suivant est un aperçu général de l'exploitation du protocole dans les entités LCSE. En cas de divergence entre celui-ci et la spécification formelle, la spécification formelle sera applicable.

8.4.1.1 Aperçu général du protocole

L'ouverture d'une voie logique est déclenchée quand la primitive de demande ESTABLISH est présentée par l'utilisateur de l'entité LCSE sortante. Un message OpenLogicalChannel, contenant les paramètres de voie logique dans le sens direct mais n'incluant ces mêmes paramètres dans le sens inverse, est envoyé vers l'entité LCSE entrante homologue, et le temporisateur T103 démarre. Si un message OpenLogicalChannelAck est reçu en réponse à un message OpenLogicalChannel, alors le temporisateur T103 s'arrête et l'utilisateur est informé par la primitive de confirmation ESTABLISH que l'ouverture de la voie logique a réussi. La voie logique peut désormais être utilisée pour transmettre les informations de l'utilisateur. Si, cependant, un message OpenLogicalChannelReject est reçu en réponse au message OpenLogicalChannel, alors le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité LCSE homologue a refusé l'établissement de la voie logique.

Si le temporisateur T103 vient à expiration pendant cette période, alors l'utilisateur est informé par la primitive d'indication RELEASE et un message CloseLogicalChannel est envoyé vers l'entité LCSE entrante homologue.

Une voie logique qui a été établie avec succès peut être fermée quand la primitive de demande RELEASE est présentée par l'utilisateur de l'entité LCSE sortante. Un message CloseLogicalChannel est envoyé à l'entité LCSE entrante homologue et le temporisateur T103 démarre. Quand un message CloseLogicalChannelAck est reçu, le temporisateur T103 est arrêté et l'utilisateur est informé que la voie logique a bien été fermée par la primitive de confirmation RELEASE.

Si le temporisateur T103 arrive en fin de temporisation pendant cette période, alors l'utilisateur est informé par la primitive d'indication RELEASE.

Avant que l'un des messages OpenLogicalChannelAck ou OpenLogicalChannelReject ait été reçu en réponse à un message OpenLogicalChannel préalablement envoyé, l'utilisateur de l'entité LCSE sortante peut fermer la voie logique en utilisant la primitive de demande RELEASE.

Avant que le message CloseLogicalChannelAck ait été reçu en réponse à un message CloseLogicalChannel préalablement envoyé, l'utilisateur de l'entité LCSE sortante peut ouvrir une nouvelle voie logique en présentant la primitive de demande ESTABLISH.

8.4.1.2 Aperçu général du protocole – Entité LCSE entrante

Quand un message OpenLogicalChannel est reçu par l'entité LCSE entrante, l'utilisateur est informé de la demande d'ouverture d'une nouvelle voie logique par la primitive d'indication ESTABLISH. L'utilisateur de l'entité LCSE entrante signale l'acceptation de la demande d'ouverture de voie logique en présentant la primitive de réponse ESTABLISH, et un message OpenLogicalChannelAck

est envoyé vers l'entité LCSE sortante homologue. La voie logique peut à présent être utilisée pour recevoir les informations de l'utilisateur. L'utilisateur de l'entité LCSE entrante signale le refus de la demande d'établissement de voie logique en présentant la primitive de demande RELEASE et un message OpenLogicalChannelReject est envoyé vers l'entité LCSE sortante homologue.

Une voie logique qui a bien été établie peut être fermée quand le message CloseLogicalChannel est reçu par l'entité LCSE entrante. L'utilisateur de l'entité LCSE entrante est informé par la primitive d'indication RELEASE, et le message CloseLogicalChannelAck est envoyé vers l'entité LCSE sortante homologue.

8.4.1.3 Résolution des litiges

Des litiges peuvent survenir quand des demandes d'ouverture de voies logiques sont déclenchées en même temps. Il peut être possible de déterminer qu'il y a un litige à partir des connaissances acquises par l'échange des capacités.

Les terminaux doivent être en mesure de détecter qu'un litige est survenu ou pourrait survenir, et ils doivent réagir comme suit:

avant que des voies logiques puissent être ouvertes, l'un des terminaux doit être choisi comme le terminal maître et l'autre comme le terminal esclave. Le protocole défini en 8.2 est un mécanisme permettant d'effectuer ce choix. Le terminal maître doit refuser immédiatement toute demande émanant du terminal esclave qu'il identifie comme une demande litigieuse. Le terminal esclave peut identifier de tels litiges, mais devra répondre à la demande émanant du terminal maître, en sachant que sa demande précédente sera refusée.

NOTE – De tels litiges pourraient être provoqués par des ressources restreintes du terminal, par exemple quand les capacités d'émission et de réception sont dépendantes, comme c'est le cas pour un terminal pouvant utiliser un grand nombre d'algorithmes audio, mais pouvant uniquement décoder avec le même algorithme que celui utilisé lors du codage.

8.4.2 Communication entre l'entité LCSE et l'utilisateur de l'entité LCSE

8.4.2.1 Primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE

Les communications entre l'entité LCSE et l'utilisateur de l'entité LCSE sont effectuées en utilisant les primitives indiquées dans le Tableau 26.

Tableau 26/H.245 Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
ESTABLISH	FORWARD_PARAM	FORWARD_PARAM	– (Note 1)	–
RELEASE	CAUSE	SOURCE CAUSE	non défini (Note 2)	–
ERROR	non défini	ERRCODE	non défini	non défini
NOTE 1 – "–" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'existe pas.				

8.4.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives ESTABLISH sont utilisées pour établir une voie logique pour les communications audiovisuelles et les transmissions de données;
- b) les primitives RELEASE sont utilisées pour libérer une voie logique;
- c) la primitive ERROR indique les erreurs de l'entité LCSE à une entité de gestion.

8.4.2.3 Définition des paramètres

La définition des paramètres de primitives indiqués dans le Tableau 26 est la suivante:

- a) le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie logique. Ce paramètre est mappé au champ des paramètres forwardLogicalChannelParameters du message OpenLogicalChannel et est transmis de façon transparente vers l'utilisateur de l'entité LCSE homologue;
- b) le paramètre SOURCE indique à l'utilisateur de l'entité LCSE l'origine de la libération de la voie logique. Le paramètre SOURCE a la valeur de "USER" ou de "LCSE", indiquant soit l'utilisateur de l'entité LCSE, soit l'entité LCSE elle-même. Cette dernière peut être le résultat d'une erreur de protocole;
- c) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité LCSE homologue a refusé une demande d'établissement de voie logique. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "LCSE";
- d) le paramètre ERRCODE indique le type d'erreur LCSE. Le Tableau 30 indique les valeurs autorisées du paramètre ERRCODE.

8.4.2.4 Etats de l'entité LCSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE, de même que l'échange de messages entre des entités LCSE homologues. Les états sont spécifiés de façon distincte pour chacune des entités LCSE sortantes et une entité LCSE entrante. Les états correspondant à une entité LCSE sortante sont:

Etat 0: RELEASED

La voie logique est libre. La voie logique ne devra pas être utilisée pour envoyer les données sortantes.

Etat 1: AWAITING ESTABLISHMENT

L'entité LCSE sortante attend l'établissement d'une voie logique avec une entité LCSE entrante homologue. La voie logique ne devra pas être utilisée pour envoyer les données sortantes.

Etat 2: ESTABLISHED

La connexion de la voie logique a été établie entre les entités LCSE homologues. La voie logique peut être utilisée pour envoyer les données sortantes.

Etat 3: AWAITING RELEASE

L'entité LCSE sortante attend de libérer une voie logique avec l'entité LCSE entrante homologue. La voie logique ne devra pas être utilisée pour envoyer les données sortantes.

Les états correspondant à une entité LCSE entrante sont les suivants:

Etat 0: RELEASED

La voie logique est libre. La voie logique ne devra pas être utilisée pour recevoir les données entrantes.

Etat 1: AWAITING ESTABLISHMENT

L'entité LCSE entrante attend d'établir une voie logique avec une entité LCSE sortante homologue. La voie logique ne devra pas être utilisée pour recevoir les données entrantes.

Etat 2: ESTABLISHED

Une connexion de voie logique a été établie entre les entités LCSE homologues. La voie logique peut être utilisée pour recevoir les données entrantes.

8.4.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE est définie ici. La séquence autorisée de primitives fait référence à des états de l'entité LCSE du point de vue de l'utilisateur de l'entité LCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités LCSE sortante et LCSE entrante, comme cela est indiqué à la Figure 10 et à la Figure 11 respectivement.

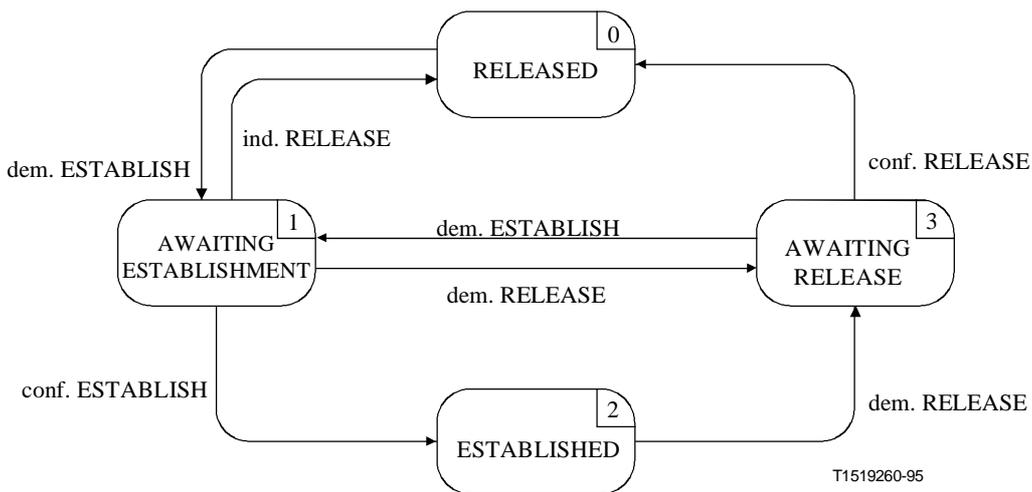


Figure 10/H.245 – Diagramme de changement d'états pour la séquence de primitives d'une entité LCSE sortante

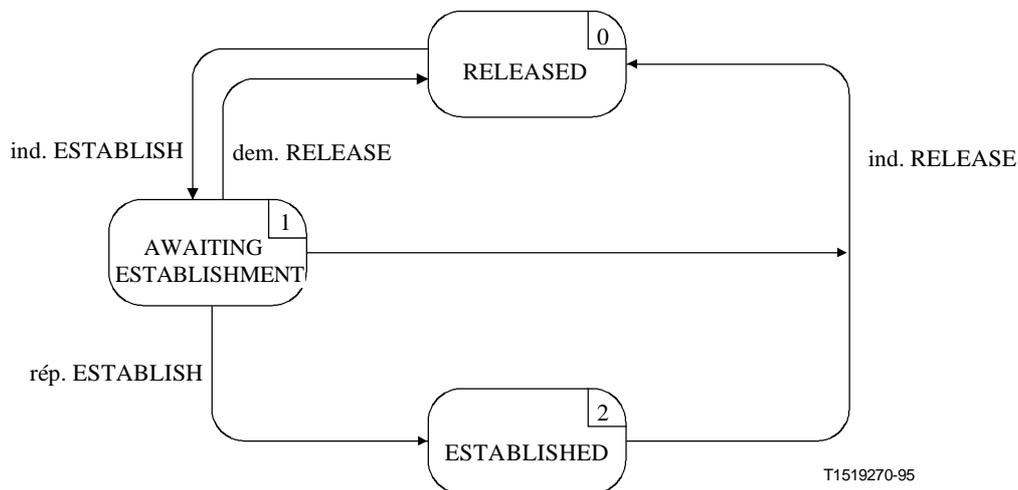


Figure 11/H.245 – Diagramme de changement d'états pour la séquence de primitives d'une entité LCSE entrante

8.4.3 Communications entre les entités LCSE homologues

8.4.3.1 Messages LCSE

Le Tableau 27 indique les messages et les champs des entités LCSE, tels qu'ils sont définis dans le paragraphe 6 et se rapportent au protocole LCSE.

Tableau 27/H.245 – Noms et champs de messages des entités LCSE

Fonction	Message	Sens	Champ
establishment	OpenLogicalChannel	S → E	forwardLogicalChannelNumber forwardLogicalChannelParameters
	OpenLogicalChannelAck	S ← E	forwardLogicalChannelNumber
	OpenLogicalChannelReject	S ← E	forwardLogicalChannelNumber cause
release	CloseLogicalChannel	S → E	forwardLogicalChannelNumber source
	CloseLogicalChannelAck	S ← E	forwardLogicalChannelNumber
S sortant E entrant			

8.4.3.2 Variables d'état des entités LCSE

La variable d'état suivante est définie pour l'entité LCSE sortante:

out_LCN

Cette variable d'état permet de distinguer entre les entités LCSE sortantes. Elle est initialisée lors du lancement de l'entité LCSE sortante. La valeur de la variable out_LCN est utilisée pour remplir le champ forwardLogicalChannelNumber des messages LCSE envoyés à partir d'une entité LCSE sortante. En ce qui concerne les messages LCSE reçus par une entité LCSE sortante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de la variable out_LCN.

La variable d'état suivante est définie pour l'entité LCSE entrante:

in_LCN

Cette variable d'état permet de distinguer entre les entités LCSE entrantes. Elle est initialisée lors du lancement de l'entité LCSE entrante. La valeur de la variable in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages LCSE envoyés par une entité LCSE entrante. En ce qui concerne les messages LCSE reçus dans une entité LCSE entrante, la valeur de champ forwardLogicalChannelNumber des messages est identique à la valeur de la variable in_LCN.

8.4.3.3 Temporisateurs LCSE

Les temporisateurs suivants sont spécifiés pour l'entité LCSE sortante:

T103

Ce temporisateur est utilisé pendant les états AWAITING ESTABLISHMENT et AWAITING RELEASE. Il spécifie le temps maximal autorisé pendant lequel aucun des messages OpenLogicalChannelAck, OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu.

8.4.4 Procédures LCSE

8.4.4.1 Introduction

La Figure 12 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités LCSE entrante et sortante.

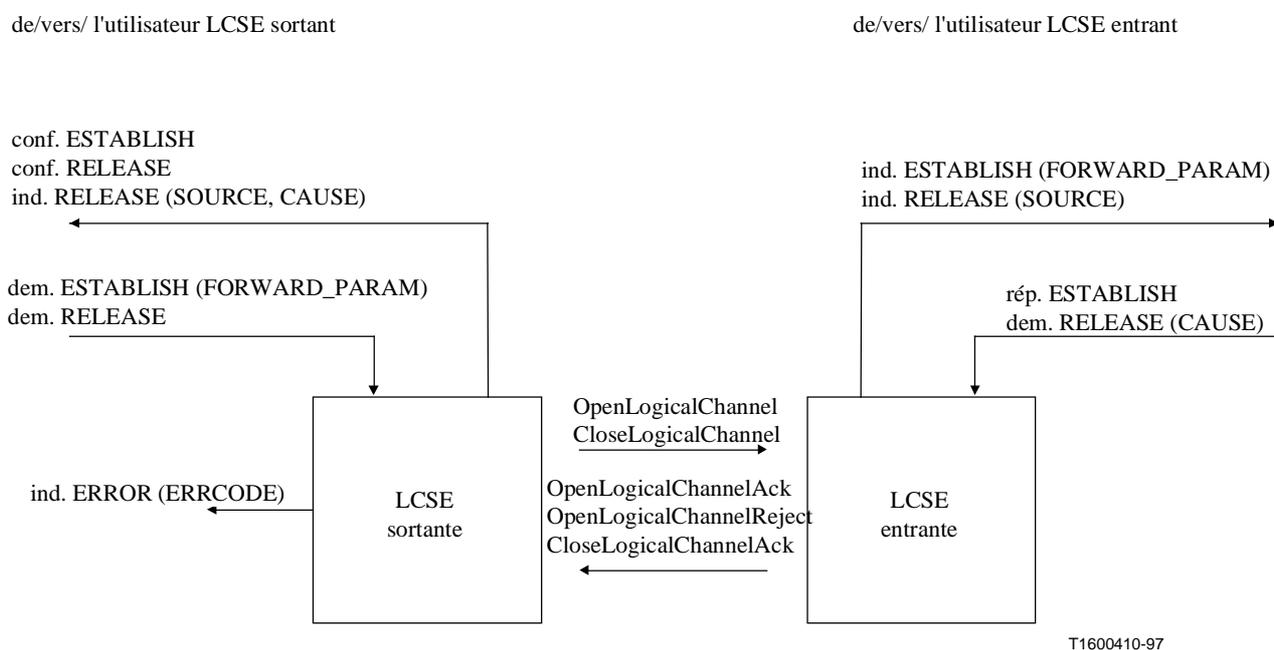


Figure 12/H.245 – Primitives et messages de l'entité de signalisation de voie logique (LCSE)

8.4.4.2 Valeurs par défaut des paramètres des primitives

Quand cela n'est pas explicitement décrit dans les diagrammes en langage SDL, les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 28.

Tableau 28/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut (Note)
ind. ESTABLISH	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
ind. RELEASE	SOURCE CAUSE	CloseLogicalChannel.source null
NOTE – Un paramètre de primitive doit être codé comme étant nul, si un champ de message indiqué n'est pas présent dans le message.		

8.4.4.3 Valeurs par défaut des champs de message

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs de message prennent des valeurs identiques à celles indiquées dans le Tableau 29.

Tableau 29/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note 1)
OpenLogicalChannel (Note 2)	forwardLogicalChannelNumber	out_LCN
	forwardLogicalChannelParameters	dem.ESTABLISH(FORWARD_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
OpenLogicalChannelReject	forwardLogicalChannelNumber cause	in_LCN dem. RELEASE(CAUSE)
CloseLogicalChannel	forwardLogicalChannelNumber source	out_LCN user
CloseLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
NOTE 1 – Un champ de message ne doit pas être codé si le paramètre de primitive correspondant est nul, c'est-à-dire est absent.		
NOTE 2 – Les paramètres reverseLogicalChannelParameters ne sont pas codés dans les procédures de signalisation de voie logique monodirectionnelle.		

8.4.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition d'erreur particulière. Le Tableau 30 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité LCSE sortante. Il n'y a pas de primitive d'indication ERROR associée à l'entité LCSE entrante.

Tableau 30/H.245 – Valeurs du paramètre ERRCODE de l'entité LCSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
message inapproprié	A	message OpenLogicalChannelAck	RELEASED
	B	message OpenLogicalChannelReject	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
aucune réponse provenant de l'entité LCSE homologue	D	fin de temporisation T103	AWAITING ESTABLISHMENT AWAITING RELEASE

8.4.4.5 Description SDL

Les procédures de l'entité LCSE entrante et de l'entité LCSE sortante sont décrites en langage SDL dans les Figures 13 et 14 respectivement.

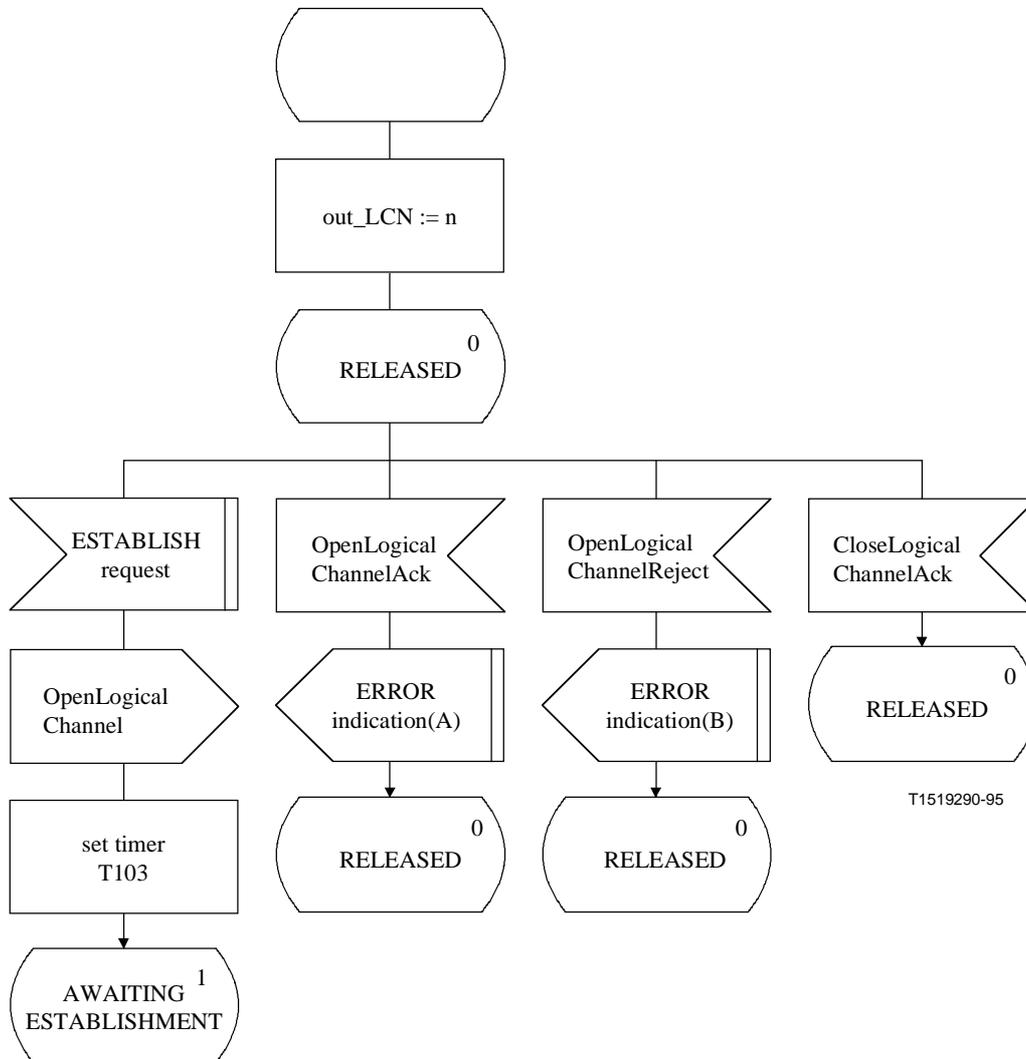


Figure 13 i)/H.245 – Description SDL de l'entité LCSE sortante

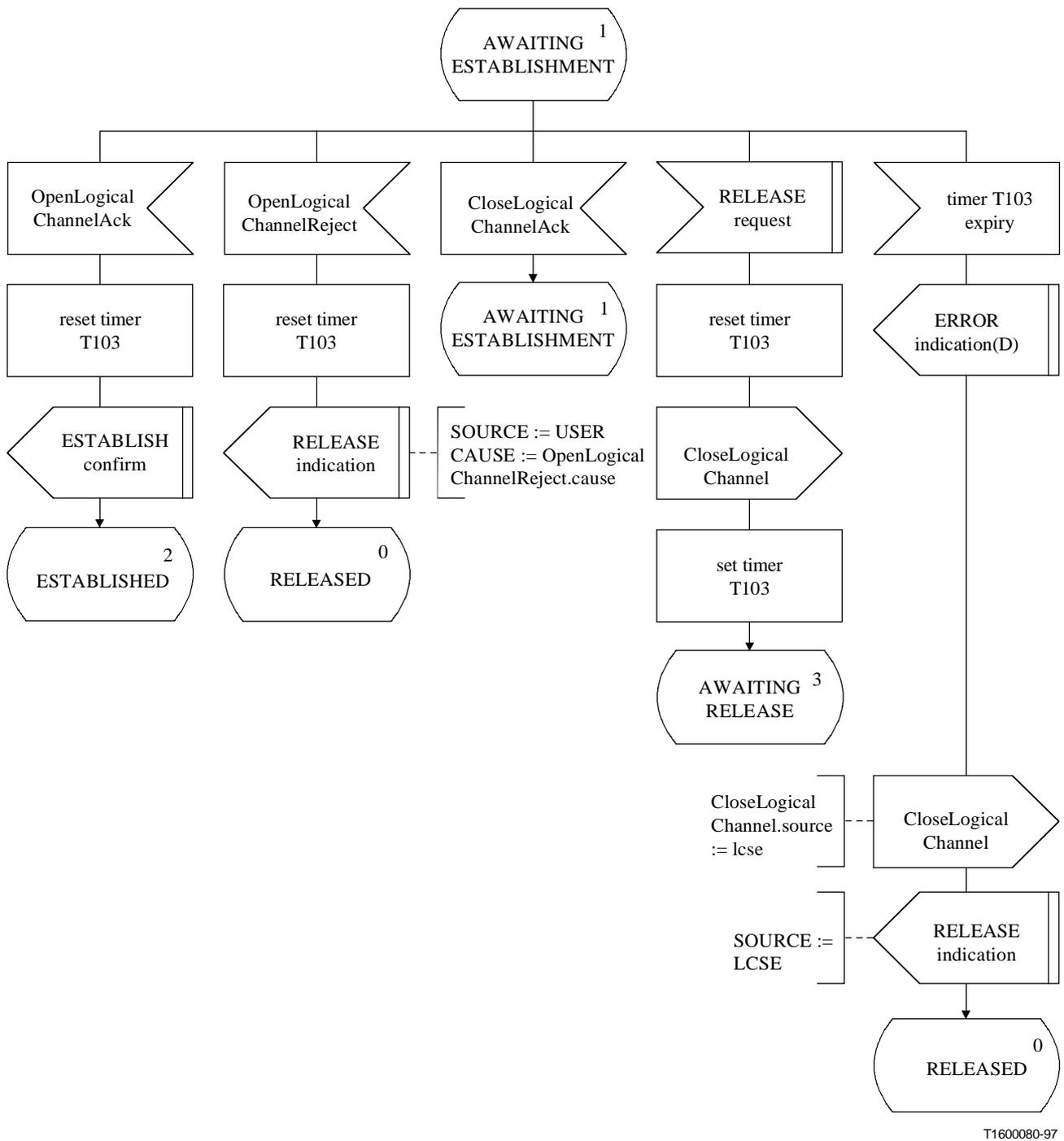
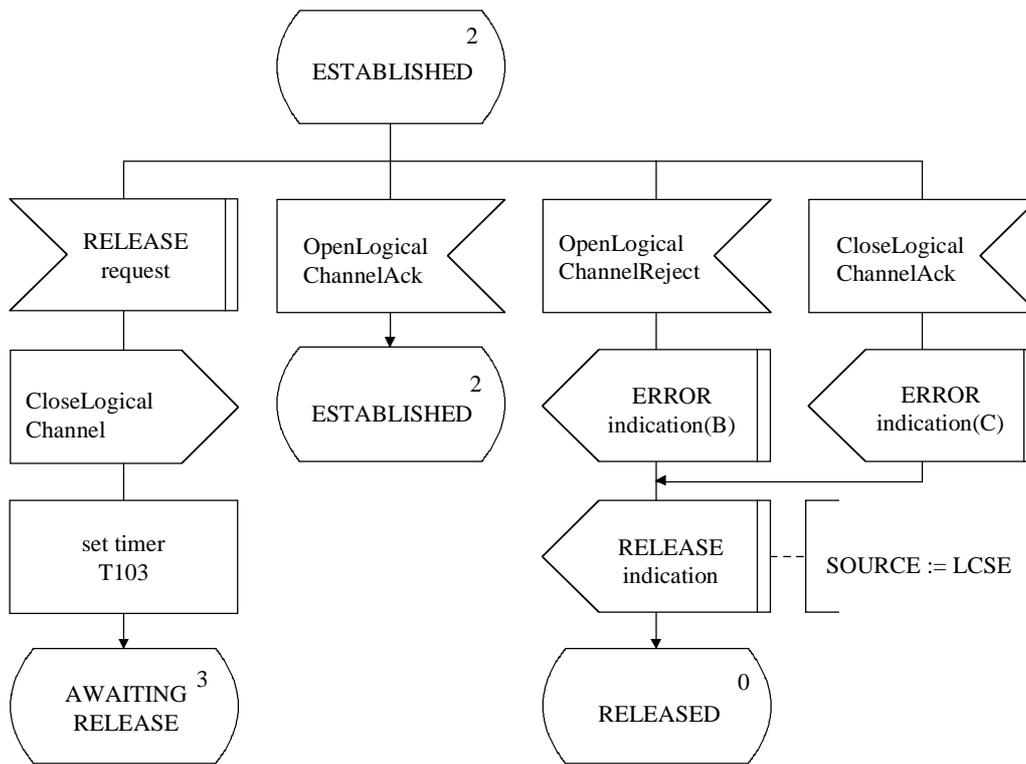


Figure 13 ii)/H.245 – Description SDL de l'entité LCSE sortante



T1519310-95

Figure 13 iii)/H.245 – Description SDL de l'entité LCSE sortante

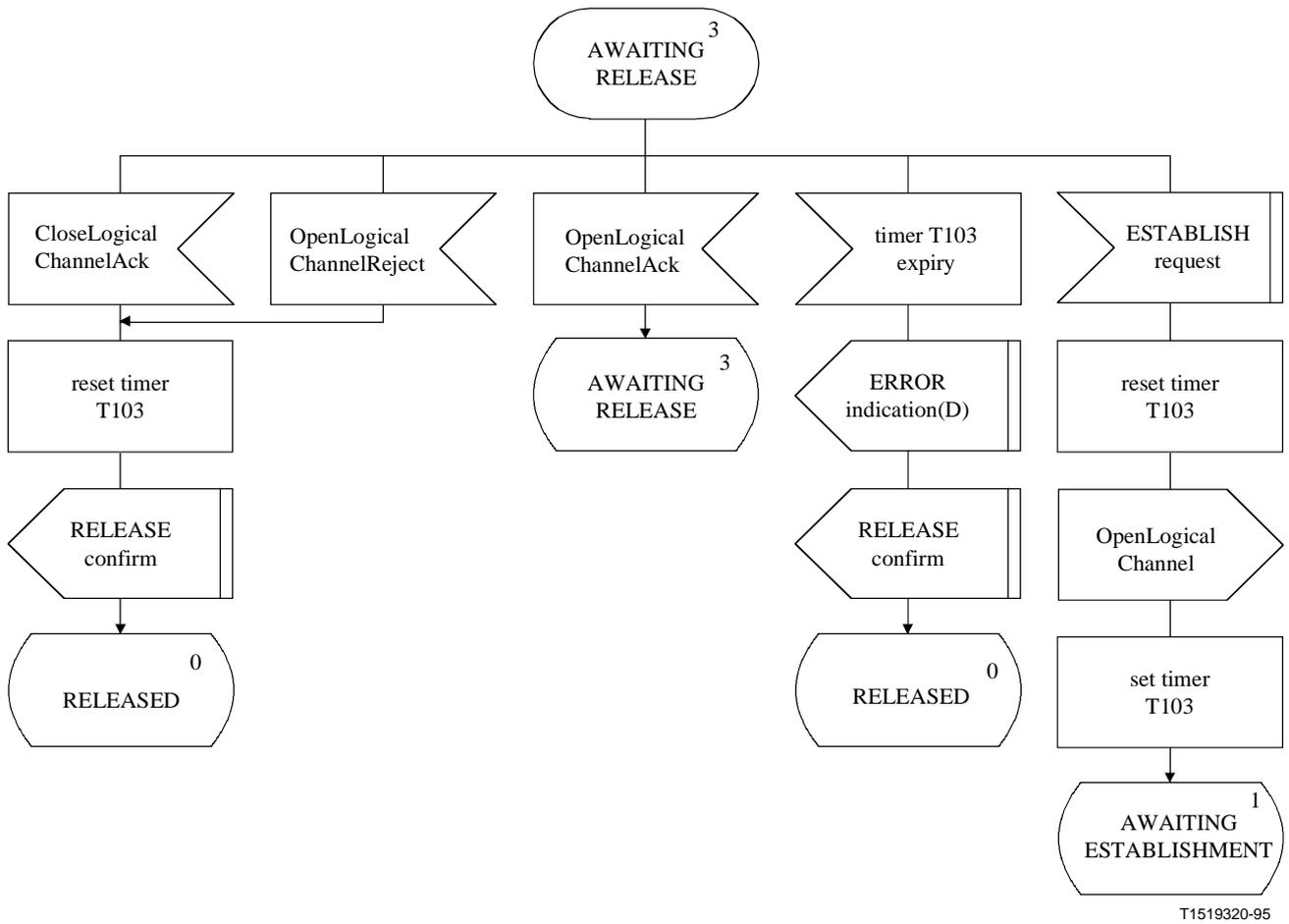
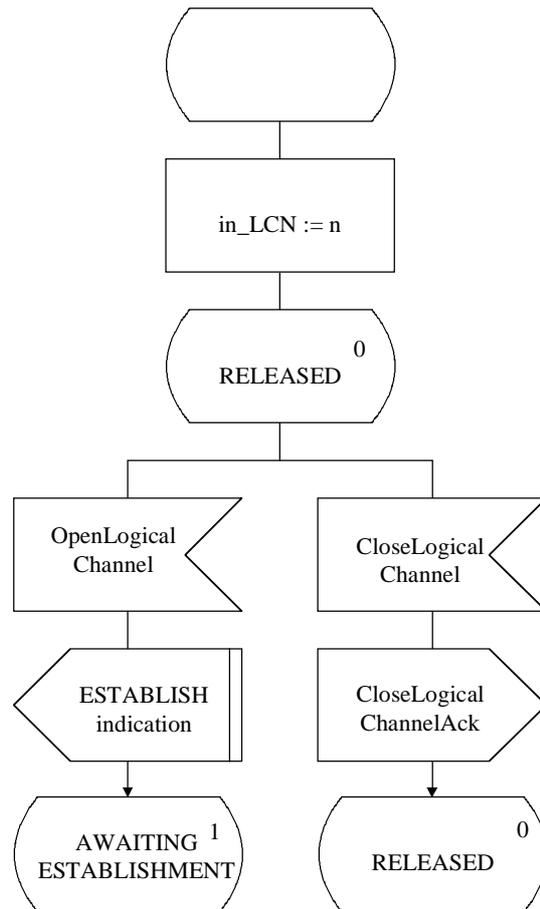
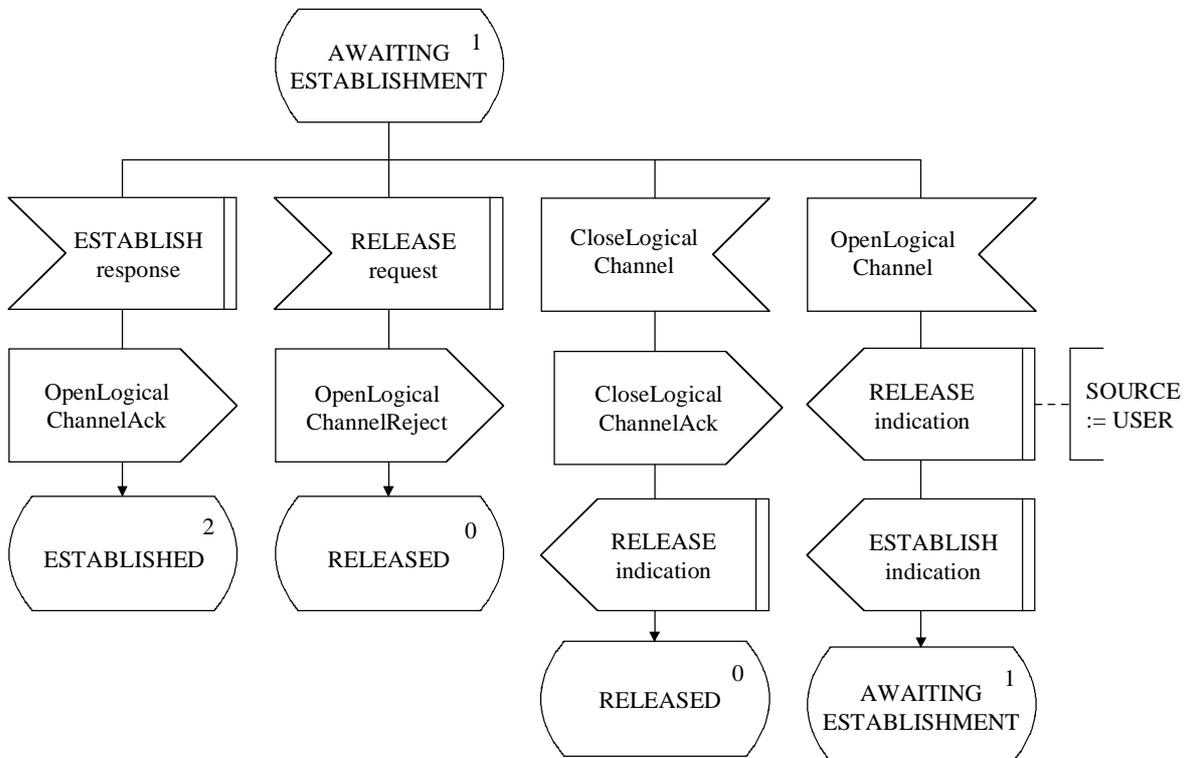


Figure 13 iv)/H.245 – Description SDL de l'entité LCSE sortante



T1519330-95

Figure 14 i)/H.245 – Description SDL de l'entité LCSE entrante



T1600090-97

Figure 14 ii)/H.245 – Description SDL de l'entité LCSE entrante

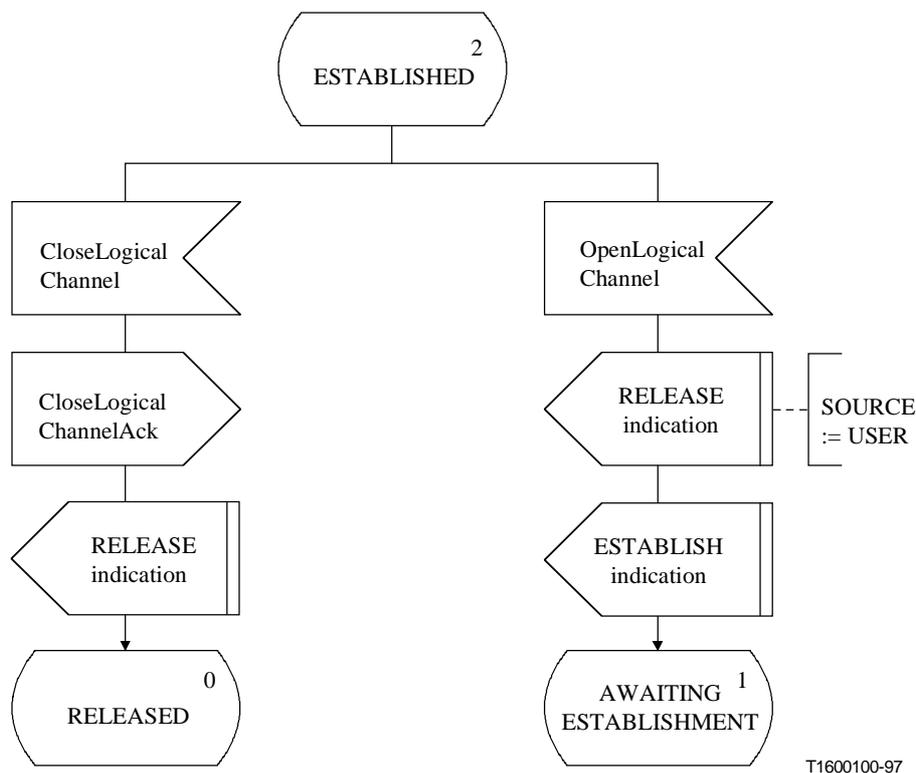


Figure 14 iii)/H.245 – Description SDL de l'entité LCSE entrante

8.5 Procédures de signalisation de la voie logique bidirectionnelle

8.5.1 Introduction

Le protocole spécifié ici permet d'obtenir une ouverture et une fermeture fiable des voies logiques bidirectionnelles utilisant des procédures avec acquittement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de la voie logique bidirectionnelle (B-LCSE). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE, de même qu'en terme d'états B-LCSE. Les informations de protocole sont transférées vers l'entité B-LCSE homologue au moyen des messages pertinents définis au paragraphe 6.

Il y a une entité B-LCSE entrante et une entité B-LCSE sortante. A chacune des extrémités entrante et sortante, il y a une instance d'entité B-LCSE pour chaque voie logique bidirectionnelle. Il n'y a pas de liaison entre une entité B-LCSE entrante et une entité B-LCSE sortante situées d'un côté, autre que celle obtenue au moyen des primitives provenant de l'utilisateur de l'entité B-LCSE et allant vers ce même utilisateur. Les conditions d'erreur des entités B-LCSE sont signalées.

Une voie logique bidirectionnelle comprend un couple de voies logiques monodirectionnelles associées. Les termes "dans le sens direct" (du côté sortant) sont utilisés pour faire référence à la transmission dans un sens à partir du terminal présentant la demande de voie logique bidirectionnelle vers l'autre terminal et "inverse" (côté entrant) est utilisé pour faire référence au sens opposé de transmission.

Les données ne devront être envoyées que sur une voie logique bidirectionnelle dans l'état ESTABLISHED (voie établie). Cependant les données peuvent être reçues sur la voie dans le sens direct quand l'entité B-LCSE entrante est dans l'état AWAITING CONFIRMATION (attente de la confirmation). Les données qui sont reçues dans d'autres états que l'état ESTABLISHED (voie

établie) et l'état AWAITING CONFIRMATION (attente de la confirmation) devront être ignorées et il faudra considérer qu'aucune défaillance ne s'est produite.

Un terminal peut refuser une demande d'ouverture de voie logique bidirectionnelle uniquement parce qu'il ne peut pas utiliser les paramètres demandés pour la voie inverse. Dans ce cas, il devra refuser la demande avec le paramètre de raison égal aux paramètres unsuitableReverseParameters et immédiatement déclencher les procédures d'établissement de la voie logique bidirectionnelle comme cela a été demandé par le terminal distant, dans lequel les paramètres dans le sens inverse sont identiques aux paramètres dans le sens direct et correspondent à un échec de la demande présentée par le terminal distant, avec les paramètres dans le sens direct que le terminal peut utiliser et dont on sait qu'ils pourront être utilisés par le terminal distant.

Le changement de mode devrait être effectué en ouvrant et en fermant les voies logiques existantes, ou en ouvrant de nouvelles voies logiques.

NOTE – Certaines Recommandations qui utilisent la présente Recommandation peuvent définir certaines voies logiques par défaut. Ces voies seront considérées comme étant à l'état ESTABLISHED dès le début de la communication et ne devront pas être ouvertes en utilisant ces procédures. Elles peuvent cependant être fermées par ces mêmes procédures et être ultérieurement réouvertes pour répondre au même objectif ou pour répondre à un objectif différent.

Un terminal qui ne peut plus traiter les signaux sur une voie logique devrait prendre des mesures appropriées pour la fermeture de la voie logique et la transmission des informations pertinentes relatives aux capacités (modifiées) vers le terminal distant.

Le texte ci-après est un aperçu général du fonctionnement du protocole des entités B-LCSE. Dans le cas de divergence entre cette spécification et la spécification formelle, cette dernière sera applicable.

8.5.1.1 Aperçu général du protocole – Entité B-LCSE sortante

L'ouverture d'une voie logique est déclenchée quand la primitive de demande ESTABLISH est présentée par l'utilisateur dans l'entité B-LCSE sortante. Un message OpenLogicalChannel, contenant les paramètres de voie logique dans les sens direct et inverse, est envoyé vers l'entité B-LCSE entrante homologue et le temporisateur T103 démarre. Si un message OpenLogicalChannelAck est reçu en réponse au message OpenLogicalChannel, alors le temporisateur T103 est arrêté, un message OpenLogicalChannelConfirm est envoyé vers l'entité B-LCSE entrante homologue et l'utilisateur est informé par la primitive de confirmation ESTABLISH que la voie logique a bien été ouverte. La voie logique peut à présent être utilisée pour émettre et recevoir les informations de l'utilisateur. Si, cependant, un message OpenLogicalChannelReject est reçu en réponse au message OpenLogicalChannel, le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité B-LCSE homologue a refusé l'établissement de la voie logique.

