

Remplacée par une version plus récente



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.245

(03/96)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructures des services audiovisuels – Procédures de
communication

**Protocole de commande pour communications
multimédias**

Recommandation UIT-T H.245
Remplacée par une version plus récente

(Antérieurement Recommandation du CCITT)

Remplacée par une version plus récente

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
INFRASTRUCTURES DES SERVICES AUDIOVISUELS	H.200–H.399
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.

Remplacée par une version plus récente

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (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 (Helsinki, 1^{er}-12 mars 1993).

La Recommandation UIT-T H.245, que l'on doit à la Commission d'études 15 (1993-1996) de l'UIT-T, a été approuvée le 19 mars 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

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 de télécommunications.

© UIT 1997

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.

Remplacée par une version plus récente

TABLE DES MATIÈRES

	<i>Page</i>
1	Domaine d'application..... 1
2	Références normatives..... 1
3	Définitions 3
4	Abréviations 4
5	Généralités 5
5.1	Choix du mode maître ou esclave 5
5.2	Echange de capacités..... 6
5.3	Procédures de signalisation du canal logique..... 6
5.4	Demande de fermeture de canal logique émanant du terminal de réception..... 7
5.5	Modification d'entrée de tableau de multiplexage de la Recommandation H.223 7
5.6	Demande de mode pour les communications audiovisuelles et les transmissions de données 7
5.7	Détermination du temps de propagation aller et retour 7
5.8	Boucles de maintenance 7
5.9	Instructions et indications d'état 7
6	Messages: syntaxe 7
7	Messages – Définitions sémantiques 33
7.1	Messages de choix du mode maître ou esclave 33
7.1.1	Choix du mode maître ou esclave 33
7.1.2	Acquittement du choix du mode maître ou esclave..... 33
7.1.3	Rejet du choix du mode maître ou esclave 33
7.1.4	Abandon du choix du mode maître ou esclave..... 33
7.2	Messages de capacités des terminaux 33
7.2.1	Présentation générale..... 33
7.2.2	Ensemble de capacités de terminaux 34
7.2.3	Acquittement de l'ensemble de capacités des terminaux..... 42
7.2.4	Refus de l'ensemble de capacités des terminaux 42
7.2.5	Libération sur temporisation de l'ensemble de capacités des terminaux 43
7.3	Messages de signalisation sur voie logique..... 43
7.3.1	Ouverture de la voie logique 43
7.3.2	Acquittement de l'ouverture de la voie logique 45
7.3.3	Rejet de l'ouverture de la voie logique 45
7.3.4	Confirmation de l'ouverture de la voie logique 45
7.3.5	Fermeture de la voie logique 46
7.3.6	Acquittement de la fermeture de la voie logique..... 46
7.3.7	Demande de fermeture de voie..... 46
7.3.8	Acquittement de demande de fermeture de voie 47
7.3.9	Refus de demande de fermeture de voie..... 47
7.3.10	Libération de demande de fermeture de voie 47
7.4	Messages de signalisation des tableaux de multiplexage 47
7.4.1	Envoi d'entrée de multiplexage..... 47
7.4.2	Acquittement d'envoi d'entrée de multiplexage 47
7.4.3	Refus d'envoi d'entrée de multiplexage 48
7.4.4	Echec du message d'envoi d'entrée de multiplexage 48
7.5	Messages de signalisation de demande de tableau de multiplexage 48
7.5.1	Demande d'entrée de multiplexage..... 48
7.5.2	Acquittement de demande d'entrée de multiplexage 48
7.5.3	Rejet de demande d'entrée de multiplexage..... 48
7.5.4	Abandon de la demande d'entrée de multiplexage 49

Remplacée par une version plus récente

Page

7.6	Messages relatifs au mode demande	49
7.6.1	Mode demande	49
7.6.2	Acquittement de mode demande	51
7.6.3	Refus de mode demande.....	51
7.6.4	Libération de mode demande	52
7.7	Messages liés au temps de propagation aller et retour	52
7.7.1	Demande de temps de propagation aller et retour	52
7.7.2	Réponse sur le temps de propagation aller et retour.....	52
7.8	Messages de boucle de maintenance.....	52
7.8.1	Demande de boucle de maintenance	52
7.8.2	Acquittement de boucle de maintenance.....	52
7.8.3	Rejet de boucle de maintenance	52
7.8.4	Commande «arrêt» de la boucle de maintenance	53
7.9	Instructions	53
7.9.1	Envoi de l'ensemble de capacités du terminal	53
7.9.2	Chiffrement	53
7.9.3	Contrôle de flux.....	53
7.9.4	Fin de session	54
7.9.5	Instructions diverses	54
7.10	Indications d'état	55
7.10.1	Fonction non disponible	55
7.10.2	Indications d'état diverses.....	55
7.10.3	Indication de gigue	56
7.10.4	Indication de décalage temporel selon H.223.....	57
7.10.5	Indication d'état de nouvelle voie virtuelle ATM.....	57
7.10.6	Données de l'utilisateur	57
8	Procédures	57
8.1	Introduction.....	57
8.1.1	Méthode de spécification.....	58
8.1.2	Communication entre l'entité de protocole et l'utilisateur de protocole.....	58
8.1.3	Communication entre entités homologues	58
8.1.4	Diagrammes SDL	58
8.1.5	Langage SDL clé.....	59
8.2	Procédures de choix du mode maître ou esclave.....	60
8.2.1	Introduction	60
8.2.2	Communication entre l'entité MSDSE et l'utilisateur de l'entité MSDSE	61
8.2.3	Communication entre les entités MSDSE homologues.....	62
8.2.4	Procédures de l'entité MSDSE	63
8.3	Procédures d'échange de capacités.....	65
8.3.1	Introduction	65
8.3.2	Communication entre l'entité CESE et l'utilisateur de l'entité CESE	70
8.3.3	Communication entre les entités CESE homologues	72
8.3.4	Procédures de l'entité CESE.....	73
8.4	Procédures de signalisation sur voies logiques unidirectionnelles	78
8.4.1	Introduction	78
8.4.2	Communication entre l'entité LCSE et l'utilisateur de l'entité LCSE	80
8.4.3	Communications entre entités LCSE homologues	82
8.4.4	Procédures de l'entité LCSE.....	83
8.5	Procédures de signalisation de canal logique bidirectionnel.....	91
8.5.1	Introduction	91
8.5.2	Communication entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE	93
8.5.3	Communication entre entités B-LCSE homologues.....	96
8.5.4	Procédures de l'entité B-LCSE.....	97
8.6	Procédures de fermeture de la voie logique.....	106
8.6.1	Introduction	106
8.6.2	Communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE.....	107
8.6.3	Communications entre les entités CLCSE homologues	109
8.6.4	Procédures de l'entité CLCSE	109