Si le temporisateur T103 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE et un message CloseLogicalChannel est envoyé à l'entité B-LCSE entrante homologue.

Une voie logique dont l'établissement a réussi peut être fermée quand la primitive de demande RELEASE est présentée par l'utilisateur de l'entité B-LCSE sortante. Un message CloseLogicalChannel est envoyé vers l'entité B-LCSE entrante homologue et le temporisateur T103 démarre. Quand un message CloseLogicalChannelAck est reçu, le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive de confirmation RELEASE que la fermeture de la voie logique a réussi.

Si le temporisateur T103 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE.

Avant que l'un des messages `OpenLogicalChannelAck` ou `OpenLogicalChannelReject` ait été reçu en réponse à un message `OpenLogicalChannel` préalablement envoyé, l'utilisateur de l'entité B-LCSE sortante peut fermer la voie logique en utilisant la primitive de demande `RELEASE`.

Avant que le message `CloseLogicalChannelAck` ait été reçu en réponse à un message `CloseLogicalChannel` préalablement envoyé, l'utilisateur de l'entité B-LCSE sortante peut ouvrir une nouvelle voie logique en présentant la primitive de demande `ESTABLISH`.

8.5.1.2 Aperçu général du protocole – Entité B-LCSE entrante

Quand un message `OpenLogicalChannel` est reçu dans l'entité B-LCSE entrante, l'utilisateur est informé de la demande d'ouverture d'une nouvelle voie logique par la primitive d'indication `ESTABLISH`. L'utilisateur de l'entité B-LCSE entrante signale l'acceptation de la demande d'ouverture de voie logique en présentant une primitive de réponse `ESTABLISH`, et un message `OpenLogicalChannelAck` est envoyé à l'entité B-LCSE sortante homologue. La voie dans le sens direct de la voie bidirectionnelle peut désormais être utilisée pour recevoir les informations de l'utilisateur. L'utilisateur de l'entité B-LCSE entrante signale le refus de la demande d'ouverture de voie logique en présentant la primitive de demande `RELEASE`, et un message `OpenLogicalChannelReject` est envoyé vers l'entité B-LCSE sortante homologue.

Quand un message `OpenLogicalChannelConfirm` est reçu dans l'entité B-LCSE entrante, l'utilisateur est informé que la voie logique bidirectionnelle est établie par la primitive de confirmation `ESTABLISH`. La voie dans le sens inverse de la voie logique bidirectionnelle peut désormais être utilisée pour transmettre les informations de l'utilisateur.

Une voie logique dont l'établissement a réussi peut être fermée quand le message `CloseLogicalChannel` est reçu dans l'entité B-LCSE entrante. L'utilisateur de l'entité B-LCSE entrante est informé par la primitive d'indication `RELEASE` et le message `CloseLogicalChannelAck` est envoyé à l'entité B-LCSE sortante homologue.

8.5.1.3 Résolution de litiges

Des litiges peuvent survenir quand des demandes d'ouverture de voies logiques sont lancées en même temps. Il est possible de déterminer qu'il y a un litige à partir des connaissances acquises par l'échange de capacités. Par ailleurs, les deux terminaux peuvent déclencher l'ouverture d'une voie logique bidirectionnelle pour répondre au même objectif, bien que les paramètres corrects demandés puissent être différents, et ces terminaux ont des capacités suffisantes pour satisfaire les deux demandes. Les terminaux doivent pouvoir détecter le moment où les deux situations surviendront et devront réagir indifféremment de la façon suivante.

Avant que les voies logiques puissent être ouvertes, l'un des terminaux sera désigné comme étant le terminal maître et l'autre comme le terminal esclave. Le protocole défini en 8.2 est un moyen de faire ce choix. Le terminal maître doit refuser sans délai toute demande présentée par le terminal esclave pour laquelle il détecte le litige. Le terminal esclave peut détecter de tels litiges, mais doit répondre à la demande présentée par le terminal maître, tout en sachant que sa demande antérieure sera refusée.

En ce qui concerne le second type de litige défini ci-dessus, il est impossible d'établir une distinction entre le cas où deux voies bidirectionnelles sont demandées et le cas où une seule de ces voies est effectivement demandée. Les terminaux devront réagir en supposant qu'une seule voie est demandée, mais un terminal pourra ultérieurement réitérer sa demande si l'hypothèse est incorrecte.

8.5.2 Communication entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE

8.5.2.1 Primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE

Les communications entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE s'effectuent en utilisant les primitives indiquées dans le Tableau 31.

Tableau 31 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
ESTABLISH (établissement)	FORWARD_PARAM REVERSE_PARAM	FORWARD_PARAM REVERSE_PARAM	REVERSE_DATA	REVERSE_DATA
RELEASE (libération)	CAUSE (raison)	SOURCE (origine) CAUSE (raison)	non défini (Note 2)	– (Note 1)
ERROR (erreur)	non défini	ERRCODE	non défini	non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'existe pas.				

8.5.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- les primitives ESTABLISH sont utilisées pour établir une voie logique pour les communications audiovisuelles et les transmissions de données;
- les primitives RELEASE sont utilisées pour libérer une voie logique;
- la primitive ERROR indique les erreurs B-LCSE à une entité de gestion.

8.5.2.3 Définition des paramètres

La définition des paramètres des primitives indiquées au Tableau 31 est la suivante:

- le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie dans le sens direct, c'est-à-dire du terminal contenant l'entité B-LCSE sortante vers le terminal contenant l'entité B-LCSE entrante. Ce paramètre est mappé au champ forwardLogicalChannelParameters du message OpenLogicalChannel et transmis de façon transparente vers l'utilisateur de l'entité LCSE homologue;
- le paramètre REVERSE_PARAM spécifie les paramètres associés à la voie inverse, c'est-à-dire du terminal contenant l'entité B-LCSE entrante vers le terminal contenant l'entité B-LCSE sortante. Ce paramètre est mappé au champ reverseLogicalChannelParameters du message OpenLogicalChannel et transmis de façon transparente vers l'utilisateur de l'entité B-LCSE homologue;
- le paramètre REVERSE_DATA spécifie certains paramètres associés à la voie inverse, c'est-à-dire du terminal contenant l'entité B-LCSE entrante vers le terminal contenant l'entité B-LCSE sortante. Ce paramètre est mappé au champ reverseLogicalChannelParameters du message OpenLogicalChannelAck et transmis de façon transparente vers l'utilisateur de l'entité B-LCSE homologue;
- le paramètre SOURCE indique à l'utilisateur de l'entité B-LCSE la source de la libération de la voie logique. Le paramètre SOURCE a la valeur de "USER" ou "B-LCSE", indiquant soit l'utilisateur de l'entité B-LCSE ou l'entité B-LCSE elle-même. Ce dernier choix peut être le résultat d'une erreur de protocole;

- e) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité B-LCSE homologue a refusé une demande d'établissement de voie logique. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "B-LCSE";
- f) le paramètre ERRCODE indique le type d'erreur de l'entité B-LCSE. Le Tableau 35 indique les valeurs autorisées pour le paramètre ERRCODE.

8.5.2.4 Etats de l'entité B-LCSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE, de même que l'échange de messages entre les entités B-LCSE homologues. Les états sont spécifiés distinctement pour chacune des entités B-LCSE entrante et sortante. Les états correspondant à une entité B-LCSE sortante sont:

Etat 0: RELEASED (libre)

La voie logique est libre. La voie logique ne devra pas être utilisée pour envoyer ou recevoir des données.

Etat 1: AWAITING ESTABLISHMENT (attente de l'établissement)

L'entité B-LCSE sortante attend d'établir une voie logique avec une entité B-LCSE entrante homologue. La voie logique ne devra pas être utilisée pour envoyer ou recevoir des données.

Etat 2: ESTABLISHED (voie logique établie)

La liaison entre les voies logiques des entités B-LCSE homologues a été établie. La voie logique peut être utilisée pour envoyer et recevoir des données.

Etat 3: AWAITING RELEASE (attente de libération)

L'entité B-LCSE sortante attend de libérer une voie logique avec l'entité B-LCSE entrante homologue. La voie logique ne devra pas être utilisée pour envoyer les données, mais les données peuvent toujours être reçues.

Les états correspondant à une entité B-LCSE entrante sont:

Etat 0: RELEASED (libre)

La voie logique est libérée. La voie logique ne devra pas être utilisée pour envoyer ou recevoir des données.

Etat 1: AWAITING ESTABLISHMENT (attente de l'établissement)

L'entité B-LCSE entrante attend d'établir une voie logique avec une entité B-LCSE sortante homologue. La voie logique ne devra pas être utilisée pour envoyer ou recevoir des données.

Etat 2: AWAITING CONFIRMATION (attente de confirmation)

L'entité B-LCSE entrante attend la confirmation que la voie logique est établie avec une entité B-LCSE sortante homologue. La voie logique ne devra pas être utilisée pour envoyer des données, mais des données peuvent être reçues.

Etat 3: ESTABLISHED (voie logique établie)

La liaison entre les voies logiques des entités B-LCSE homologues a été établie. La voie logique peut être utilisée pour envoyer et recevoir des données.

8.5.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE est définie ici. La séquence autorisée des primitives fait référence à des états de l'entité B-LCSE du point

de vue de l'utilisateur de l'entité B-LCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités B-LCSE sortante et B-LCSE entrante, comme cela est indiqué dans les Figures 15 et 16 respectivement.

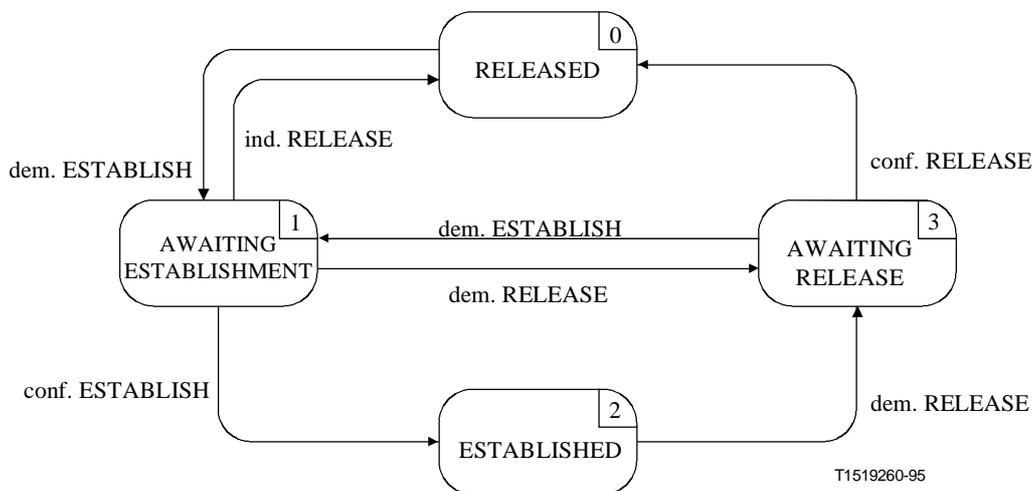


Figure 15/H.245 – Diagramme de changement d'états pour la séquence de primitives dans l'entité B-LCSE sortante

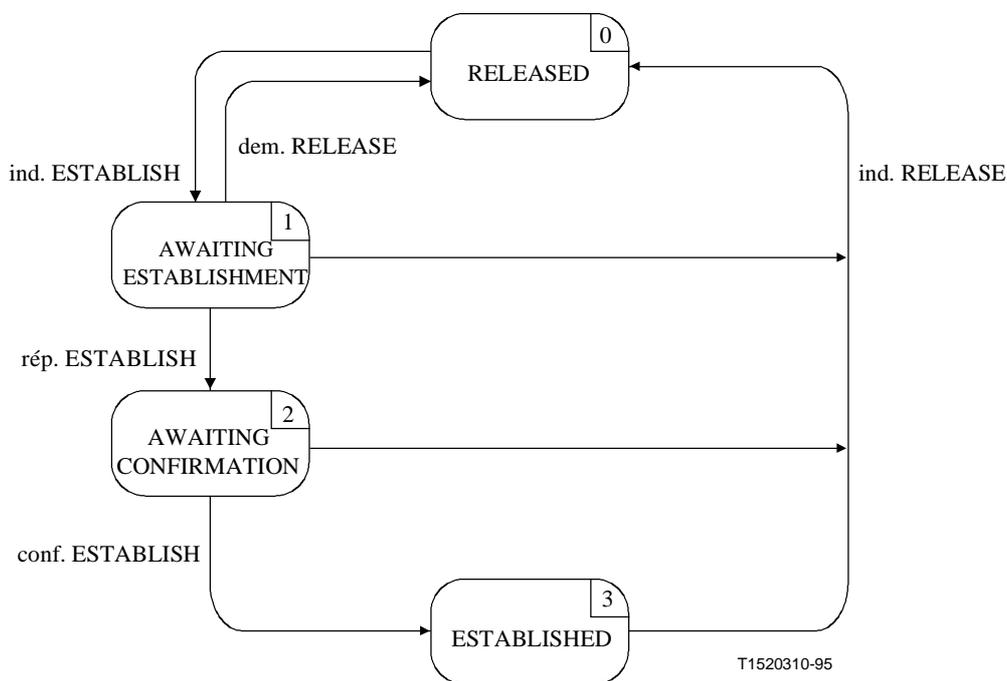


Figure 16/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité B-LCSE entrante

8.5.3 Communication entre les entités B-LCSE homologues

8.5.3.1 Messages B-LCSE

Le Tableau 32 indique les messages et champs des entités B-LCSE, tels qu'ils ont été définis au paragraphe 6 et qui concernent le protocole des entités B-LCSE.

Tableau 32/H.245 – Noms et champs de message des entités B-LCSE

Fonction	Message	Sens	Champ
establishment	OpenLogicalChannel	S → E	forwardLogicalChannelNumber forwardLogicalChannelParameters reverseLogicalChannelParameters
	OpenLogicalChannelAck	S ← E	forwardLogicalChannelNumber reverseLogicalChannelParameters
	OpenLogicalChannelReject	S ← E	forwardLogicalChannelNumber cause
	OpenLogicalChannelConfirm	S → E	forwardLogicalChannelNumber
release	CloseLogicalChannel	S → E	forwardLogicalChannelNumber
	CloseLogicalChannelAck	S ← E	source forwardLogicalChannelNumber
S sortant E entrant			

8.5.3.2 Variables d'état de l'entité B-LCSE

Les variables d'état suivantes sont définies dans l'entité B-LCSE sortante:

out_LCN

Cette variable d'état distingue entre elles les entités B-LCSE sortantes. Elle est initialisée pendant l'initialisation des entités B-LCSE sortantes. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages B-LCSE envoyés à partir d'une entité B-LCSE sortante. En ce qui concerne les messages B-LCSE reçus dans une entité B-LCSE sortante, le message de valeur de champ forwardLogicalChannelNumber est identique à la valeur du paramètre out_LCN.

La variable d'état suivante est définie dans l'entité B-LCSE entrante:

in_LCN

Cette variable d'état est utilisée pour distinguer les entités B-LCSE entrantes. Elle est initialisée lors de l'initialisation de l'entité B-LCSE entrante. La valeur du paramètre in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages B-LCSE envoyés à partir d'une entité B-LCSE entrante. En ce qui concerne les messages B-LCSE reçus dans l'entité B-LCSE entrante, la valeur de champ du message forwardLogicalChannelNumber est identique à la valeur du paramètre in_LCN.

8.5.3.3 Temporisateurs B-LCSE

Le temporisateur suivant est spécifié pour l'entité B-LCSE entrante et l'entité B-LCSE sortante:

T103

Dans l'entité B-LCSE sortante, ce temporisateur est utilisé pendant les états AWAITING ESTABLISHMENT (ATTENTE DE L'ETABLISSEMENT) et AWAITING RELEASE (ATTENTE DE LA LIBERATION). Il spécifie le temps maximal pendant lequel aucun des messages OpenLogicalChannelAck, OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu.

Dans l'entité B-LCSE entrante, ce temporisateur est utilisé pendant l'état AWAITING CONFIRMATION (ATTENTE DE CONFIRMATION). Il spécifie le temps maximal pendant lequel aucun message de confirmation OpenLogicalChannelConfirm ne peut être reçu.

8.5.4 Procédures B-LCSE

8.5.4.1 Introduction

La Figure 17 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités B-LCSE entrante et sortante.

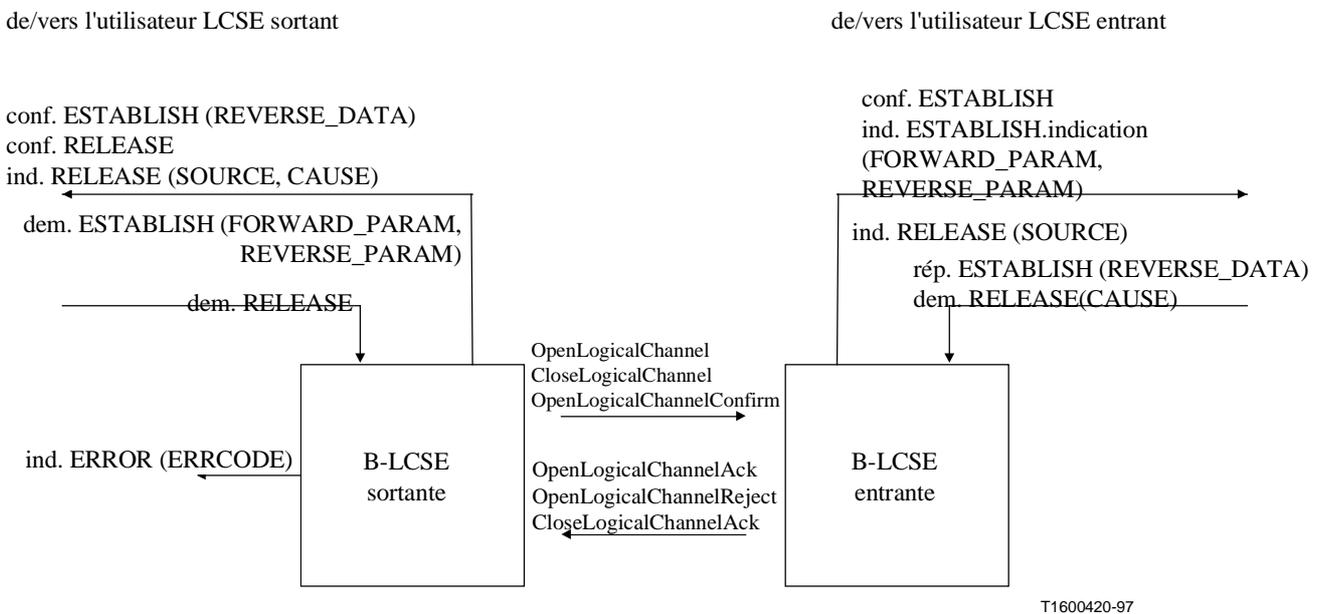


Figure 17/H.245 – Primitives et messages dans l'entité de signalisation de voie logique bidirectionnelle

8.5.4.2 Valeurs par défaut des paramètres de primitives

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent les valeurs indiquées dans le Tableau 33.

Tableau 33/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut (Note)
ind. ESTABLISH	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
	REVERSE_PARAM	OpenLogicalChannel.reverseLogicalChannelParameters
conf. ESTABLISH	REVERSE_DATA	OpenLogicalChannelAck.reverseLogicalChannelParameters
ind.RELEASE	SOURCE	CloseLogicalChannel.source
	CAUSE	null
NOTE – Un paramètre de primitive devra être codé comme nul si un champ de message indiqué n'est pas présent dans le message.		

8.5.4.3 Valeurs par défaut du champ de message

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les champs de messages prennent les valeurs indiquées dans le Tableau 34.

Tableau 34/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note)
OpenLogicalChannel	forwardLogicalChannelNumber	out_LCN
	forwardLogicalChannelParameters	dem. ESTABLISH(FORWARD_PARAM)
	reverseLogicalChannelParameters	dem. ESTABLISH(REVERSE_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
	reverseLogicalChannelParameters	rép. ESTABLISH(REVERSE_DATA)
OpenLogicalChannelReject	forwardLogicalChannelNumber cause	in_LCN dem. RELEASE(CAUSE)
OpenLogicalChannelConfirm	forwardLogicalChannelNumber	out_LCN
CloseLogicalChannel	forwardLogicalChannelNumber	out_LCN
	source	user
CloseLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
NOTE – Un champ de message ne devra pas être codé si le paramètre de primitive correspondant est "null", c'est à-dire est absent.		

8.5.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition particulière d'erreur. Le Tableau 35 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE sortante et le Tableau 36 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE entrante.

Tableau 35/H.245 – Valeurs des paramètres ERRCODE de l'entité B-LCSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	État
message inapproprié	A	OpenLogicalChannelAck	RELEASED
	B	OpenLogicalChannelReject	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
pas de réponse provenant de l'entité B-LCSE homologue	D	expiration du temporisateur T103	AWAITING ESTABLISHMENT AWAITING RELEASE

Tableau 36/H.245 – Valeurs des paramètres ERRCODE de l'entité B-LCSE entrante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
message inapproprié	E	OpenLogicalChannelConfirm	AWAITING ESTABLISHMENT
pas de réponse provenant de l'entité B-LCSE homologue	F	expiration du temporisateur T103	AWAITING CONFIRMATION

8.5.4.5 Description SDL

Les procédures de l'entité B-LCSE entrante et de l'entité B-LCSE sortante sont décrites en langage SDL dans les Figures 18 et 19 respectivement.

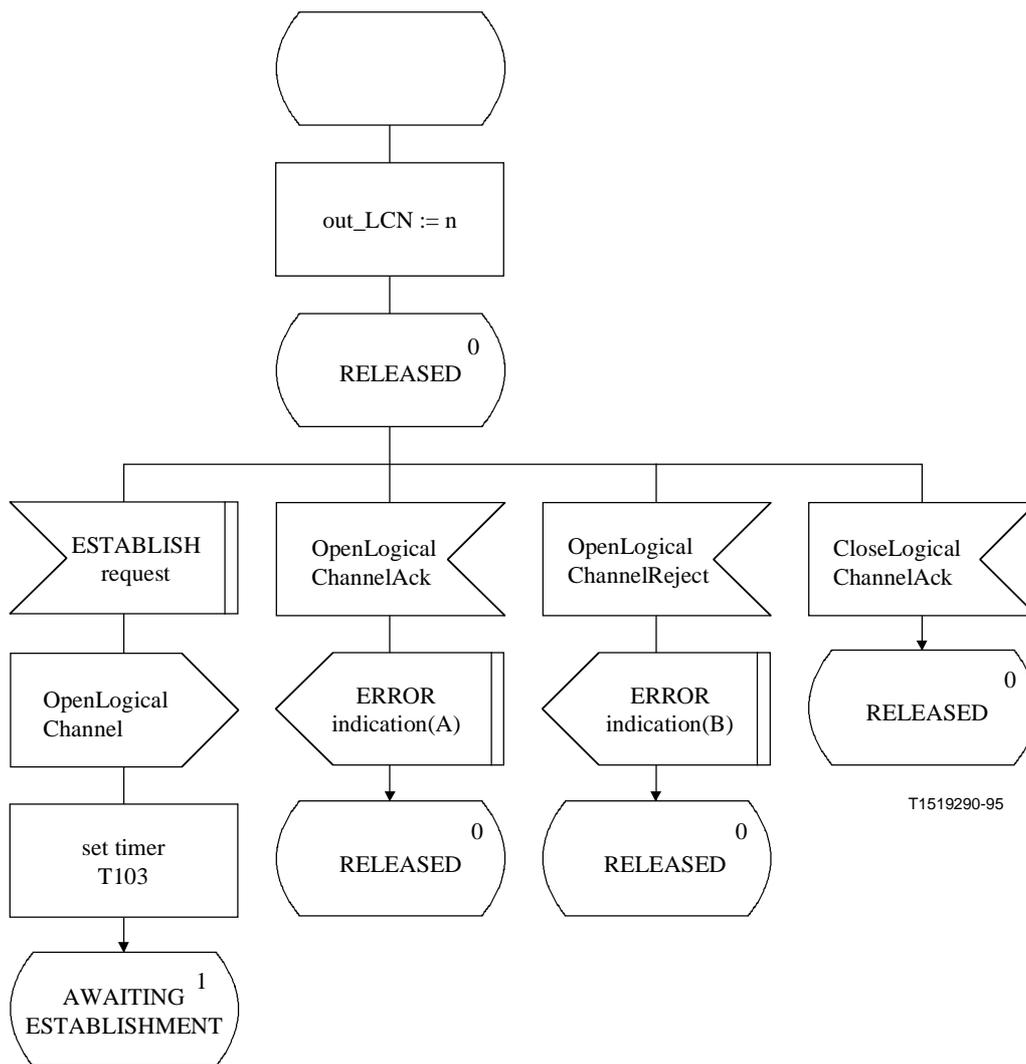


Figure 18 i)/H.245 – Description SDL de l'entité B-LCSE sortante

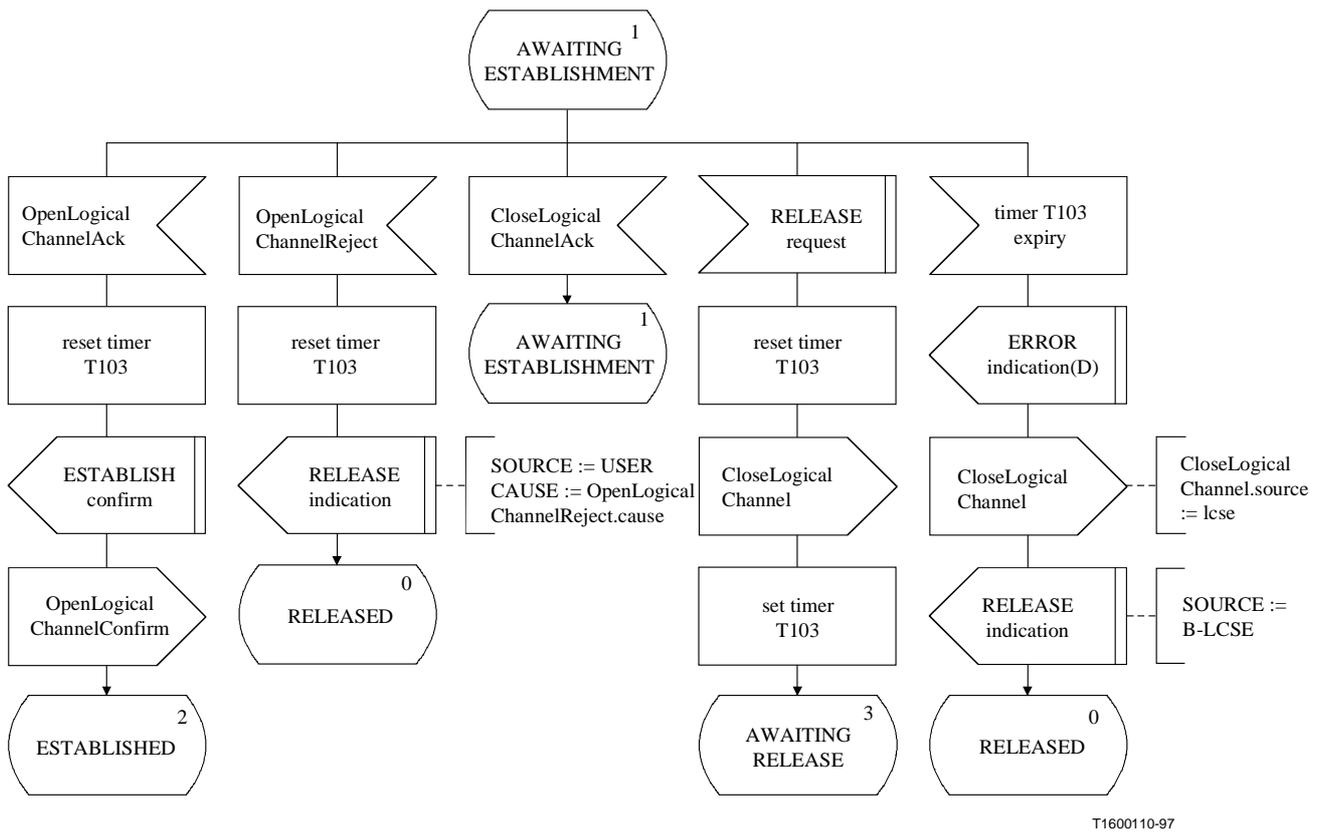


Figure 18 ii)/H.245 – Description SDL de l'entité B-LCSE sortante

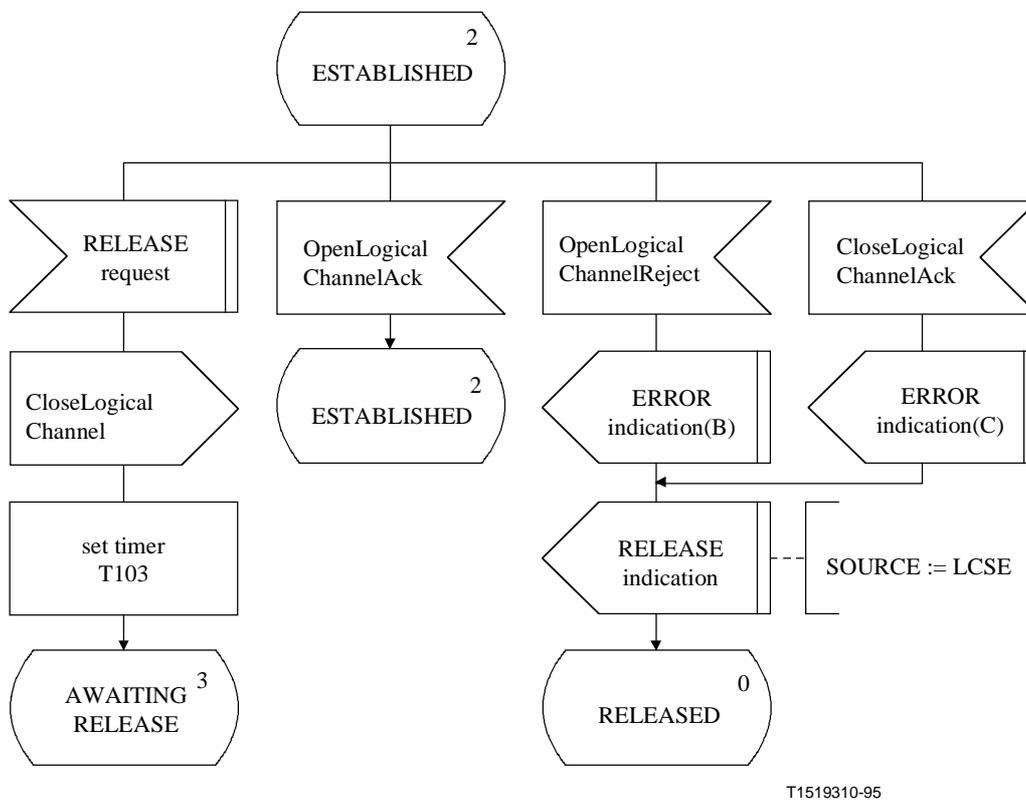


Figure 18 iii)/H.245 – Description SDL de l'entité B-LCSE sortante

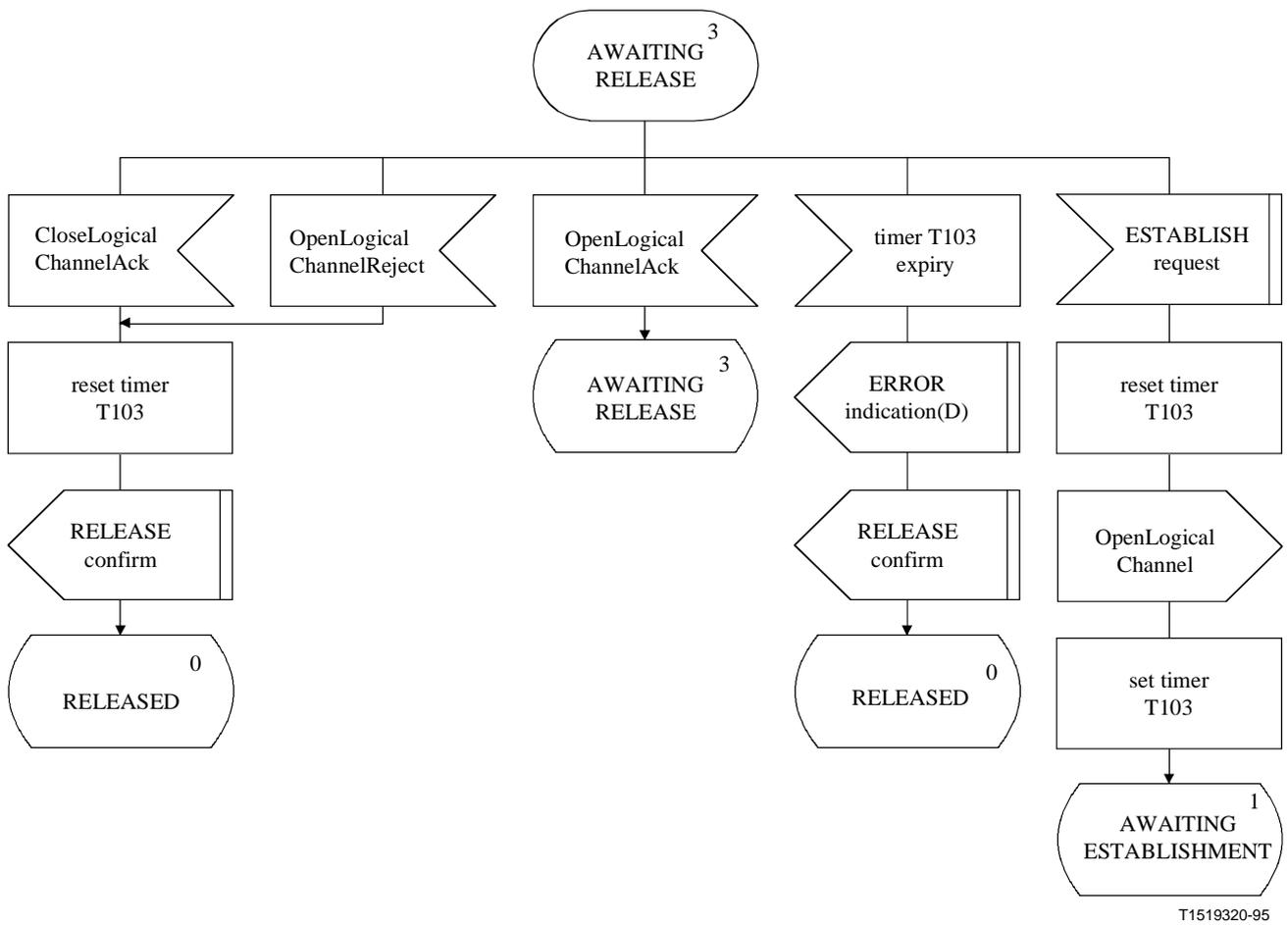


Figure 18 iv)/H.245 – Description SDL de l'entité B-LCSE sortante

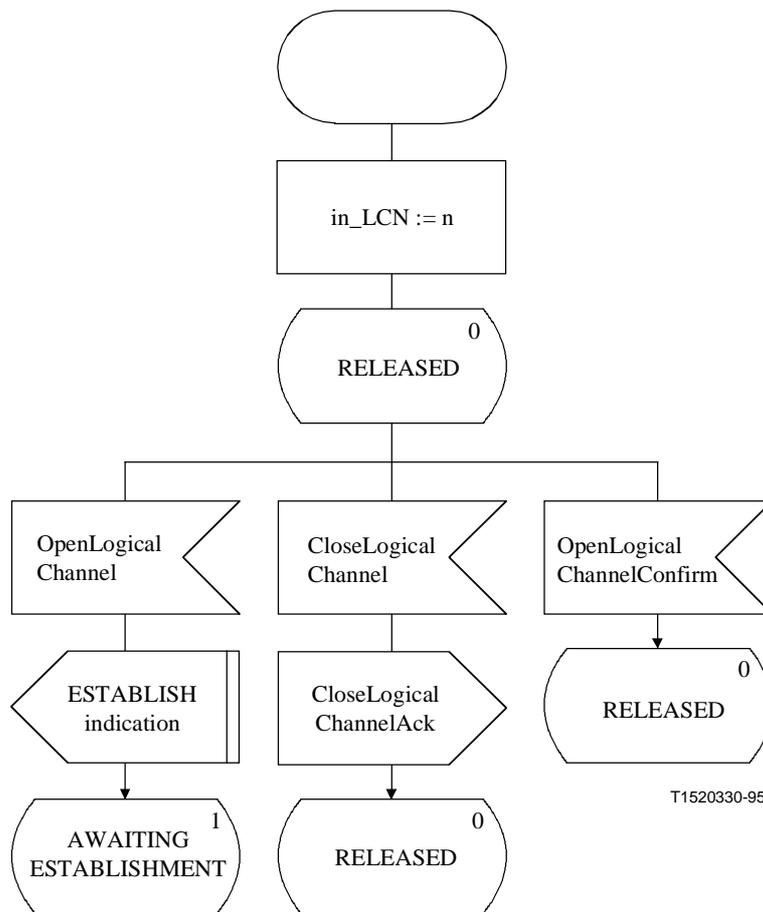


Figure 19 i)/H.245 – Description SDL de l'entité B-LCSE entrante

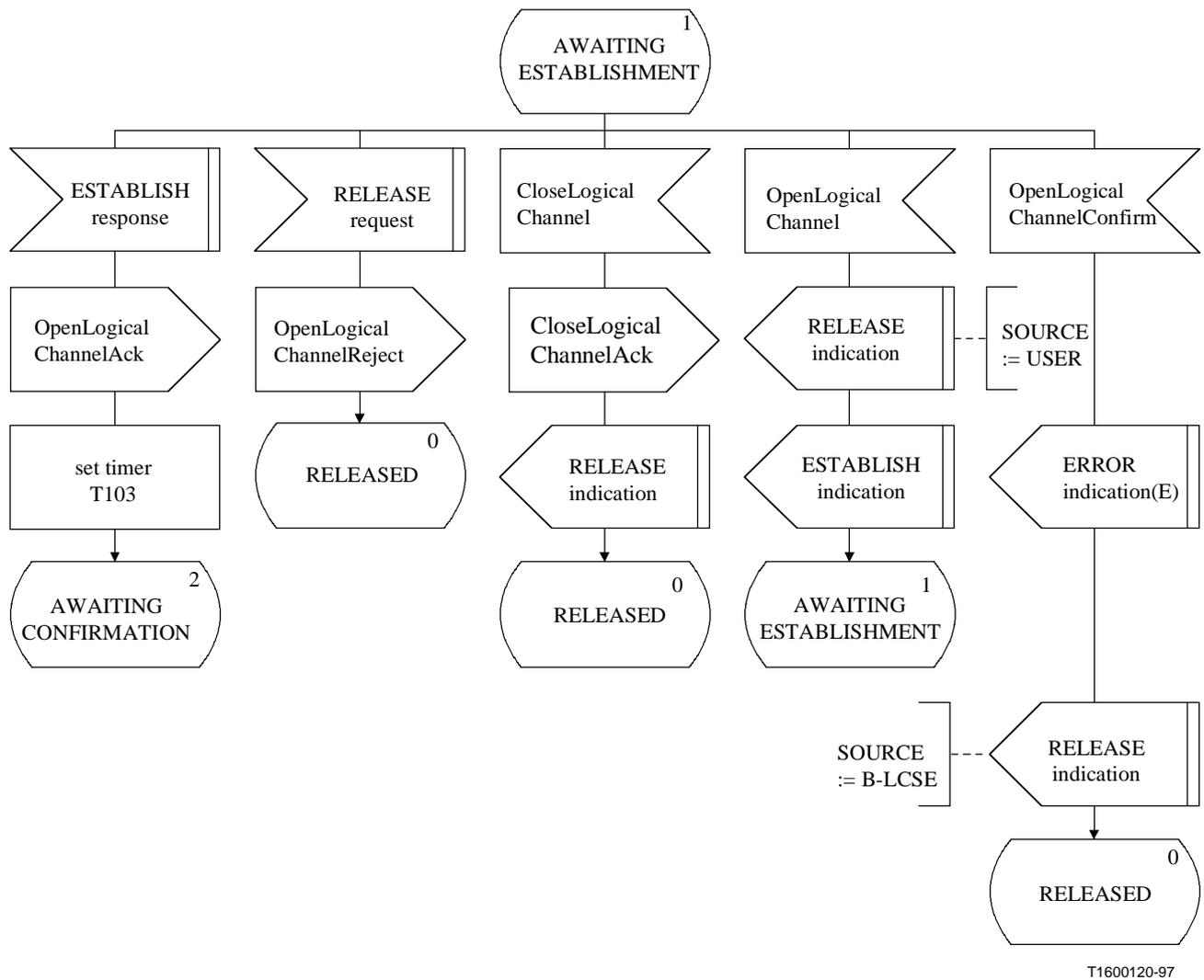


Figure 19 ii)/H.245 – Description SDL de l'entité B-LCSE entrante

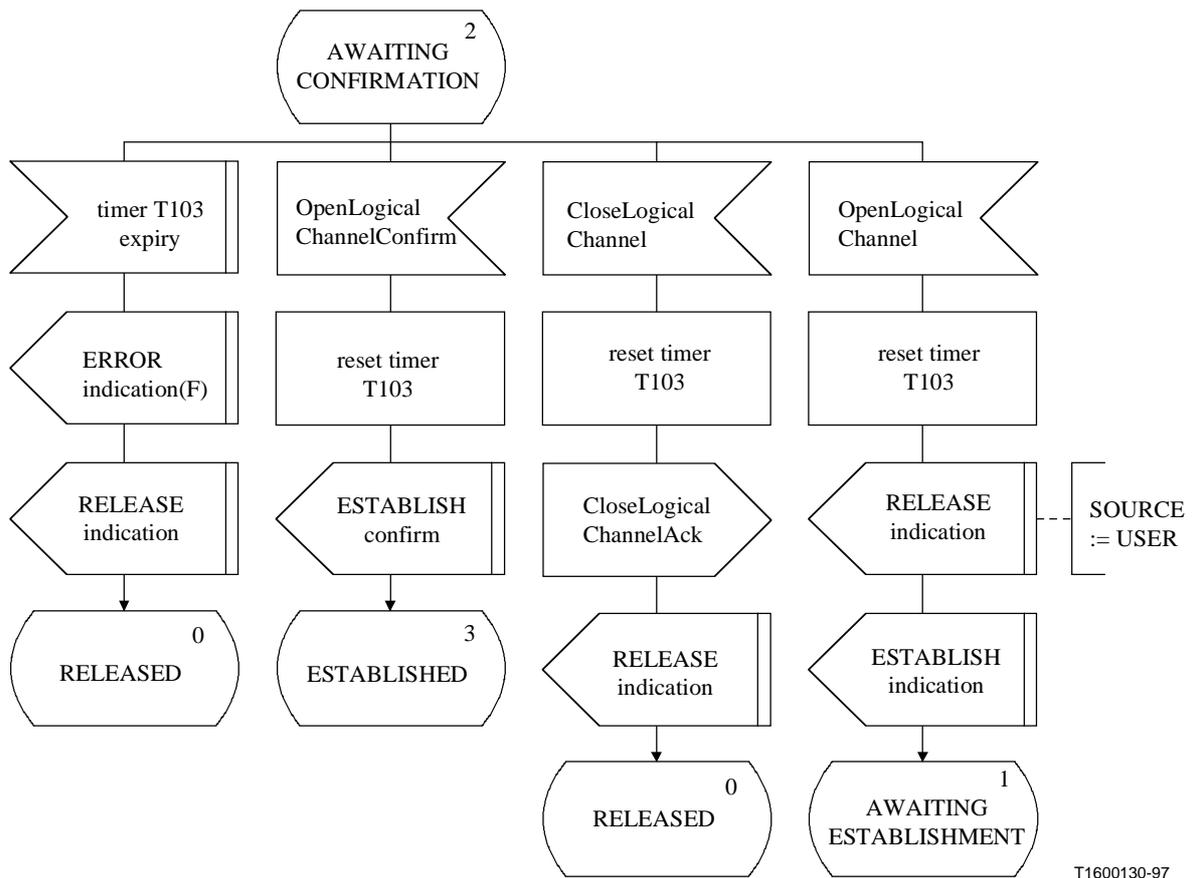


Figure 19 iii)/H.245 – Description SDL de l'entité B-LCSE entrante

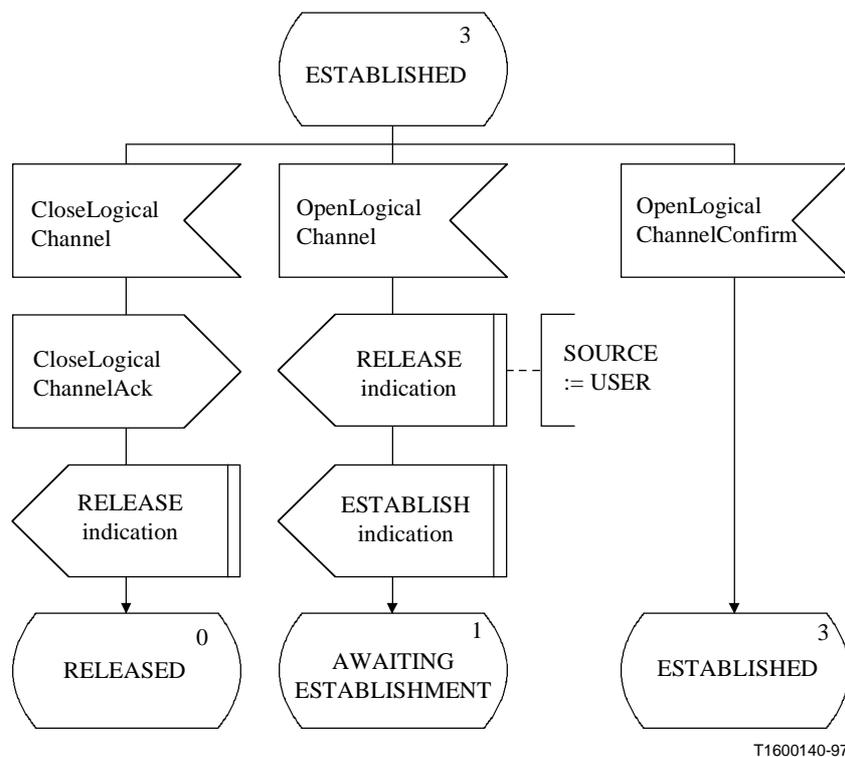


Figure 19 iv)/H.245 – Description SDL de l'entité B-LCSE entrante

8.6 Procédures de fermeture de la voie logique

8.6.1 Introduction

Ces procédures sont utilisées par un terminal pour demander à un terminal distant de fermer une voie logique. Il convient de noter que ces procédures sont uniquement des procédures de demande de fermeture; la voie logique est effectivement fermée en utilisant les procédures LCSE et B-LCSE. Ces procédures sont désignées ici comme l'entité de signalisation de fermeture de voie logique (CLCSE, *close logical channel signalling entity*). Elles sont spécifiées en termes de primitives et d'états à l'interface entre l'entité CLCSE et l'utilisateur de l'entité CLCSE. Les informations de protocole sont transférées vers l'entité CLCSE homologue au moyen des messages pertinents définis dans le paragraphe 6. Il y a une entité CLCSE sortante et une entité CLCSE entrante. A chacune des extrémités entrante et sortante, il y a une instance d'entité CLCSE pour chaque voie logique.