Remplacée par une version plus récente

Page

8.7	Procédures relatives au tableau de multiplexage selon la Recommandation H.223.....	114
8.7.1	Introduction	114
8.7.2	Communication entre l'entité MTSE et l'utilisateur de l'entité MTSE.....	116
8.7.3	Communications entre les entités MTSE homologues.....	118
8.7.4	Procédures de l'entité MTSE	119
8.8	Procédures de demande d'entrée de multiplexage.....	120
8.8.1	Introduction	120
8.8.2	Communication entre l'entité RMESE et l'utilisateur de l'entité RMESE.....	126
8.8.3	Communication entre entités RMESE homologues	128
8.8.4	Procédures de l'entité RMESE	128
8.9	Procédures de mode demande.....	130
8.9.1	Introduction	130
8.9.2	Communication entre l'entité MRSE et l'utilisateur de l'entité MRSE.....	133
8.9.3	Communication entre les entités MRSE homologues	136
8.9.4	Procédures de l'entité MRSE.....	136
8.10	Procédures liées au temps de propagation aller et retour	142
8.10.1	Introduction	142
8.10.2	Communication entre l'entité RTDSE et l'utilisateur de l'entité RTDSE.....	143
8.10.3	Communication entre les entités RTDSE homologues	144
8.10.4	Procédures de l'entité RTDSE	145
8.11	Procédures de boucle de maintenance.....	147
8.11.1	Introduction	147
8.11.2	Communication entre l'entité MLSE et l'utilisateur de l'entité MLSE.....	148
8.11.3	Communication entre entités MLSE homologues.....	150
8.11.4	Procédures de l'entité MLSE	151
	Annexe A – Affectations d'identificateur d'objet.....	158
	Appendice I – Aperçu général de la notation ASN.1	159
I.1	Introduction à la notation ASN.1	159
I.2	Types de données de base en notation ASN.1	159
I.3	Types de structures de données.....	161
I.4	Type d'identificateur d'objet.....	161
	Appendice II – Exemples de procédures H.245.....	162
II.1	Introduction.....	162
II.2	Entité de signalisation de choix entre mode maître ou esclave	162
II.3	Entité de signalisation d'échange de capacités.....	167
II.4	Entité de signalisation de voie logique.....	169
II.5	Entité de signalisation de clôture de voie logique.....	172
II.6	Entité de signalisation de table de multiplexage	173
II.7	Entité de signalisation de demande de mode.....	175
II.8	Entité de signalisation de temps de propagation aller et retour.....	177
	Appendice III – Résumé des temporisateurs et compteurs de procédure	180
III.1	Temporisateurs.....	180
III.2	Compteurs	180

Remplacée par une version plus récente

RÉSUMÉ

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 de réception de transmission, ainsi que la préférence de mode du côté réception, la signalisation de la voie logique, et les instructions 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.

Remplacée par une version plus récente

Recommandation H.245

PROTOCOLE DE COMMANDE POUR COMMUNICATIONS MULTIMÉDIAS

(Genève, 1996)

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 de réception et de transmission, ainsi que la préférence de mode du côté réception, la signalisation de la voie logique, et les instructions 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. Ces différents systèmes partagent 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 l'acheminement des données correctes.

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 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 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 (1995), *Structure de trame pour un canal à débit de 64 à 1 920 kbit/s pour les téléservices audiovisuels*.
- [6] Recommandation UIT-T H.222.0 (1996) | ISO/CEI 13818-1:1997, *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 environnement ATM*.
- [8] Recommandation UIT-T H.223 (1996), *Protocole de multiplexage des communications multimédias à faible débit*.

Remplacée par une version plus récente

- [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 couches multiples définis dans la Recommandation H.221.*
- [10] Recommandation UIT-T H.230 (1995), *Signaux de commande et d'indication synchrones de 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 des 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 (1996) | ISO/CEI 13818-2:1997, *Enregistrement de l'identificateur 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 (1996), *Systèmes et équipements terminaux visiophoniques H.320 aux environnements RNIS à large bande.*
- [18] Recommandation UIT-T H.324 (1996), *Terminal pour communication multimédia à faible débit.*
- [19] Recommandations UIT-T I.363.1, I.363.3, I.363.5 (1996), *Spécification de la couche ATM du RNIS-LB: AAL de type 1, AAL de type 3/4, AAL de type 5.*
- [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 les cas de moyens non normalisés.*
- [23] Recommandation T.51 du CCITT (1992), *Jeux de caractères codés non latins pour les services de télématique.*
- [24] Recommandation UIT-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 liaison 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 du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 3: audio.*

Remplacée par une version plus récente

- [35] ISO/CEI 13818-3:1995, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 3: son.*
- [36] ISO/CEI 13818-6:1996, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 6: extensions pour DSM-CC.*
- [37] ISO/CEI TR 9577:1993, *Technologies de l'information – Télécommunications et échange d'informations entre systèmes – Identification du protocole dans la couche réseau.*

3 Définitions

La présente Recommandation définit les termes suivants:

- 3.1 canal logique bidirectionnel:** un canal bidirectionnel comprend deux voies de transmission associées entre deux terminaux, une dans chaque sens de transmission.
- 3.2 capacité:** un terminal a une capacité spécifique s'il peut coder et transmettre ou recevoir et décoder un signal particulier.
- 3.3 canal:** liaison monodirectionnelle entre deux terminaux ou deux extrémités.
- 3.4 commande:** message qui nécessite une action mais pas de réponse explicite.
- 3.5 flux élémentaire:** terme générique désignant un train binaire codé vidéo, audio ou autre.
- 3.6 entrée:** désigne des éléments d'ensembles ou de tableaux (ensembles de capacités et tableaux de multiplexage, par exemple).
- 3.7 dans la bande:** les messages dans la bande sont ceux qui sont transportés dans le canal ou la voie logique auxquels ils se rapportent.
- 3.8 indication:** message contenant des informations mais ne nécessitant pas d'action ou de réponse.
- 3.9 voie logique:** voie unidirectionnelle ou bidirectionnelle pour la transmission de l'information.
- 3.10 numéro de voie logique:** nombre qui identifie une voie logique unique.
- 3.11 signalisation de voie logique:** ensemble des procédures utilisées pour ouvrir et fermer des canaux logiques.
- 3.12 terminal maître:** terminal défini comme étant maître par la procédure de choix du mode maître-esclave définie dans cette Recommandation, ou par toute autre procédure.
- 3.13 type de support:** une forme particulière d'information transmise à un utilisateur, ou données contenant cette information: la vidéo, l'audio et le texte sont des exemples de types de support.
- 3.14 mode:** ensemble de flux élémentaires qu'un terminal transmet, à l'intention de transmettre ou voudrait recevoir.
- 3.15 communication multimédia:** transmission ou réception de signaux correspondant à au moins deux types de support.
- 3.16 hors norme:** qui n'est pas conforme à une norme nationale ou internationale à laquelle il est fait référence dans cette Recommandation.
- 3.17 multipoint:** interconnexion simultanée d'au moins trois terminaux pour permettre à plusieurs sites de communiquer par l'utilisation d'équipements de commande multipoint (ponts de conférence) qui dirigent d'une manière centralisée les flux d'information.
- 3.18 demande:** message qui induit une action du terminal distant et nécessite une réponse immédiate de ce terminal.

Remplacée par une version plus récente

- 3.19 réponse:** message apportant la réponse à une demande.
- 3.20 session:** période de communication conversationnelle ou non conversationnelle entre deux terminaux (par exemple consultation d'une base de données).
- 3.21 terminal asservi:** terminal désigné comme étant celui qui est asservi par la procédure de choix de mode maître-esclave définie dans cette Recommandation, ou par toute autre procédure.
- 3.22 prise en charge:** possibilité de fonctionner dans un mode donné; cependant une spécification de prise en charge d'un mode donné ne signifie pas que ce mode doit être effectivement utilisé en permanence; à moins d'être interdits, d'autres modes peuvent être utilisés après négociation.
- 3.23 terminal:** équipement d'extrémité quelconque, par exemple terminal d'un utilisateur ou tout autre système de communication tel qu'un pont de conférence ou un serveur d'informations.
- 3.24 voie logique unidirectionnelle:** voie pour la transmission d'un seul flux élémentaire de trafic d'un terminal à un autre.

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
ASN.1	notation de syntaxe abstraite un (<i>abstract syntax notation one</i>)
ATM	mode de transfert asynchrone (<i>asynchronous transfer mode</i>)
B-LCSE	entité de signalisation de canal logique bidirectionnel (<i>bi-directional logical channel signalling entity</i>)
CESE	entité de signalisation d'échange de capacités (<i>capability exchange signalling entity</i>)
CLCSE	entité de signalisation de fermeture de la voie logique (<i>close logical channel signalling entity</i>)
CIF	format intermédiaire commun (d'une image vidéo: voir les Recommandations H.261 et H.263) (<i>common intermediate format</i>)
CPCS	sous-couche de convergence de parties communes (de la couche d'adaptation ATM 5) (<i>common part convergence sublayer</i>)
DTMF	tonalités multifréquences (<i>dual tone multi-frequency</i>)
DSM-CC	moyens d'enregistrement numérique – Télécommande et contrôle (<i>digital storage media – command and control</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>initialisation 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>)
MCU	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>)

Remplacée par une version plus récente

MSDSE	entité de signalisation pour le choix du mode maître ou asservi (<i>master slave determination signalling entity</i>)
MTSE	entité de signalisation du tableau de multiplexage (<i>multiplex table signalling entity</i>)
MRSE	entité de signalisation de demande de mode (<i>mode request 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>)
PDU	unité de données de protocole (<i>protocol data unit</i>)
PID	identificateur de paquet (se référer à la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>packet identifier</i>)
QCIF	quart de format CIF (<i>quarter CIF</i>)
RMESE	entité de signalisation de demande d'entrée de multiplexage (<i>request multiplex entry signalling entity</i>)
RTDSE	entité de signalisation du temps de propagation aller et retour (<i>round trip delay signalling entity</i>)
RTGC	réseau téléphonique général commuté
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 binaire 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 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 paragraphe détaillant les procédures définit les protocoles utilisant les messages définis dans les autres paragraphes.

Bien que tous les messages et toutes les procédures définis dans cette 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 ou esclave

Des conflits peuvent surgir lorsque deux terminaux en communication déclenchent simultanément des événements semblables et qu'un seul de ces événements est possible ou souhaité, par exemple lorsque, compte tenu des ressources disponibles, l'événement ne peut se produire qu'une seule fois. Pour résoudre de tels conflits, un terminal agira en tant que maître pendant que l'autre agira en tant qu'esclave. Des dispositions précisent comment le terminal maître et le terminal esclave doivent réagir en cas de conflit.

Remplacée par une version plus récente

La procédure de choix du mode maître ou esclave permet aux terminaux en communication de définir quel sera le terminal maître et quel sera le terminal esclave. Une fois choisi, le statut des terminaux reste inchangé pendant toute la durée de la communication.

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.

On transmet toutes les capacités d'un terminal à recevoir et à décoder différents signaux à l'autre terminal par la transmission de son ensemble de ses capacités.

Les capacités de réception décrivent les capacités qu'a le terminal de recevoir et de traiter les flux d'information 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 un 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 qu'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 permettent l'envoi simultané de plus d'un flux d'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 de la vidéo à plus haute résolution peut être décodée quand un algorithme audio plus simple est utilisé; ou que, soit deux séquences vidéo à faible résolution, soit 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 tandis que 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, si sa signification est connue.

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

5.3 Procédures de signalisation du canal logique

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

Une partie de ce protocole se rapporte à l'ouverture de canaux bidirectionnels. Pour éviter l'apparition éventuelle de conflits lorsque deux terminaux déclenchent simultanément des événements semblables, un terminal est défini comme le terminal maître, et l'autre comme le terminal esclave. Un protocole permet de déterminer quel est le terminal maître et quel est le terminal esclave. Cependant, des systèmes utilisant la présente Recommandation peuvent spécifier la procédure définie dans cette Recommandation ou d'autres moyens de choisir quel sera le terminal maître et quel sera le terminal esclave.

Remplacée par une version plus récente

5.4 Demande de fermeture de canal logique émanant du terminal de réception

Un canal logique est ouvert ou fermé depuis l'extrémité où est situé l'émetteur. Un mécanisme permet à un terminal de réception de demander la fermeture d'un canal logique entrant. Le terminal d'émission peut accepter ou rejeter la demande de fermeture du canal logique. Un terminal peut, par exemple, utiliser les procédures décrites ici pour demander la fermeture d'un canal logique entrant qui, pour une raison quelconque, ne peut pas être décodé. Ces procédures peuvent aussi être utilisées par le terminal qui n'a pas ouvert ce canal, pour demander la fermeture d'un canal logique bidirectionnel.

5.5 Modification d'entrée de tableau de multiplexage de la Recommandation H.223

Le tableau de multiplexage de la Recommandation H.223 permet d'associer chaque octet d'un message MUX H.223 à un numéro particulier de canal logique. Le tableau de multiplexage de la Recommandation H.223 peut comporter jusqu'à 15 entrées. Un mécanisme permet au terminal d'émission de spécifier les nouvelles entrées de tableau de multiplexage H.223 et de les signaler au récepteur. Un terminal de réception peut aussi demander la retransmission d'une entrée de tableau de multiplexage.

5.6 Demande de mode pour les communications audiovisuelles et les transmissions de données

Une fois que le protocole d'échange de capacités est achevé, les deux terminaux connaissent les capacités de chaque partenaire pour la transmission et la réception comme cela est spécifié dans les descripteurs de capacités qui ont été échangés. Il n'est pas nécessaire qu'un terminal déclare toutes ses capacités; il ne doit déclarer que celles qu'il souhaite utiliser.

Un terminal peut indiquer ses capacités d'émission. Un terminal recevant des capacités d'émission du terminal distant peut demander qu'un mode particulier lui soit transmis. Un terminal indique qu'il ne veut pas que son mode d'émission soit contrôlé 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 d'émission et un terminal de réception. Un mécanisme permet de mesurer ce temps de propagation aller et retour. Ce mécanisme peut aussi être utile pour déterminer si le terminal distant fonctionne toujours.