Si un terminal ne peut pas traiter les signaux entrants, il peut utiliser ces procédures pour demander la fermeture des voies logiques correspondantes.

Un terminal qui répond de façon positive, c'est-à-dire, en présentant une primitive de réponse CLOSE, devra déclencher la fermeture de la voie logique en envoyant la primitive de demande RELEASE à l'entité LCSE ou B-LCSE appropriée dès que cela est possible.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

8.6.1.1 Aperçu général du protocole – Entité CLCSE sortante

Une procédure de fermeture de demande de voie logique est déclenchée quand la primitive de demande CLOSE est présentée par l'utilisateur de l'entité CLCSE sortante. Un message RequestChannelClose est envoyé vers l'entité CLCSE entrante homologue, et le temporisateur T108 démarre. Si, cependant, un message RequestChannelCloseReject est reçu en réponse au message RequestChannelClose, le temporisateur est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité CLCSE homologue a refusé la fermeture de la voie logique.

Si le temporisateur T108 arrive à expiration, l'utilisateur de l'entité CLCSE sortante est informé par la primitive d'indication REJECT et un message RequestChannelCloseRelease est envoyé.

8.6.1.2 Aperçu général du protocole – Entité CLCSE entrante

Quand un message RequestChannelClose est reçu dans l'entité CLCSE entrante, l'utilisateur est informé de la demande de fermeture de la voie logique par la primitive d'indication CLOSE. L'utilisateur de l'entité CLCSE entrante signale l'acceptation de demande de fermeture de la voie logique en présentant une primitive de réponse CLOSE, et un message RequestChannelCloseAck est envoyé vers l'entité CLCSE sortante homologue. L'utilisateur de l'entité CLCSE entrante signale le refus de la demande de voie logique en présentant la primitive de demande REJECT, et un message RequestChannelCloseReject est envoyé à l'entité CLCSE sortante homologue.

8.6.2 Communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE

8.6.2.1 Primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE

La communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE s'effectue en utilisant les primitives indiquées dans le Tableau 37.

Tableau 37/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
CLOSE	– (Note 1)	–	–	–
REJECT	CAUSE	SOURCE CAUSE	non défini (Note 2)	non défini
NOTE 1 – "–" signifie l'absence de paramètres.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

8.6.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives CLOSE sont utilisées pour demander la fermeture d'une voie logique;
- b) les primitives REJECT sont utilisées pour refuser la fermeture d'une voie logique.

8.6.2.3 Définition des paramètres

La définition des paramètres des primitives indiquées dans le Tableau 37 est la suivante:

- a) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE prend la valeur de "USER" ou "PROTOCOL". Ce dernier cas peut survenir par suite d'une fin de temporisation;
- b) le paramètre CAUSE indique le motif de refus de la fermeture d'une voie logique. Le paramètre CAUSE est absent quand le paramètre SOURCE indique "PROTOCOL".

8.6.2.4 Etats pour une entité CLCSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE.

Les états pour une entité CLCSE sortante sont:

Etat 0: IDLE

L'entité CLCSE est à l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité CLCSE attend une réponse provenant de l'entité CLCSE distante.

Les états correspondant à une entité CLCSE entrante sont:

Etat 0: IDLE

L'entité CLCSE est à l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité CLCSE attend une réponse provenant de l'utilisateur de l'entité CLCSE.

8.6.2.5 Diagramme de changement d'états

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités CLCSE entrante et sortante, comme cela est indiqué aux Figures 20 et 21 respectivement.

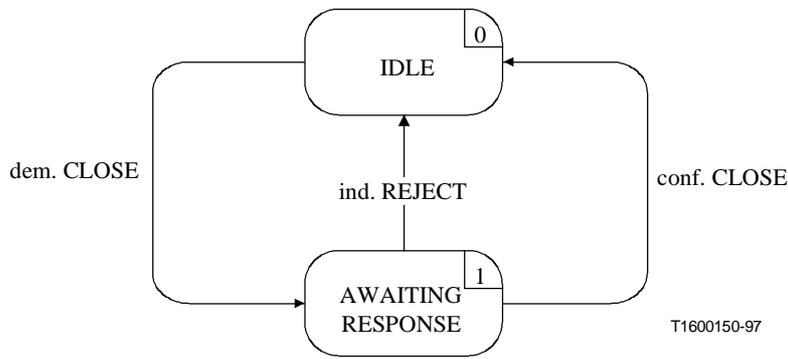


Figure 20/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité CLCSE sortante

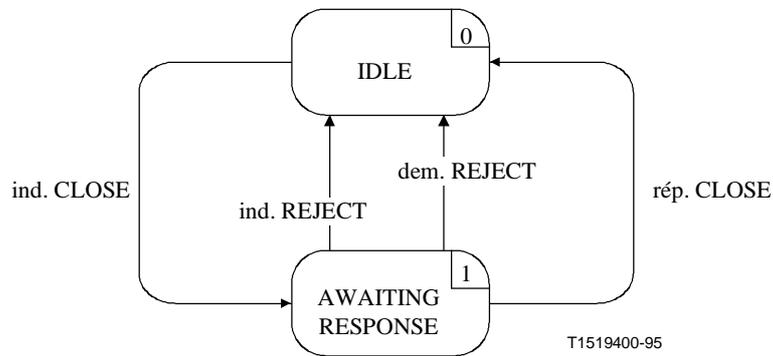


Figure 21/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité CLCSE entrante

8.6.3 Communication entre les entités CLCSE homologues

8.6.3.1 Messages

Le Tableau 38 indique les messages et les champs des entités CLCSE tels qu'ils ont été définis dans le paragraphe 6 et se rapportent au protocole CLCSE.

Tableau 38/H.245 – Noms et champs des messages des entités CLCSE

Fonction	Message	Sens	Champ
transfer	RequestChannelClose	S → E	forwardLogicalChannelNumber
	RequestChannelCloseAck	S ← E	forwardLogicalChannelNumber
	RequestChannelCloseReject	S ← E	forwardLogicalChannelNumber
reset	RequestChannelCloseRelease	S → E	forwardLogicalChannelNumber
S sortant E entrant			

8.6.3.2 Variables d'état CLCSE

La variable d'état suivante est définie dans l'entité CLCSE sortante:

out_LCN

Cette variable d'état établit une distinction entre les entités CLCSE sortantes. Elle est initialisée lors de l'initialisation des entités CLCSE sortantes. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages CLCSE envoyés à partir d'une entité CLCSE sortante. En ce qui concerne les messages CLCSE reçus dans l'entité CLCSE sortante, la valeur de champ du message forwardLogicalChannelNumber est identique à celle de la valeur out_LCN.

La variable d'état suivante est définie dans l'entité CLCSE entrante:

in_LCN

Cette variable d'état établit une distinction entre les entités CLCSE entrantes. Elle est initialisée lors de l'initialisation de l'entité CLCSE entrante. La valeur du paramètre in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages CLCSE envoyés à partir d'une entité CLCSE entrante. En ce qui concerne les messages CLCSE reçus dans l'entité CLCSE entrante, la valeur de champ du message forwardLogicalChannelNumber est identique à la valeur du paramètre in_LCN.

8.6.3.3 Temporisateurs CLCSE

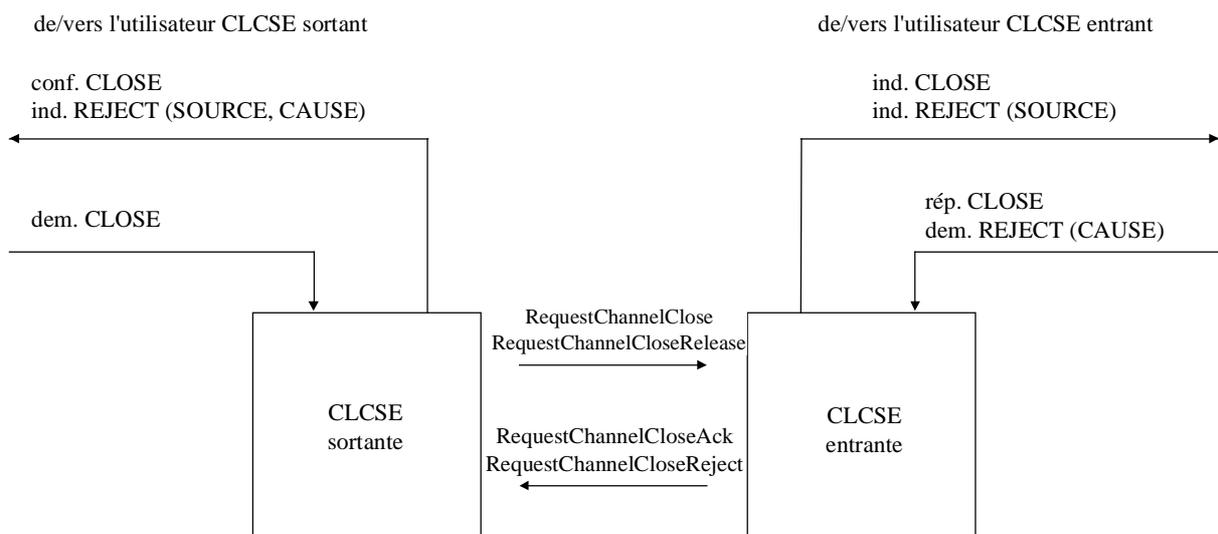
Le temporisateur suivant est spécifié pour l'entité CLCSE sortante:

T108

Ce temporisateur est utilisé à l'état AWAITING RESPONSE (attente de la réponse). Il spécifie le temps maximal pendant lequel aucun des messages RequestChannelCloseAck ou RequestChannelCloseReject ne peut être reçu.

8.6.4 Procédures CLCSE

La Figure 22 récapitule les primitives CLCSE et leurs paramètres, de même que les messages, pour chacune des entités CLCSE entrante et sortante.



T1519410-95

Figure 22/H.245 – Primitives et messages dans l'entité de signalisation de fermeture de voie logique (CLCSE)

8.6.4.1 Valeurs par défaut des paramètres de primitives

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 39.

Tableau 39/H.245 – Valeurs des paramètres de primitives par défaut

Primitive	Paramètre	Valeur par défaut
ind. REJECT	SOURCE CAUSE	USER null

8.6.4.2 Valeurs par défaut des champs de messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs de messages prennent des valeurs identiques à celles indiquées dans le Tableau 40.

Tableau 40/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut
RequestChannelClose	forwardLogicalChannelNumber	out_LCN
RequestChannelCloseAck	forwardLogicalChannelNumber	in_LCN
RequestChannelCloseReject	forwardLogicalChannelNumber cause	in_LCN dem. REJECT (CAUSE)
RequestChannelCloseRelease	forwardLogicalChannelNumber	out_LCN

8.6.4.3 Description SDL

Les procédures de l'entité CLCSE sortante et de l'entité CLCSE entrante sont décrites en langage SDL aux Figures 23 et 24 respectivement.

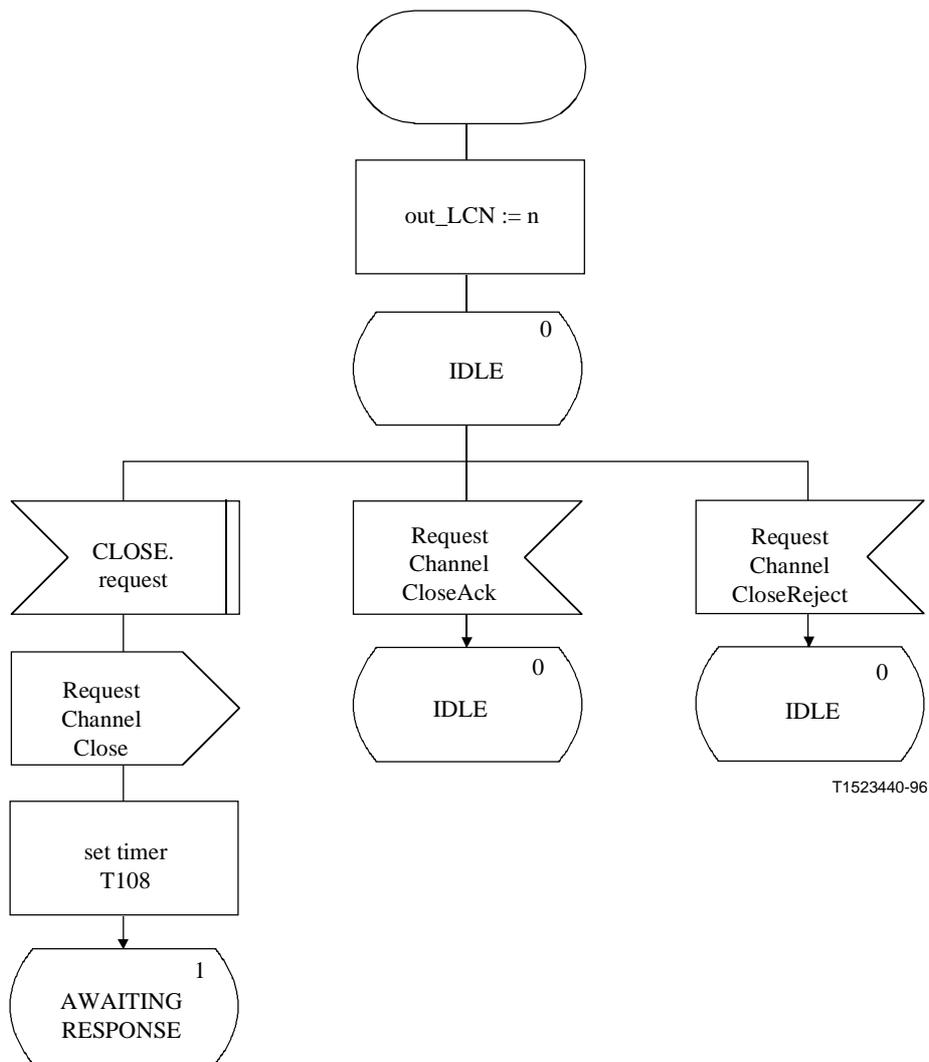
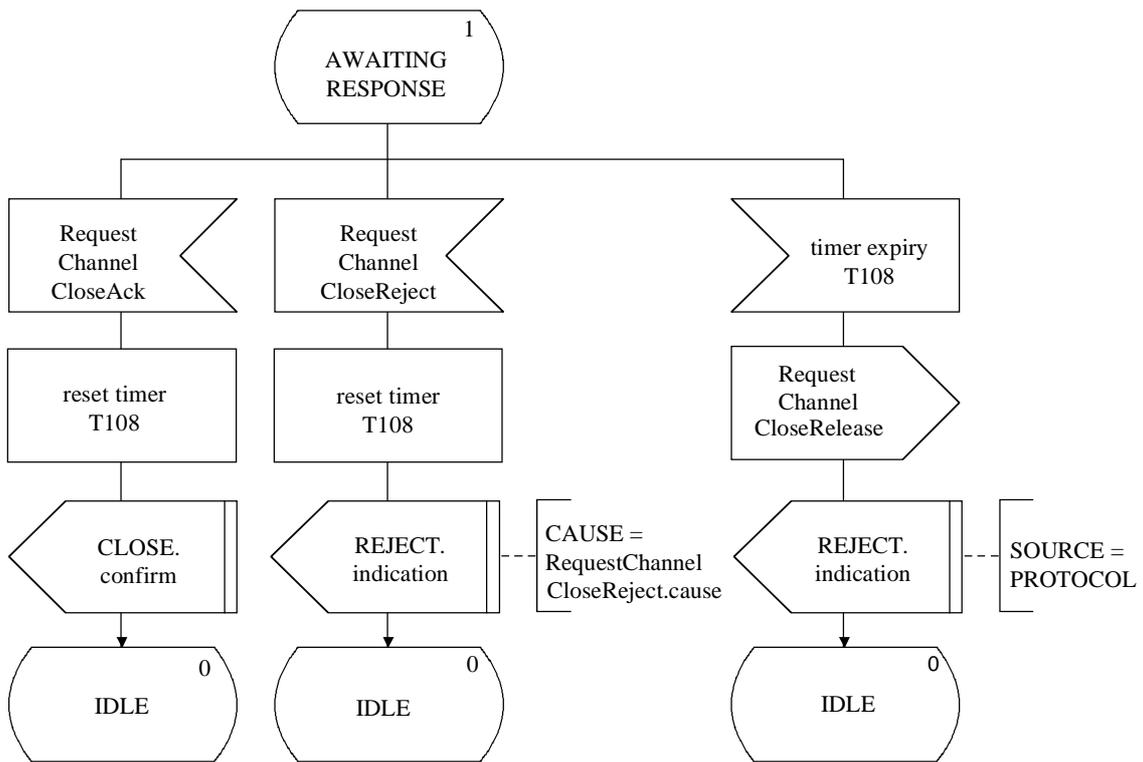
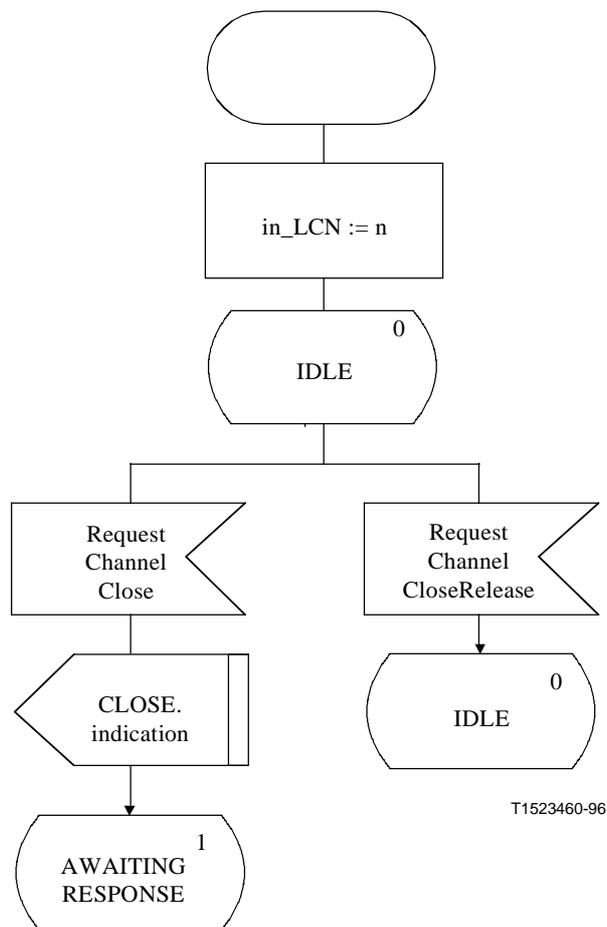


Figure 23 i)/H.245 – Description SDL de l'entité CLCSE sortante



T1523450-96

Figure 23 ii)/H.245 – Description SDL de l'entité CLCSE sortante



T1523460-96

Figure 24 i)/H.245 – Description SDL de l'entité CLCSE entrante

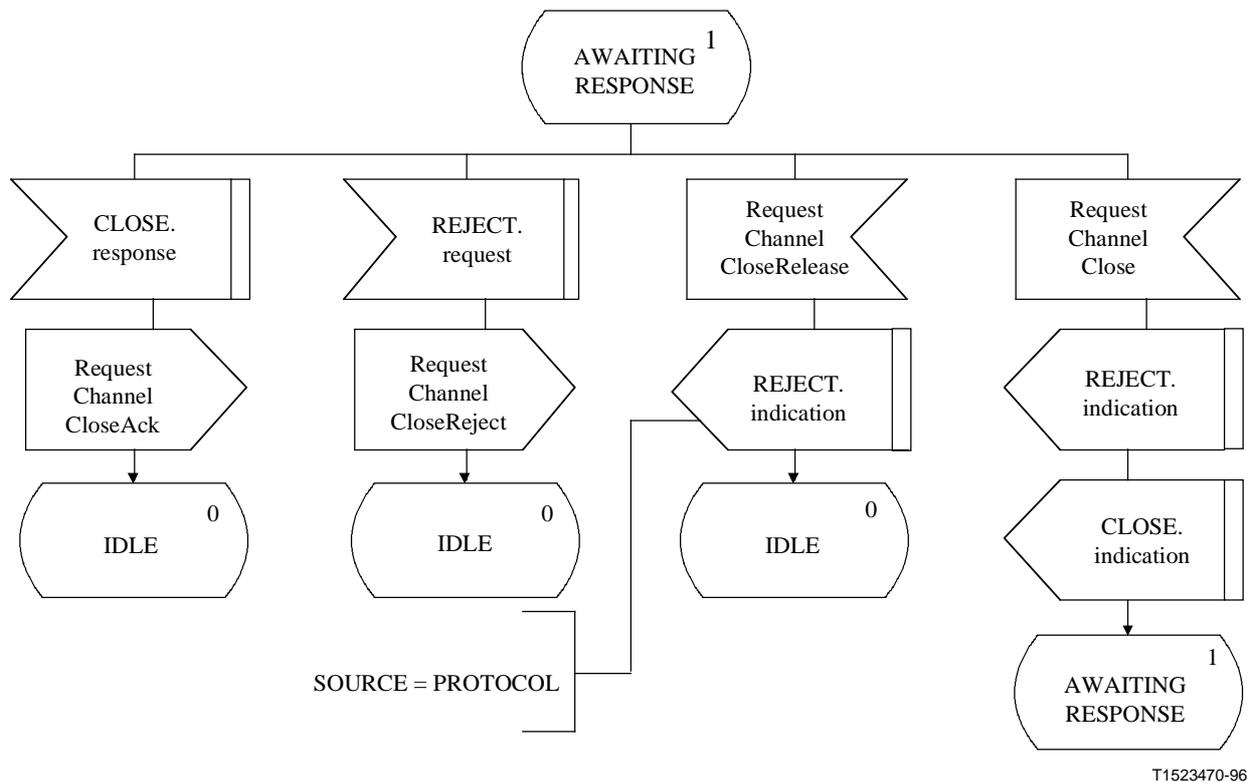


Figure 24 ii)/H.245 – Description SDL de l'entité CLCSE entrante

8.7 Procédures du tableau de multiplexage H.223

8.7.1 Introduction

Le tableau de multiplexage permet d'associer chaque octet d'une trame MUX-PDU selon H.223 [8] à un numéro de voie logique donné. Le tableau de multiplexage selon H.223 peut contenir jusqu'à 16 entrées, numérotées de 0 à 15. Les entrées de tableau comprises entre 1 et 15 devront être envoyées par les émetteurs aux récepteurs comme cela est spécifié dans les procédures suivantes.

Les procédures décrites ici sont désignées comme l'entité de signalisation du tableau de multiplexage (MTSE, *multiplex table signalling entity*). Elles sont spécifiées en termes de primitives et d'états à l'interface entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les informations de protocole sont transmises au moyen des messages pertinents définis dans le paragraphe 6.

On distingue une entité MTSE sortante et une entité MTSE entrante. Il existe une instance d'entité MTSE pour chaque entrée du tableau de multiplexage.

Un terminal émetteur utilise ce protocole pour indiquer à un terminal distant une ou plusieurs nouvelles entrées du tableau de multiplexage. Le terminal distant peut accepter ou refuser les nouvelles entrées du tableau de multiplexage. Si le terminal distant accepte une entrée du tableau de multiplexage, l'entrée précédente ayant un numéro d'entrée donné est remplacé par la nouvelle entrée.

L'émetteur peut désactiver une entrée de tableau de multiplexage en envoyant un descripteur MultiplexEntrydescriptor sans elementList. L'émetteur ne devra aucunement utiliser une entrée de tableau de multiplexage qui est désactivée. Avant d'émettre un paramètre MultiplexEntrySend, l'émetteur devra cesser d'utiliser les entrées qui sont décrites. Il ne devra pas recommencer à utiliser ces entrées avant d'avoir reçu un acquittement MultiplexEntrySendAck. Cette procédure est utilisée

parce que si l'utilisation des entrées du tableau de multiplexage ne cesse pas avant l'envoi du paramètre MultiplexEntrySend, les erreurs peuvent susciter une ambiguïté dans le récepteur.

L'émetteur devra cesser d'utiliser des entrées désactivées avant d'envoyer le paramètre MultiplexEntrySend indiquant que ces entrées ont été désactivées. Des entrées désactivées peuvent être de nouveau utilisées à tout moment en envoyant le message MultiplexEntrySend en vue d'activer cette entrée. La désactivation des entrées dont l'émetteur n'a plus besoin peut augmenter la probabilité de détecter les erreurs dans le champ des codes de multiplexage H.223.

NOTE – Alors que certaines entrées du tableau de multiplexage sont mises à jour, il est possible de continuer à utiliser d'autres entrées (actives). Une entrée de tableau de multiplexage peut également être effacée dans le même paramètre MultiplexEntrySend qui est utilisé pour modifier d'autres entrées du tableau de multiplexage.

Au début de la communication, sauf spécification contraire dans une Recommandation appropriée, seule l'entrée de tableau 0 est disponible pour la transmission, les entrées 1 à 15 étant désactivées.

Une procédure de demande d'entrée de multiplexage peut être utilisée à tout moment pour élucider la question de la retransmission des entrées spécifiées dans les tableaux de multiplexage à partir du terminal distant, par exemple par suite d'une interruption ou de toute autre motif d'incertitude.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

8.7.1.1 Aperçu général du protocole – Entité MTSE sortante

Une procédure de demande d'entrée de tableau de multiplexage est déclenchée quand la primitive de demande TRANSFER est présentée par l'utilisateur de l'entité MTSE sortante. Un message MultiplexEntrySend est envoyé vers l'entité MTSE entrante homologue, et le temporisateur T104 démarre; Si un message MultiplexEntrySendAck est reçu en réponse au message MultiplexEntrySend, alors le temporisateur T104 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER que la demande d'envoi de l'entrée du tableau de multiplexage a réussi. Si, cependant, un message MultiplexEntrySendReject est reçu en réponse au message MultiplexEntrySend, le temporisateur est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité MTSE homologue a refusé d'accepter l'entrée du tableau de multiplexage.

Si le temporisateur T104 arrive en fin de temporisation, l'utilisateur de l'entité MTSE est informé par la primitive d'indication REJECT et un message MultiplexEntrySendRelease est envoyé.

Seuls les messages MultiplexEntrySendAck et MultiplexEntrySendReject répondant au message MultiplexEntrySend le plus récent sont acceptés. Les réponses à des messages précédents MultiplexEntrySend sont omises.

Une nouvelle procédure de demande d'envoi de l'entrée du tableau de multiplexage doit être déclenchée par la primitive de demande TRANSFER par l'utilisateur de l'entité MTSE sortante avant qu'un message MultiplexEntrySendAck ou MultiplexEntrySendReject ait été reçu.

8.7.1.2 Aperçu général du protocole – Entité MTSE entrante

Quand un message MultiplexEntrySend est reçu dans l'entité MTSE entrante, l'utilisateur est informé de la demande d'envoi de l'entrée du tableau de multiplexage en présentant la primitive d'indication TRANSFER. L'utilisateur de l'entité MTSE entrante signale le refus de l'entrée du tableau de multiplexage en présentant la primitive de réponse TRANSFER et un message MultiplexEntrySendAck est envoyé vers l'entité MTSE sortante homologue. L'utilisateur de l'entité MTSE entrante signale le refus de l'entrée du tableau de multiplexage en présentant la primitive de

demande REJECT, et un message MultiplexEntrySendReject est envoyé à l'entité MTSE sortante homologue.

Un nouveau message MultiplexEntrySend peut être reçu avant que l'utilisateur de l'entité MTSE entrante ait répondu à un message MultiplexEntrySend précédent. L'utilisateur de l'entité MTSE entrante est informé par la primitive d'indication REJECT, suivie de la primitive d'indication TRANSFER, et l'utilisateur de l'entité MTSE entrante répond à la nouvelle entrée du tableau de multiplexage.

Si un message MultiplexEntrySendRelease est reçu avant que l'utilisateur de l'entité MTSE entrante ait répondu à un message MultiplexEntrySend précédent, l'utilisateur de l'entité MTSE entrante est informé par la primitive d'indication REJECT, et l'entrée précédente du tableau de multiplexage est ignorée.

8.7.2 Communication entre l'entité MTSE et l'utilisateur de l'entité MTSE

8.7.2.1 Primitives entre l'entité MTSE et l'utilisateur de l'entité MTSE

La communication entre l'entité MTSE et l'utilisateur de l'entité MTSE s'effectue en utilisant les primitives indiquées dans le Tableau 41.

Tableau 41/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	MUX-DESCRIPTOR	MUX-DESCRIPTOR	– (Note 1)	–
REJECT	CAUSE	SOURCE CAUSE	non défini (Note 2)	non défini
NOTE 1 – "–" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

8.7.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- les primitives TRANSFER sont utilisées pour le transfert des entrées du tableau de multiplexage;
- les primitives REJECT sont utilisées pour refuser une entrée du tableau de multiplexage et pour mettre fin à un transfert d'entrée de tableau de multiplexage.

8.7.2.3 Définition des paramètres

La définition des paramètres des primitives indiquées dans le Tableau 41 est la suivante:

- le paramètre MUX-DESCRIPTOR est une entrée du tableau de multiplexage. Ce paramètre est mappé au champ MultiplexEntryDescriptor du message MultiplexEntrySend et transmis de façon transparente de l'utilisateur de l'entité MTSE dans l'entité MTSE sortante vers l'utilisateur de l'entité MTSE dans l'entité MTSE entrante. Il peut y avoir plusieurs descripteurs MUX-DESCRIPTORS associés à la primitive TRANSFER;
- le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE prend la valeur "USER" ou "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;

- c) le paramètre CAUSE indique le motif du refus d'ouverture d'une entrée de tableau de multiplexage. Le paramètre CAUSE est absent quand le paramètre SOURCE indique "PROTOCOL".

8.7.2.4 Etats de l'entité MTSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les états sont spécifiés de façon distincte pour chacune des entités MTSE entrante et sortante. Les états correspondant à une entité MTSE sortante sont:

Etat 0: IDLE

Il n'y a pas de transfert de l'entité MTSE en cours. L'entrée du tableau de multiplexage peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'utilisateur de l'entité MTSE a demandé le transfert d'une entrée du tableau de multiplexage, et une réponse provenant de l'entité MTSE homologue est attendue. L'entrée du tableau de multiplexage ne devra pas être utilisée par l'émetteur.

Les états correspondant à une entité MTSE entrante sont:

Etat 0: IDLE

Il n'y a pas de transfert de l'entité MTSE en cours. L'entrée du tableau de multiplexage peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'entité MTSE homologue a transféré une entrée du tableau de multiplexage, et une réponse provenant de l'utilisateur de l'entité MTSE est attendue. L'entrée du tableau de multiplexage peut ne pas être utilisée par l'émetteur.

8.7.2.5 Diagramme de changement d'états

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités MTSE sortante et entrante, comme cela est indiqué dans les Figures 25 et 26 respectivement.

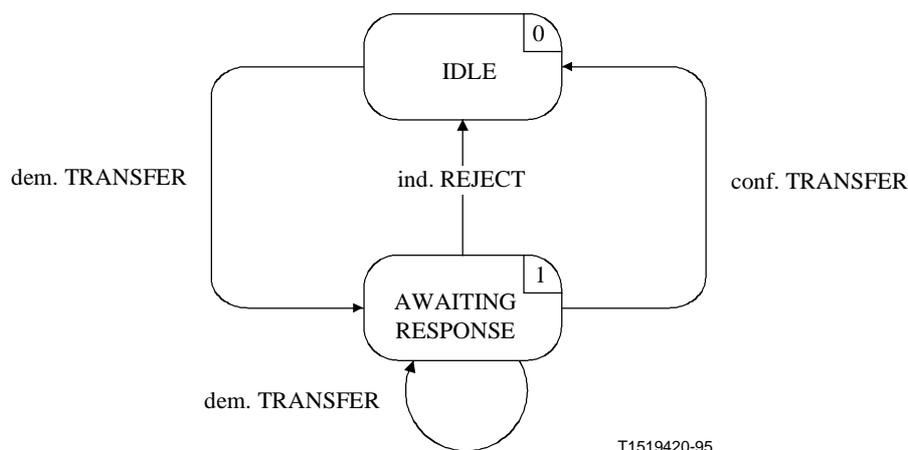


Figure 25/H.245 – Diagramme de changement d'états pour la séquence de primitives dans l'entité MTSE sortante

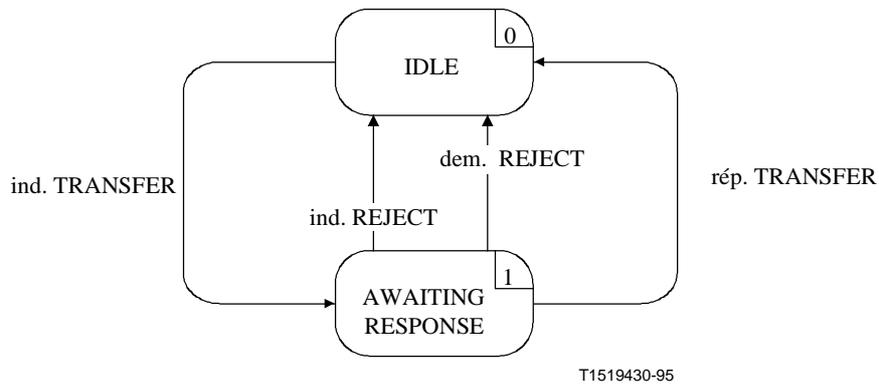


Figure 26/H.245 – Diagramme de changement d'états pour une séquence de primitives de l'entité MTSE entrante

8.7.3 Communication entre les entités MTSE homologues

8.7.3.1 Messages

Le Tableau 42 indique les messages et les champs de l'entité MTSE tels qu'ils sont définis dans le paragraphe 6 et se rapportent au protocole MTSE.

Tableau 42/H.245 – Noms et champs de message de l'entité MTSE

Fonction	Message	Sens	Champ
transfer	MultiplexEntrySend	S → E	sequenceNumber multiplexEntryDescriptors multiplexTableEntryNumber multiplexEntryDescriptors elementList
	MultiplexEntrySendAck	S ← E	sequenceNumber multiplexTableEntryNumber
reject	MultiplexEntrySendReject	S ← E	sequenceNumber multiplexTableEntryNumber rejectionDescriptions.cause
reset	MultiplexEntrySendRelease	S → E	multiplexTableEntryNumber
S sortant E entrant			

8.7.3.2 Variables d'état de l'entité MTSE

Les variables d'état suivantes sont définies dans l'entité MTSE sortante:

out_ENUM

Cette variable d'état établit une distinction entre les entités MTSE sortantes; Elle est initialisée pendant l'initialisation des entités MTSE sortantes. La valeur de out_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages MTSE envoyés à partir d'une entité MTSE sortante. En ce qui concerne les messages MTSE reçus dans une entité MTSE sortante, la valeur de champ multiplexTableEntryNumber des messages est identique à la valeur de out_ENUM.

out_SQ

Cette variable d'état est utilisée pour indiquer le message MultiplexEntrySend qui a été envoyé le plus récemment. Elle est augmentée de un et mappé au champ sequenceNumber du message MultiplexEntrySend avant la transmission d'un message MultiplexEntrySend. L'opération arithmétique effectuée sur la variable out_SQ est modulo 256.

Les variables d'état suivantes sont définies dans l'entité MTSE entrante:

in_ENUM

Cette variable d'état est utilisée pour établir une distinction entre les entités MTSE entrantes. Elle est initialisée lors de l'initialisation des entités MTSE entrantes. La valeur de in_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages MTSE envoyés à partir d'une entité MTSE entrante. En ce qui concerne les messages MTSE reçus dans une entité MTSE entrante, la valeur de champ multiplexTableEntryNumber du message est identique à la valeur de in_ENUM.

In_SQ

Cette variable d'état est utilisée pour enregistrer la valeur du champ sequenceNumber du message MultiplexEntrySend qui a été reçu en dernier. Les messages MultiplexEntrySendAck et MultiplexEntrySendReject ont leurs champs sequenceNumbers définis à la valeur de in_SQ, avant d'être envoyés à l'entité MTSE homologue.

8.7.3.3 Temporiseurs MTSE

Le temporisateur suivant est spécifié pour l'entité MTSE sortante:

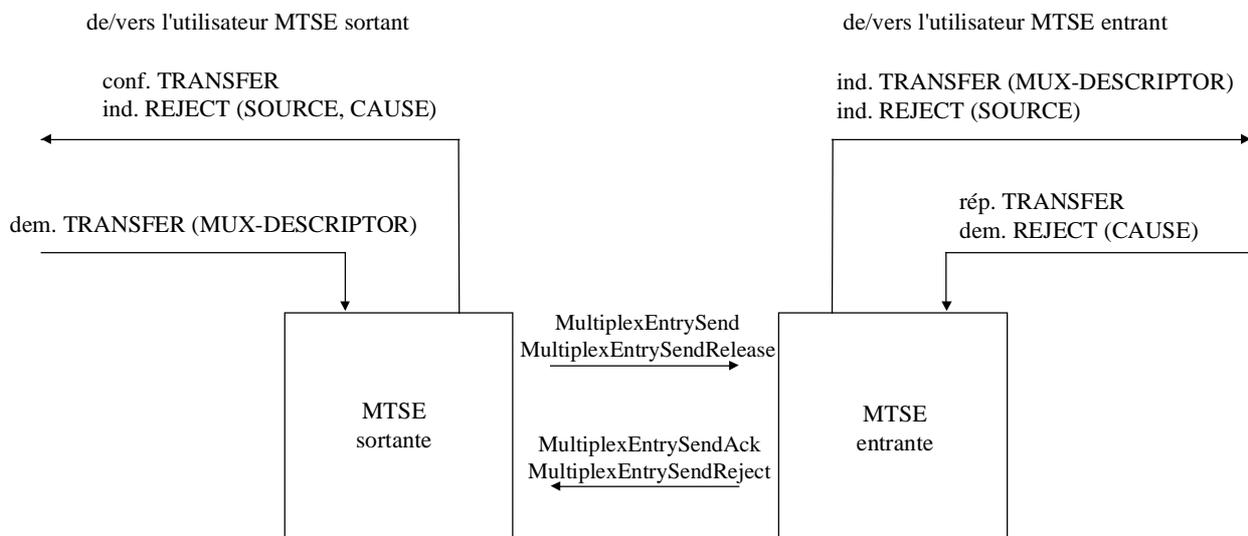
T104

Ce temporisateur est utilisé à l'état AWAITING RESPONSE. Il spécifie le temps maximal durant lequel aucun message MultiplexEntrySendAck ou MultiplexEntrySendReject ne peut être reçu.

8.7.4 Procédures MTSE

8.7.4.1 Introduction

La Figure 27 récapitule les primitives et leurs paramètres, de même que les messages et les champs pertinents pour chacune des entités MTSE entrante et sortante.



T1519440-95

Figure 27/H.245 – Primitives et messages dans l'entité de signalisation de tableau de multiplexage (MTSE)

8.7.4.2 Valeurs par défaut des paramètres de primitives

Quand cela n'a pas été explicitement décrit dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 43.

Tableau 43/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut
ind. TRANSFER	MUX-DESCRIPTOR	MultiplexEntrySend.multiplexEntryDescriptors.elementList
ind. REJECT	SOURCE CAUSE	USER null

8.7.4.3 Valeurs par défaut des champs de message

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs de messages prennent des valeurs identiques à celles indiquées dans le Tableau 44.

Tableau 44/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note)
MultiplexEntrySend	sequenceNumber multiplexEntryDescriptors. multiplexTableEntryNumber multiplexEntryDescriptors.elementList	out_SQ out_ENUM dem. TRANSFER (MUX-DESCRIPTOR)
MultiplexEntrySendAck	sequenceNumber multiplexTableEntryNumber	in_SQ in_ENUM

Tableau 44/H.245 – Valeurs par défaut des champs de message (*fin*)

Message	Champ	Valeur par défaut (Note)
MultiplexEntrySendReject	sequenceNumber	in_SQ
	rejectionDescriptions.multiplexTableEntryNumber	in_ENUM
	rejectionDescriptions.cause	dem.REJECT (CAUSE)
MultiplexEntrySendRelease	multiplexTableEntryNumber	out_ENUM
NOTE – Un champ de message ne doit pas être codé si le paramètre de primitive correspondant est nul, c'est-à-dire n'est pas présent.		

8.7.4.4 Description SDL

Les procédures de l'entité MTSE sortante et de l'entité MTSE entrante sont décrites en langage SDL dans les Figures 28 et 29 respectivement.

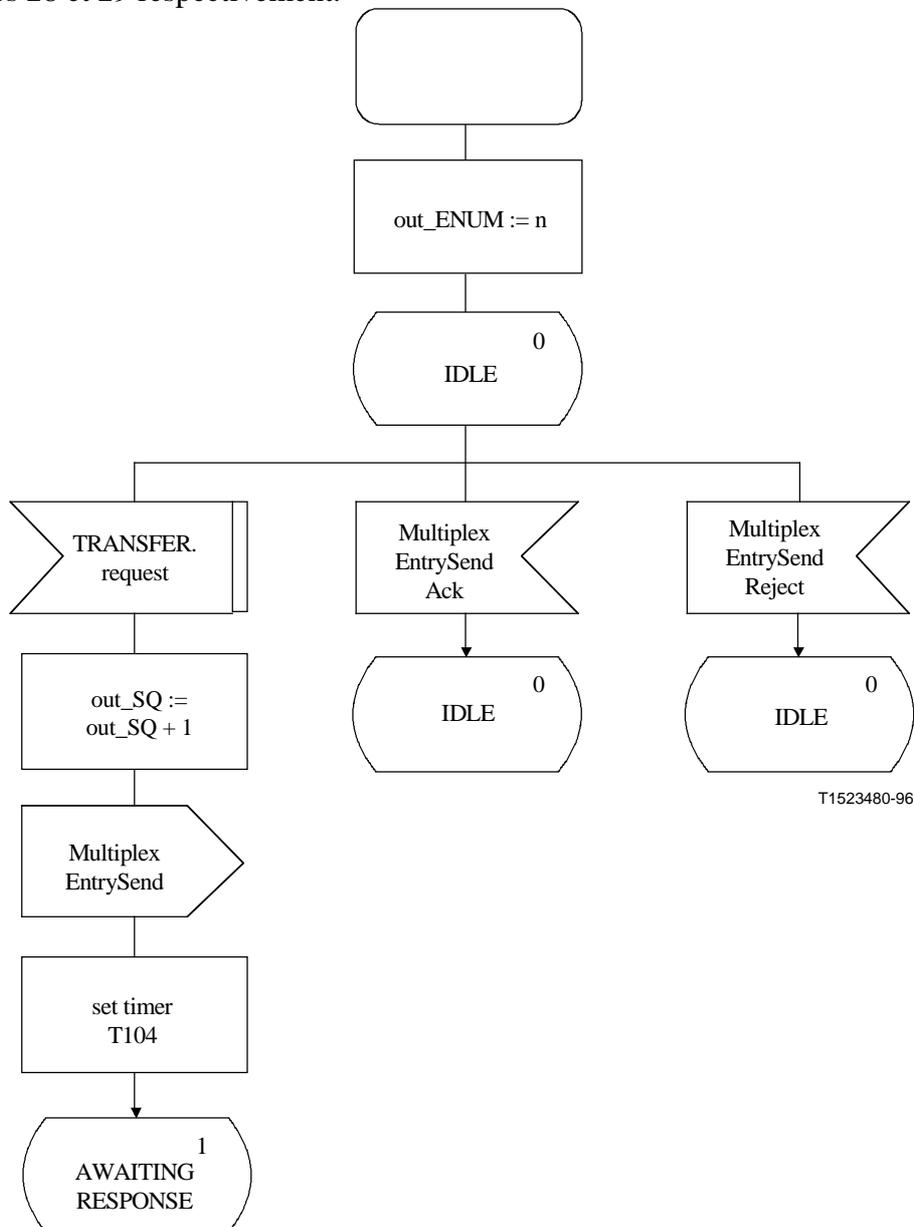


Figure 28 i)/H.245 – Description SDL dans l'entité MTSE sortante

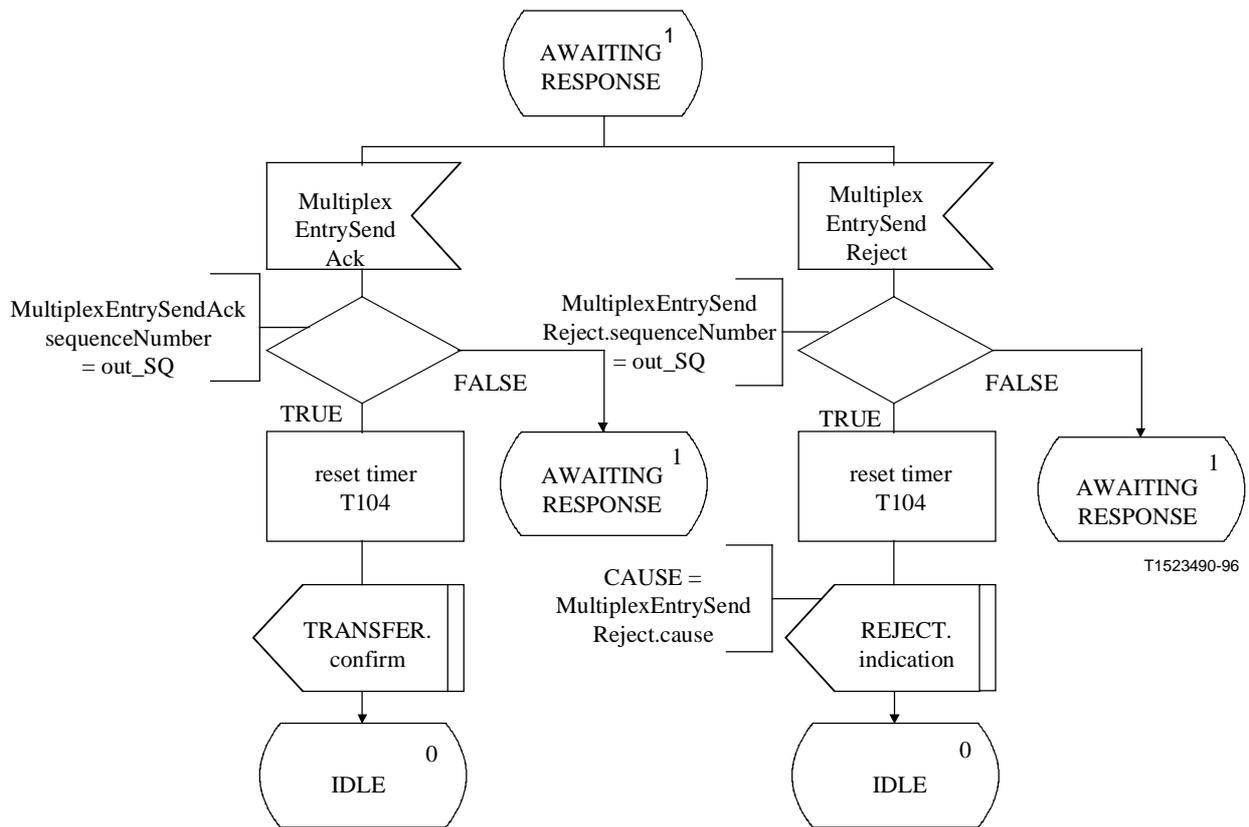
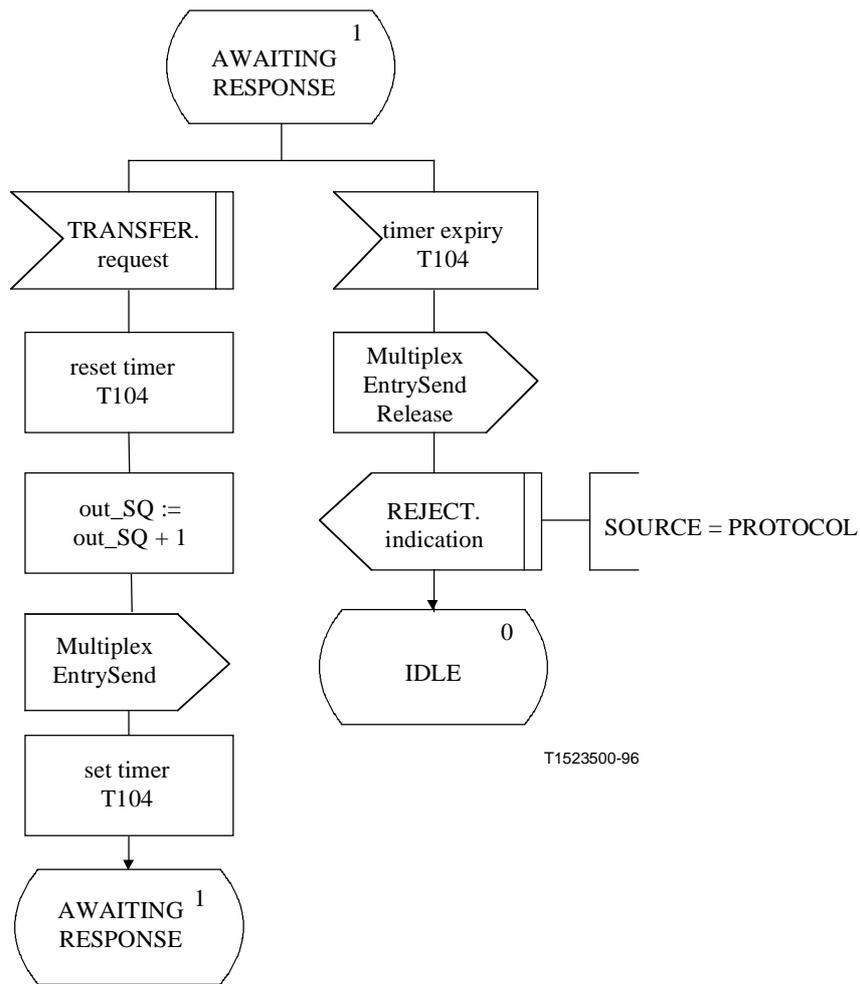


Figure 28 ii)/H.245 – Description SDL dans l'entité MTSE sortante



T1523500-96

Figure 28 iii)/H.245 – Description SDL dans l'entité MTSE sortante

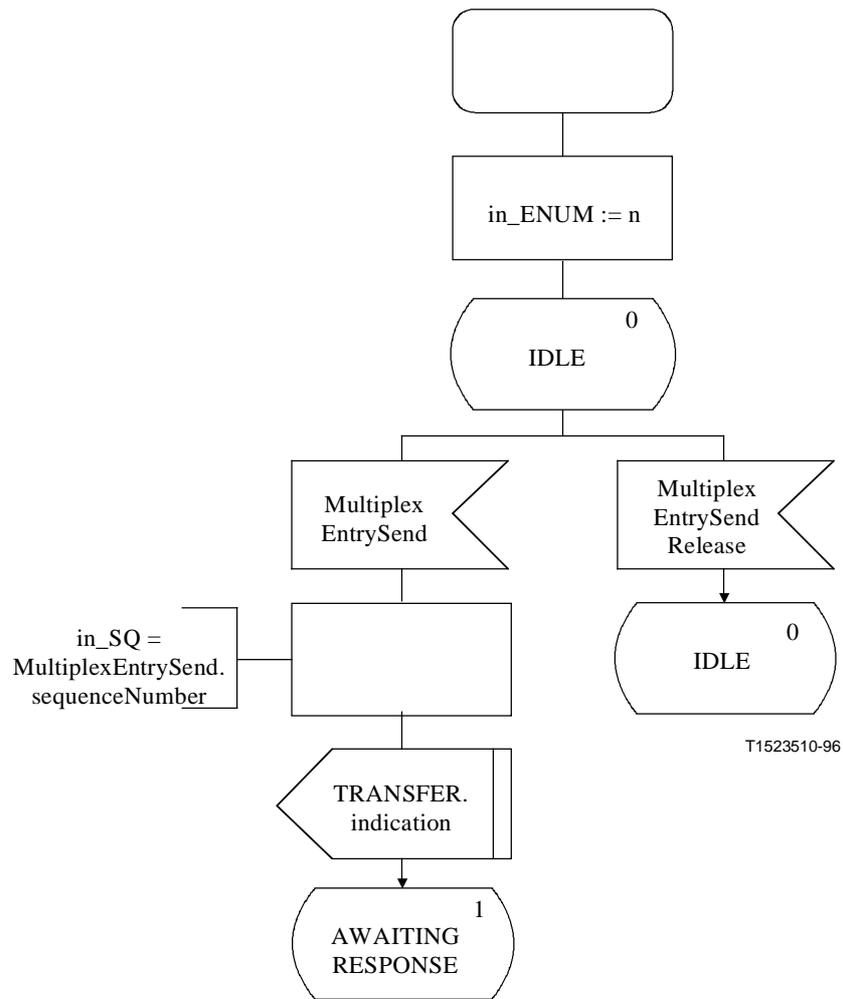


Figure 29 i)/H.245 – Description SDL dans l'entité MTSE entrante

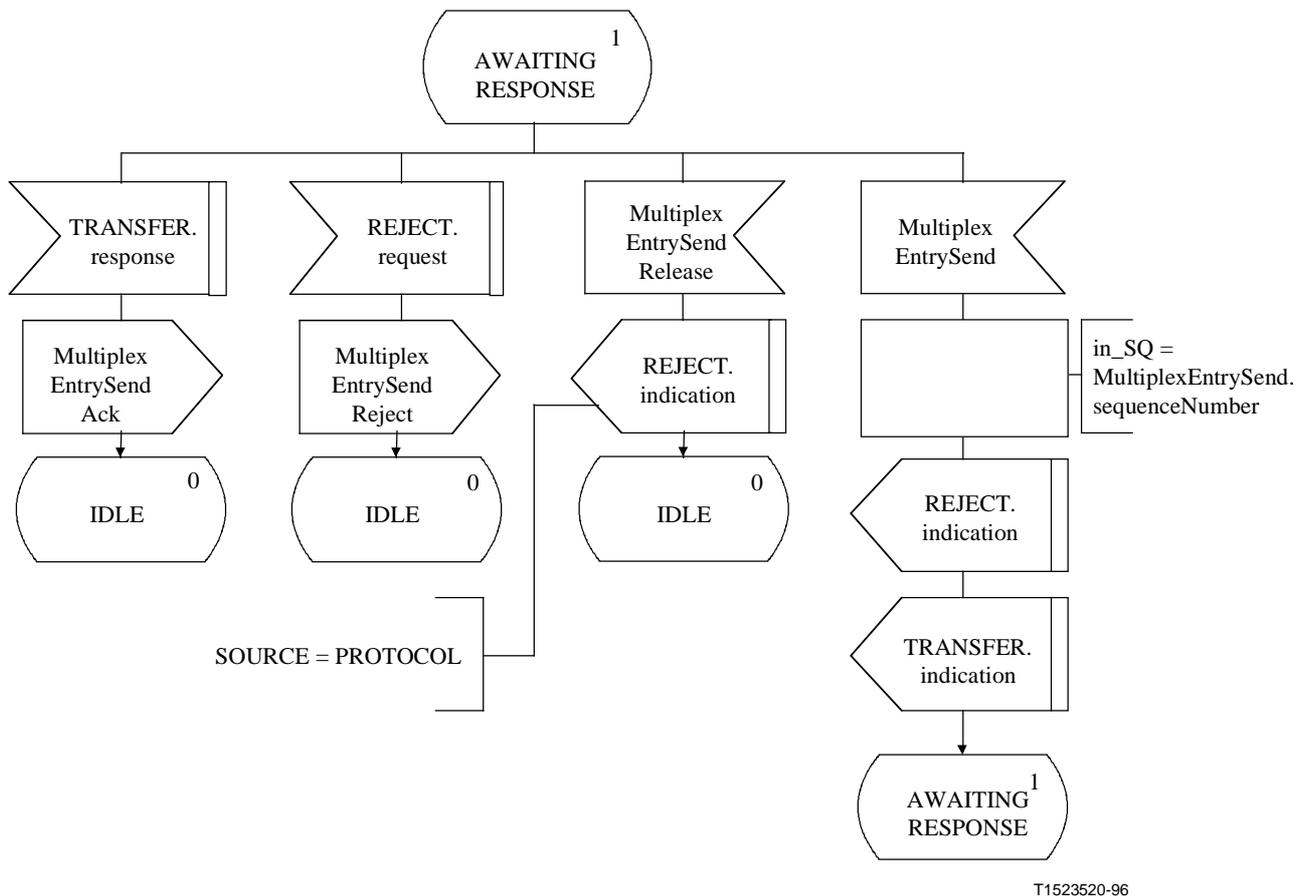


Figure 29 ii)/H.245 – Description SDL dans l'entité MTSE entrante

8.8 Procédures relatives à la demande d'entrée de multiplexage

8.8.1 Introduction

Ces procédures sont utilisées par un terminal pour demander la retransmission d'un ou plusieurs descripteurs MultiplexEntryDescriptors. Ces procédures sont désignées comme demande d'entité de signalisation de l'entrée de multiplexage (RMESE, *request multiplex entry signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité RMESE et l'utilisateur de l'entité RMESE. Les informations de protocole sont transmises à l'entité RMESE homologue au moyen des messages pertinents définis au paragraphe 6. Il y a une entité RMESE sortante et une entité RMESE entrante. Il existe une instance d'entité RMESE pour chaque entrée de tableau de multiplexage.

Un terminal qui apporte une telle réponse positive, c'est-à-dire en présentant une primitive de réponse SEND, doit déclencher les procédures du Tableau de multiplexage pour envoyer l'entrée de tableau de multiplexage dès que possible.

Le texte suivant présente un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

NOTE – Le présent protocole a été défini de sorte qu'il y a une entité RMESE indépendante pour chaque entrée du tableau de multiplexage, et la syntaxe a été définie pour permettre à un message unique de transmettre les informations relatives à une ou plusieurs entrées du tableau de multiplexage. La façon dont les messages sont élaborés est une décision relative aux réalisations: un terminal peut par exemple répondre à un message RequestMultiplexEntry demandant l'envoi de trois entrées avec un, deux ou trois messages de réponse.

8.8.1.1 Aperçu général du protocole – Entité RMESE sortante

Une procédure de demande d'entrée de multiplex est lancée quand la primitive de demande SEND est présentée par l'utilisateur dans l'entité RMESE sortante. Un message RequestMultiplexEntry est envoyé vers l'entité RMESE entrante homologue, et le temporisateur T107 démarre. Si un message RequestMultiplexEntryAck est reçu en réponse au message RequestMultiplexEntry, alors le temporisateur T107 est arrêté, et l'utilisateur est informé par la primitive de confirmation SEND que la procédure de demande d'entrée de multiplexage a réussi. Si, cependant, un message RequestMultiplexEntryReject est reçu en réponse au message RequestMultiplexEntry, alors le temporisateur T 107 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité RMESE homologue a refusé d'envoyer l'entrée de multiplexage.

Si le temporisateur T107 arrive à expiration, alors l'utilisateur de l'entité RMESE est informé par la primitive d'indication REJECT et un message RequestMultiplexEntryRelease est envoyé.

8.8.1.2 Aperçu général du protocole – Entité RMESE entrante

Quand un message RequestMultiplexEntry est reçu dans l'entité RMESE entrante, l'utilisateur est informé de la demande d'entrée de multiplexage par la primitive d'indication SEND. L'utilisateur de l'entité RMESE signale l'acceptation de la demande d'entrée de multiplexage en présentant la primitive de réponse SEND, et un message RequestMultiplexEntryAck est envoyé vers l'entité RMESE sortante homologue. L'utilisateur de l'entité RMESE entrante signale le refus de la demande d'entrée de multiplexage en présentant la primitive de demande REJECT, et un message RequestMultiplexEntryReject est envoyé vers l'entité RMESE sortante homologue.

8.8.2 Communication entre l'entité RMESE et l'utilisateur de l'entité RMESE

8.8.2.1 Primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE

La communication entre l'entité RMESE, et l'utilisateur de l'entité RMESE, est effectuée en utilisant les primitives indiquées dans le Tableau 45.

Tableau 45/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
SEND	– (Note 1)	–	–	–
REJECT	CAUSE	SOURCE CAUSE	non défini (Note 2)	non défini

NOTE 1 – "–" signifie aucun paramètre.
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.

8.8.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives SEND sont utilisées pour demander la transmission d'une entrée de multiplexage;
- b) les primitives REJECT sont utilisées pour demander la transmission d'une entrée de multiplexage.

8.8.2.3 Définition des paramètres

La définition des paramètres de primitives indiquées au Tableau 45 est la suivante:

- a) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur de "USER" ou de "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;
- b) le paramètre CAUSE indique le motif de refus d'envoi d'une entrée de tableau de multiplexage. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

8.8.2.4 Etats de l'entité RMESE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE.

Les états correspondant à une entité RMESE sortante sont:

Etat 0: IDLE

L'entité RMESE est dans l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité RMESE attend une réponse de l'entité RMESE distante.

Les états correspondant à une entité RMESE entrante sont:

Etat 0: IDLE

L'entité RMESE est dans l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité RMESE attend une réponse de l'utilisateur de l'entité RMESE.

8.8.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE est définie ici. Les séquences autorisées sont spécifiées séparément pour chacune des entité RMESE entrante et sortante, comme l'indiquent respectivement les Figures 30 et 31.

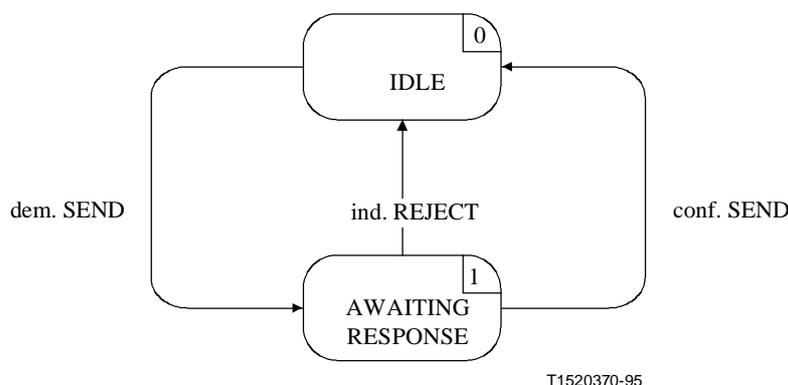


Figure 30/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité RMESE sortante

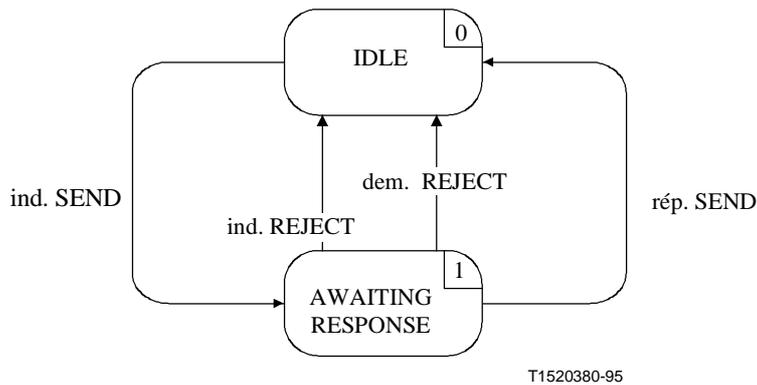


Figure 31/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité RMESE entrante

8.8.3 Communication entre les entités RMESE homologues

8.8.3.1 Messages

Le Tableau 46 indique les messages et les champs de l'entité RMESE, tels qu'ils sont définis dans le paragraphe 6 et se rapportent au protocole de l'entité RMESE.

Tableau 46/H.245 – Noms et champs de message de l'entité RMESE

Fonction	Message	Sens	Champ
transfer	RequestMultiplexEntry	S → E	multiplexTableEntryNumber
	RequestMultiplexEntryAck	S ← E	multiplexTableEntryNumber
	MultiplexEntryEntryReject	S ← E	multiplexTableEntryNumber rejectionDescriptions.cause
reset	RequestMultiplexEntryRelease	S → E	
S sortant			
E entrant			

8.8.3.2 Variables d'état de l'entité RMESE

Les variables d'état suivantes sont définies dans l'entité RMESE sortante:

out_ENUM

Cette variable d'état permet d'établir une distinction entre les entités RMESE sortantes. Elle est initialisée lors de l'initialisation de l'entité RMESE sortante. La valeur de la variable out_ENUM est utilisée pour remplir le champ multiplexTableEntryNumber de messages d'entités RMESE envoyés à partir d'une entité RMESE sortante. En ce qui concerne les messages RMESE reçus par une entité RMESE sortante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de out_ENUM.

La variable d'état suivante est définie dans l'entité RMESE entrante:

in_ENUM

Cette variable d'état permet d'établir une distinction entre les entités RMESE entrantes. Elle est initialisée lors de l'initialisation de l'entité RMESE entrante. La valeur de la variable in_ENUM est utilisée pour remplir le champ multiplexTableEntryNumber de messages d'entités RMESE envoyés à

partir d'une entité RMESE entrante. En ce qui concerne les messages RMESE reçus par une entité RMESE entrante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de in_ENUM.

8.8.3.3 Temporisateurs RMESE

Le temporisateur suivant est spécifié pour l'entité RMESE sortante:

T107

Ce temporisateur est utilisé pendant l'état AWAITING RESPONSE (attente de la réponse). Il spécifie le temps maximal pendant lequel aucun message RequestMultiplexEntryAck ou RequestMultiplexEntryReject ne peut être reçu.

8.8.4 Procédures RMESE

La Figure 32 récapitule les primitives des entités RMESE et leurs paramètres, de même que les messages, pour chacune des entités RMESE entrante et sortante.

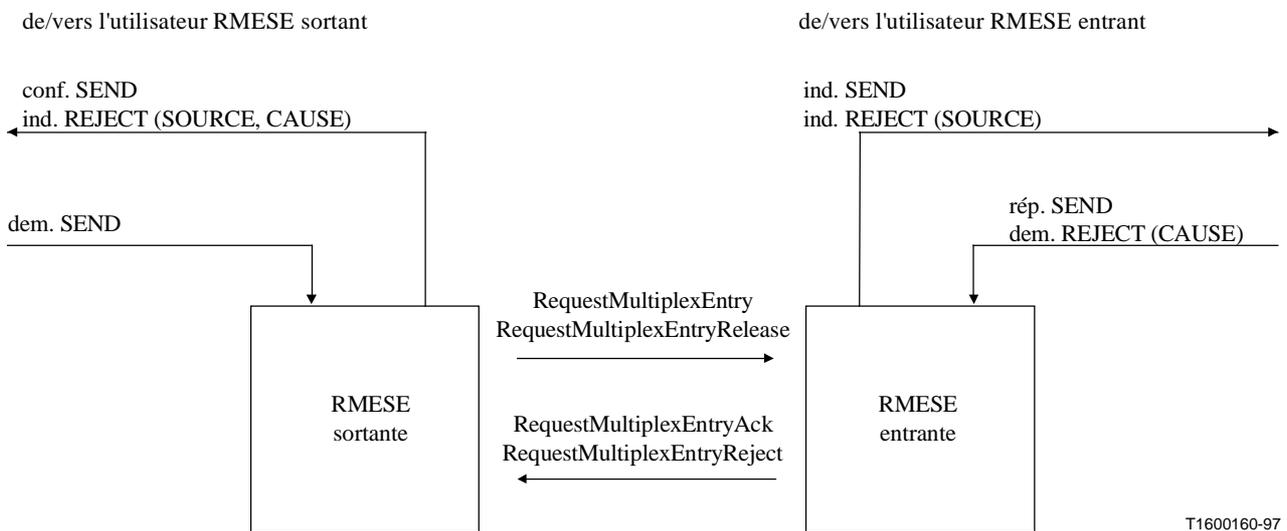


Figure 32/H.245 – Primitives et messages de l'entité de signalisation de demande d'entrée de multiplexage (RMESE)

8.8.4.1 Valeurs par défaut des paramètres de primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs qui sont indiquées dans le Tableau 47, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau 47/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
ind. REJECT	SOURCE CAUSE	USER null

8.8.4.2 Valeurs par défaut des champs de messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs de messages prennent des valeurs identiques à celles indiquées dans le Tableau 48.

Tableau 48/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut
RequestMultiplexEntry	multiplexTableEntryNumber	out_ENUM
RequestMultiplexEntryAck	multiplexTableEntryNumber	in_ENUM
RequestMultiplexEntryReject	multiplexTableEntryNumber cause	in_ENUM dem. REJECT (CAUSE)
RequestMultiplexEntryRelease	multiplexTableEntryNumber	out_ENUM

8.8.4.3 Descriptions SDL

Les procédures des entités RMESE entrantes et des entités RMESE sortantes sont exprimées en langage SDL dans les Figures 33 et 34 respectivement.

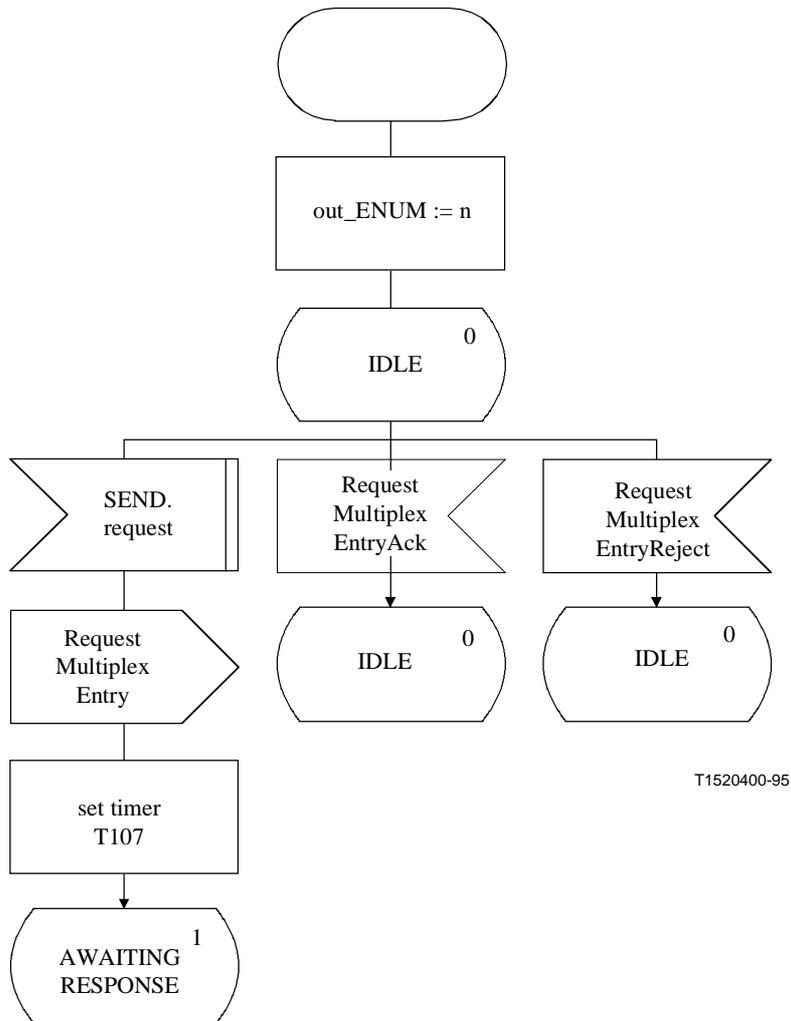


Figure 33 i)/H.245 – Description SDL des entités RMESE sortantes

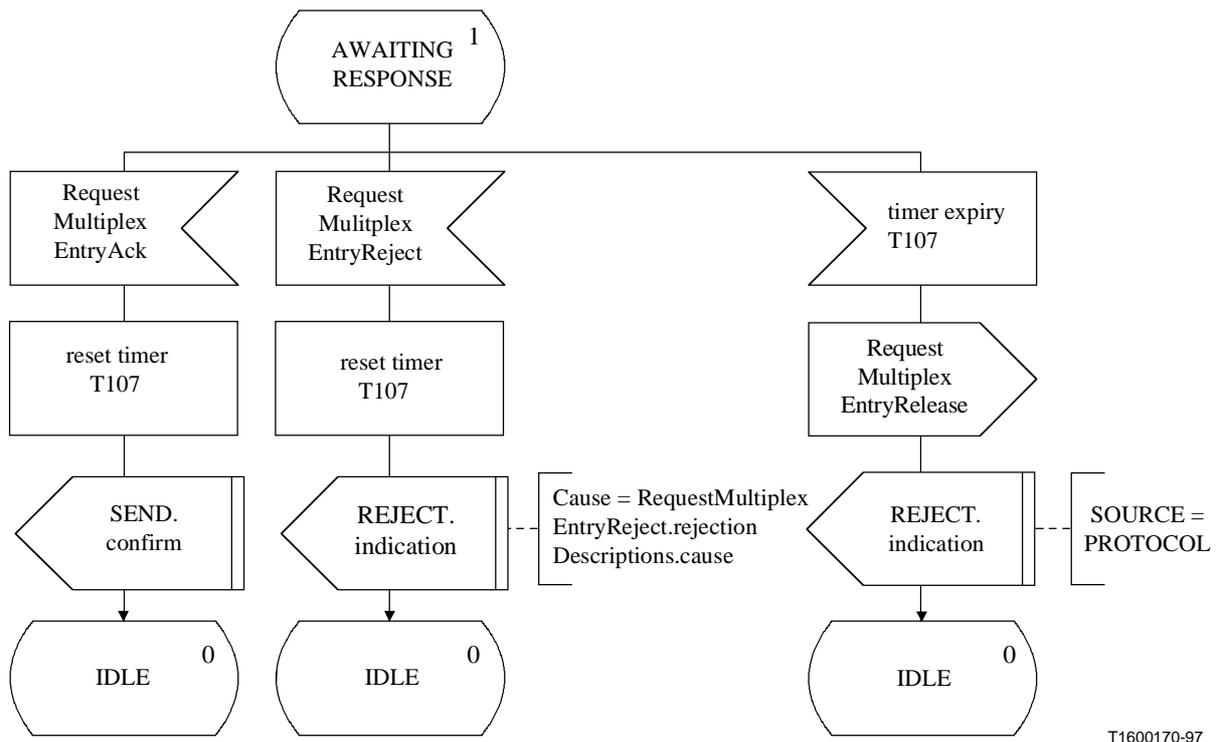


Figure 33 ii)/H.245 – Description SDL des entités RMESE sortantes

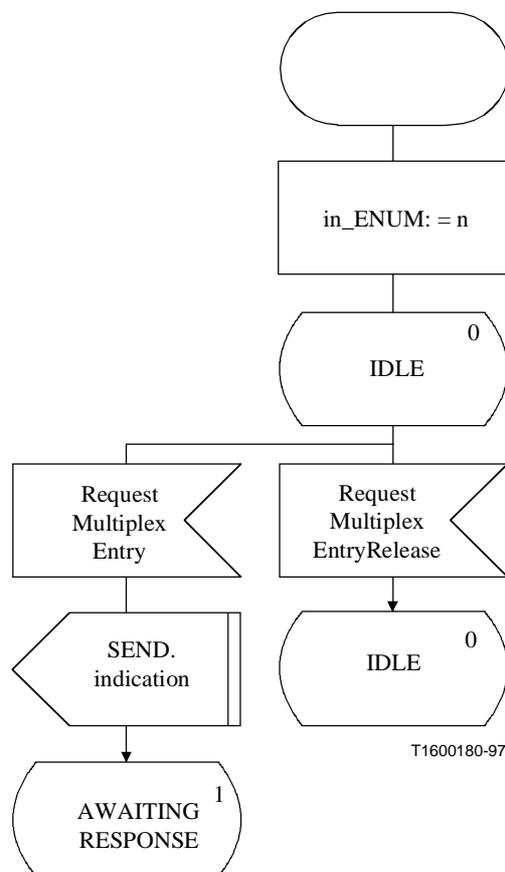


Figure 34 i)/H.245 – Description SDL des entités RMESE entrantes

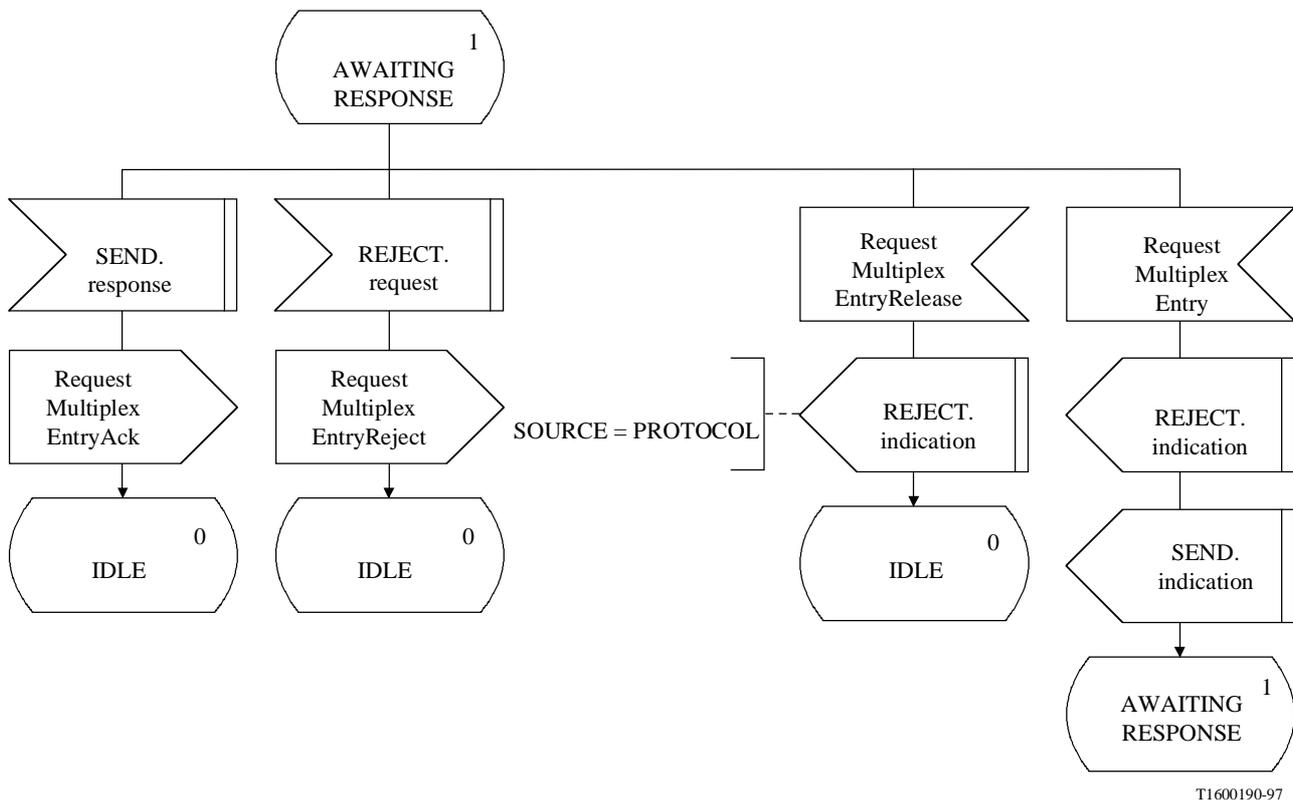


Figure 34 ii)/H.245 – Description SDL des entités RMESE entrantes

8.9 Procédures de demande de mode

8.9.1 Introduction

Les procédures décrites ici permettent à un terminal de demander à un terminal distant d'utiliser un mode particulier de fonctionnement dans son sens d'émission. On fait référence ici à ces procédures sous le nom de l'entité de signalisation de demande de mode (MRSE, *mode request signalling entity*). Elles sont spécifiées en termes de primitives et d'états à l'interface entre l'entité MRSE et l'utilisateur de l'entité MRSE. Les informations de protocole sont transmises vers l'entité MRSE homologue par l'intermédiaire de messages pertinents définis dans le paragraphe 6. Il y a une entité MRSE sortante et une entité MRSE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité MRSE par appel.

Un terminal qui apporte une telle réponse positive, c'est-à-dire en présentant la primitive de réponse TRANSFER, devra déclencher les procédures de signalisation de voie logique pour définir le mode approprié de transmission dès que possible.

Si les capacités actuellement disponibles reçues du terminal distant contiennent une ou plusieurs capacités d'émission, un terminal peut choisir un mode qu'il préfère lui avoir transmis en utilisant les procédures de demande de mode. Un terminal dont les capacités actuellement disponibles contiennent une ou plusieurs capacités d'émission et qui reçoit une telle demande devrait satisfaire la demande.

Une demande de mode ne doit pas être envoyée à un terminal dont les capacités actuellement disponibles ne contiennent pas de capacités d'émission, c'est-à-dire que le terminal ne souhaite pas être contrôlé à distance et ne doit pas non plus être contrôlé de cette façon. Si ce terminal reçoit cependant une demande de mode, il peut satisfaire cette demande.

Un terminal qui reçoit la commande `multipointModeCommand` doit satisfaire tous les demandes de mode, jusqu'à ce que cette commande soit annulée par la réception de la commande `cancelMultipointModeCommand`. Une demande de mode peut être envoyée vers un terminal dont les capacités actuellement disponibles ne contiennent pas de capacités de transmission quand une commande `multipointModeCommand` a été préalablement envoyée.

La demande de mode peut inclure des voies qui sont déjà ouvertes. Par exemple, dans le cas où une voie pour G.723.1 serait actuellement ouverte et un terminal souhaiterait recevoir une voie G.728 supplémentaire, il enverrait une demande de mode contenant à la fois les voies G.723.1 et G.728. L'absence de la demande de voie G.723.1 indiquerait que G.723.1 n'est plus souhaité.

NOTE – La description de la demande de mode spécifie un mode complet. Si, par exemple, la vidéo est en cours de transmission et si une demande de mode n'incluant pas de spécification pour la vidéo est demandé, alors ceci exige l'interruption de la transmission vidéo.

Si une source alimente plusieurs récepteurs, elle peut être incapable de répondre à des signaux reçus tels que des demandes d'émission dans un mode particulier.

Le texte suivant donne un aperçu général du fonctionnement du protocole MRSE. En cas de divergence avec la spécification formelle, la spécification formelle sera applicable.

8.9.1.1 Aperçu général du protocole – Entité MRSE sortante

Une procédure de demande de mode est déclenchée quand la primitive de demande `TRANSFER` est présentée par l'utilisateur dans l'entité MRSE sortante. Un message `RequestMode` est envoyé vers l'entité MRSE entrante homologue, et le temporisateur T109 démarre. Si un message `RequestModeAck` est reçu en réponse au message `RequestMode`, alors le temporisateur T109 est arrêté et l'utilisateur est informé par la primitive de confirmation `TRANSFER` que la demande de mode a réussi. Si, cependant, un message `RequestModeReject` est reçu en réponse au message `RequestMode`, le temporisateur T109 est arrêté et l'utilisateur est informé par la primitive d'indication `REJECT` que l'utilisateur de l'entité MRSE homologue a refusé d'accepter la demande de mode.

Si le temporisateur T109 arrive à expiration, l'entité MRSE sortante est informée par la primitive d'indication `REJECT` et un message `RequestModeRelease` est envoyé.

Seuls les messages `RequestModeAck` et `RequestModeReject` répondant au message `RequestMode` le plus récent sont acceptés. Les messages envoyés en réponse à des messages précédents `RequestMode` sont omis.