5.8 Boucles de maintenance

Des procédures permettent d'établir des boucles de maintenance. Il est possible de spécifier la boucle d'un canal logique unique sous la forme d'une boucle numérique ou d'une boucle décodée, ainsi que la boucle de l'ensemble du multiplexage.

5.9 Instructions et indications d'état

Les instructions et indications d'état sont fournies pour différents besoins comme par exemple les indications d'activité ou d'inactivité des signaux audio et vidéo pour l'utilisateur, la demande de rafraîchissement accéléré pour la commutation de source dans les applications multipoint. Ni les instructions, ni les indications d'état ne mettent à jour les messages de réponse à partir du terminal distant. Les instructions rendent une action obligatoire sur le terminal distant alors que les indications d'état ne font qu'apporter des informations et ne rendent aucune action obligatoire.

Une instruction est définie pour permettre le contrôle du débit binaire des canaux logiques et de l'ensemble du multiplexage à partir du terminal distant. Cela répond à différents besoins: interfonctionnement avec des terminaux faisant appel à des multiplexages dans lesquels seul un nombre fini de débits binaires est disponible; applications multipoint pour lesquelles une adaptation entre les débits de différentes sources est nécessaire; et contrôle de flux dans les réseaux encombrés.

6 Messages: syntaxe

Ce paragraphe spécifie la syntaxe des messages utilisant la notation définie en ASN.1 [30]. Les messages seront codés pour la transmission en appliquant les règles de codage compactes spécifiées en [31] en utilisant la variante de base alignée. Le bit le plus significatif de chaque octet est transmis en premier, comme indiqué dans la Recommandation X.691.

Remplacée par une version plus récente

MULTIMEDIA-SYSTEM-CONTROL DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- Exporter tous les symboles

-- =====
-- Messages du niveau supérieur
-- =====

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

-- Un message de demande entraîne une action et nécessite une réponse immédiate

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

-- Un message de réponse est la réponse à un message de demande

```
ResponseMessage ::=CHOICE
{
    nonStandard           NonStandardMessage,
    masterSlaveDeterminationAck MasterSlaveDeterminationAck,
    masterSlaveDeterminationReject MasterSlaveDeterminationReject,
    terminalCapabilitySetAck TerminalCapabilitySetAck,
    terminalCapabilitySetReject TerminalCapabilitySetReject,
    openLogicalChannelAck OpenLogicalChannelAck,
    openLogicalChannelReject OpenLogicalChannelReject,
    closeLogicalChannelAck CloseLogicalChannelAck,
    ...
}
```

Remplacée par une version plus récente

requestChannelCloseAck	RequestChannelCloseAck,
requestChannelCloseReject	RequestChannelCloseReject,
multiplexEntrySendAck	MultiplexEntrySendAck,
multiplexEntrySendReject	MultiplexEntrySendReject,
requestMultiplexEntryAck	RequestMultiplexEntryAck,
requestMultiplexEntryReject	RequestMultiplexEntryReject,
requestModeAck	RequestModeAck,
requestModeReject	RequestModeReject,
roundTripDelayResponse	RoundTripDelayResponse,
maintenanceLoopAck	MaintenanceLoopAck,
maintenanceLoopReject	MaintenanceLoopReject,
...	

}

-- Une PDU de commande induit une action mais ne nécessite pas de réponse explicite

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

-- Un message d'indication est une information qui ne nécessite ni action ni réponse

IndicationMessage	::=CHOICE
{	
nonStandard	NonStandardMessage,
functionNotSupported	FunctionNotSupported,
masterSlaveDeterminationRelease	MasterSlaveDeterminationRelease,
terminalCapabilitySetRelease	TerminalCapabilitySetRelease,
openLogicalChannelConfirm	OpenLogicalChannelConfirm,
requestChannelCloseRelease	RequestChannelCloseRelease,
multiplexEntrySendRelease	MultiplexEntrySendRelease,
requestMultiplexEntryRelease	RequestMultiplexEntryRelease,
requestModeRelease	RequestModeRelease,
miscellaneousIndication	MiscellaneousIndication,
jitterindication	Jitterindication,
h223Skewindication	H223Skewindication,
}	

Remplacée par une version plus récente

```
newATMVCindication      NewATMVCindication,
userinput               UserinputIndication,
...
}

-- Le numéro de séquence est défini ici de la manière dont il est utilisé dans un certain nombre de PDU

SequenceNumber          ::=INTEGER (0..255)

-- =====
-- Définitions PDU hors normes
-- =====

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

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

NonStandardIdentifier   ::=CHOICE
{
  object                OBJECT IDENTIFIER,
  h221NonStandard       SEQUENCE
  {
    t35CountryCode      INTEGER (0..255),           -- code T.35 de pays
    t35Extension         INTEGER (0..255),         -- code du ressort national
    manufacturerCode    INTEGER (0..65535)        -- code du ressort national
  }
}

-- =====
-- Définitions relatives au choix du mode maître ou esclave
-- =====

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

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

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

Remplacée par une version plus récente

```
MasterSlaveDeterminationRelease ::=SEQUENCE
{
  ...
}

-- =====
-- Définitions relatives à l'échange de capacités
-- =====

TerminalCapabilitySet ::=SEQUENCE
{
  sequenceNumber          SequenceNumber,

  protocolIdentifier      OBJECT IDENTIFIER,
  -- recevra la valeur
  -- {itu-t(0) recommendation(0)h(8) 245 version(0) 1}

  multiplexCapability     MultiplexCapability OPTIONAL,

  capabilityTable         SET SIZE (1..256) OF CapabilityTableEntry OPTIONAL,

  capabilityDescriptors   SET SIZE (1..256) OF CapabilityDescriptor OPTIONAL,
  ...
}

V75 Capability ::=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
  }
}
```

Remplacée par une version plus récente

```
{
  highestEntryNumberProcessed      CapabilityTableEntryNumber,
  noneProcessed                    NULL
},
...
},
...
}
TerminalCapabilitySetRelease      ::=SEQUENCE
{
  ...
}
```

-- =====
-- Définitions relatives à l'échange de capacités: description des capacités de niveau supérieur
-- =====

```
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),           -- unités milliseconde
    ...
  },
  ...
}
```

-- =====
-- Définitions relatives à l'échange de capacités: capacités de multiplexage
-- =====

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

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


Remplacée par une version plus récente

```

h223MultiplexTableCapability          CHOICE
{
    basic                              NULL,
    enhanced                            SEQUENCE
    {
        maximumNestingDepth            INTEGER (1..15),
        maximumElementListSize        INTEGER (2..255),
        maximumSubElementListSize     INTEGER (2..255),
        ...
    }
},
...
}
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),
    maxWindowSizeCapabilityCapability  INTEGER (1..127),
    v75Capability                    V75Capability,
    ...
}

-- =====
-- Définitions relatives à l'échange de capacités: capacités vidéo
-- =====

VideoCapability                        ::=CHOICE
{
    nonStandard                       NonStandardParameter ,
    h261VideoCapability                H261VideoCapability,
    h262VideoCapability                H262VideoCapability,
    h263VideoCapability                H263VideoCapability,
    is11172VideoCapability             IS11172VideoCapability,
    ...
}