Une nouvelle procédure de demande de mode peut être déclenchée par la primitive de demande `TRANSFER` par l'utilisateur de l'entité MRSE sortante avant qu'un message `RequestModeAck` ou `RequestModeReject` ait été reçu.

8.9.1.2 Aperçu général du protocole – Entité MRSE entrante

Quand un message `RequestMode` est reçu dans l'entité MRSE entrante, l'utilisateur est informé de la demande de mode par la primitive d'indication `TRANSFER`. L'utilisateur de l'entité MRSE entrante accepte la demande de mode en présentant la primitive de réponse `TRANSFER`, et un message `RequestModeAck` est envoyé vers l'entité MRSE sortante homologue. L'utilisateur de l'entité MRSE entrante signale le refus de la demande de mode en présentant la primitive de demande `REJECT`, et un message `RequestModeReject` est envoyé à l'entité MRSE sortante homologue.

Un nouveau message `RequestMode` peut être reçu avant que l'utilisateur de l'entité MRSE entrante ait répondu à un message `RequestMode` antérieur. L'utilisateur de l'entité MRSE entrante est informé par la primitive d'indication `REJECT`, suivie de la primitive d'indication `TRANSFER`, et l'utilisateur de l'entité MRSE entrante répond à la nouvelle entrée du tableau de multiplexage.

Si un message RequestModeRelease est reçu avant que l'utilisateur de l'entité MRSE entrante ait répondu à un message RequestMode antérieur, l'utilisateur de l'entité MRSE entrante est informé par la primitive d'indication REJECT, et la demande de mode antérieure est omise.

8.9.2 Communication entre l'entité MRSE et l'utilisateur de l'entité MRSE

8.9.2.1 Primitives entre l'entité MRSE et l'utilisateur de l'entité MRSE

La communication entre l'entité MRSE et l'utilisateur de l'entité MRSE s'effectue en utilisant les primitives indiquées dans le Tableau 49.

Tableau 49/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	MODE-ELEMENT	MODE-ELEMENT	MODE-PREF	MODE-PREF
REJECT	CAUSE	SOURCE CAUSE	non défini (Note)	non défini
NOTE – "non défini" signifie que cette primitive n'est pas définie.				

8.9.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives TRANSFER sont utilisées pour le transfert de la demande de mode;
- b) les primitives REJECT sont utilisées pour refuser une demande de mode.

8.9.2.3 Définition des paramètres

La définition des paramètres de primitives indiqués dans le Tableau 49 est la suivante:

- a) le paramètre MODE-ELEMENT spécifie un élément de mode. Ce paramètre est mappé au champ requestedModes du message RequestMode et transmis de façon transparente de l'utilisateur de l'entité MRSE sortante vers l'utilisateur de l'entité MRSE entrante. Ce paramètre est obligatoire. Il peut y avoir plusieurs éléments MODE-ELEMENT associés aux primitives TRANSFER;
- b) le paramètre MODE-PREF informe l'utilisateur de l'utilisation ou de la non-utilisation de la demande de mode en premier choix. Il est appliqué au champ de réponse du message RequestModeAck et transmis de façon transparente de l'utilisateur de l'entité RMSE entrante vers l'utilisateur de l'entité RMSE sortante. Il a deux valeurs "MOST-PREFERRED" et "LESS-PREFERRED";
- c) le paramètre SOURCE indique la source de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur de "USER" ou de "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;
- d) le paramètre CAUSE indique le motif du refus de rejet d'une demande de mode. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

8.9.2.4 Etats de l'entité MRSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité MRSE et l'utilisateur de l'entité MRSE. Les états pour une entité MRSE sortante sont:

Etat 0: IDLE

L'entité MRSE est à l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité MRSE attend une réponse de l'entité MRSE distante.

Les états correspondant à une entité MRSE entrante sont:

Etat 0: IDLE

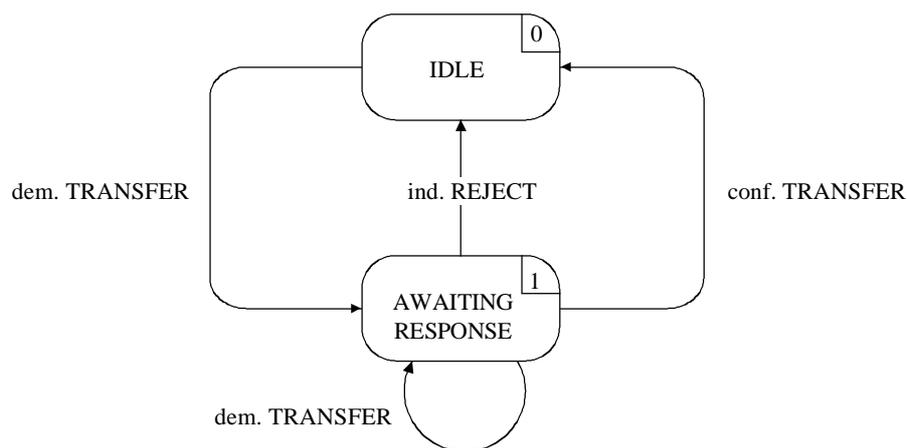
L'entité MRSE est à l'état "idle".

Etat 1: AWAITING RESPONSE

L'entité MRSE attend une réponse de l'utilisateur de l'entité MRSE.

8.9.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité MRSE et l'utilisateur de l'entité MRSE est définie ici. Les séquences autorisées sont spécifiées séparément pour chacune des entités MRSE sortante et entrante, comme l'indiquent respectivement les Figures 35 et 36.



T1519480-95

Figure 35/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité MRSE sortante

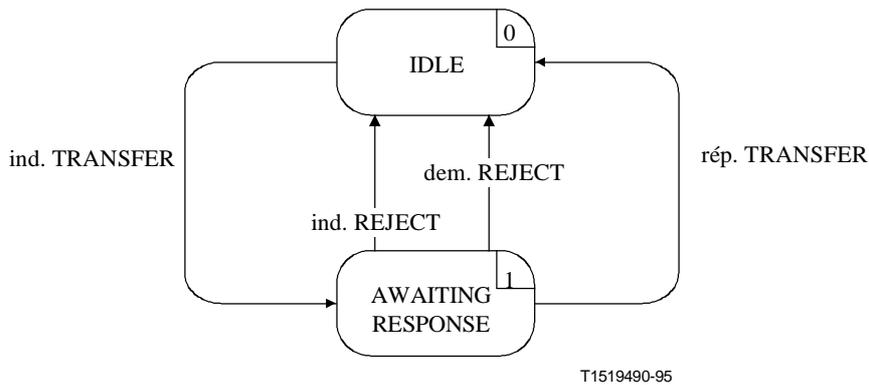


Figure 36/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité MRSE entrante

8.9.3 Communication entre les entités MRSE homologues

8.9.3.1 Messages

Le Tableau 50 indique les messages et les champs de l'entité MRSE tels qu'ils ont été définis dans le paragraphe 6 et se rapportent au protocole de l'entité MRSE.

Tableau 50/H.245 – Noms et champs des messages de l'entité MRSE

Fonction	Message	Sens	Champ
mode request	RequestMode	S → E	sequenceNumber requestedModes
	RequestModeAck	S ← E	sequenceNumber response
	RequestModeReject	S ← E	sequenceNumber cause
reset	RequestModeRelease	S → E	–

S sortant
E entrant

8.9.3.2 Variables d'état de l'entité MRSE

Les variables d'état suivantes sont définies dans l'entité MRSE sortante:

out_SQ

Cette variable d'état est utilisée pour indiquer le message RequestMode le plus récent. Elle est augmentée de un et mappée au champ sequenceNumber du message RequestMode avant la transmission du message RequestMode. L'opération arithmétique exécutée sur la variable out_SQ est modulo 256.

Les variables d'état suivantes sont définies dans l'entité MRSE entrante:

in_SQ

Cette variable d'état est utilisée pour enregistrer la valeur du champ sequenceNumber du message RequestMode reçu le plus récemment. Les messages RequestModeAck et RequestModeReject ont

leurs champs `sequenceNumber` mis à la valeur de la variable `in_SQ`, avant d'être envoyés vers l'entité MRSE homologue.

8.9.3.3 Temporisateurs MRSE

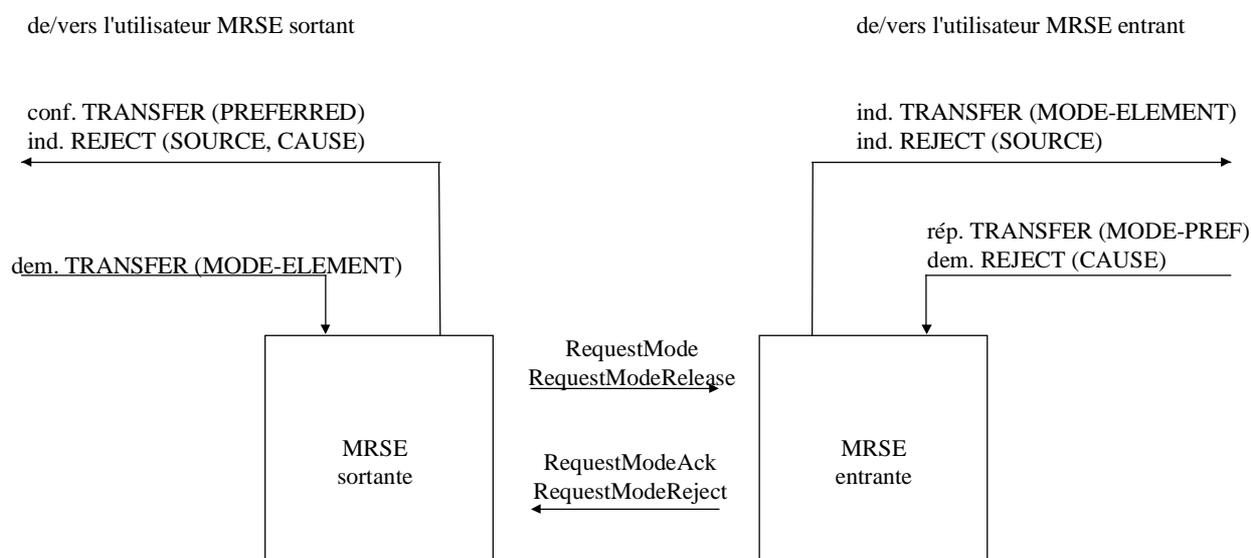
Le temporisateur suivant est spécifié pour l'entité MRSE sortante:

T109

Ce temporisateur est utilisé pendant l'état `AWAITING RESPONSE`. Il spécifie le temps maximal pendant lequel aucun message `RequestModeAck` ou `RequestModeReject` ne peut être reçu.

8.9.4 Procédures MRSE

La Figure 37 récapitule les primitives des entités MRSE et leurs paramètres, de même que les messages, pour chacune des entités MRSE entrante et sortante.



T1519500-95

Figure 37/H.245 – Primitives et messages dans l'entité de signalisation de la demande de mode (MRSE)

8.9.4.1 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 51, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau 51/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeurs par défaut
ind. TRANSFER	MODE-ELEMENT	RequestMode.requestedModes
conf. TRANSFER	MODE-PREF	RequestModeACK.response
ind. REJECT	SOURCE	USER
	CAUSE	null

8.9.4.2 Valeurs par défaut des champs de messages

Les champs de message prennent des valeurs identiques à celles indiquées dans le Tableau 52, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau 52/H.245 – Valeurs par défaut des champs de messages

Message	Champ	Valeurs par défaut
RequestMode	sequenceNumber requestedModes	out_SQ dem. TRANSFER(MODE-ELEMENT)
RequestModeAck	sequenceNumber response	in_SQ rép. TRANSFER(MODE-PREF)
RequestModeReject	sequenceNumber cause	in_SQ dem. REJECT(CAUSE)
RequestModeRelease	–	–

8.9.4.3 Description SDL

Les procédures de l'entité MRSE sortante et de l'entité MRSE entrante sont décrites en langage SDL dans les Figures 38 et 39 respectivement.

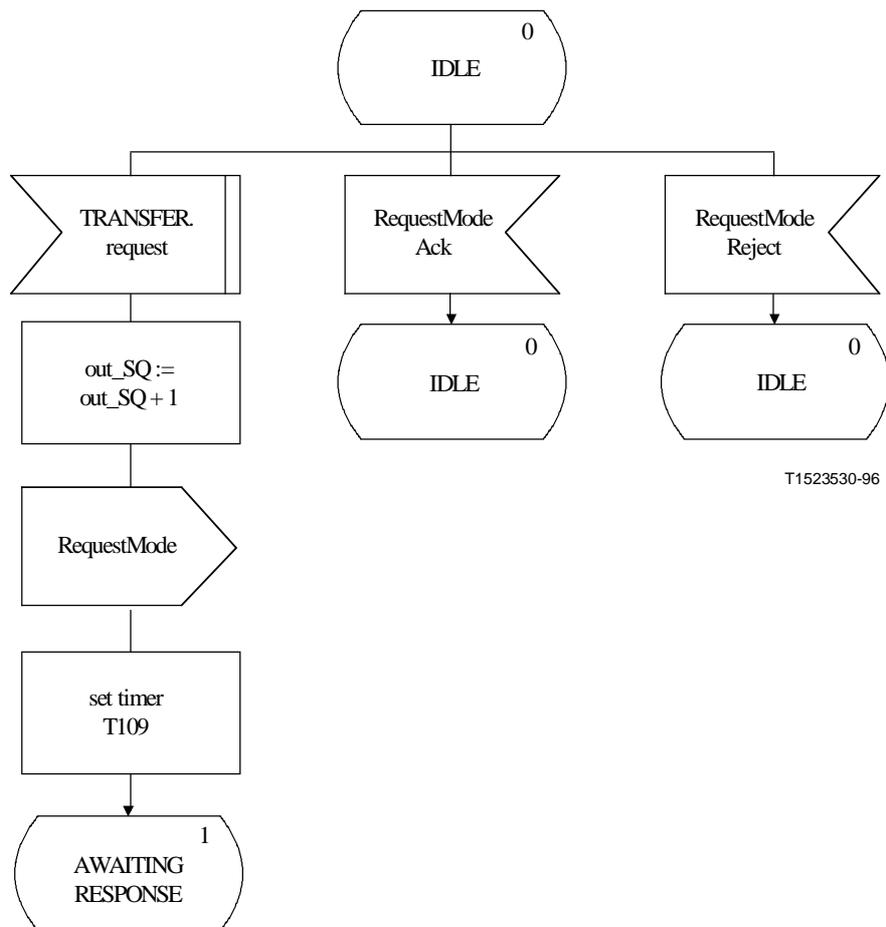


Figure 38 i)/H.245 – Description SDL de l'entité MRSE sortante

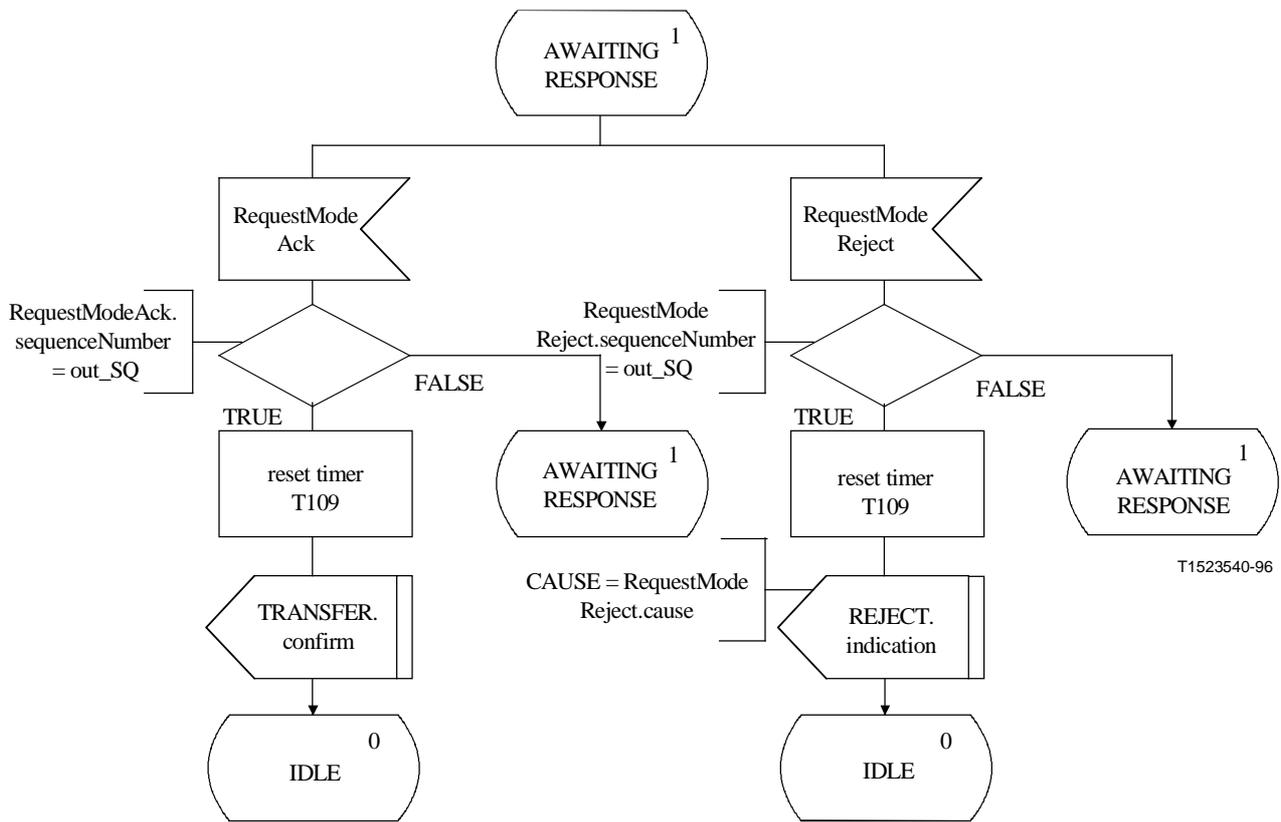


Figure 38 ii)/H.245 – Description SDL de l'entité MRSE sortante

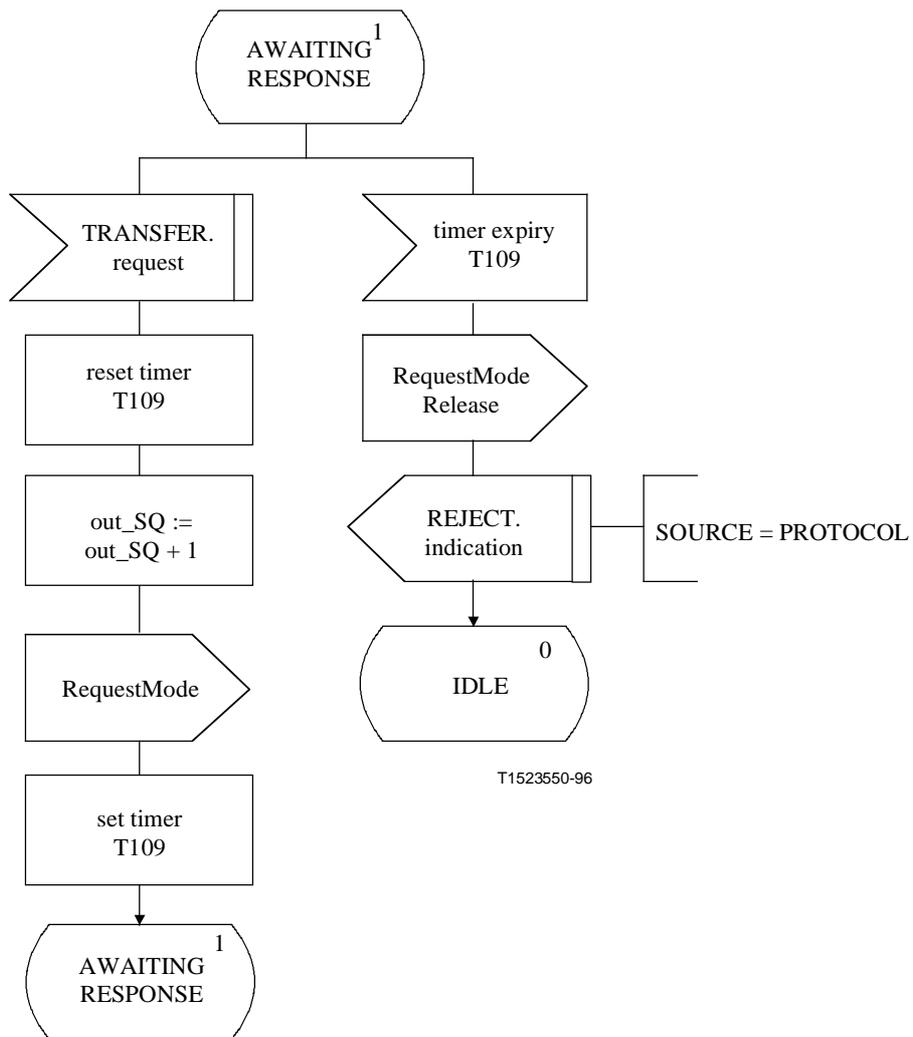


Figure 38 iii)/H.245 – Description SDL de l'entité MRSE sortante

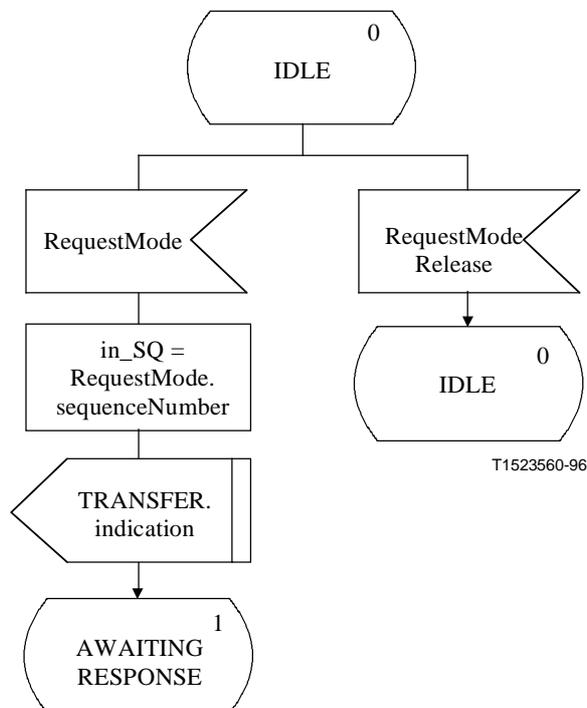


Figure 39 i)/H.245 – Description SDL de l'entité MRSE entrante

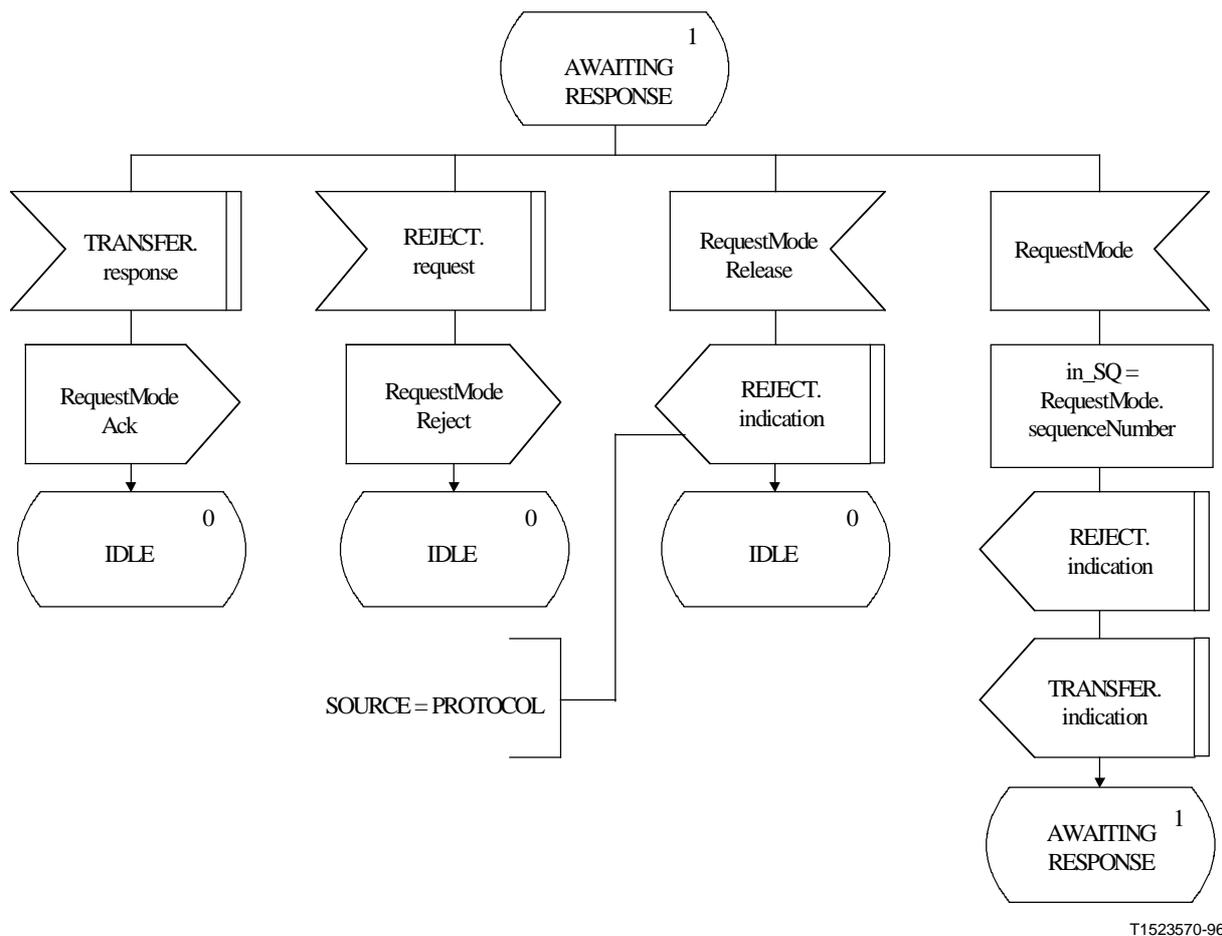


Figure 39 ii)/H.245 – Description SDL de l'entité MRSE entrante

8.10 Procédures liées au temps de propagation aller et retour

8.10.1 Introduction

Les procédures décrites ici permettent de déterminer le temps de propagation aller et retour entre deux terminaux en cours de communication. Cette fonction permet également à un utilisateur de H.245 de déterminer si une entité homologue du protocole H.245 est toujours active.

Il est fait référence à la fonction décrite ici sous le nom d'entité de signalisation du temps de propagation aller et retour (RTDSE, *round trip delay signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité RTDSE et l'utilisateur de l'entité RTDSE. Il y a une instance de l'entité RTDSE dans chacun des terminaux maître et esclave. Les deux types de terminaux maître et esclave peuvent procéder à la détermination du temps de propagation aller et retour.

Le texte suivant donne un aperçu général du fonctionnement du protocole dans les entités RTDSE. En cas de divergence avec la spécification formelle, la spécification formelle sera applicable.

8.10.1.1 Aperçu général du protocole – Entités RTDSE

Une procédure de détermination du temps de propagation aller et retour est déclenchée quand la primitive de demande TRANSFER est présentée par l'utilisateur de l'entité RTDSE. Un message

RoundTripDelayRequest est envoyé vers l'entité RTDSE homologue, et le temporisateur T105 démarre. Si un message RoundTripDelayResponse est reçu en réponse au message RoundTripDelayRequest, le temporisateur T105 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER du temps de propagation aller et retour, qui correspond à la valeur du temporisateur T105.

Si un message RoundTripDelayRequest est reçu à tout moment à partir de l'entité RTDSE homologue, un message RoundTripDelayResponse est envoyé immédiatement à l'entité RTDSE homologue.

Si le temporisateur T105 expire, l'utilisateur de l'entité RTDSE est informé par la primitive d'indication EXPIRY.

Seul un message RoundTripDelayResponse répondant au message RoundTripDelayRequest le plus récent est accepté. Les messages envoyés en réponse à des messages RoundTripDelayRequest antérieurs sont omis.

Une nouvelle procédure de détermination du temps de propagation aller et retour peut être déclenchée au moyen de la primitive de demande TRANSFER par l'utilisateur de l'entité RTDSE avant qu'un message RoundTripDelayResponse ait été reçu.

8.10.2 Communication entre l'entité RTDSE et l'utilisateur de l'entité RTDSE

8.10.2.1 Primitives entre l'entité RTDSE et l'utilisateur de l'entité RTDSE

La communication entre l'entité RTDSE et l'utilisateur RTDSE s'effectue en utilisant les primitives indiquées dans le Tableau 53. Ces primitives sont destinées à définir les procédures RTDSE et ne sont pas supposées spécifier une réalisation ou apporter des contraintes.

Tableau 53/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	– (Note 1)	non défini (Note 2)	non défini	DELAY
EXPIRY	non défini	–	non défini	non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

8.10.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) la primitive TRANSFER est utilisée pour demander la détermination du temps de propagation aller et retour et pour en donner un compte rendu;
- b) la primitive EXPIRY indique qu'aucune réponse n'a été reçue du terminal homologue.

8.10.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau 53 est la suivante:

- a) le paramètre DELAY indique le résultat du temps de propagation aller et retour.

8.10.2.4 Etats de l'entité RTDSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité RTDSE et l'utilisateur de l'entité RTDSE.

Etat 0: IDLE

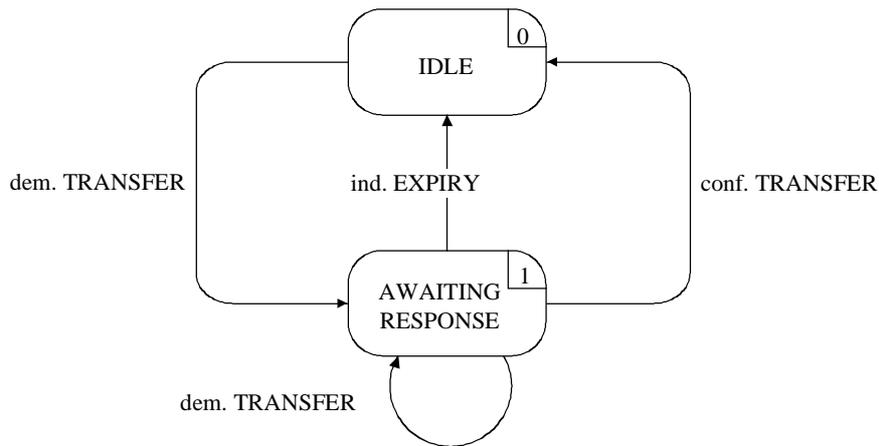
Il n'y a pas de requête RTDSE transfer en cours.

Etat 1: AWAITING RESPONSE

L'utilisateur de l'entité RTDSE a demandé la mesure du temps de propagation aller et retour. Une réponse de l'entité RTDSE homologue est attendue.

8.10.2.5 Diagramme de changement d'états

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité RTDSE et l'utilisateur de l'entité RTDSE. Les séquences autorisées sont indiquées à la Figure 40.



T1519510-95

Figure 40/H.245 – Diagramme de changement d'états pour la séquence de primitives de l'entité RTDSE

8.10.3 Communication entre les entités RTDSE homologues

8.10.3.1 Messages

Le Tableau 54 indique les messages et les champs de l'entité RTDSE tels qu'ils ont été définis dans le paragraphe 6 et se rapportent au protocole RTDSE.

Tableau 54/H.245 – Noms et champs des messages RTDSE

Fonction	Message	Champ
transfer	RoundTripDelayRequest	sequenceNumber
	RoundTripDelayResponse	sequenceNumber

8.10.3.2 Variables d'état de l'entité RTDSE

Les variables d'état suivantes de l'entité RTDSE sont définies:

out_SQ

Cette variable d'état est utilisée pour indiquer le message RoundTripDelayRequest le plus récent. Elle est augmentée de un et mappée au champ sequenceNumber du message RoundTripDelayRequest avant la transmission d'un message RoundTripDelayRequest. L'opération arithmétique effectuée sur la variable d'état out_SQ est modulo 256.

8.10.3.3 Temporisateurs RTDSE

Le temporisateur suivant est spécifié pour l'entité RTDSE:

T105

Ce temporisateur est utilisé à l'état AWAITING RESPONSE. Il spécifie le temps maximal pendant lequel aucun message RoundTripDelayResponse ne peut être reçu.

8.10.4 Procédures RTDSE

8.10.4.1 Introduction

La Figure 41 récapitule les primitives RTDSE et leurs paramètres, de même que les messages.

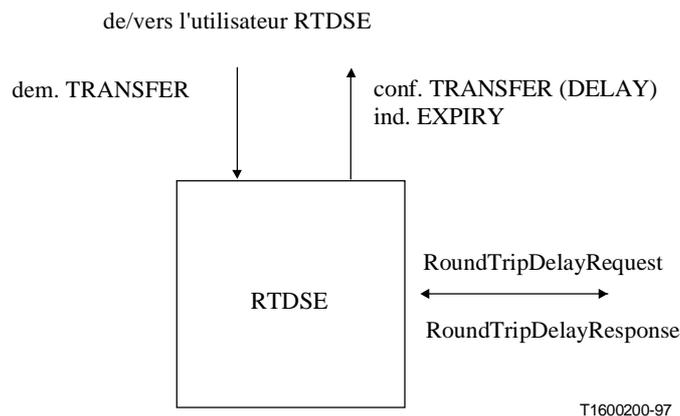


Figure 41/H.245 – Primitives et messages de l'entité RTDSE

8.10.4.2 Valeurs par défaut des paramètres de primitives

Lorsque cela n'est pas explicitement décrit dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 55.

Tableau 55/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut
conf. TRANSFER	DELAY	valeur initiale du temporisateur T105 minorée de la valeur du temporisateur T105
ind. EXPIRY	–	–

NOTE – Les temporisateurs sont définis pour le compte à rebours jusqu'à zéro. Le paramètre DELAY indique le temps de temporisation, correspondant à la différence de valeur entre la valeur initiale et la valeur gardée par le temporisateur.

8.10.4.3 Valeurs par défaut des champs de message

Quand cela n'est pas explicitement décrit dans les messages SDL, les champs de messages prennent des valeurs identiques à celles indiquées dans le Tableau 56.

Tableau 56/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut
RoundTripDelayRequest	sequenceNumber	out_SQ
RoundTripDelayResponse	sequenceNumber	RoundTripDelayRequest.sequenceNumber

8.10.4.4 Description SDL

Les procédures des entités RTDSE sont décrites sous forme de diagrammes SDL dans la Figure 42.

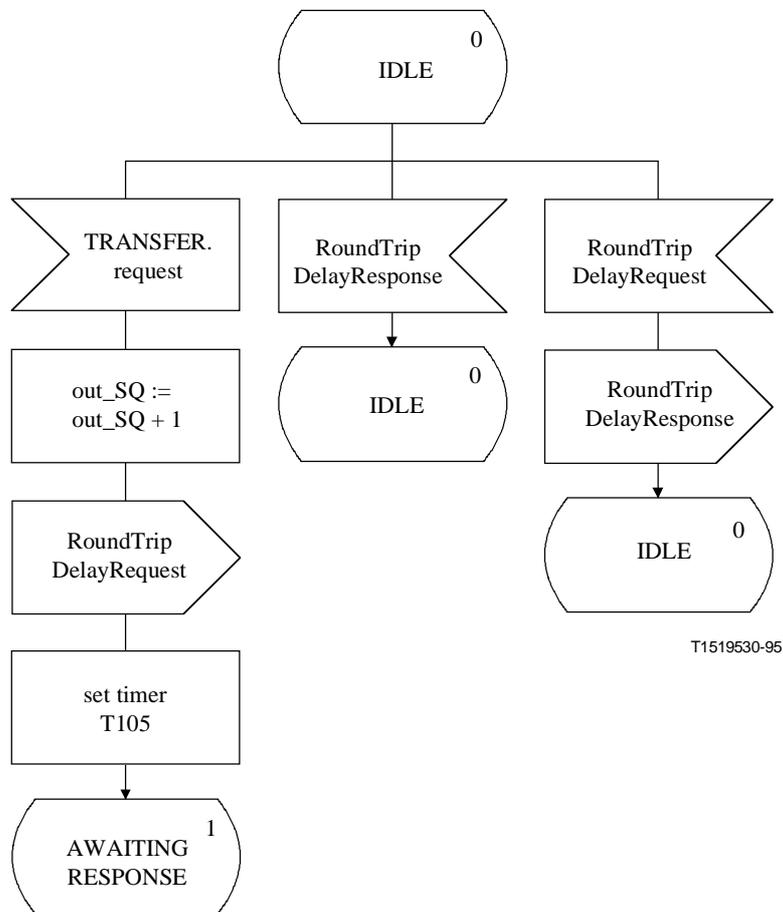


Figure 42 i)/H.245 – Description SDL de l'entité RTDSE

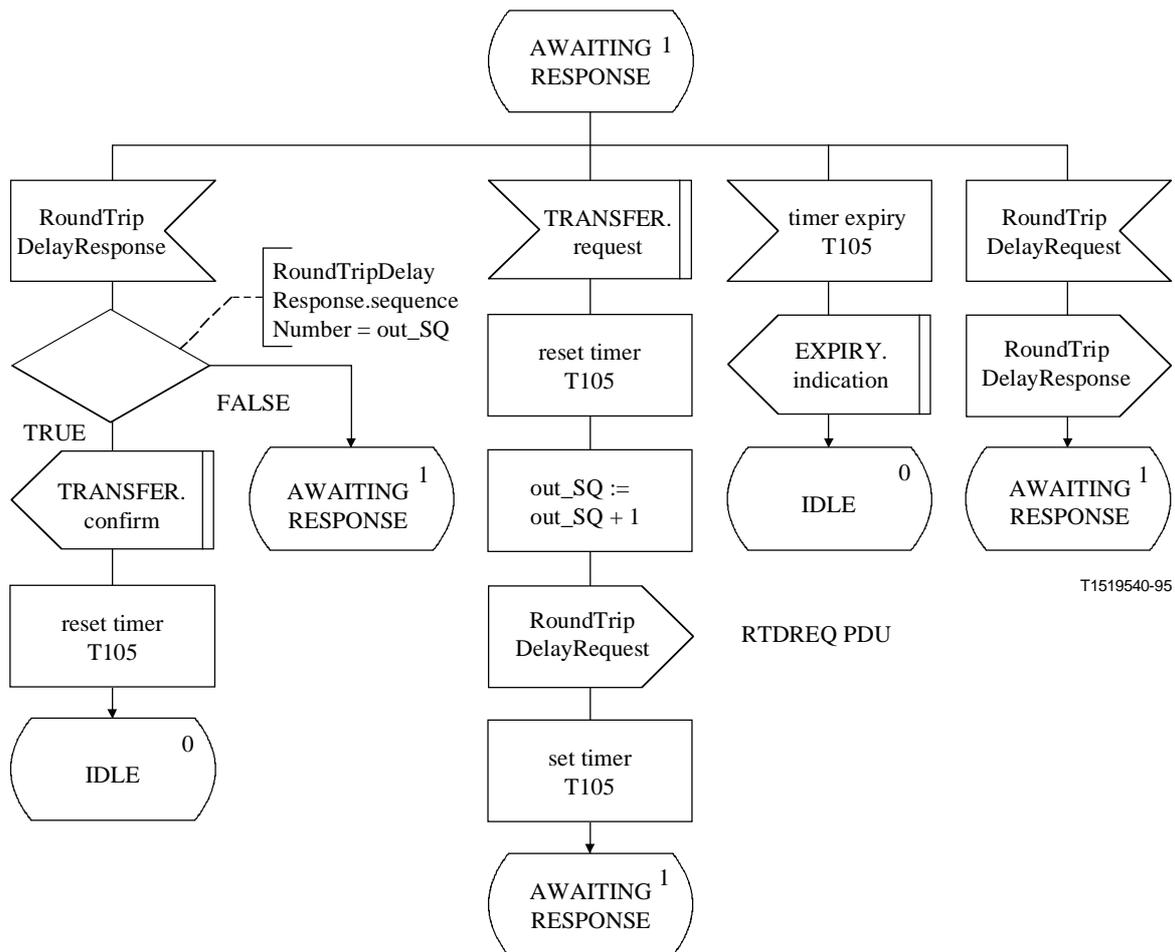


Figure 42 ii)/H.245 – Description SDL de l'entité RTDSE

8.11 Procédures relatives à la boucle de maintenance

8.11.1 Introduction

Le protocole spécifié ici assure la fiabilité de fonctionnement des boucles de maintenance en utilisant des procédures d'acquiescement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de la boucle de maintenance (MLSE, *maintenance loop signalling entity*). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité MLSE et l'utilisateur de l'entité MLSE, de même qu'en termes d'états MLSE. Les informations relatives au protocole sont transmises vers l'entité MLSE homologue au moyen des messages pertinents définis au paragraphe 6.

Il y a une entité MLSE sortante et une entité MLSE entrante. A chacune des extrémités entrante et sortante, il y a une instance d'entité MLSE pour chaque voie logique bidirectionnelle, et une instance pour la boucle du système. Il n'y a pas de liaison entre une entité MLSE entrante et une entité MLSE sortante située du même côté, autre que celle obtenue au moyen de primitives vers l'utilisateur de l'entité MLSE et à partir de ce même utilisateur. Les conditions d'erreur des entités MLSE sont indiquées.

Le terminal contenant l'entité MLSE entrante devra mettre en boucle les données appropriées pendant qu'il est à l'état LOOPED, et non à un autre moment. Le terminal contenant l'entité MLSE sortante devra être capable de recevoir les données en boucle pendant qu'il est dans un état

quelconque, mais pendant qu'il est à l'état LOOPED, il devrait recevoir des données en boucle uniquement.

NOTE – Le message MaintenanceLoopOffCommand s'applique à toutes les entités MLSE. Il est toujours utilisé pour arrêter toutes les boucles de maintenance.

Le texte suivant est un aperçu général du fonctionnement du protocole MLSE. En cas de divergence avec la spécification formelle, cette dernière sera applicable.

8.11.1.1 Aperçu général du protocole – Sortant

L'établissement d'une boucle de maintenance est déclenchée quand la primitive de demande LOOP est présentée par l'utilisateur dans l'entité MLSE sortante. Un message MaintenanceLoopRequest est envoyé vers l'entité MLSE entrante homologue, et le temporisateur T102 démarre. Si un message MaintenanceLoopAck est reçu en réponse à un message MaintenanceLoopRequest, le temporisateur T102 est arrêté et l'utilisateur est informé par la primitive de confirmation LOOP que la boucle de maintenance a été établie avec succès. Si, cependant, un message MaintenanceLoopReject est reçu en réponse à un message MaintenanceLoopRequest, le temporisateur T102 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité MLSE homologue a refusé l'établissement de la boucle de maintenance.

Si le temporisateur T102 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE, et un message MaintenanceLoopOffCommand est envoyé vers l'entité MLSE entrante homologue. Ceci annulera toutes les boucles de maintenance, et plus spécifiquement celle relative à l'entité MLSE elle-même.

Une boucle de maintenance qui a été établie avec succès peut être annulée quand la primitive de demande RELEASE est présentée par l'utilisateur de l'entité MLSE sortante. Un message MaintenanceLoopOffCommand est envoyé vers l'entité MLSE entrante homologue.

Avant qu'un des messages MaintenanceLoopAck ou MaintenanceLoopReject ait été reçu en réponse à un message MaintenanceLoopRequest préalablement envoyé, l'utilisateur de l'entité MLSE sortante peut annuler la boucle de maintenance en utilisant la primitive de demande RELEASE.

8.11.1.2 Aperçu général du protocole – Entrant

Quand un message MaintenanceLoopRequest est reçu dans l'entité MLSE entrante, l'utilisateur est informé de la demande d'établissement d'une boucle de maintenance par la primitive d'indication LOOP. L'utilisateur de l'entité MLSE entrante signale l'acceptation de la demande d'établissement de la boucle de maintenance en présentant la primitive de réponse LOOP, et un message MaintenanceLoopAck est envoyé vers l'entité MLSE sortante homologue. La boucle de maintenance doit à présent être exécutée. L'utilisateur de l'entité MLSE entrante signale le rejet de la demande d'établissement de la boucle de maintenance en présentant la primitive de demande RELEASE, et un message MaintenanceLoopReject est envoyé à l'entité MLSE sortante homologue.

Une boucle de maintenance qui a été établie avec succès peut être annulée quand le message MaintenanceLoopOffCommand est reçu dans l'entité MLSE entrante. L'utilisateur de l'entité MLSE entrante est informé par la primitive d'indication RELEASE.

8.11.2 Communication entre l'entité MLSE et l'utilisateur de l'entité MLSE

8.11.2.1 Primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE

La communication entre l'entité MLSE et l'utilisateur de l'entité MLSE est effectuée au moyen des primitives indiquées dans le Tableau 57.

Tableau 57/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
LOOP	LOOP_TYPE	LOOP_TYPE	– (Note 1)	–
RELEASE	CAUSE	SOURCE CAUSE	non défini (Note 2)	non défini
ERROR	non défini	ERRCODE	non défini	non défini
NOTE 1 – "–" signifie l'absence de paramètres NOTE 2 – "non défini" signifie que cette primitive n'existe pas.				

8.11.2.2 Définition des primitives

La définition de ces paramètres est la suivante:

- a) les primitives LOOP sont utilisées pour établir une boucle de maintenance;
- b) les primitives RELEASE sont utilisées pour annuler une boucle de maintenance;
- c) la primitive ERROR indique les erreurs MLSE à une entité de gestion.

8.11.2.3 Définition des paramètres

La définition des paramètres de primitives indiquée au Tableau 57 est la suivante:

- a) le paramètre LOOP_TYPE spécifie les paramètres associés à la boucle de maintenance. Il a les valeurs de "SYSTEM", "MEDIA", and "LOGICAL_CHANNEL". Ce paramètre, de même que le numéro de voie logique, déterminent la valeur du champ de type du message MaintenanceLoopRequest qui est alors transmis de façon permanente vers l'utilisateur de l'entité MLSE homologue;
- b) le paramètre SOURCE indique à l'utilisateur de l'entité MLSE la source de la libération de la boucle de maintenance. Le paramètre SOURCE a la valeur de "USER" or "MLSE", indiquant l'utilisateur de l'entité MLSE ou l'entité MLSE elle-même. Cette dernière peut survenir par suite d'une erreur de protocole;
- c) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité MLSE homologue a rejeté une demande d'établissement de boucle de protocole. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "MLSE";
- d) le paramètre ERRCODE indique le type d'erreur MLSE. Le Tableau 61 indique les valeurs autorisées du paramètre ERRCODE.

8.11.2.4 Etats MLSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE, de même que les messages entre les entités MLSE homologues. Les états sont spécifiés séparément pour chaque entité MLSE entrante et pour chaque entité MLSE sortante. Les états correspondant à une entité MLSE sortante sont:

Etat 0: NOT LOOPED

Il n'y a pas de boucle de maintenance.

Etat 1: AWAITING RESPONSE

L'entité MLSE sortante attend l'établissement d'une boucle de maintenance avec une entité MLSE entrante homologue.

Etat 2: LOOPED

La boucle de maintenance entre les entités MLSE homologues a été établie. Toutes les données reçues sur le canal approprié devraient être des données en boucle.

Les états correspondant à une entité MLSE entrante sont:

Etat 0: NOT LOOPED

Il n'y a pas de boucle de maintenance.

Etat 1: AWAITING RESPONSE

L'entité MLSE entrante attend l'établissement d'une boucle de maintenance avec une entité MLSE sortante. Les données appropriées ne devront pas être mises en boucle.

Etat 2: LOOPED

Une boucle de maintenance a été établie entre des entités MLSE homologues. Toutes les données reçues sur le canal approprié devront être mises en boucle.

8.11.2.5 Diagramme de changement d'états

La séquence autorisée de primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE est définie ici. La séquence autorisée de primitives concerne les états de l'entité MLSE du point de vue de l'utilisateur de l'entité MLSE. Les séquences sont spécifiées séparément pour chacune des entités MLSE entrante et sortante, comme cela est indiqué dans les Figures 43 et 44 respectivement.

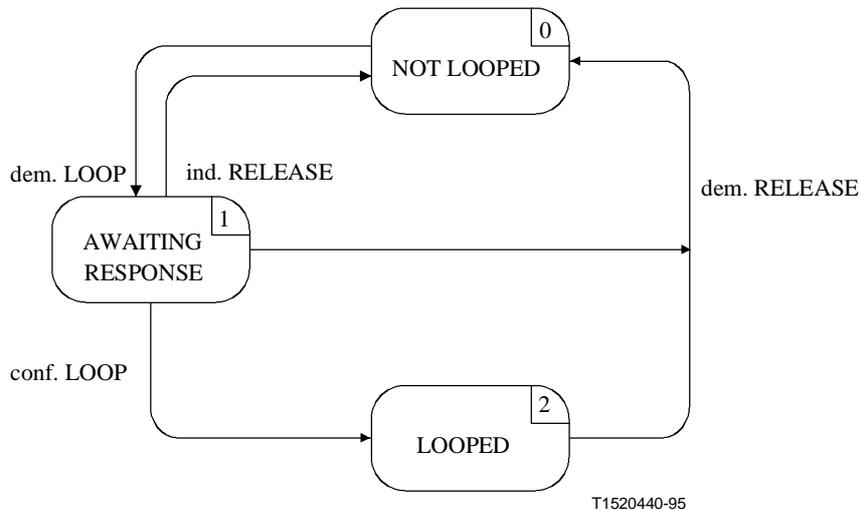


Figure 43/H.245 – Diagramme de changement d'états pour une séquence de primitives de l'entité MLSE sortante

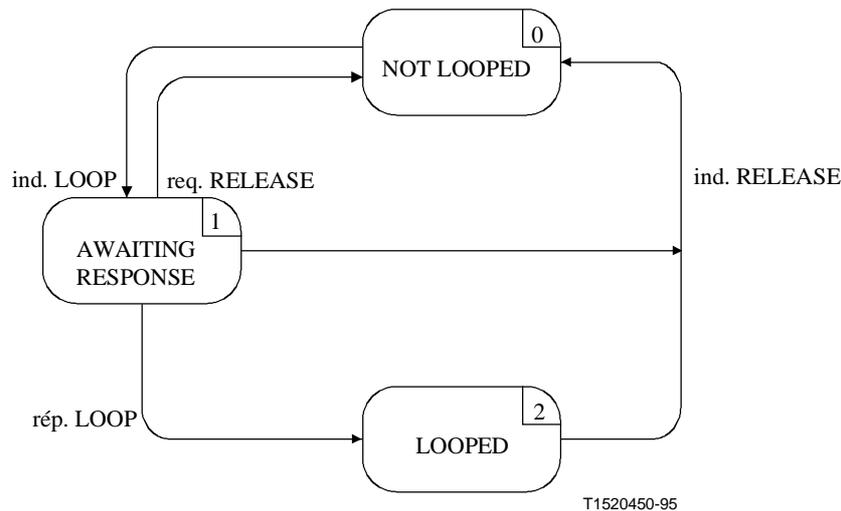


Figure 44/H.245 – Diagramme de changement d'états pour une séquence de primitives de l'entité MLSE entrante

8.11.3 Communication entre les entités MLSE homologues

8.11.3.1 Messages MLSE

Le Tableau 58 indique les messages et champs des entités MLSE définies dans le paragraphe 6, qui se rapportent au protocole MLSE.

Tableau 58/H.245 – Noms et champs de messages MLSE

Fonction	Message	Sens	Champ
establish	MaintenanceLoopRequest	S → E	type
	MaintenanceLoopAck	S ← E	type
	MaintenanceLoopReject	S ← E	type cause
release	MaintenanceLoopOffCommand	S → E	–
S sortant E entrant			

8.11.3.2 Variables d'état MLSE

La variable d'état suivante est définie dans l'entité MLSE sortante:

out_MLN

Cette variable d'état est utilisée pour établir une distinction entre les entités MLSE sortantes. Elle est initialisée lors de l'initialisation des entités MLSE sortantes. La valeur de out_MLN est utilisée pour définir le champ de type des messages MaintenanceLoopRequest envoyés à partir d'une entité MLSE sortante.

La variable d'état suivante est définie dans l'entité MLSE sortante:

in_MLN

Cette variable d'état établit une distinction entre les entités MLSE entrantes. Elle est initialisée lors de l'initialisation des entités MLSE entrantes. En ce qui concerne les messages MaintenanceLoopRequest reçus dans une entité MLSE entrante, la valeur de champ du type de message est compatible avec la valeur du paramètre in_MLN.

in_TYPE

Cette variable d'état enregistre la valeur du paramètre LOOP_TYPE quand la demande MaintenanceLoopRequest est reçue. Cette variable d'état contribue à positionner la valeur du champ de type dans le message MaintenanceLoopAck.

8.11.3.3 Temporiseurs MLSE

Le temporisateur suivant est spécifié pour l'entité MLSE sortante:

T102

Ce temporisateur est utilisé à l'état AWAITING RESPONSE. Il spécifie le temps maximal durant lequel aucun message MaintenanceLoopAck ou MaintenanceLoopReject ne peut être reçu.

8.11.4 Procédures MLSE

8.11.4.1 Introduction

La Figure 45 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités MLSE entrante et sortante.

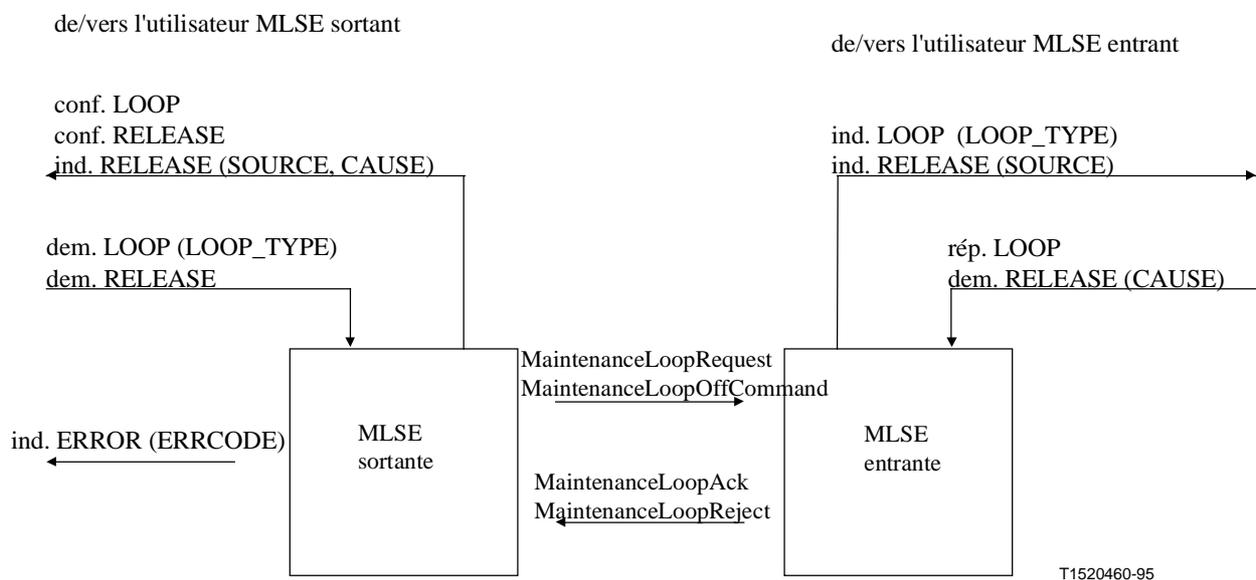


Figure 45/H.245 – Primitives et messages dans l'entité de signalisation de la boucle de maintenance

8.11.4.2 Valeurs par défaut des paramètres de primitives

Quand cela n'a pas été explicitement décrit dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent des valeurs identiques à celles indiquées dans le Tableau 59.

Tableau 59/H.245 – Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut (note)
ind. LOOP	LOOP_TYPE	MaintenanceLoopRequest.type
ind. RELEASE	SOURCE	USER
	CAUSE	MaintenanceLoopReject.cause
NOTE – Un paramètre par défaut devra être codé comme mis à la valeur "null" si un champ de message indiqué n'est pas présent dans le message.		

8.11.4.3 Valeurs par défaut des champs de message

Quand cela n'a pas été explicitement décrit dans les diagrammes SDL, les champs de message prennent des valeurs identiques à celles indiquées dans le Tableau 60.

Tableau 60/H.245 – Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note 1)
MaintenanceLoopRequest	type	dem. LOOP(LOOP_TYPE) et out_MLN (Note 2)
MaintenanceLoopAck	type	in_LOOP et in_MLN (Note 3)
MaintenanceLoopReject	type	in_LOOP et in_MLN (Note 3)
	cause	dem. RELEASE(CAUSE)
MaintenanceLoopOffCommand	–	–
NOTE 1 – Un champ de message ne devra pas être codé si le paramètre de primitive correspondant est nul, c'est-à-dire absent.		
NOTE 2 – La valeur du champ de type est déduite du paramètre LOOP_TYPE et du numéro de voie logique.		
NOTE 3 – La valeur du champ de type est déduite des variables d'état in_LOOP et in_MLN.		

8.11.4.4 Valeurs des paramètres ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition particulière d'erreur. Le Tableau 61 indique les valeurs que le paramètre ERRCODE peut prendre dans les entités MLSE sortantes. Il n'y a pas de primitive d'indication ERROR associée à l'entité MLSE entrante.

Tableau 61/H.245 – Valeurs des paramètres ERRCODE dans l'entité MLSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
message inapproprié	A	MaintenanceLoopAck	LOOPED
pas de réponse de l'entité MLSE homologue	B	expiration de la temporisation T102	AWAITING RESPONSE

8.11.4.5 Description SDL

Les procédures de l'entité MLSE entrante et de l'entité MLSE sortante sont décrites en langage SDL dans les Figures 46 et 47 respectivement.

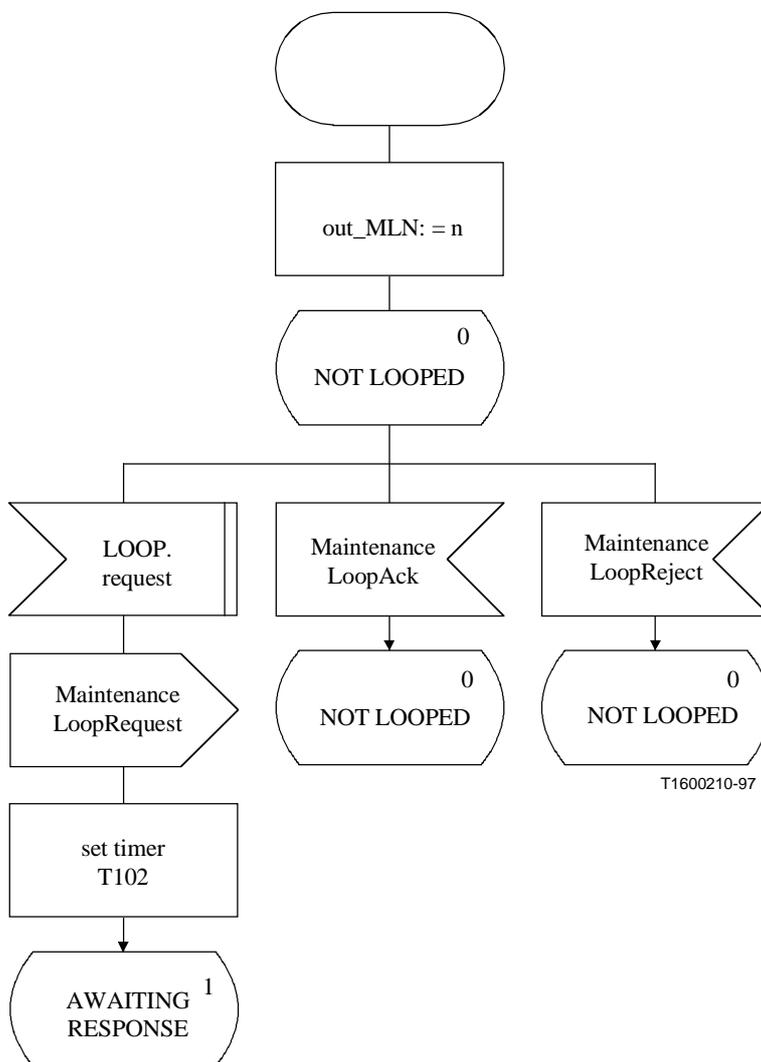


Figure 46 i)/H.245 – Description SDL des entités MLSE sortantes

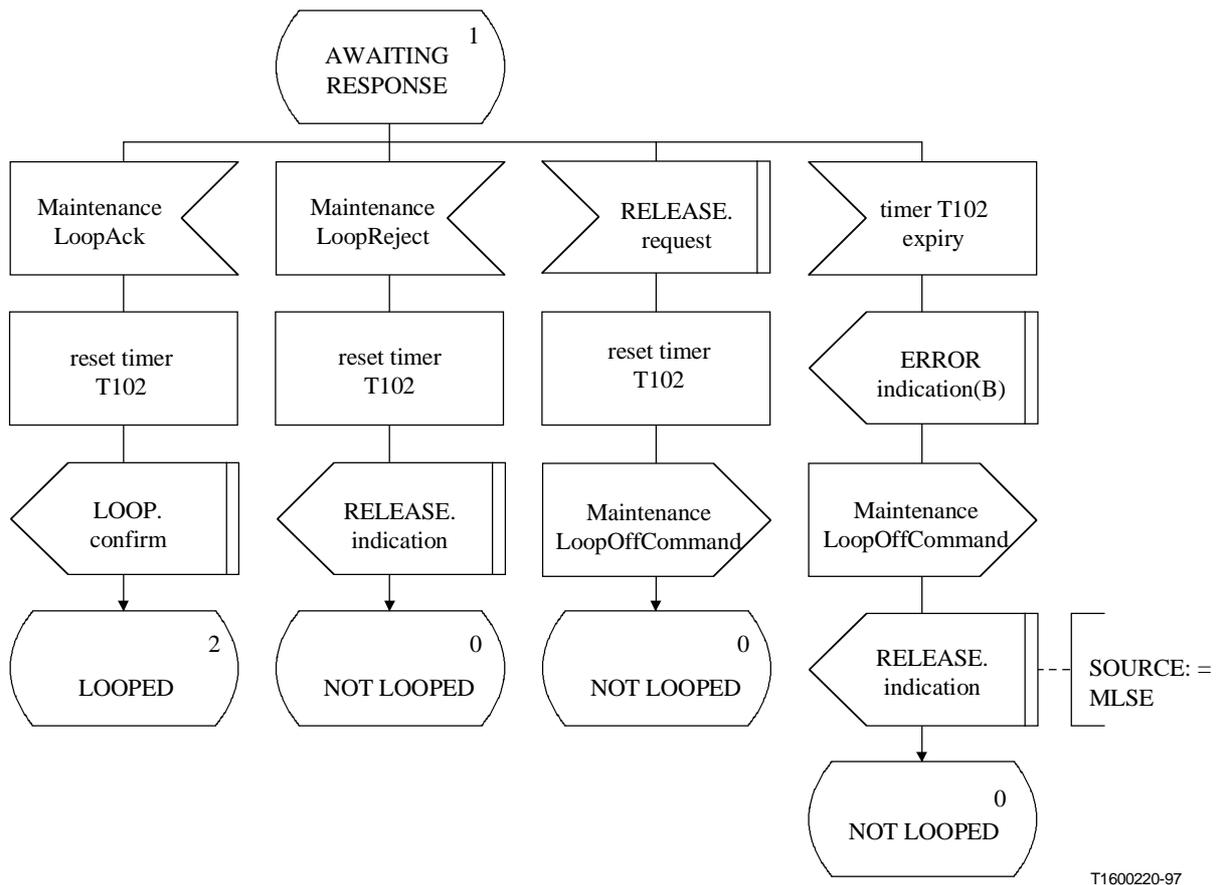


Figure 46 ii)/H.245 – Description SDL des entités MLSE sortantes

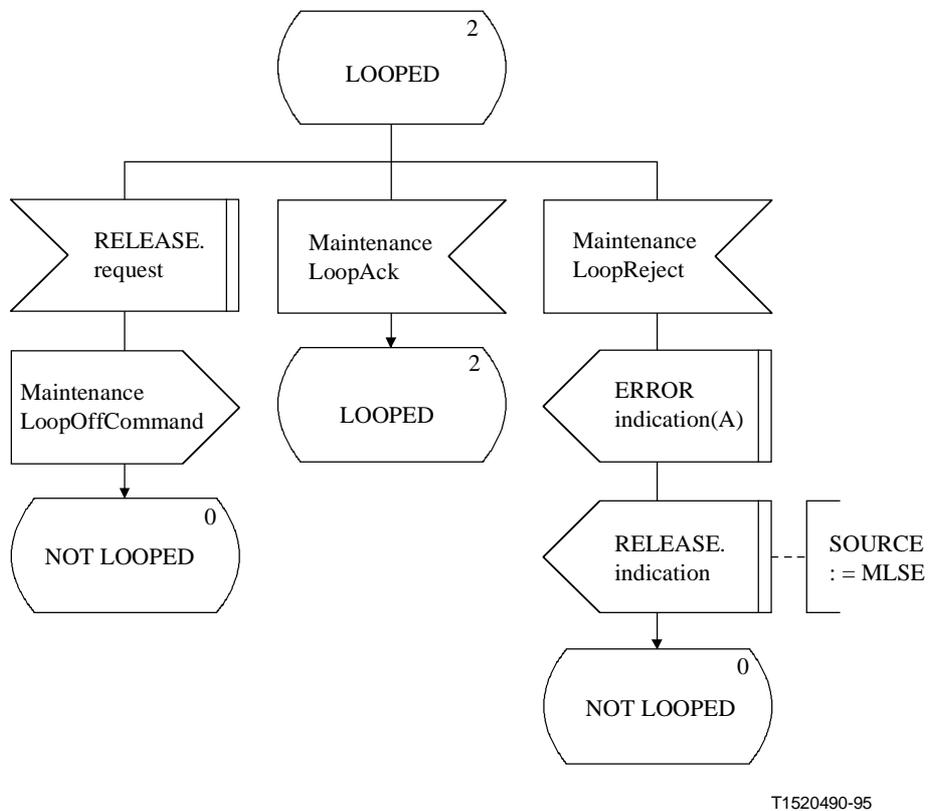


Figure 46 iii)/H.245 – Description SDL des entités MLSE sortantes

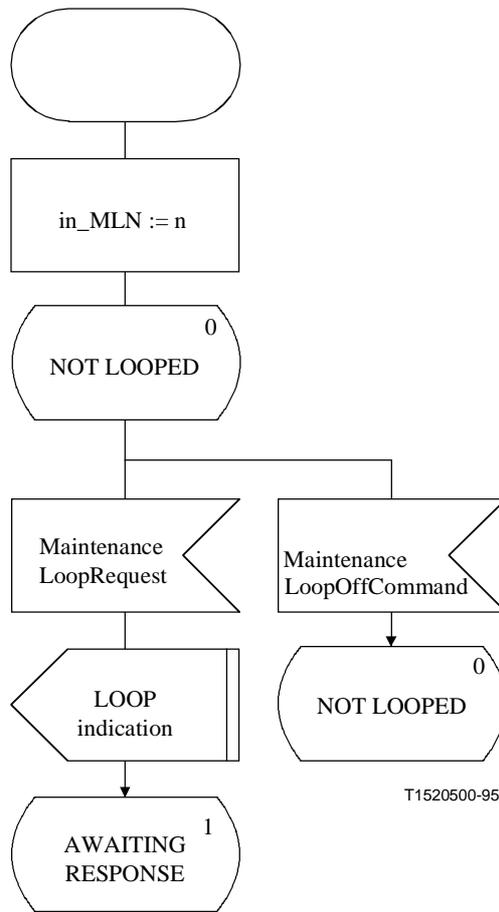


Figure 47 i)/H.245 – Description SDL des entités MLSE entrantes

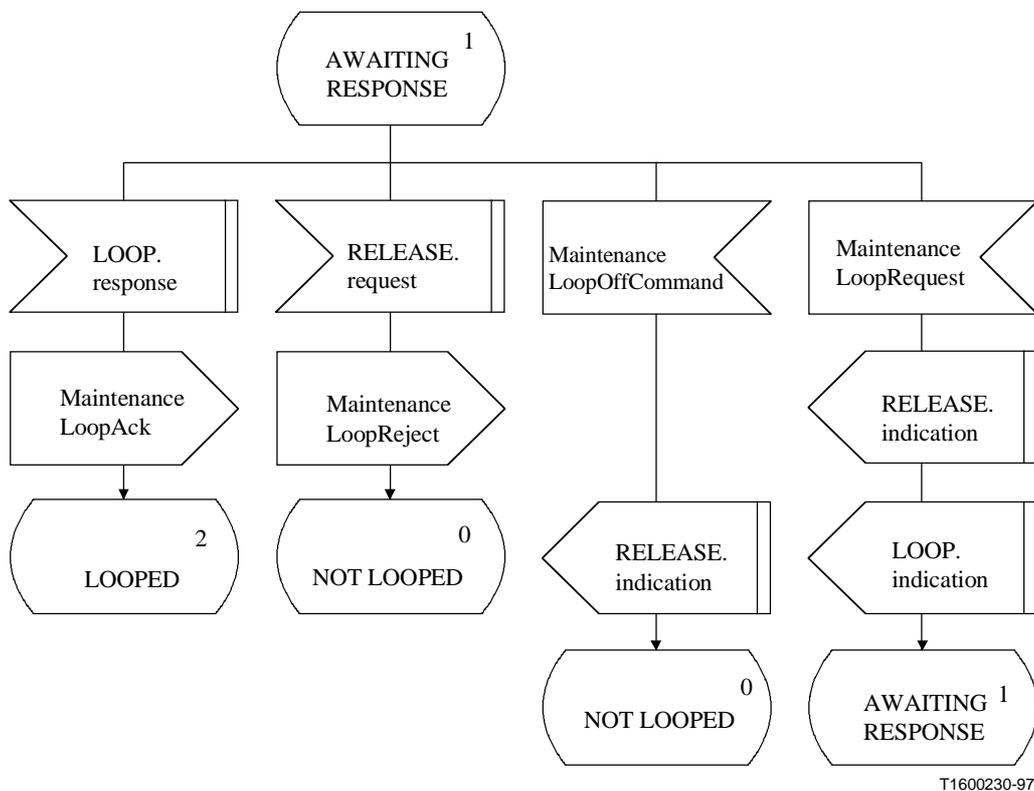


Figure 47 ii)/H.245 – Description SDL des entités MLSE entrantes

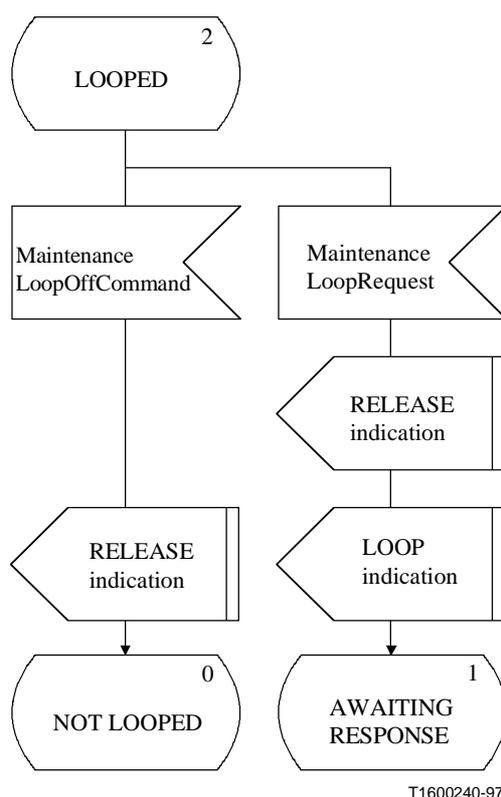


Figure 47 iii)/H.245 – Description SDL des entités MLSE entrantes

ANNEXE A

Affectations d'identificateur d'objet

Le Tableau A.1 énumère l'affectation des identificateurs d'objet définis en vue de leur utilisation dans la présente Recommandation.

Tableau A.1

Valeur d'identificateur d'objet	Description
{itu-t (0) recommendation (0) h (8) 245 version (0) 1 }	cet identificateur d'objet est utilisé pour indiquer la version de la présente Recommandation utilisée comme protocole de commande du système multimédia. Ceci indique la première version de la présente Recommandation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 2 }	cet identificateur d'objet est utilisé pour indiquer la version de la présente Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, deux versions normalisées ont été définies. Ceci indique la seconde version de la présente Recommandation.

APPENDICE I

Aperçu général de la syntaxe ASN.1

I.1 Introduction à la syntaxe ASN.1

La notation de syntaxe abstraite un (ASN.1, *abstract syntax notation one*) est un langage de spécification de données. Ce langage a été initialement normalisé par la Recommandation X.409 faisant partie de la série des Recommandations de messagerie électronique X.400. La Recommandation X.208 et, plus récemment, la Recommandation X.680 sont issues de la Recommandation initiale. La syntaxe ASN.1 permet de spécifier sans ambiguïté des structures de données complexes, y compris celles contenant des champs de longueur variable, des champs facultatifs, de même que des récurrences.

Les Recommandations ci-dessus ne traitent que des spécifications ASN.1 relatives à la syntaxe et à la sémantique. Le codage binaire des structures de données est traité dans d'autres Recommandations, notamment la Recommandation X.690 (règles de codage de base ou BER) et X.691 (règles de codage compacté ou PER). Les règles de codage de base (BER, *basic encoding rules*) permettent le déchiffrement de données par des systèmes ayant une connaissance générale de la notation ASN.1 mais pas des détails de la spécification utilisée pour la mise en forme des données. En d'autres termes, les types de données sont codés, de même que les valeurs de données. Les règles de codage compacté (PER, *packed encoding rules*) sont plus efficaces, étant donné que seules les valeurs de données sont codées et que le codage est conçu avec très peu de redondance. Cette méthode peut être utilisée quand on prévoit que les données seront conformes à une structure donnée à la fois pour l'émetteur et le récepteur.

Les réalisations conformes à la présente Recommandation utilisent les règles de codage compacté (PER). Étant donné que les deux correspondants savent que les messages seront conformes à la spécification H.245, il n'est pas nécessaire de coder cette spécification dans les messages. À des fins de simplification, la variante alignée sur des octets du codage PER est utilisée. Cela exige que les champs de données dont la longueur est égale ou supérieure à huit bits soient alignés sur des limites d'octets, et qu'un nombre entier d'octets soit pris. L'alignement est effectué en ajoutant des zéros aux données à l'avant des champs de grande longueur.

I.2 Types de données de base de la syntaxe ASN.1

Le type de données le plus simple est BOOLEAN (booléen) qui prend les valeurs FAUX et VRAI. Ces valeurs sont codées sur un seul bit prenant respectivement les valeurs 0 et 1. Par exemple, le paramètre **segmentableFlag** BOOLEAN est codé

Valeur	Codage
FAUX	0
VRAI	1

Le type de données essentiel est INTEGER (nombre entier), qui représente des valeurs de nombres entiers. Les nombres du type INTEGER peuvent être illimités comme dans l'expression:

bitRate INTEGER

ou ils peuvent être limités à une gamme de valeurs, comme par exemple:

maximumA12SDUSize INTEGER (0..65535)

Les nombres entiers limités sont codés différemment selon la taille de la gamme. Supposons que N soit le nombre de valeurs dans la gamme, c'est-à-dire la limite supérieure moins la limite inférieure plus un. Le nombre entier limité sera codé selon l'une des cinq façons suivantes compte tenu de la valeur de N:

N	Codage
1	pas de bits nécessaires
2-255	un champ compris entre 1 et 8 bits non aligné sur l'octet
256	un champ de 8 bits aligné sur l'octet
257-65536	un champ de 16 bits alignés sur octets
supérieur	comme le nombre minimal d'octets alignés précédé par le codage du nombre d'octets selon la règle ci-dessus

Dans tous les cas, le nombre qui est effectivement utilisé est la valeur devant être codée moins la limite inférieure de la gamme. Dans ces exemples, les "bits d'alignement sur octets" représentent de 0 à 7 bits nuls qui sont ajoutés à la chaîne codée de sorte que le champ suivant puisse commencer à la frontière d'un octet.

firstGOB INTEGER (0..17)

Valeur	Codage
0	00000
3	00011

h233IVResponseTime INTEGER (0..255)

Valeur	Codage
3	bits d'alignement sur l'octet 00000011
254	bits d'alignement sur l'octet 11111110

skew (décalage temporel) INTEGER (0..4095)

Valeur	Codage
3	bits d'alignement sur octets 00000000 00000011
4095	bits d'alignement sur octets 00001111 11111111

Les valeurs de nombres entiers illimités (complément à 2) qui peuvent être représentés par un nombre d'octets inférieur ou égal à 127 octets sont codées sur le nombre minimal d'octets nécessaires. Le nombre d'octets (la longueur) est codé(e) comme un octet aligné qui précède le nombre lui-même. Par exemple,

- 1 bits d'alignement sur octets 00000001 11111111
- 0 bits d'alignement sur octets 00000001 00000000
- 128 bits d'alignement sur octets 00000010 00000000 10000000
- 1000000 bits d'alignement sur octets 00000011 00001111 01000010 01000000

La notation ASN.1 accepte différents types de chaînes de caractères. Ces chaînes sont constituées, par exemple, de suites de longueur variable contenant des bits, des octets, ou d'autres types de données courtes. Elles sont généralement codées sous forme d'une longueur suivie par les données. La longueur peut être codée comme un nombre entier illimité ou un nombre entier limité si le type SIZE (taille) de la chaîne est spécifié. Par exemple,

data OCTET STRING

Etant donné que la longueur de la chaîne d'octet n'est pas limitée, elle devra être codée comme un *nombre entier semi-limité* (ayant une borne inférieure mais pas de borne supérieure). Tout d'abord, on ajoute des bits de remplissage pour obtenir un codage aligné sur octets. Le reste du codage est défini comme suit:

Longueur	Codage
0 à 127	longueur de 8 bits suivie par les données
128 à 16K-1	longueur de 16 bits avec le bit de poids fort (MSB) mis à 1, les données à suivre
16K à 32K-1	11000001, 16K de données, ensuite coder le reste
32K à 48K-1	11000010, 32K de données, ensuite coder le reste
48K à 64K-1	11000011, 48K de données, ensuite coder le reste
64K ou plus	11000100, 64K de données, ensuite coder le reste

Cette méthode est appelée "fragmentation". Noter que si la longueur est un multiple de 16K, alors la représentation se terminera par l'octet zéro indiquant une chaîne de longueur zéro.

I.3 Types de structures de données

La notation ASN.1 comprend plusieurs types de structures de données ou de conteneurs de données qui sont similaires dans leur concept à l'union, la structure et au tableau du langage C. Ces types sont respectivement CHOICE, SEQUENCE et SEQUENCE OF. Dans tous les cas, le codage commence par des bits spécifiques au conteneur, suivis du codage normal du contenu.

CHOICE (choix) est utilisé pour sélectionner exactement un type dans un groupe de types de données. Par exemple,

```

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

```

Un numéro d'index est attribué à chaque choix, en commençant par zéro. L'index du choix réel est codé comme un nombre entier limité. L'index est suivi par le codage du choix réel ou n'est suivi par aucun élément si le choix est **NULL**. Si le marqueur d'extension est présent (comme ci-dessus), l'index est précédé par un bit qui est zéro si le choix réel s'effectue à partir de la liste initiale.

SEQUENCE est simplement une association de types de données différents. Les éléments constitutifs de la séquence peuvent être OPTIONAL (facultatifs). Le codage est très simple. S'il y a un marqueur d'extension, le premier bit indique la présence d'éléments supplémentaires. Celui-ci est

suivi par une série de bits, un par élément facultatif indiquant la présence d'éléments. On ajoute ensuite le codage des composantes de la séquence. Par exemple:

```

H261VideoCapability ::= SEQUENCE
{
    qcifMPI INTEGER (1..4) OPTIONAL, -- unités 1/29,97 Hz
    cifMPI INTEGER (1..4) OPTIONAL, -- unités 1/29,97 Hz
    temporalSpatialTradeOffCapability BOOLEAN,
    ...
}

```

Le codage comprend un bit pour le marqueur d'extension, deux bits pour les champs facultatifs, deux bits pour tout champ facultatif présent, un bit pour les paramètres booléens et toutes les données supplémentaires. Noter que cette séquence n'a pas de bits d'alignement sur octets.

Les types SEQUENCE OF et SET OF décrivent un ensemble de composantes similaires (tableau). SEQUENCE OF implique que l'ordre des éléments est important, tandis que SET OF indique que cet ordre est arbitraire. Le codage PER est le même pour les deux types.

Ces types peuvent avoir une contrainte de taille SIZE ou un nombre illimité d'éléments. Si ce nombre est connu à priori et est inférieur à 64K, il n'est pas codé. Sinon, le nombre effectif de composantes est codé de façon à obtenir une longueur limitée ou semi-limitée. Ceci est suivi par le codage des données. Si la longueur est au moins égale à 16K et est codée, la liste de données sera fragmentée comme la chaîne d'octets. Dans ce cas, les fragments sont séparés après un certain nombre de champs de composantes (16K, 32K, etc.), et non pas après un certain nombre d'octets.

L4 Type d'identificateur d'objet

Le type de valeur est normalement indiqué dans la spécification ASN.1 de sorte que les seules informations devant être codées et transmises sont les données elles-mêmes. Cependant, il est parfois souhaitable de coder le type de données, de même que la valeur des données. Par exemple, l'identificateur **protocolIdentifier** contient

```

protocolIdentifier OBJECT IDENTIFIER (identificateur d'objet)
    -- devra être mis à la valeur
    -- {itu-t (0) recommendation (0) h (8) 245 version (0) 1}

```

Ce paramètre est codé comme les données codées avec les règles de codage de base (BER) (X.690) précédées par la longueur de ce codage en octets. La longueur est codée comme un ensemble de nombres semi-limités (voir l'exemple OCTET STRING ci-dessus). Ce qui suit illustre comment ce codage est effectué.

Le premier octet indique la longueur la longueur du codage qui suit.

Les deux premières composantes de l'identificateur d'objet sont associées comme 40*première composante + seconde composante, auquel cas 40*0+0 = 0. Les autres composantes sont codées sans changement. Chacune est codée en une série d'octets, le premier bit de chaque octet indiquant s'il y en a davantage. Ainsi:

0 → 0000 0000,

8 → 0000 0100,

tandis que 245 qui est supérieur à 127 devient 1000 0001 0111 0101.

Ainsi, l'ensemble du codage en numérotation hexadécimale comprend les sept octets 06000881 750001.

APPENDICE II

Exemples de procédures H.245

II.1 Introduction

Le présent appendice donne des exemples des procédures définies dans le paragraphe 8. La Figure II.1-1 montre les symboles utilisés dans les diagrammes utilisés dans le présent appendice.

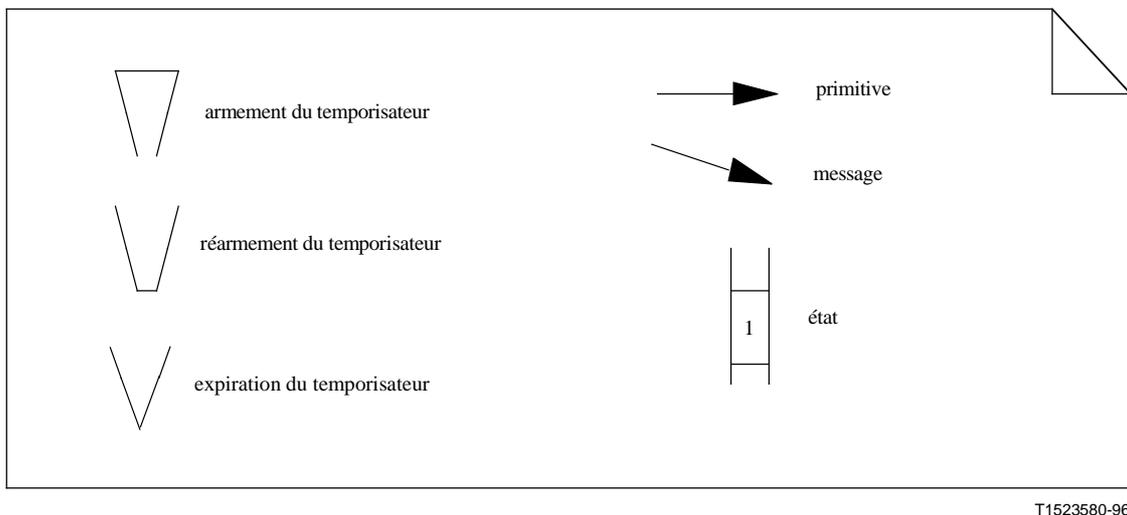


Figure II.1-1 – Symboles utilisés dans les figures

II.2 Entité de signalisation de choix du mode maître ou esclave

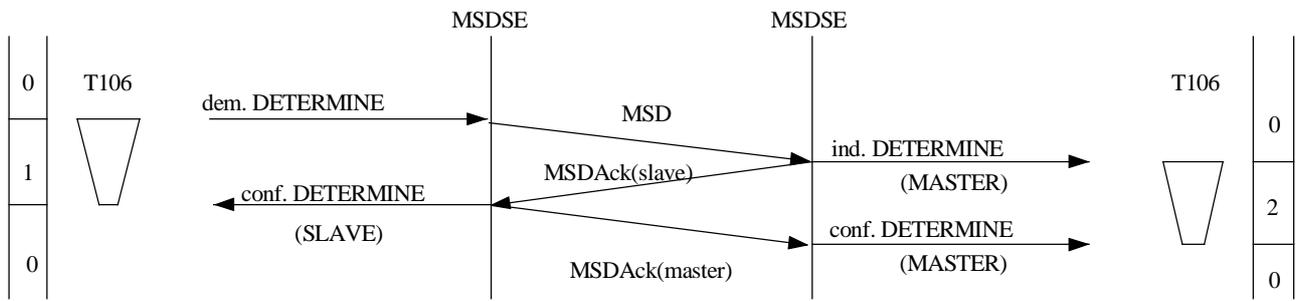
Dans les figures suivantes, les messages sont représentés par les noms abrégés indiqués dans le Tableau II.2-1.

Tableau II.2-1/H.245 – Noms abrégés du choix maître ou esclave

Message	Nom utilisé dans les exemples
MasterSlaveDetermination	MSD
MasterSlaveDeterminationAck	MSDAck
MasterSlaveDeterminationReject	MSDReject
MasterSlaveDeterminationRelease	MSDRelease

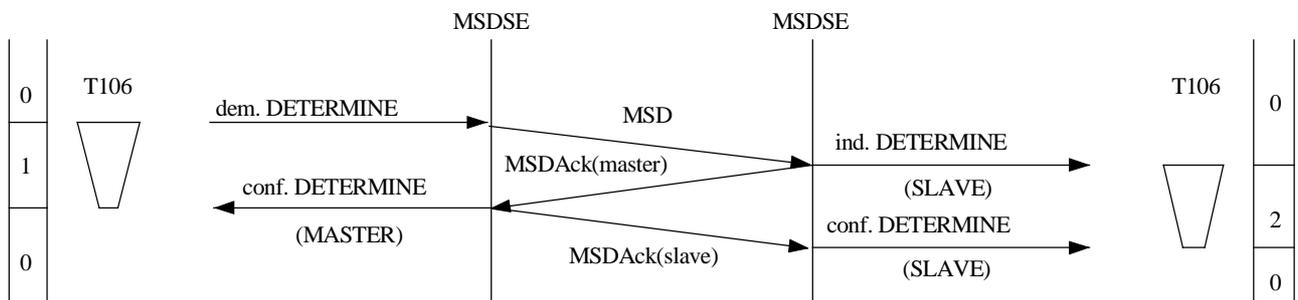
Dans les Figures II.2-1 à II.2-5, les états IDLE, OUTGOING AWAITING RESPONSE, et INCOMING AWAITING RESPONSE sont respectivement désignés par "0", "1", et "2".