H261VideoCapability                   ::=SEQUENCE
{
    qcifMPI                           INTEGER (1..4) OPTIONAL,      -- unité: 1/29,97 Hz
    cifMPI                             INTEGER (1..4) OPTIONAL,    -- unité: 1/29,97 Hz
    temporalSpatialTradeOffCapability  BOOLEAN,
    maxBitRate                         INTEGER (1..19200),         -- unité: 100 bit/s
    stillimageTransmission             BOOLEAN,                    -- Annexe D/H.261
    ...
}

H262VideoCapability                   ::=SEQUENCE
{
    profileAndLevel-SPatML             BOOLEAN,
    profileAndLevel-MPatLL             BOOLEAN,
    profileAndLevel-MPatML             BOOLEAN,
    profileAndLevel-MPatH-14           BOOLEAN,
    profileAndLevel-MPatHL             BOOLEAN,
}

```

Remplacée par une version plus récente

```
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, -- unité: 400 bits/s
vbvBufferSize                   INTEGER (0.. 262143) OPTIONAL,      -- unité: 16 384 bits
samplesPerLine                  INTEGER (0..16383) OPTIONAL,      -- unité: échantillon/ligne
linesPerFrame                   INTEGER (0..16383) OPTIONAL,      -- unité: ligne/image
framesPerSecond                 INTEGER (0..15) OPTIONAL,         -- frame_rate_code
luminanceSampleRate            INTEGER (0..4294967295) OPTIONAL, -- unité: échantillon/s
...
}

H263VideoCapability              ::=SEQUENCE
{
    sqcifMPI                     INTEGER (1..32) OPTIONAL,        -- unité: 1/29,97 Hz
    qcifMPI                     INTEGER (1..32) OPTIONAL,        -- unité: 1/29,97 Hz
    cifMPI                      INTEGER (1..32) OPTIONAL,        -- unité: 1/29,97 Hz
    cif4MPI                     INTEGER (1..32) OPTIONAL,        -- unité: 1/29,97 Hz
    cif16MPI                    INTEGER (1..32) OPTIONAL,        -- unité: 1/29,97 Hz
    maxBitRate                   INTEGER (1..192400),             -- unité: 100 bit/s
    unrestrictedVector           BOOLEAN,
    arithmeticCoding             BOOLEAN,
    advancedPrediction           BOOLEAN,
    pbFrames                     BOOLEAN,
    temporalSpatialTradeOffCapability BOOLEAN,
    hrd-B                        INTEGER (0..524287) OPTIONAL,    -- unité: 128 bits
    bppMaxKb                     INTEGER (0..65535) OPTIONAL,    -- unité: 1024 bits
    ...
}

IS11172VideoCapability          ::=SEQUENCE
{
    constrainedBitstream         BOOLEAN,
    videoBitRate                 INTEGER (0..1073741823) OPTIONAL, -- unité: 400 bits/s
    vbvBufferSize               INTEGER (0..262143) OPTIONAL,    -- unité: 16 384 bits
    samplesPerLine              INTEGER (0..16383) OPTIONAL,      -- unité: échantillon/ligne
    linesPerFrame               INTEGER (0..16383) OPTIONAL,      -- unité: échantillon/image
    pictureRate                 INTEGER (0..15) OPTIONAL,         -- unité: échantillon/s
    luminanceSampleRate         INTEGER (0..4294967295) OPTIONAL, -- unité: échantillon/s
    ...
}

-- =====
-- Définitions relatives à l'échange de capacités: capacités audio
-- =====

-- Pour un multiplexage H.222, les nombres entiers indiquent la taille de la mémoire tampon
-- STD en unités de 256 octets
-- Pour un multiplexage H.223, les nombres entiers indiquent le nombre maximal de trames
-- audio par unité AL-SDU

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),
}
```

Remplacée par une version plus récente

g723	SEQUENCE	
{		
maxAl-sduAudioFrames	INTEGER (1..256),	
silenceSuppression	BOOLEAN	
},		
g728	INTEGER (1..256),	
g729	INTEGER (1..256),	
g729Annex A	INTEGER (1..256),	
is11172AudioCapability	IS11172AudioCapability,	
is13818AudioCapability	IS13818AudioCapability,	
...		
}		
IS11172AudioCapability	::=SEQUENCE	
{		
audioLayer1	BOOLEAN,	
audioLayer2	BOOLEAN,	
audioLayer3	BOOLEAN,	
audioSampling32k	BOOLEAN,	
audioSampling44k1	BOOLEAN,	
audioSampling48k	BOOLEAN,	
singleChannel	BOOLEAN,	
twoChannels	BOOLEAN,	
bitRate	INTEGER (1..448),	-- unité: 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),	-- unité: kbit/s
...		
}		

Remplacée par une version plus récente

-- =====
-- Définitions relatives à l'échange de capacités: capacités de données
-- =====

```
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                DataModeProtocol,
      nlpidData                    OCTET STRING
    },
    dsvdControl                    NULL,
    h222DataPartitioning           DataProtocolCapability,
    ...
  },
  maxBitRate                       INTEGER (0..4294967295),      -- unité: 100 bit/s
  ...
}

DataProtocolCapability            ::=CHOICE
{
  nonStandard                      NonStandardParameter,
  v14buffered                      NULL,
  v42lapm                          NULL,
  ...
  hdlcFrameTunneling              NULL,
  h310SeparateVCStack              NULL,
  h310SingleVCStack               NULL,
  transparent                       NULL,
  ...
}

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,
    jbig200x200Progr                 BOOLEAN,
    jbig300x300Seq                   BOOLEAN,
    jbig300x300Progr                 BOOLEAN,
  }
}
```

Remplacée par une version plus récente

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

-- =====
-- Définitions relatives à la signalisation de canal logique
-- =====

-- Le terme 'Forward' désigne le sens de transmission vers l'avant, depuis le terminal qui a
-- fait demande initiale de canal logique en direction de l'autre terminal; le terme 'reverse'
-- désigne le sens de transmission opposé, dans le cas d'une demande de canal bidirectionnel.

OpenLogicalChannel ::=SEQUENCE
{
    forwardLogicalChannelNumber LogicalChannelNumber,

    forwardLogicalChannelParameters SEQUENCE
    {
        portNumber INTEGER (0..65535) OPTIONAL,
        dataType DataType,
        multiplexParameters CHOICE
        {
            h222LogicalChannelParameters H222LogicalChannelParameters,
            h223LogicalChannelParameters H223LogicalChannelParameters,
            v76LogicalChannelParameters V76LogicalChannelParameters,
            ...
        },
        ...
    },
    ...
},
-- Est utilisé pour spécifier la voie opposée dans le cas d'une demande d'ouverture de canal bidirectionnel

reverseLogicalChannelParameters SEQUENCE
{
    dataType DataType,
    multiplexParameters CHOICE
    {
        -- Les paramètres H.222 ne sont jamais présents dans le sens opposé
        h223LogicalChannelParameters H223LogicalChannelParameters,
        v76LogicalChannelParameters V76LogicalChannelParameters,
        ...
    }OPTIONAL, -- Non présent dans le cas de la Recommandation H.222
    ...
}OPTIONAL, -- Non présent dans le cas d'une demande de canal unidirectionnel
...
}

LogicalChannelNumber ::=INTEGER (1..65535)

V75Parameters ::=SEQUENCE
{
    audioHeaderPresent BOOLEAN
}
}
```