Dans les Figures II.2-1 à II.2-5, la valeur paramétrique associée aux primitives d'indication DETERMINE et de confirmation DETERMINE est celle du paramètre TYPE. La valeur de champ associée au message MasterSlaveDeterminationAck est celle du champ correspondant au choix.



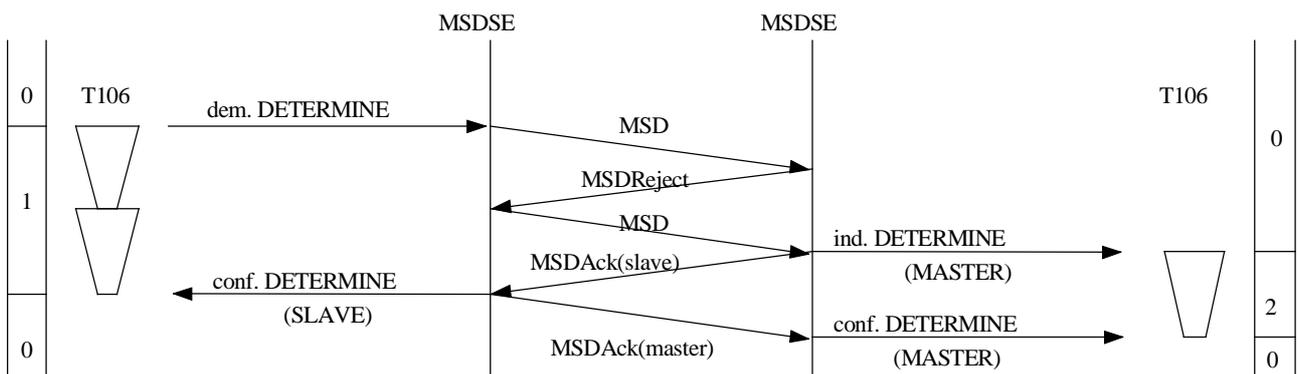
T1523590-96

Figure II.2-1/H.245 – Choix du mode maître ou esclave – Terminal maître dans l’entité MSDSE distante



T1523600-96

Figure II.2-2/H.245 – Choix du mode maître ou esclave – Esclave dans l’entité MSDSE distante



T1523610-96

Figure II.2-3/H.245 – Choix du mode maître ou esclave – La première tentative a donné un résultat indéterminé – La deuxième tentative a réussi

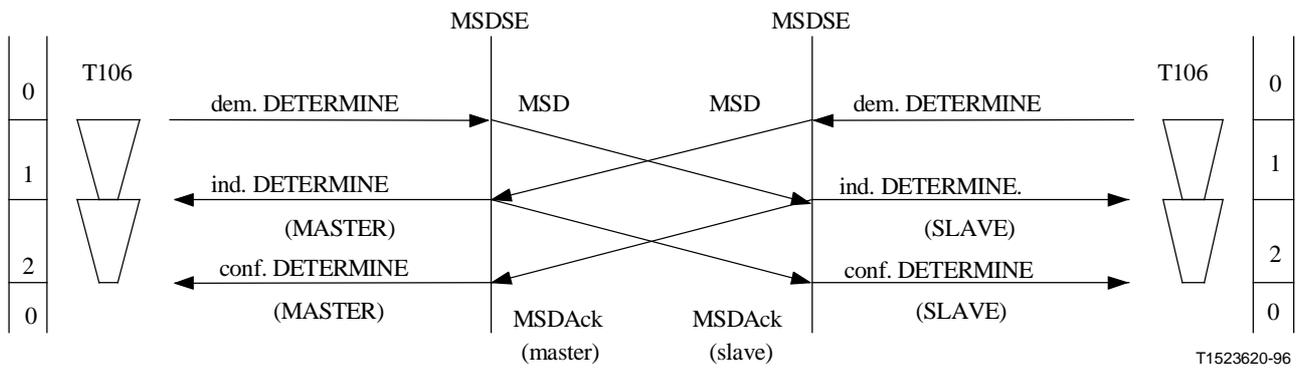


Figure II.2-4/H.245 – Choix du mode maître ou esclave – Choix simultané

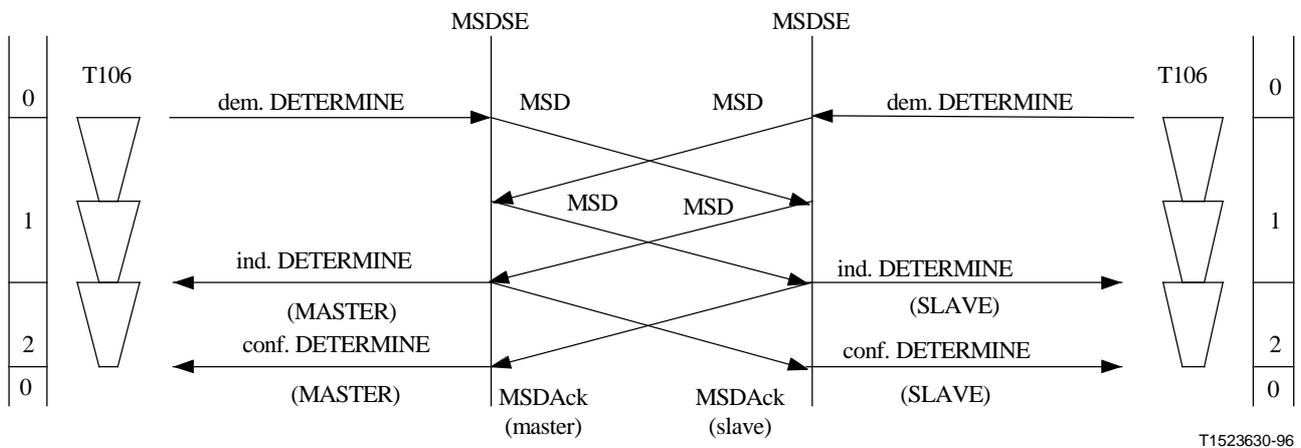


Figure II.2-5/H.245 – Choix du mode maître ou esclave – Choix simultané mais avec retour de résultat indéterminé à la première tentative

Dans la Figure II.2-6, le temporisateur local T106 est arrivé à expiration. Seul le terminal de droite connaît son statut. Ce terminal est en mesure de recevoir de nouvelles commandes mais ne peut rien demander à l'autre terminal qui dépend de la connaissance du résultat du choix de statut. Le terminal de gauche ne peut ni accepter ni lancer de nouvelles procédures. Une deuxième procédure de choix de statut doit être lancée.

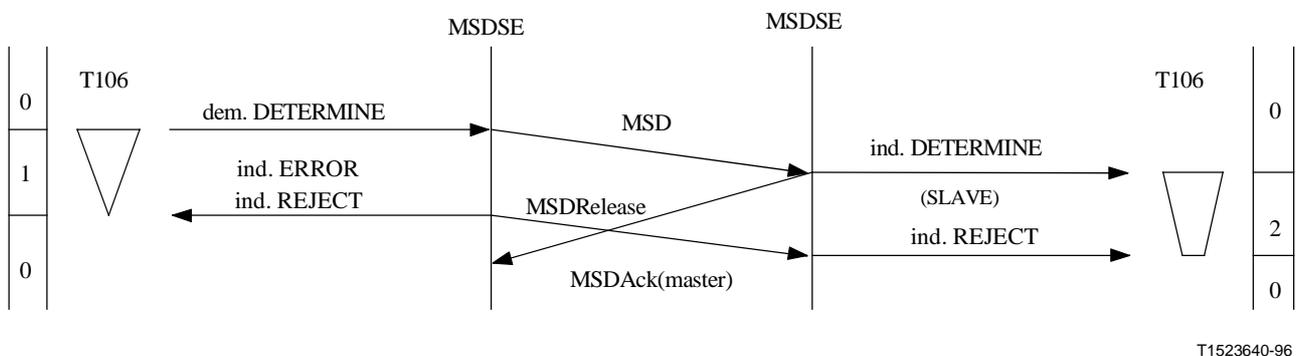


Figure II.2-6/H.245 – Choix du mode maître ou esclave – Temporisateur local T106 à expiration avec configuration d'esclave dans l'entité distante

Dans la Figure II.2-7, le temporisateur distant T106 est venu à expiration pendant l'état INCOMING AWAITING ACKNOWLEDGEMENT. Les deux terminaux connaissent leur statut. Le terminal de gauche peut recevoir et émettre des commandes. Le terminal distant ne sait cependant pas si le terminal local est prêt à recevoir, et il ne peut pas émettre des commandes dépendant de la connaissance du résultat du choix de statut. Une deuxième procédure de choix du statut devrait être lancée.

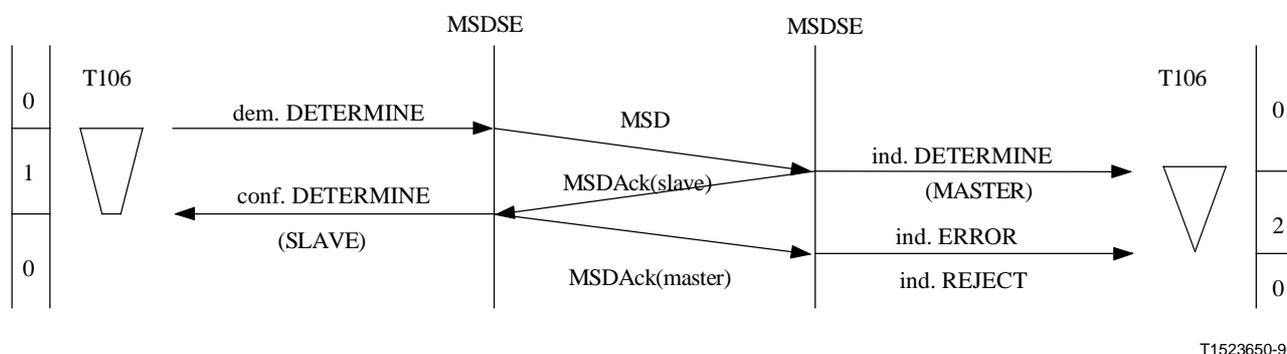


Figure II.2-7/H.245 – Choix du mode maître ou esclave – Temporisateur distant T106 à expiration avec entité maîtresse du côté distant

Dans la Figure II.2-8, le temporisateur distant T106 est arrivé à expiration pendant l'état OUTGOING AWAITING ACKNOWLEDGEMENT d'une procédure de détermination simultanée. Les deux terminaux connaissent leur statut. Le terminal de droite peut recevoir et émettre des commandes. Cependant, le terminal de gauche ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre de commandes qui dépendent de la connaissance du résultat de choix de statut. Le terminal de gauche peut recevoir de telles commandes. Une deuxième procédure de détermination de statut doit être lancée.

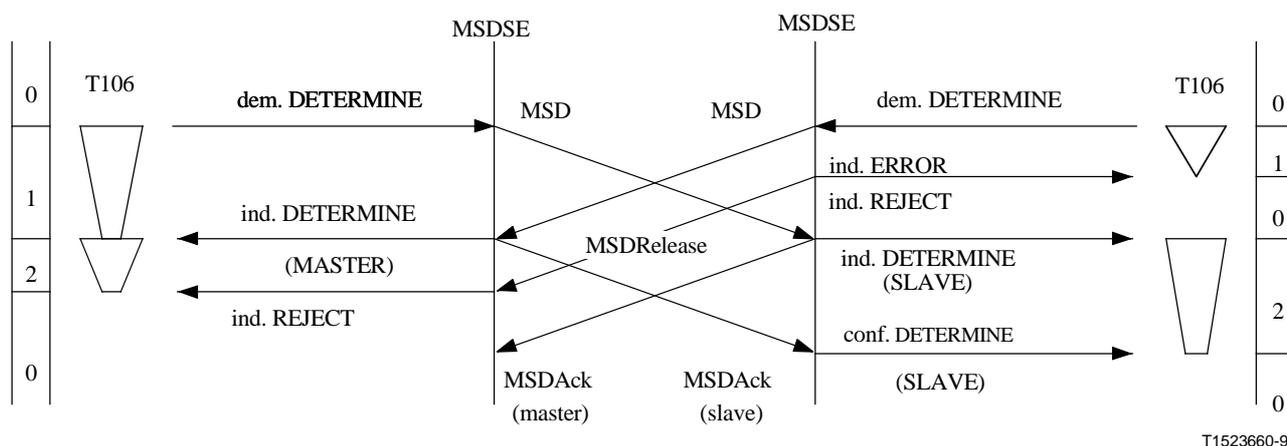


Figure II.2-8/H.245 – Choix du mode maître ou esclave – Procédures de choix simultanées avec expiration du temporisateur T106 de l'entité esclave

Dans la Figure II.2-9, le temporisateur distant T106 est arrivé à expiration au cours de l'état INCOMING AWAITING ACKNOWLEDGEMENT d'une procédure de détermination simultanée. Les deux terminaux connaissent leur statut. Le terminal de gauche peut recevoir et émettre des commandes. Cependant, le terminal de droite ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre de commandes qui dépendent de la connaissance du résultat du choix de statut. Le terminal de droite peut recevoir de telles commandes. Une deuxième procédure de choix de statut doit être lancée.

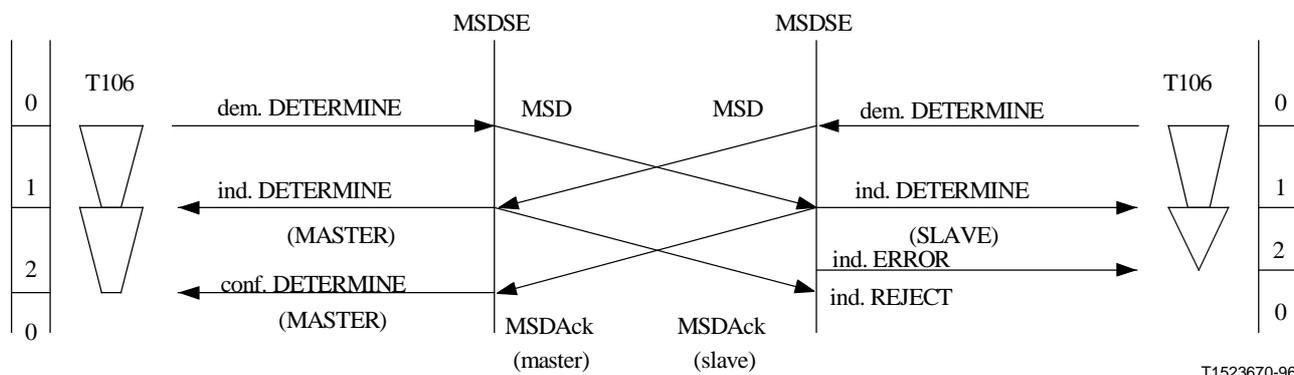


Figure II.2-9/H.245 – Choix du mode maître ou esclave – Procédures de détermination simultanée avec expiration du temporisateur T106 au cours de l'état INCOMING AWAITING ACKNOWLEDGEMENT

Dans la Figure II.2-10, un résultat indéterminé a été obtenu N100 fois. Dans ce cas, N100 = 3.

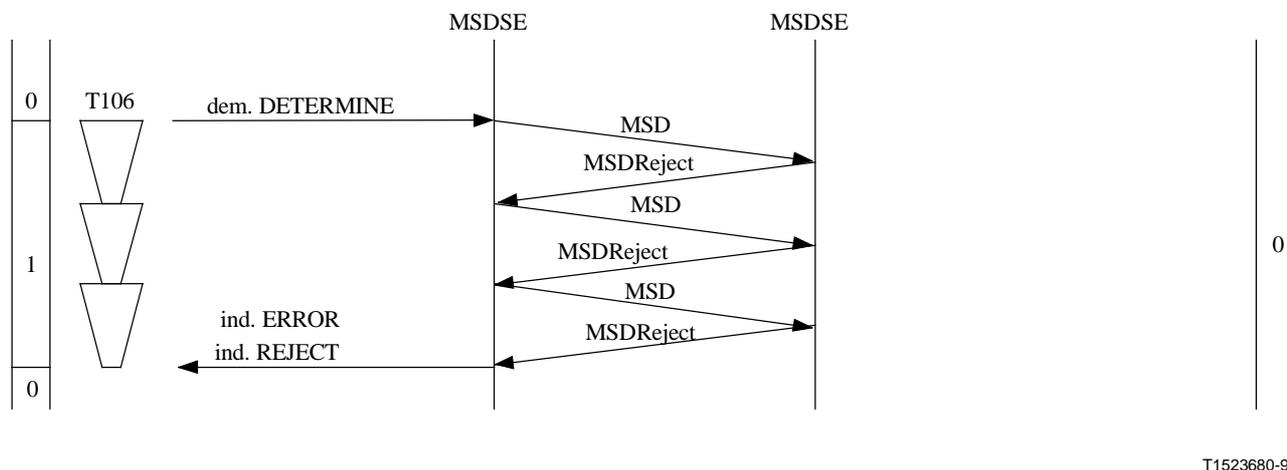
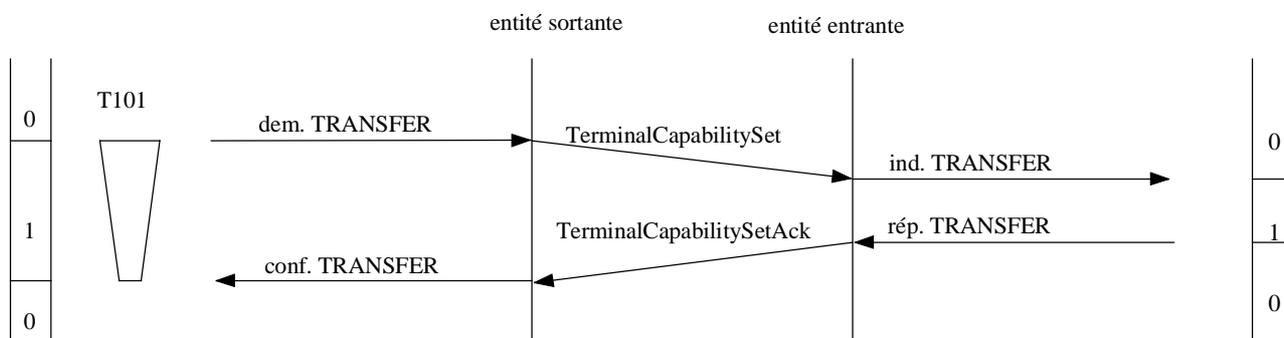


Figure II.2-10/H.245 – Choix du mode maître ou esclave – Résultat indéterminé avec N100 = 3

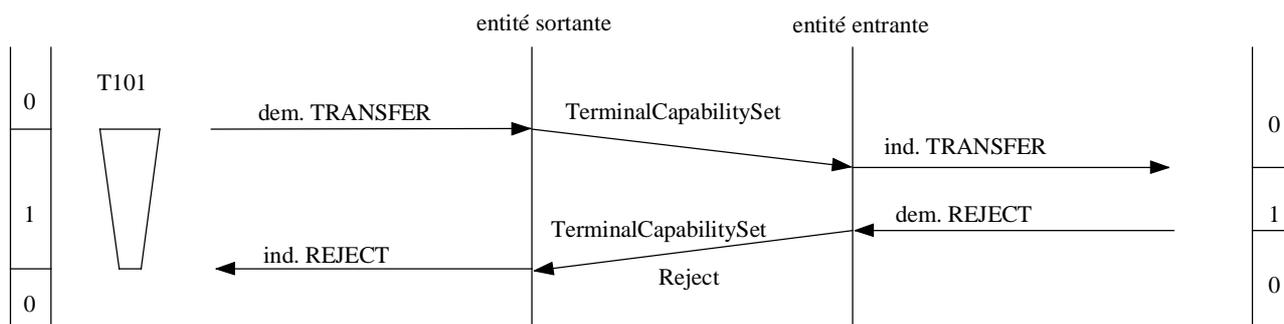
II.3 Entité de signalisation d'échange de capacités (CESE)

Les Figures II.3-1 à II.3-4 illustrent les procédures relatives à l'entité CESE. Les états IDLE et AWAITING RESPONSE sont respectivement désignés par "0" et "1".



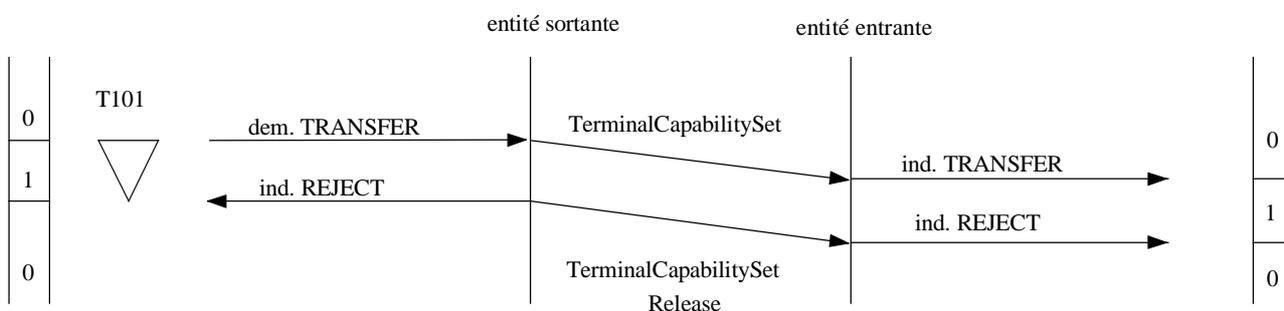
T1523690-96

Figure II.3-1/H.245 – Echange de capacités avec acceptation provenant de l'utilisateur de l'entité CESE homologue entrante



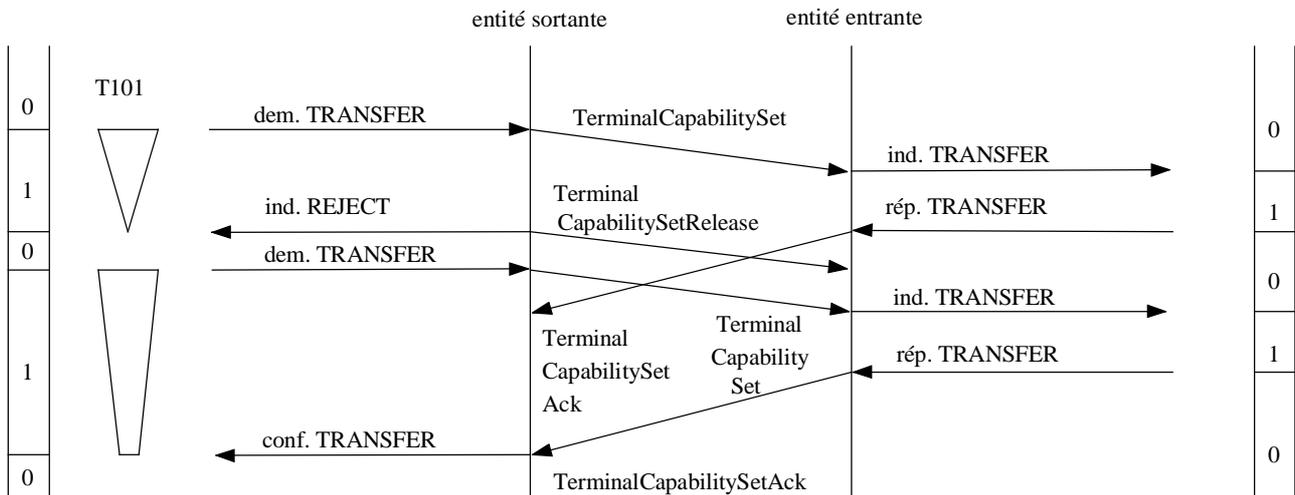
T1523700-96

Figure II.3-2/H.245 – Echange de capacités avec rejet provenant de l'utilisateur de l'entité CESE homologue entrante



T1523710-96

Figure II.3-3/H.245 – Echange de capacités avec expiration du temporisateur T101 – Le message TerminalCapabilitySetRelease message arrive dans l'entité CESE entrante avant la réponse provenant de l'utilisateur de l'entité CESE entrante

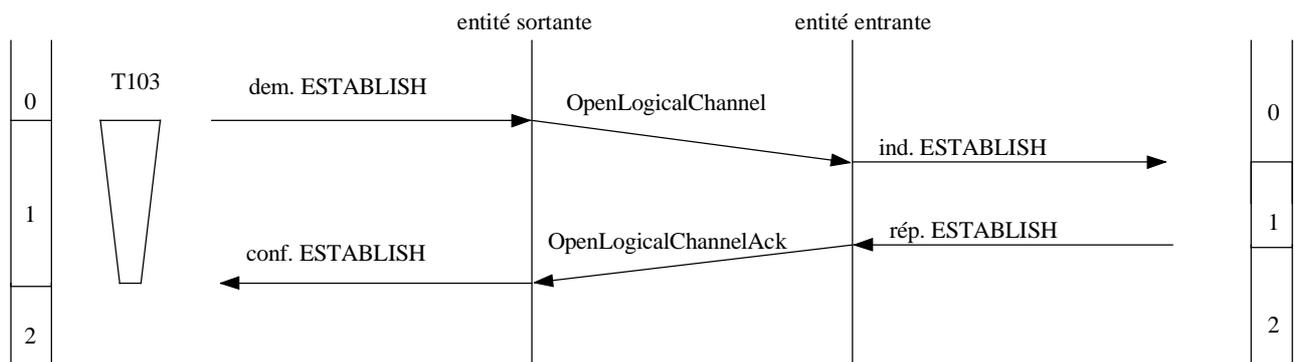


T1523720-96

Figure II.3-4/H.245 – Echange de capacités avec expiration du temporisateur T101 suivi d'un deuxième échange de capacités – Le message TerminalCapabilitySetRelease arrive dans l'entité CESE entrante après la réponse provenant de l'utilisateur de l'entité CESE entrante – Dans l'entité CESE sortante, le message TerminalCapabilitySetAck message, envoyé en réponse au premier message TerminalCapabilitySet, est omis – Seul le second échange de capacités a réussi

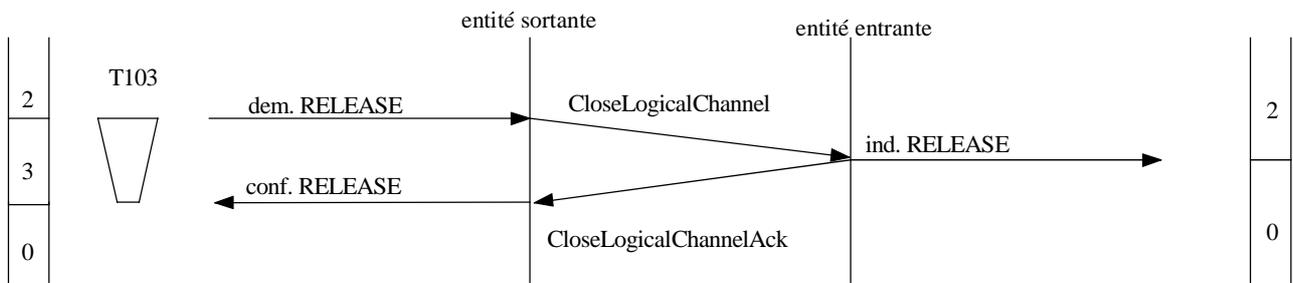
II.4 Entité de signalisation de la voie logique (LCSE)

Les Figures II.4-1 à II.4-7 illustrent les procédures relatives à l'entité LCSE. Les états suivants d'une entité LCSE sortante: RELEASED, AWAITING ESTABLISHMENT, ESTABLISHED, et AWAITING RELEASE sont respectivement désignés par "0", "1", "2", et "3". Les états suivants d'entité LCSE entrante: RELEASED, AWAITING ESTABLISHMENT, et ESTABLISHED, sont respectivement désignés par "0", "1", et "2".



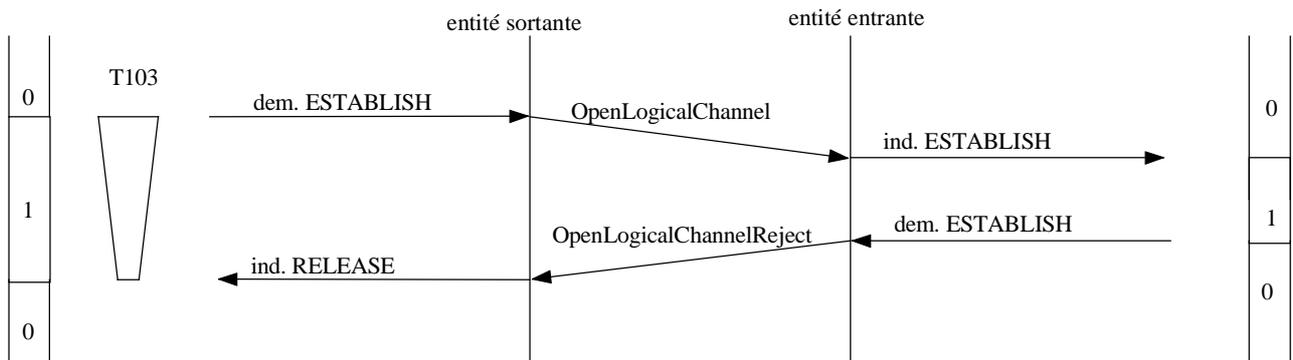
T1523730-96

Figure II.4-1/H.245 – Etablissement de la voie logique



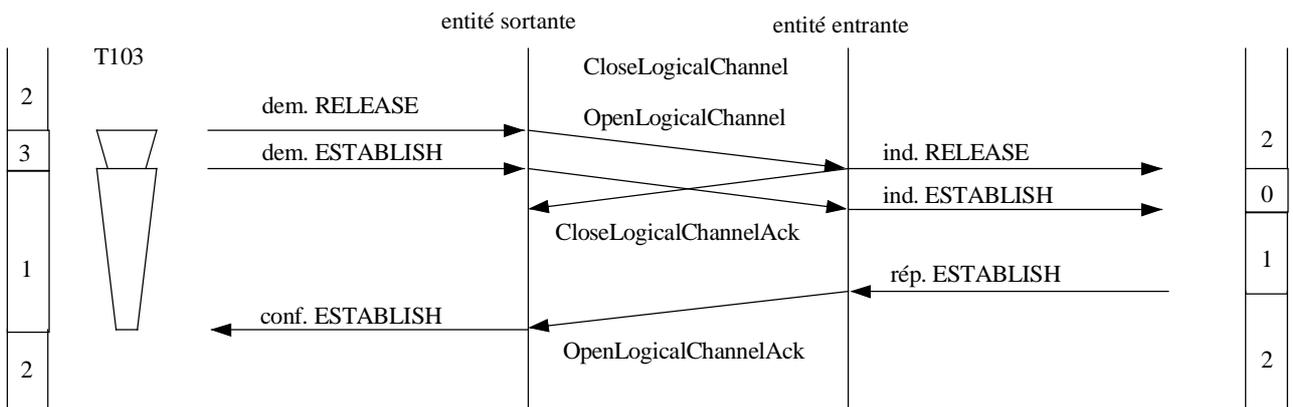
T1523740-96

Figure II.4-2/H.245 – Libération de la voie logique



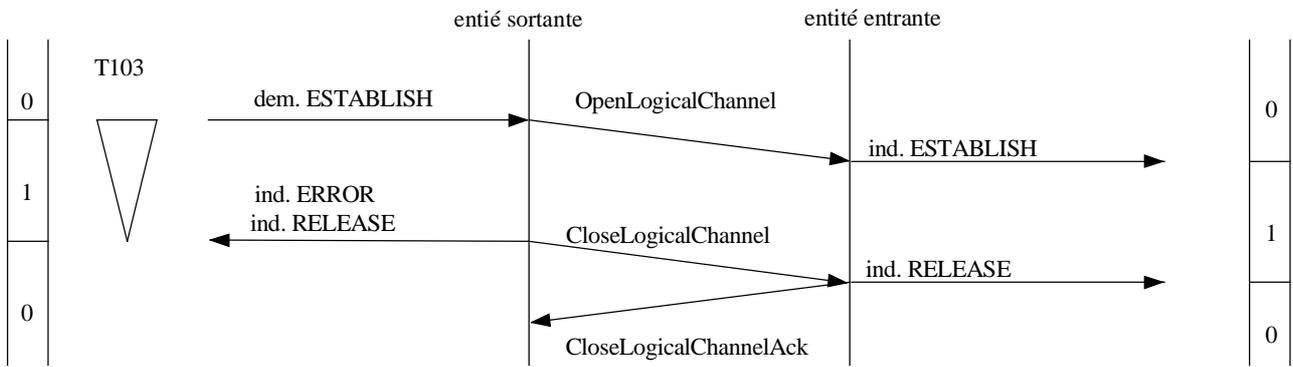
T1523750-96

Figure II.4-3/H.245 – Rejet d'établissement de la voie logique par l'utilisateur de l'entité LCSE homologue



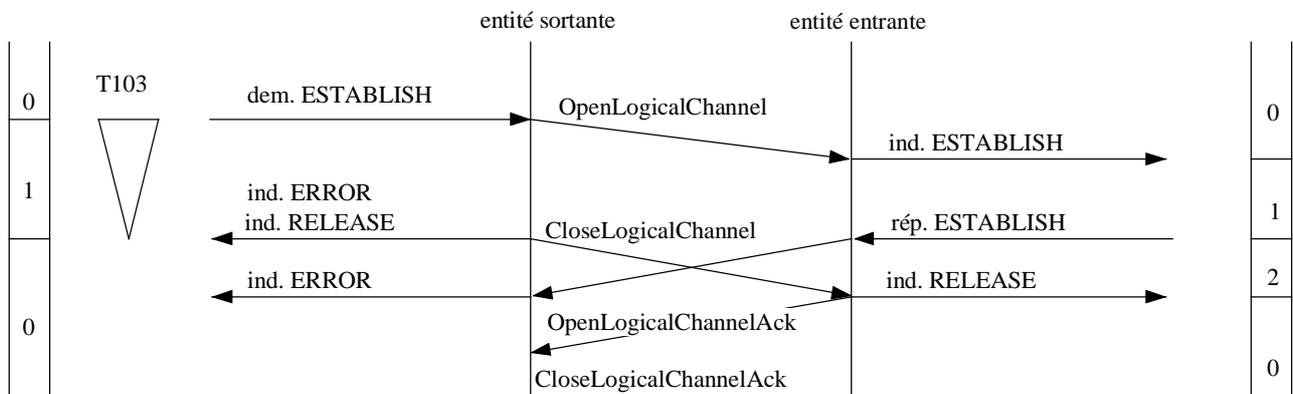
T1523760-96

Figure II.4-4/H.245 – Libération de voie logique suivie du rétablissement immédiat



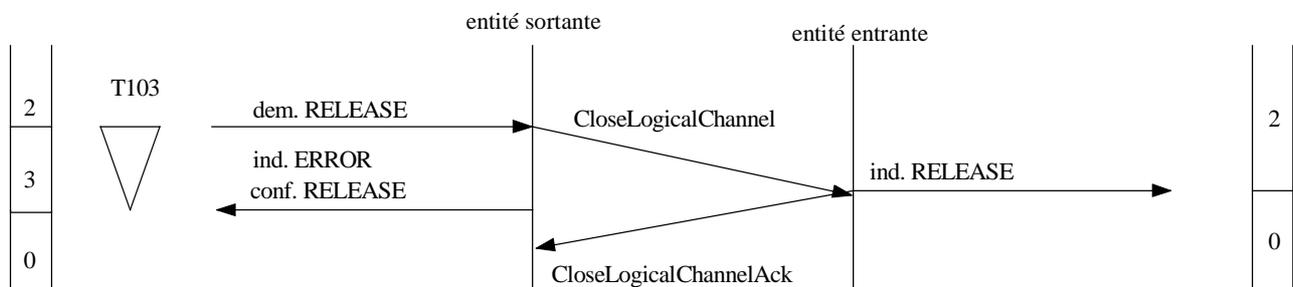
T1523770-96

Figure II.4-5/H.245 – Demande d’établissement de la voie logique avec expiration du temporisateur T103 en raison d’un retard de la réponse provenant de l’utilisateur d’entité LCSE homologue entrante



T1523780-96

Figure II.4-6/H.245 – Demande d’établissement de voie logique avec expiration du temporisateur T103 après la transmission du message OpenLogicalChannelAck dans l’entité LCSE entrante, mais avant la réception du message OpenLogicalChannelAck dans l’entité LCSE sortante

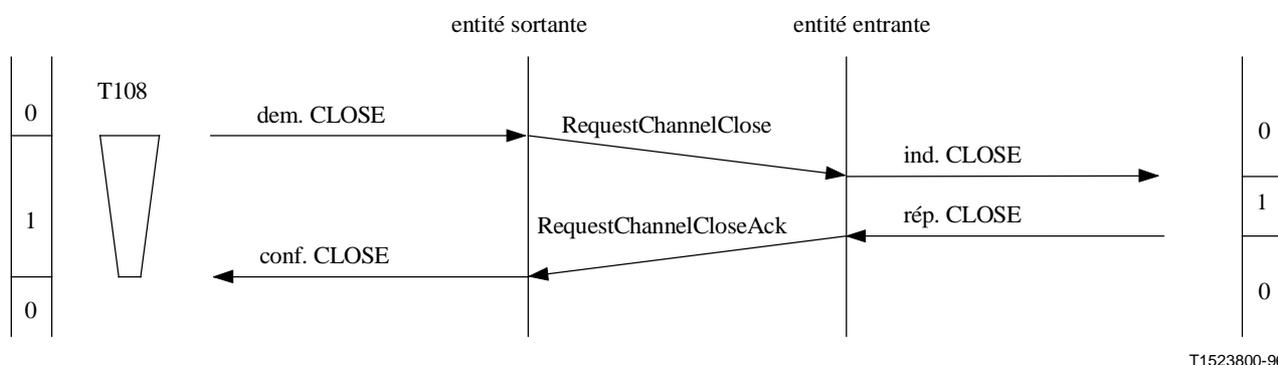


T1523790-96

Figure II.4-7/H.245 – Demande de libération de voie logique avec expiration du temporisateur T103

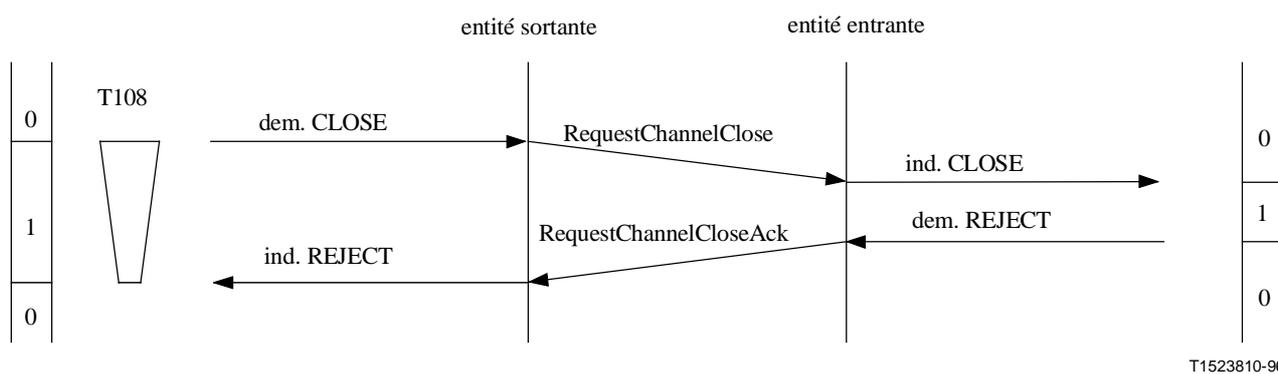
II.5 Entité de signalisation de fermeture de la voie logique (CLCSE)

Les Figures II.5-1 à II.5-4 illustrent les procédures relatives à l'entité CLCSE. Les états IDLE et AWAITING RESPONSE sont désignés par "0" et "1" respectivement.