Remplacée par une version plus récente

DataType	::=CHOICE	
{		
nonStandard	NonStandardParameter,	
nullData	NULL,	
videoData	VideoCapability,	
audioData	AudioCapability,	
data	DataApplicationCapability,	
encryptionData	EncryptionMode,	
...		
}		
H222LogicalChannelParameters	::=SEQUENCE	
{		
virtualChannelID	INTEGER (0..65535),	
subChannelID	INTEGER (0..8191),	
pcr-pid	INTEGER (0..8191) OPTIONAL,	
programDescriptors	OCTET STRING OPTIONAL,	
streamDescriptors	OCTET STRING OPTIONAL,	
...		
}		
H223LogicalChannelParameters	::=SEQUENCE	
{		
adaptationLayerType	CHOICE	
{		
nonStandard	NonStandardParameter,	
al1Framed	NULL,	
al1NotFramed	NULL,	
al2WithoutSequenceNumbers	NULL,	
al2WithSequenceNumbers	NULL,	
al3	SEQUENCE	
{		
controlFieldOctets	INTEGER (0..2),	
sendBufferSize	INTEGER (0..16777215)	<i>-- unité: octet</i>
},		
...		
},		
segmentableFlag	BOOLEAN,	
...		
}		
V76LogicalChannelParameters	::=SEQUENCE	
{		
hdlcParameters	V76HDLCPParameters,	
suspendResume	CHOICE	
{		
wAddress	NULL,	
woAddress	NULL,	
...		
},		
ulH	BOOLEAN,	
mode	CHOICE	
{		
eRM	SEQUENCE	
{		
windowSize	INTEGER (1..127),	
recovery	CHOICE	
{		
rej	NULL,	
sREJ	NULL,	
mSREJ	NULL,	
...		
},		
...		
},		
...		
},		
uNERM	NULL,	
...		
},		

Remplacée par une version plus récente

v75Parameters	V75Parameters,
...	
}	
V76HDLParameters	::=SEQUENCE
{	
crcLength	CRCLength,
n401	INTEGER (1..127),
loopbackTestProcedure	BOOLEAN,
...	
}	
CRCLength	::=CHOICE
{	
crc8bit	NULL,
crc16bit	NULL,
crc32bit	NULL,
...	
}	
OpenLogicalChannelAck	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
reverseLogicalChannelParameters	SEQUENCE
{	
reverseLogicalChannelNumber	LogicalChannelNumber
portNumber	INTEGER (0..65535) OPTIONAL,
multiplexParameters	CHOICE
{	
h222LogicalChannelParameters	H222LogicalChannelParameters,
-- Les paramètres H.223 ne sont jamais présents dans le sens opposé	
...	
} OPTIONAL,	-- Non présent dans le cas de la Recommandation H.223
...	
} OPTIONAL,	-- Non présent dans le cas d'une demande de canal unidirectionnel
...	
}	
}	
OpenLogicalChannelReject	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
cause	CHOICE
{	
unspecified	NULL,
unsuitableReverseParameters	NULL,
dataTypeNotSupported	NULL,
dataTypeNotAvailable	NULL,
unknownDataType	NULL,
dataTypeALCombinationNotSupported	NULL,
...	
},	
...	
}	
OpenLogicalChannelConfirm	::=SEQUENCE
{	
forwardLogicalChannelNumber	LogicalChannelNumber,
...	
}	
CloseLogicalChannel	::=SEQUENCE
{	
logicalChannelNumber	LogicalChannelNumber,
source	CHOICE
{	
user	NULL,
lcse	NULL
},	
...	
}	

Remplacée par une version plus récente

```
CloseLogicalChannelAck ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    ...
}

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

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

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

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

-- =====
-- Définitions relatives au tableau de multiplexage de la Recommandation H.223
-- =====

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),
        untilClosingFlag NULL
    }
}
-- répétition du type
-- utilisé pour le dernier
-- élément
```

Remplacée par une version plus récente

MultiplexTableEntryNumber	::=INTEGER (1..15)
MultiplexEntrySendAck { sequenceNumber multiplexTableEntryNumber ... }	::=SEQUENCE SequenceNumber, SET SIZE (1..15) OF MultiplexTableEntryNumber, ...
MultiplexEntrySendReject { sequenceNumber rejectionDescriptions ... }	::=SEQUENCE SequenceNumber, SET SIZE (1..15) OF MultiplexEntryRejectionDescriptions, ...
MultiplexEntryRejectionDescriptions { multiplexTableEntryNumber cause { unspecifiedCause descriptorTooComplex ... }, ... }	::=SEQUENCE MultiplexTableEntryNumber, CHOICE NULL, NULL, ...
MultiplexEntrySendRelease { multiplexTableEntryNumber ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...
RequestMultiplexEntry { entryNumbers ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...
RequestMultiplexEntryAck { entryNumbers ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...
RequestMultiplexEntryReject { entryNumbers rejectionDescriptions ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, SET SIZE (1..15) OF RequestMultiplexEntryRejectionDescriptions, ...
RequestMultiplexEntryRejectionDescriptions { multiplexTableEntryNumber cause { unspecifiedCause ... }, ... }	::=SEQUENCE MultiplexTableEntryNumber, CHOICE NULL, ...
RequestMultiplexEntryRelease { entryNumbers ... }	::=SEQUENCE SET SIZE (1..15) OF MultiplexTableEntryNumber, ...

Remplacée par une version plus récente

-- Définitions des demandes de modes

-- RequestMode est une liste des modes, classés par ordre de préférence, qu'un terminal aimerait
-- se voir transmis

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
{
 ...
}

-- Définitions des demandes de modes: description des modes

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

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

Remplacée par une version plus récente

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

=====
-- Définitions des demandes de modes: modes vidéo
=====

VideoMode ::=CHOICE
{
    nonStandard NonStandardParameter,
    h261VideoMode H261VideoMode,
    h262VideoMode H262VideoMode,
    h263VideoMode H263VideoMode,
    is11172VideoMode IS11172VideoMode,
    ...
}

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

H262VideoMode ::=SEQUENCE
{
    profileAndLevel CHOICE
    {
        profileAndLevel-SPatML NULL,
        profileAndLevel-MPatLL NULL,
        profileAndLevel-MPatML NULL,
        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,
        ...
    },
    ...
}
```

Remplacée par une version plus récente