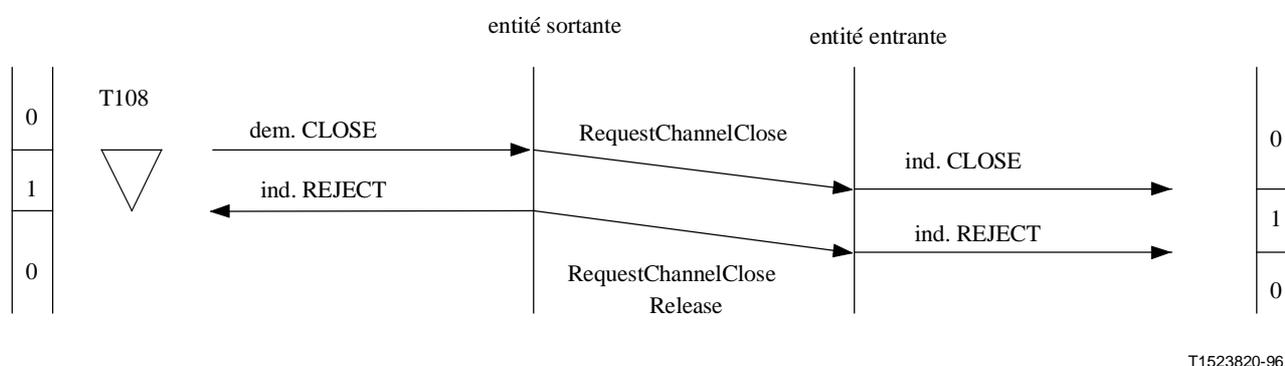
T1523800-96

Figure II.5-1/H.245 – Demande de fermeture de voie logique



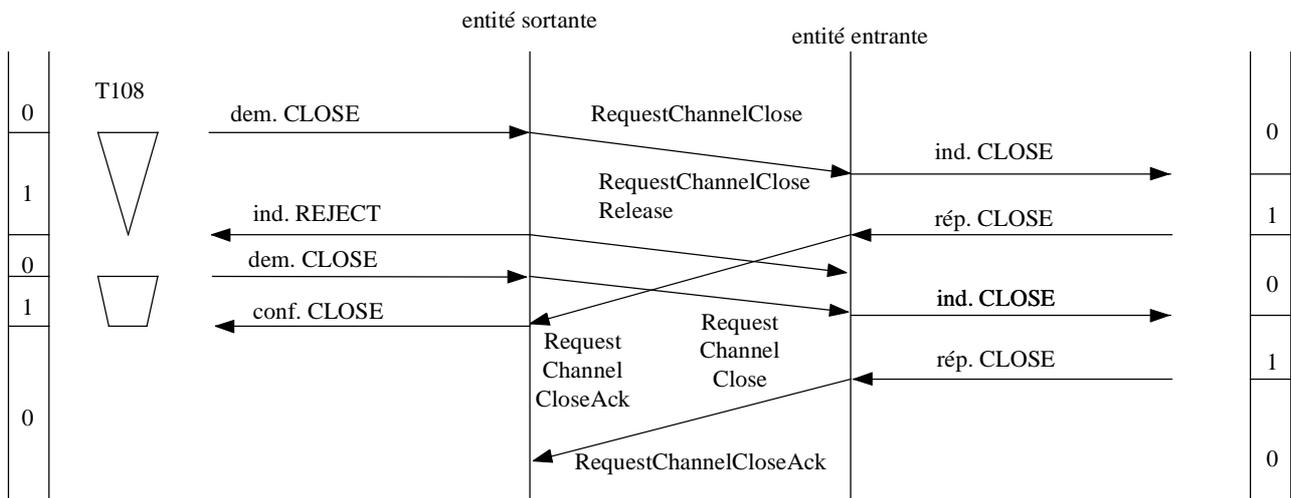
T1523810-96

Figure II.5-2/H.245 – Demande de fermeture de voie logique avec rejet provenant de l'utilisateur de l'entité CLCSE homologue entrante



T1523820-96

Figure II.5-3/H.245 – Demande de fermeture de voie logique avec expiration du temporisateur T108 – Le message RequestChannelCloseRelease arrive dans l'entité CLCSE entrante avant la réponse provenant de l'utilisateur de l'entité CLCSE entrante

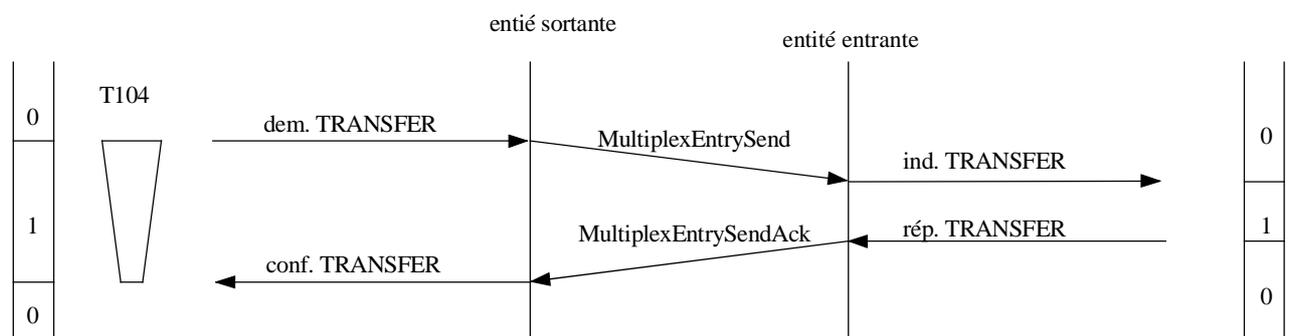


T1523830-96

Figure II.5-4/H.245 – Demande de fermeture de voie logique avec expiration du temporisateur T108, suivie d’une deuxième demande de fermeture de voie logique – La demande de fermeture de voie est confirmée dès la réception du premier message RequestChannelClose

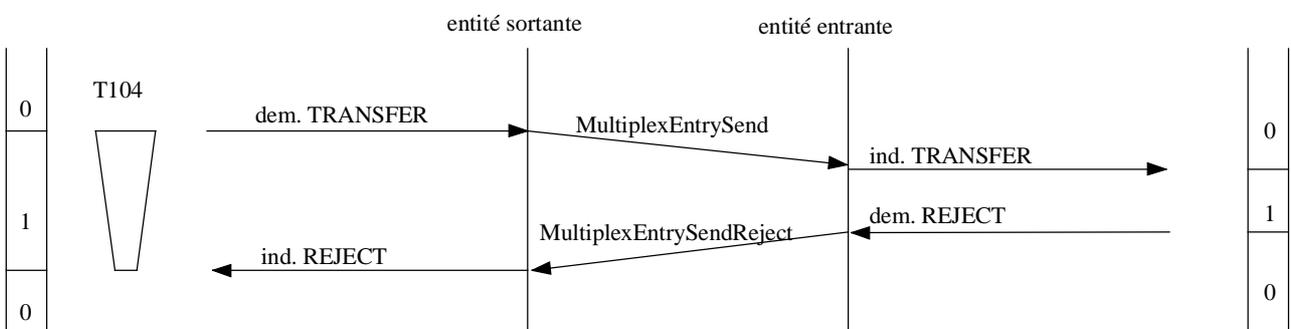
II.6 Entité de signalisation du tableau de multiplexage (MTSE)

Les Figures II.6-1 à II.6-5 illustrent les procédures relatives à l'entité MTSE. Les états IDLE et AWAITING RESPONSE sont respectivement désignés par "0" et "1".



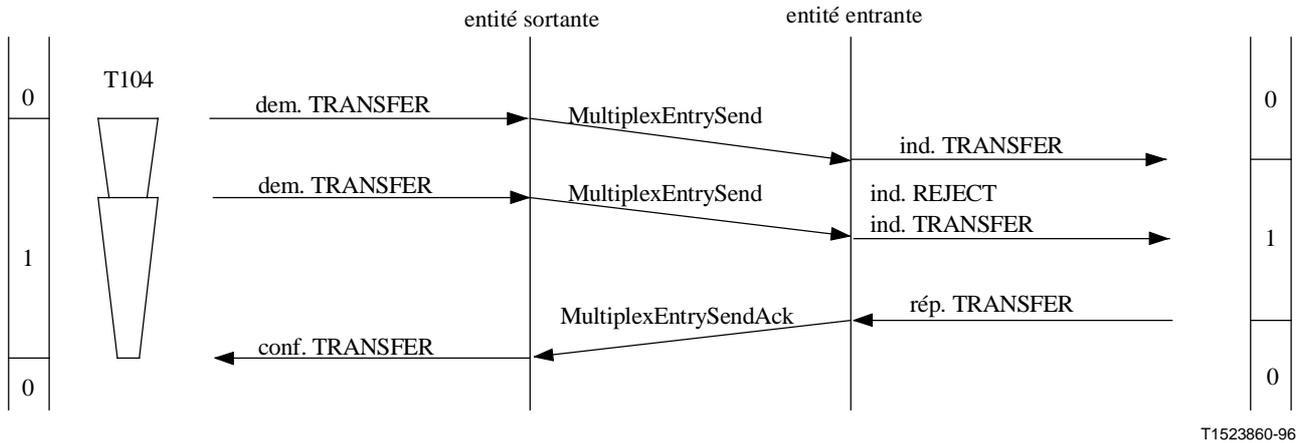
T1523840-96

Figure II.6-1/H.245 – Aboutissement de la demande d’envoi du tableau de multiplexage



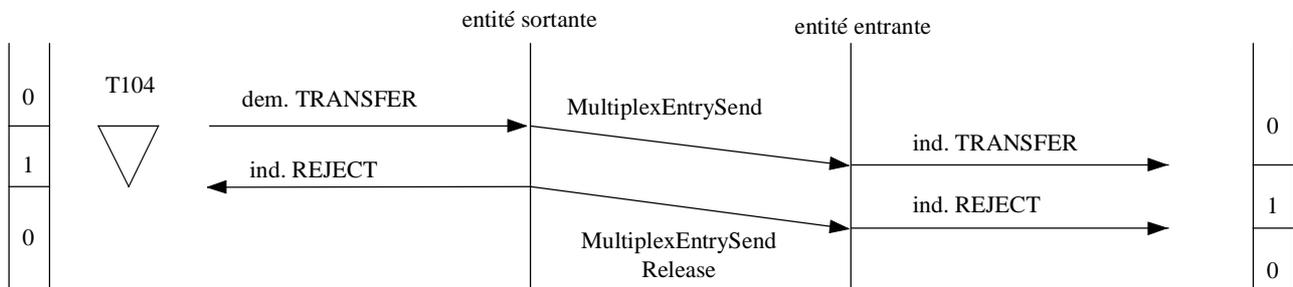
T1523850-96

Figure II.6-2/H.245 – Demande d’envoi du tableau de multiplexage avec rejet provenant de l’utilisateur de l’entité MTSE homologue



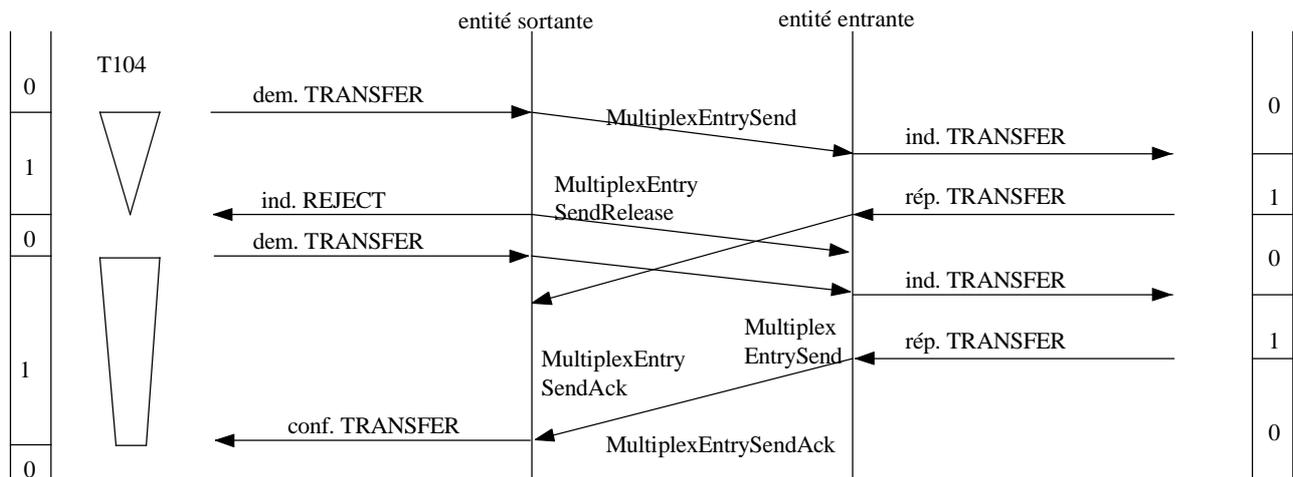
T1523860-96

Figure II.6-3/H.245 – Demande d’envoi du tableau de multiplexage avec une deuxième demande d’envoi de tableau de multiplexage avant l’acquittement de la première demande – La première demande n’a pas abouti



T1523870-96

Figure II.6-4/H.245 – Demande d’envoi de tableau de multiplexage avec expiration du temporisateur T104

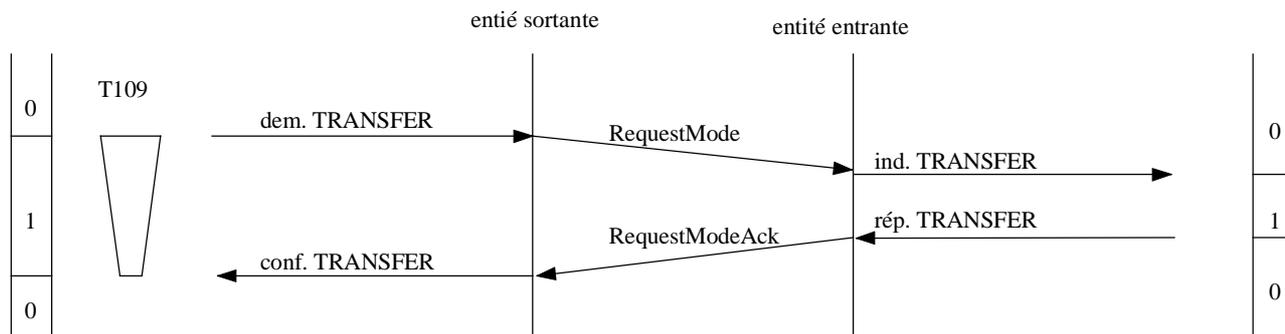


T1523880-96

Figure II.6-5/H.245 – Demande de table de multiplexage avec expiration du temporisateur T104 suivie d’une deuxième demande d’envoi de table de multiplexage – Le premier message MultiplexEntrySendAck est ignoré dans l’entité MTSE sortante – Seule la deuxième demande a abouti

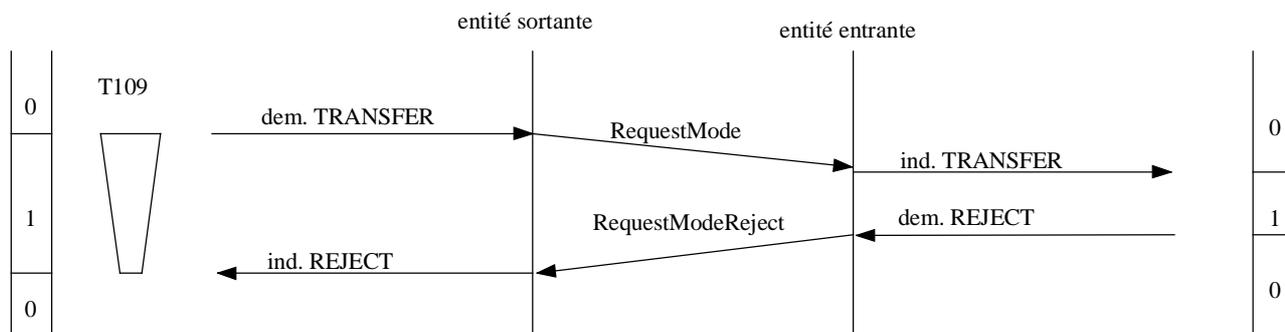
II.7 Entité de signalisation de demande de mode (MRSE)

Les Figures II.7-1 à II.7-5 illustrent les échanges entre entités MRSE. Les états IDLE et AWAITING RESPONSE states sont désignés par "0" et "1" respectivement.



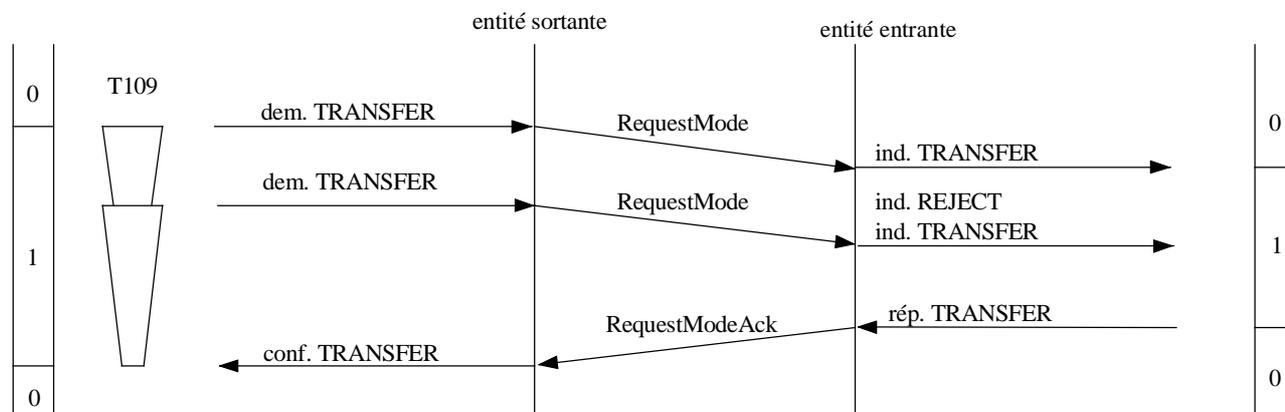
T1523890-96

Figure II.7-1/H.245 – Aboutissement de demande de mode



T1523900-96

Figure II.7-2/H.245 – Demande de mode avec rejet provenant de l'utilisateur de l'entité MRSE homologue



T1523910-96

Figure II.7-3/H.245 – Demande de mode avec deuxième demande de mode avant acquittement de la première demande – La première demande a échoué

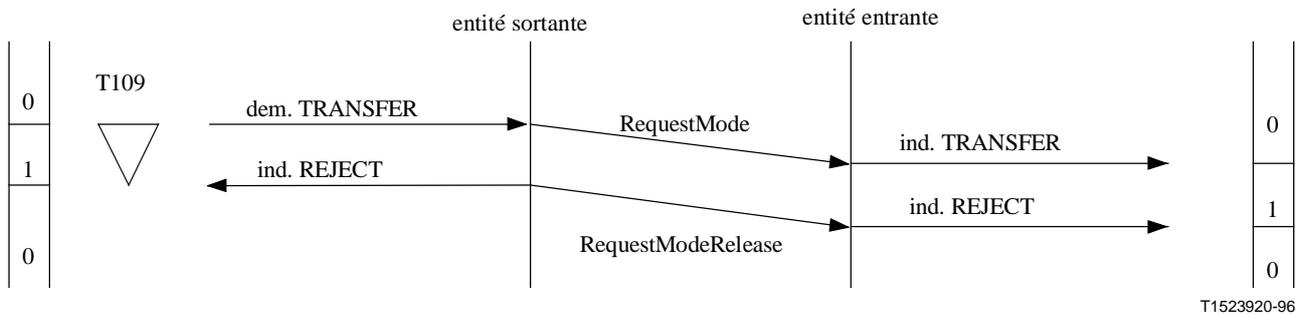


Figure II.7-4/H.245 – Demande de mode non suivie d’effet avec expiration du temporisateur T109 – La demande de mode a échoué

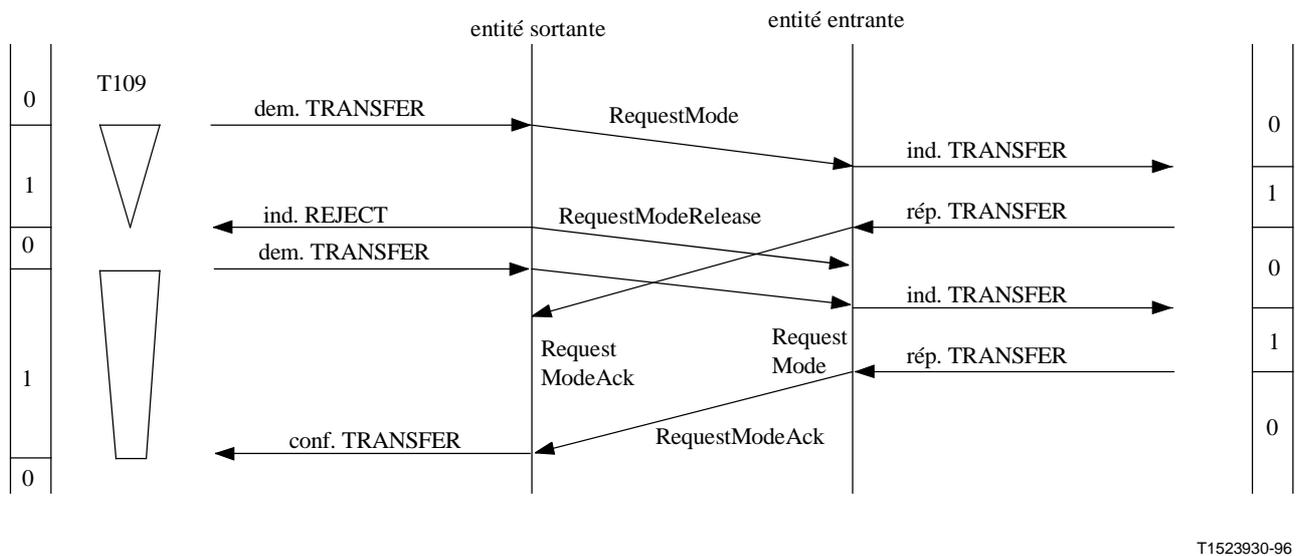


Figure II.7-5/H.245 – Demande de mode avec expiration du temporisateur T109, suivie d’une deuxième demande de mode – Le premier message RequestModeAck est omis dans l’entité MRSE sortante – Seule la deuxième demande a réussi

II.8 Entité de signalisation de temps de propagation aller et retour (RTDSE)

Les Figures II.8-1 à II.8-4 illustrent les procédures d’entité RTDSE. Les états IDLE et AWAITING RESPONSE de l’entité sont désignés respectivement par "0" et "1".

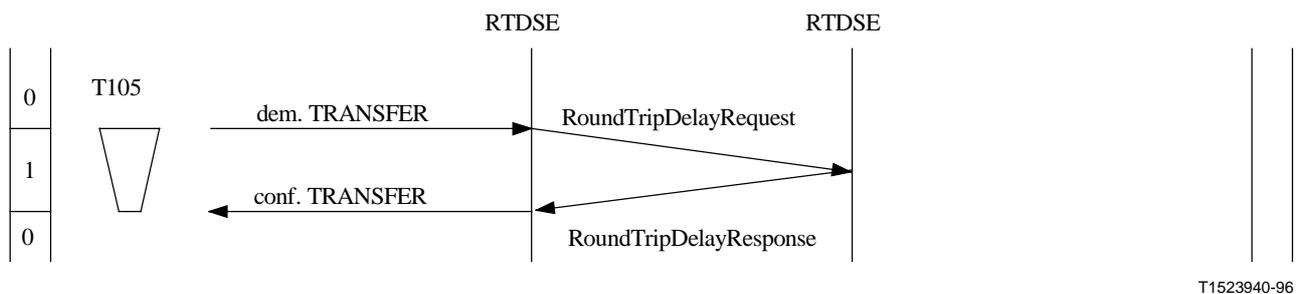


Figure II.8-1/H.245 – Procédure de détermination du temps de propagation aller et retour

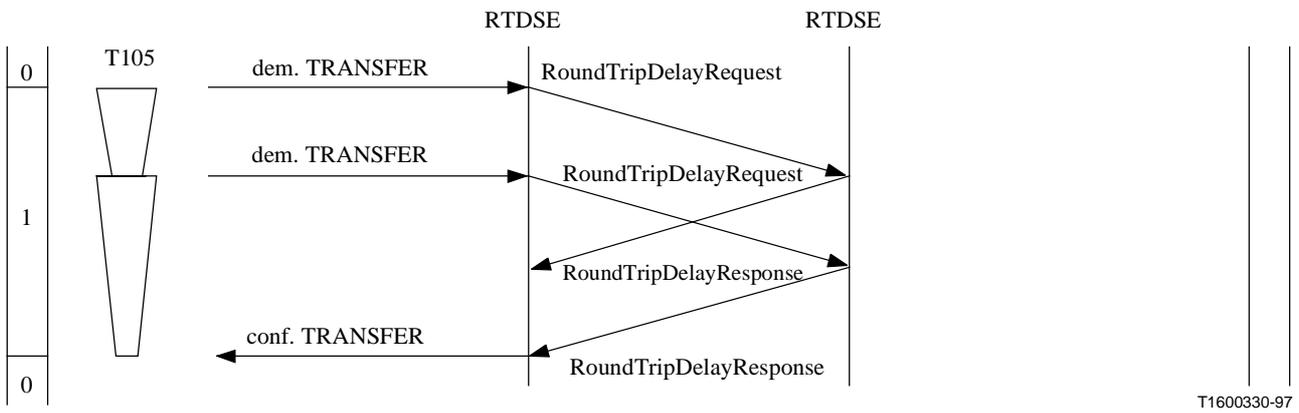


Figure II.8-2/H.245 – Procédure de détermination du temps de propagation aller et retour avec procédure de détermination du temps de propagation aller et retour antérieure non acquittée et en attente

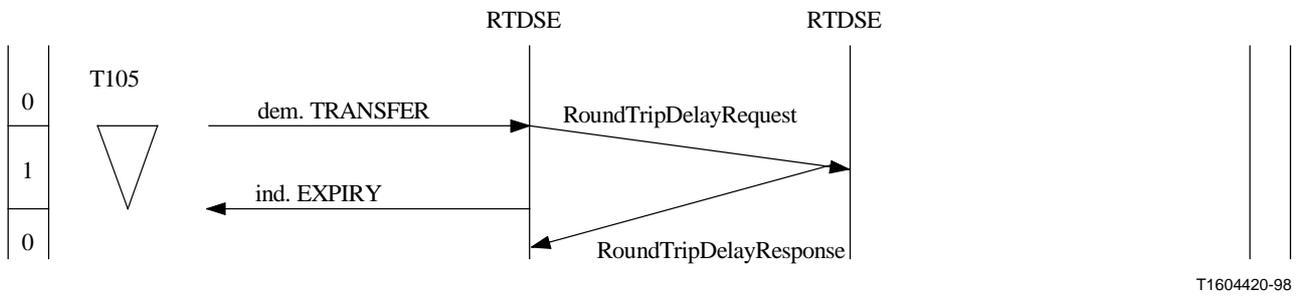


Figure II.8-3/H.245 – Procédure de détermination du temps de propagation aller et retour avec expiration du temporisateur T105

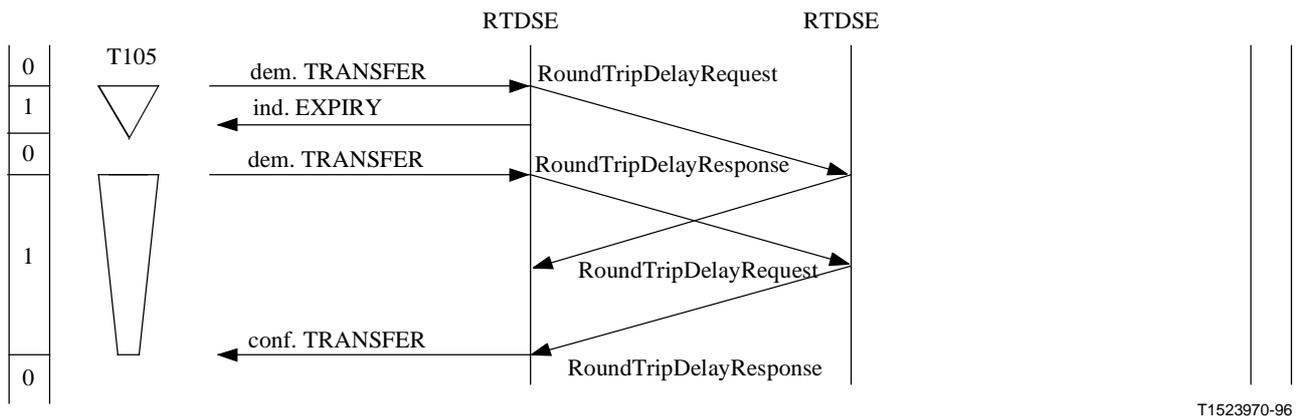


Figure II.8-4/H.245 – Procédure de détermination du temps de propagation aller et retour avec expiration du temporisateur T105, suivie d’une deuxième procédure de détermination du temps de propagation aller et retour – Le message RoundTripDelayResponse issu de la première procédure arrive pendant la deuxième procédure et est omis

II.9 Entité de signalisation de voie logique bidirectionnelle (B-LCSE)

Les Figures II.9-1 à II.9-7 illustrent les procédures des entités B-LCSE. Les états RELEASED, AWAITING ESTABLISHMENT, ESTABLISHED, et AWAITING RELEASE de l'entité B-LCSE sortante sont désignés par "0", "1", "2" et "3" respectivement. Les états RELEASED, AWAITING ESTABLISHMENT, AWAITING CONFIRMATION et ESTABLISHED de l'entité B-LCSE entrante sont désignés par "0", "1", "2" et "3" respectivement.

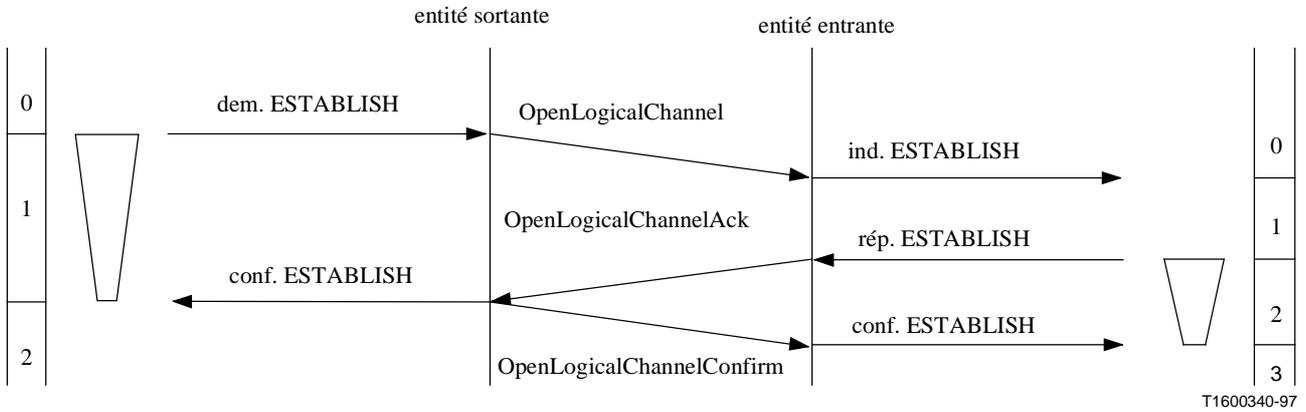


Figure II.9-1/H.245 – Etablissement des voies logiques bidirectionnelles

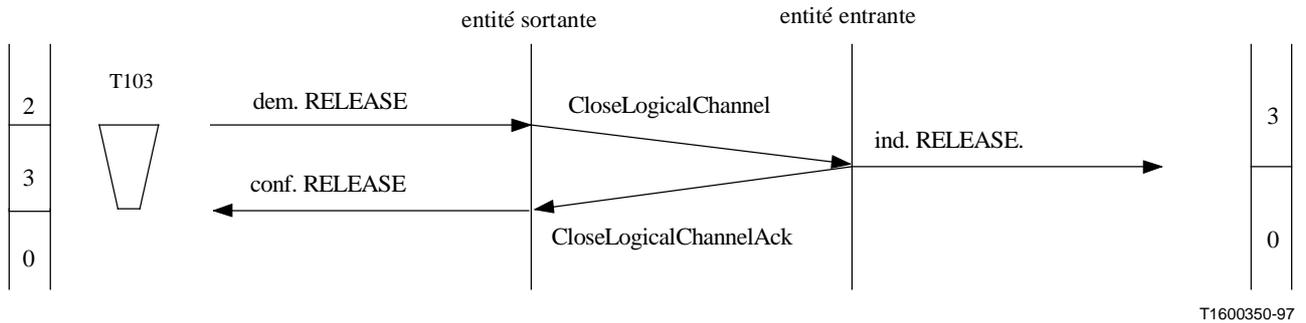


Figure II.9-2/H.245 – Libération de la voie logique bidirectionnelle

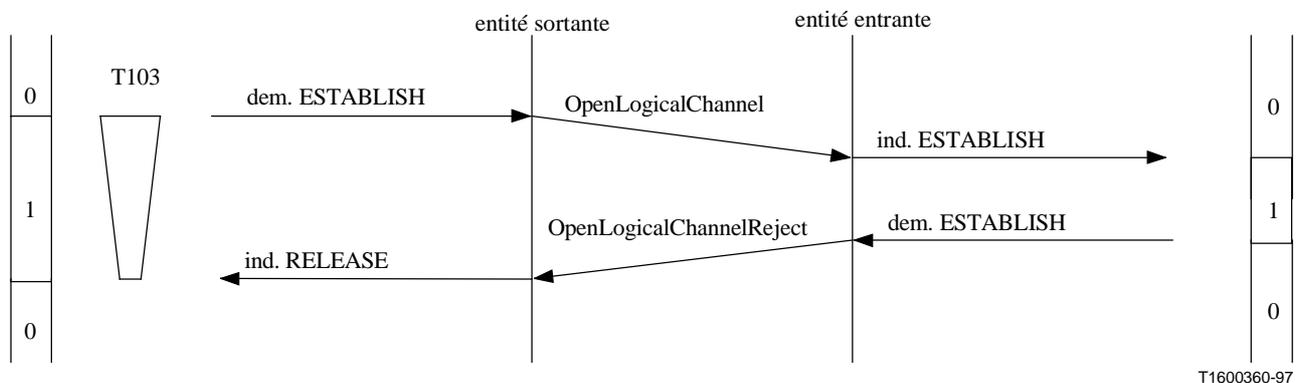


Figure II.9-3/H.245 – Rejet d'établissement de la voie logique bidirectionnelle par l'utilisateur de l'entité B-LCSE homologue

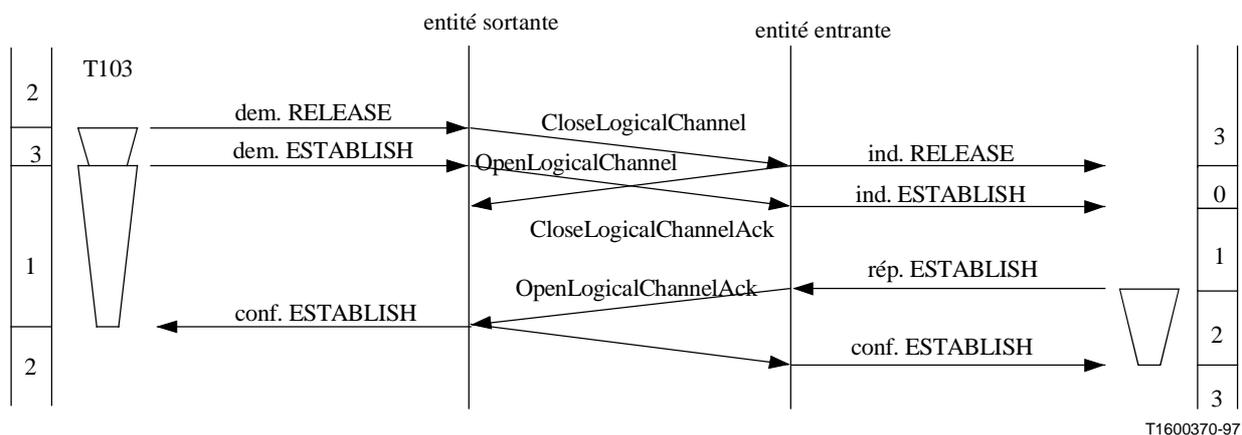


Figure II.9-4/H.245 – Libération de voie logique bidirectionnelle suivie par un rétablissement immédiat

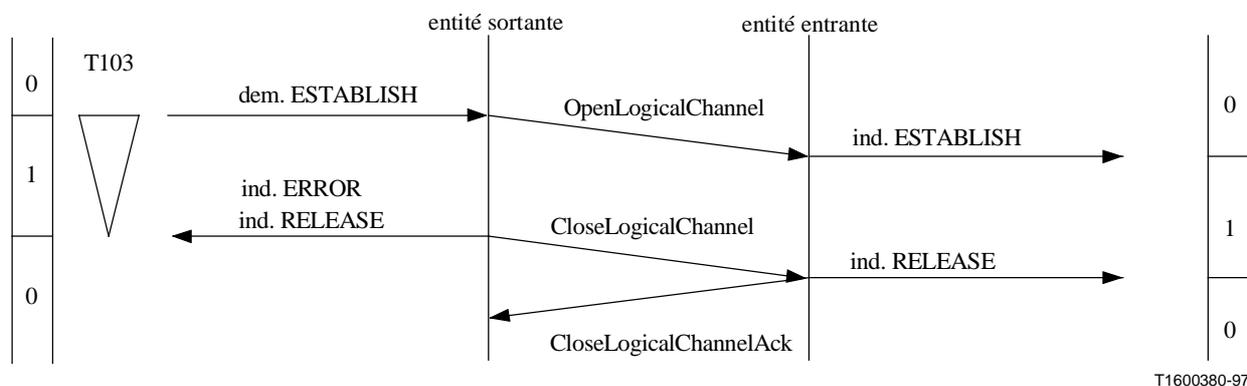


Figure II.9-5/H.245 – Demande d'établissement de voie logique bidirectionnelle avec fin de temporisation T103 du côté émetteur par suite d'une réaction lente de l'utilisateur de l'entité B-LCSE entrante homologue

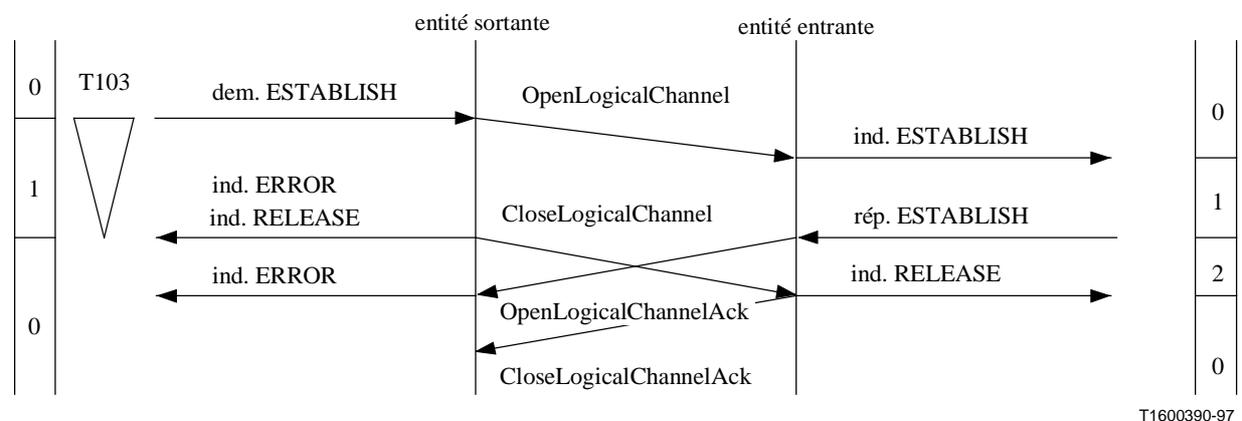


Figure II.9-6/H.245 – Demande d'établissement de voie logique bidirectionnelle avec fin de temporisation T103 du côté sortant – Le temporisateur T013 du côté sortant est arrivé en fin de temporisation après la transmission du message OpenLogicalChannelAck dans l'entité B-LCSE entrante, mais avant la réception du message OpenLogicalChannelAck dans l'entité B-LCSE sortante

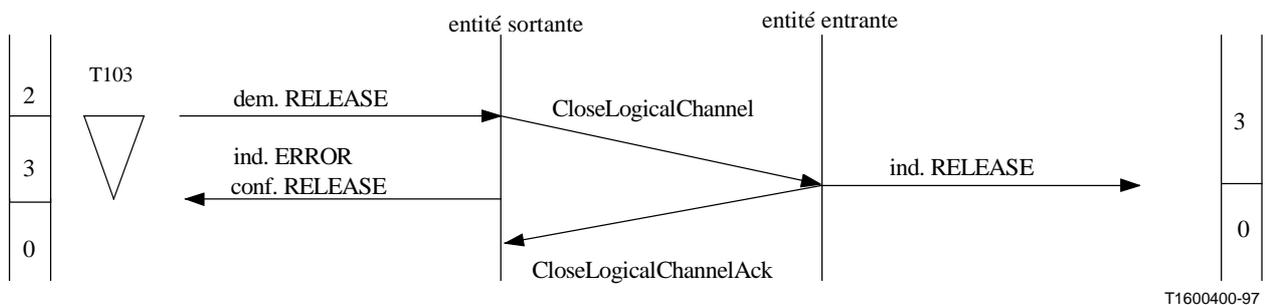


Figure II.9-7/H.245 – Demande de libération de voie logique bidirectionnelle avec fin de temporisation T103 du côté sortant

APPENDICE III

Récapitulatif des temporisateurs et des compteurs utilisés dans les procédures

Le présent appendice énumère les temporisateurs et les compteurs spécifiés dans le paragraphe 8.

La présente Recommandation ne définit pas les valeurs à mettre dans ces temporisateurs. Les valeurs peuvent être définies dans d'autres Recommandations telles que H.310, H.323 et H.324.

III.1 Temporisateurs

Le Tableau III.1 récapitule les temporisateurs spécifiés dans la présente Recommandation.

Tableau III.1/ H.245 – Temporisateurs de procédure

Temporisateur	Procédure	Définition
T106	choix du mode maître ou esclave	ce temporisateur est utilisé dans l'état OUTGOING AWAITING RESPONSE et pendant l'état INCOMING AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message d'acquiescement ne peut être reçu.
T101	échange de capacités	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message TerminalCapabilitySetAck ou TerminalCapabilitySetReject ne peut être reçu.
T103	signalisation de voie logique monodirectionnelle ou de voie logique bidirectionnelle	ce temporisateur est utilisé dans les états AWAITING ESTABLISHMENT et AWAITING RELEASE. Il spécifie la durée maximale pendant laquelle aucun message OpenLogicalChannelAck, OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu.
T108	SEND Logical Channel (émission sur la voie logique)	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RequestMultiplexEntryAck ou RequestMultiplexEntryReject ne peut être reçu.
T104	table de multiplexage H.223	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message MultiplexEntrySendAck ou MultiplexEntrySendReject ne peut être reçu.

Tableau III.1/ H.245 – Temporisateurs de procédure (fin)

Temporisateur	Procédure	Définition
T109	demande de mode	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RequestModeAck ou RequestModeReject ne peut être reçu.
T105	Round Trip Delay	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RoundTripDelayResponse ne peut être reçu.
T107	Request Multiplex Entry	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RequestMultiplexEntryAck ou RequestMultiplexEntryReject ne peut être reçu.
T102	Maintenance Loop	ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message MaintenanceLoopAck ou MaintenanceLoopReject ne peut être reçu.

III.2 Compteurs

Le Tableau III.2 indique les compteurs spécifiés dans la présente Recommandation.

Tableau III.2/ H.245 – Compteurs de procédure

Temporisateur	Procédure	Définition
N100	choix du mode maître ou esclave	ce compteur spécifie le nombre maximal d'envois de messages MasterSlaveDetermination au cours de l'état OUTGOING AWAITING RESPONSE.

APPENDICE IV

Extensions à la Recommandation H.245

La présente Recommandation est un "document évolutif" utilisé par de nombreuses Recommandations sur les systèmes, notamment H.310, H.323, H.324 et V.70, pour lesquelles des extensions autorisant une compatibilité ascendante sont susceptibles d'être retenues à chaque réunion de la Commission d'études 16 de l'UIT-T. Le présent appendice explique les procédures qui devraient être utilisées pour apporter des extensions à la présente Recommandation.

Pour une période considérée, une seule syntaxe H.245 est en vigueur. Aucune autre Recommandation de l'UIT-T ne devrait inclure des variantes de syntaxe H.245 de façon normative.

Les demandes d'extensions à la présente Recommandation devraient être soumises, sous forme de contribution blanche ou de document de liaison, à la Commission d'études 16 de l'UIT-T, et une copie sera envoyée dès que possible au rapporteur et au rédacteur de la présente Recommandation. De telles demandes devraient inclure:

- 1) les spécifications générales de la syntaxe qui doit être élaborée par le rédacteur de la présente Recommandation ou la syntaxe proposée par référence à la version actuellement approuvée;
- 2) la sémantique proposée pour le paragraphe 7;
- 3) les procédures proposées pour le paragraphe 8 si de nouvelles procédures sont demandées.

Toutes les extensions à la présente Recommandation doivent permettre une compatibilité ascendante avec ses versions antérieures. La syntaxe, la sémantique et les procédures ne peuvent pas être modifiées. La signification de la syntaxe préexistante ne peut pas être modifiée.

Les demandes devraient être soumises dès que possible pour tenir compte des compléments apportés par les experts de la présente Recommandation dans la Commission d'études 16. Il est entendu que la syntaxe exacte demandée peut être modifiée en raison de:

- 1) la vérification de la syntaxe correcte ASN.1;
- 2) l'harmonisation avec d'autres demandes concomitantes pour des compléments à la présente Recommandation;
- 3) la compatibilité ascendante avec des versions préexistantes de la présente Recommandation;
- 4) l'avis des experts pour l'ajout de nouvelles fonctions apportées à la structure actuelle de H.245.

Le rédacteur de la présente Recommandation étudiera toutes les demandes d'ajout et proposera le texte final de ses versions complètes en vue de l'approbation par la Commission d'études 16 selon la procédure prévue dans la Résolution 1. Lors de l'approbation de chaque version de la présente Recommandation par la Commission d'études, son numéro de version dans **protocolIdentifieur** sera augmenté de un pour identifier la nouvelle version.

Il convient de remarquer que la Commission d'études 16 a l'intention de n'accepter que des compléments harmonisés avec la présente Recommandation qui émanent du rédacteur de cette même Recommandation.

SÉRIES 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
Série T	Terminaux des services télématiques
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

12751

Imprimé en Suisse

Genève, 1998