```

videoBitRate          INTEGER(0..1073741823) OPTIONAL, -- unité: 400 bits/sec
vbvBufferSize         INTEGER(0..262143) OPTIONAL,   -- unité: 16 384 bits
samplesPerLine        INTEGER(0..16383) OPTIONAL,    -- unité: éch./ligne
linesPerFrame         INTEGER(0..16383) OPTIONAL,    -- unité: ligne/image
framesPerSecond       INTEGER(0..15) OPTIONAL,      -- frame_rate_code
luminanceSampleRate   INTEGER(0..4294967295) OPTIONAL, -- unité: éch./s
...
}

```

```

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

```

```

IS11172VideoMode ::=SEQUENCE
{
  constrainedBitStream
  videoBitRate     INTEGER (0..1073741823) OPTIONAL, -- unité: 400 bits/s
  vbvBufferSize    INTEGER (0..262143) OPTIONAL,   -- unité: 16 384 bits
  samplesPerLine   INTEGER (0..16383) OPTIONAL,    -- unité: éch./ligne
  linesPerFrame    INTEGER (0..16383) OPTIONAL,    -- unité: éch./image
  pictureRate      INTEGER (0..15) OPTIONAL,
  luminanceSampleRate
  INTEGER (0..4294967295) OPTIONAL, -- unité: éch./s
  ...
}

```

```

-- =====
-- Définitions des demandes de modes: modes audio
-- =====

```

```

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,
  v76              NULL,
  g723             CHOICE
  {
    noSilenceSuppressionLowRate
    noSilenceSuppressionHighRate
    silenceSuppressionLowRate
    silenceSuppressionHighRate
  },
  ...
}

```

Remplacée par une version plus récente

```

is11172AudioMode
is13818AudioMode

...
}

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), -- unité: 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,

```

Remplacée par une version plus récente

```
multilingual                BOOLEAN,

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

=====
-- Définitions des demandes de modes: modes données
=====

DataMode                    ::=SEQUENCE
{
  application                CHOICE
  {
    nonStandard              NonStandardParameter,
    t120                     DataProtocolCapability,
    dsm-cc                   DataProtocolCapability,
    userData                 DataProtocolCapability,
    t84                      DataProtocolCapability,
    t434                    DataProtocolCapability,
    h224                    DataProtocolCapability,
    nIpid                   SEQUENCE
    {
      nIpidProtocol          DataProtocolCapability,
      nIpidData              OCTET STRING
    },
    dsvdControl              NULL,
    h222DataPartitioning    DataProtocolCapability,
    ...
  }
  bitRate                   INTEGER (0..4294967295),     -- unité: 100 kbit/s
  ...
}

=====
-- Définitions des demandes de modes: modes de chiffrement
=====

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

=====
-- Définitions des temps de propagation aller-retour
=====

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

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

Remplacée par une version plus récente

=====
-- Définition des boucles de maintenance
=====

```
MaintenanceLoopRequest ::=SEQUENCE
{
    sequenceNumber      SequenceNumber,
    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
{
    ...
}
```

=====
-- Définitions des messages de commande
=====

=====
-- Message de commande: envoi de l'ensemble de capacités du terminal
=====

```
SendTerminalCapabilitySet ::=CHOICE
{
    specificRequest      SEQUENCE
    {
        multiplexCapability BOOLEAN,
        capabilityTableEntryNumbers SET SIZE(1..65535) OF CapabilityTableEntry Number OPTIONAL,
        capabilityDescriptorNumbers SET SIZE(1..256) OF CapabilityDescriptorNumber OPTIONAL,
        ...
    },

```

Remplacée par une version plus récente

```
genericRequest          NULL,
...
}

-----
-- Messages de commande: chiffrement
-----

EncryptionCommand      ::=CHOICE
{
    encryptionSE        OCTET STRING,          -- selon H.233, mais pas de protection
                                                -- d'erreurs
    encryptionIVRequest NULL,
    encryptionAlgorithmID SEQUENCE
    {
        h233AlgorithmIdentifier SequenceNumber,
        associatedAlgorithm      NonStandardParameter
    },
    ...
}

-----
-- Message de commande: contrôle de flux
-----

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

-----
-- Message de commande: modification ou fin de session
-----

EndSessionCommand      ::=CHOICE
{
    nonStandard         NonStandardParameter,

    disconnect         NULL,
    gstnOptions        CHOICE
    {
        telephonyMode     NULL,
        v8bis             NULL,
        v34DSVD           NULL,
        v34DuplexFAX      NULL,
        v34H324           NULL,
        ...
    },
    ...
}
```

Remplacée par une version plus récente

-- Message de commande: diverses commandes de type H.230

```
MiscellaneousCommand ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    type CHOICE
    {
        equaliseDelay NULL, -- identique à ACE H.230
        zeroDelay NULL, -- identique à ACZ H.230
        multipointModeCommand NULL,
        cancelMultipointModeCommand NULL,
        videoFreezePicture NULL,
        videoFastUpdatePicture NULL,
        videoFastUpdateGOB SEQUENCE
        {
            firstGOB INTEGER (0..17),
            numberOfGOBs INTEGER (1..18)
        },
        videoTemporalSpatialTradeOff INTEGER (0..31), -- exige une valeur de
        -- compromis
        videoSendSyncEveryGOB NULL,
        videoSendSyncEveryGOBCancel NULL,
        ...
    },
    ...
}
```

-- Définitions des messages d'indication

-- Message d'indication: fonction non prise en charge

-- L'indication suivante sert à renvoyer une demande, réponse ou commande complète non reconnue

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

-- Message d'indication: diverses indications de type H.230

```
Miscellaneousindication ::=SEQUENCE
{
    logicalChannelNumber LogicalChannelNumber,
    type CHOICE
    {
        logicalChannelActive NULL, -- identique à AIA et VIA H.230
        logicalChannelInactive NULL, -- identique à AIM et VIS H.230

        multipointConference NULL,
        cancelMultipointConference NULL,

        multipointZeroComm NULL, -- identique à MIZ H.230
        cancelMultipointZeroComm NULL, -- identique à cancel MIZ H.230
    }
}
```

Remplacée par une version plus récente

multipointSecondaryStatus	NULL,	-- identique à MIS H.230
cancelMultipointSecondaryStatus	NULL,	-- identique à canal MIS H.230
videoindicateReadyToActivate	NULL,	-- identique à VIR H.230
videoTemporalSpatialTradeOff	INTEGER (0..31),	-- identique à la valeur de -- compromis actuelle

```

...
},
...
}

```

```

-----
-- Message d'indication: indication de gigue
-----

```

```

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

```

```

-----
-- Message d'indication: canal logique H.223 asymétrique
-----

```

```

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

```

```

-----
-- Message d'indication: indication de nouveau canal virtuel ATM
-----

```

```

NewATMVCindication ::=SEQUENCE
{
    resourceID INTEGER(0..65535),
    bitRate INTEGER(1..65535), -- unité: 64 kbits/s
    bitRateLockedToPCRClock BOOLEAN,
    bitRateLockedToNetworkClock BOOLEAN,
    aal CHOICE
    {
        aal1 SEQUENCE
        {
            clockRecovery CHOICE
            {
                nullClockRecovery NULL,
                srtsClockRecovery NULL,
                adaptiveClockRecovery NULL,
                ...
            },

```

Remplacée par une version plus récente

```
errorCorrection          CHOICE
{
  nullErrorCorrection    NULL,
  longInterleaver        NULL,
  shortInterleaver       NULL,
  errorCorrectionOnly     NULL,
  ...
},
structuredDataTransfer  BOOLEAN,
partiallyFilledCells    BOOLEAN,
...
},
aal5                     SEQUENCE
{
  forwardMaximumSDUSize  INTEGER (0..65535),      -- unité: octet
  backwardMaximumSDUSize INTEGER (0..65535),      -- unité: octet
  ...
},
...
},
multiplex                CHOICE
{
  NoMultiplex            NULL,
  transportStream        NULL,
  programStream          NULL,
  ...
},
...
}
```

```
-- =====
-- Message d'indication: entrée utilisateur
-- =====
```

```
UserInputIndication    ::=CHOICE
{
  nonStandard           NonStandardParameter,
  alphanumeric          GeneralString,
  ...
}
```

END

Remplacée par une version plus récente

7 Messages – Définitions sémantiques

Ce paragraphe donne les définitions sémantiques et les contraintes applicables aux éléments de syntaxe définis dans le paragraphe précédent.

MultimediaSystemControlMessage: est un 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 aboutit à une action du terminal distant et requiert une réponse immédiate de ce terminal. Le message nonStandardRequest 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 nonStandardResponse 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 nonStandardCommand 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 nonStandardIndication 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.

NonStandardIdentifiant: est 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*) *= attribués par pays. Les codes de fabricant sont les mêmes que ceux qui sont attribués pour les besoins de la Recommandation H.320 [17].

7.1 Messages de choix du mode maître ou esclave

Cet ensemble de messages est utilisé par un protocole pour déterminer quel est le terminal maître et quel 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.

Le type de terminal terminalType est un nombre qui identifie les différents types de terminaux – tels que terminaux, unités MCU ou centres têtes de lignes, par exemple. L'affectation de valeurs aux types de terminaux ne relève pas de la présente Recommandation.

Le nombre statusDeterminationNumber est un nombre aléatoire situé dans la gamme 0..2²⁴-1.

7.1.2 Acquiescement du choix du mode maître ou esclave

Ce message est utilisé pour confirmer si le terminal est le terminal maître ou le terminal esclave, comme l'indique le résultat de la procédure. Quand on effectue un choix de type maître, le terminal recevant ce message est le terminal maître et quand le choix est du type esclave, le terminal recevant le message est le terminal esclave.

7.1.3 Rejet du choix du mode maître ou esclave

Ce message est utilisé pour rejeter le message de choix du mode maître-esclave MasterSlaveDetermination. Quand le motif est du type identicalNumbers, le rejet est dû au fait que les nombres aléatoires sont équivalents et que les types de terminaux sont identiques.

7.1.4 Abandon du choix du mode maître ou esclave

Ce message est envoyé en cas de temporisation.

7.2 Messages de capacités des terminaux

Cet ensemble de messages est destiné à garantir la sûreté des échanges de capacités entre les deux terminaux.

7.2.1 Présentation générale

Le terminal émetteur affecte un numéro à chaque mode distinct dans lequel un terminal peut fonctionner et ce numéro figure dans un tableau de capacités capabilityTable. Par exemple, une réalisation conforme aux Recommandations G.723 et G.728 pour l'audio, et CIF H.263 pour la vidéo, se verrait attribuer des nombres distincts.

Remplacée par une version plus récente

Ces numéros de capacités sont groupés dans des structures `AlternativeCapabilitySet`. Chaque structure `AlternativeCapabilitySet` indique que le terminal peut fonctionner selon exactement un mode décrit par l'ensemble de capacités. Par exemple, une liste `AlternativeCapabilitySet` {G.711, G.723.1, G.728} signifie que le terminal peut fonctionner dans n'importe lequel de ces modes audio, mais pas plus d'un à la fois.

Ces structures `AlternativeCapabilitySet` sont groupées dans des structures `simultaneousCapabilities`. Chaque structure `simultaneousCapability` indique un ensemble de modes 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 l'un ou l'autre des deux codecs vidéo simultanément avec n'importe lequel des codecs audio. La structure `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 pour la Recommandation H.261, un autre canal vidéo selon l'une des Recommandations H.261 ou H.263, et un canal audio selon l'une des Recommandations G.711, G.723.1 ou G.728.

NOTE – Les véritables capacités enregistrées dans le tableau `capabilityTable` sont souvent plus complexes que celles présentées ici. Par exemple, chaque capacité correspondant à la Recommandation H.263 fournit des détails qui incluent la capacité de fonctionner selon les différents formats d'image à des intervalles d'images donnés et la possibilité d'utiliser des modes de codage facultatifs.

L'ensemble de toutes les capacités du terminal est décrit par un ensemble de structures `CapabilityDescriptor`, chacune d'elles ayant une structure `simultaneousCapabilities` et un indice `capabilityDescriptorNumber` uniques. Par l'envoi de plus d'un descripteur `CapabilityDescriptor`, le terminal peut signaler des dépendances entre des modes de fonctionnement en décrivant différents ensembles de modes pouvant être utilisés simultanément. Par exemple, un terminal émettant deux structures `CapabilityDescriptor`, l'une comme dans l'ensemble précédent selon {{H.261, H.263}, {G.711, G.723.1, G.728}} et l'autre selon {{ H.262} et { G.711}} signifie que le terminal peut également faire fonctionner le codec vidéo selon la Recommandation H.262, mais uniquement avec le codec audio de faible complexité selon la Recommandation G.711.

Les terminaux peuvent ajouter dynamiquement des capacités durant une session de communications en délivrant des structures de type `CapabilityDescriptor` supplémentaires, ou éliminer des capacités en envoyant des structures `CapabilityDescriptor` révisées. Tous les terminaux devront transmettre au moins une structure `CapabilityDescriptor`.

7.2.2 Ensemble de capacités de terminaux

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

Le paramètre `sequenceNumber` est utilisé pour numéroter des instances de structures `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 en vigueur. L'Annexe A énumère les identificateurs d'objet définis en vue de leur utilisation dans le cadre de la présente Recommandation.

La structure `multiplexCapability` contient des capacités se rapportant au multiplexage et à l'interface de réseau. Un terminal devra inclure la structure `multiplexCapability` dans le premier ensemble de capacités du terminal `TerminalCapacitySet` qu'il transmet. Les capacités V.75 décrivent les capacités de l'entité de commande V.75. L'en-tête audio indique les capacités de l'en-tête audio V.75.

7.2.2.1 Tableau de capacités

Un tableau de capacités est une liste numérotée de capacités. Un terminal doit pouvoir effectuer toutes les fonctions énumérées dans son tableau de capacités, mais ne doit pas nécessairement effectuer simultanément plus d'une d'entre elles.

Une structure `TerminalCapabilitySet` peut contenir zéro élément `CapabilityTableEntry` ou davantage. Au début, aucun élément de tableaux n'est défini. Quand un élément `CapabilityTableEntry` est reçu, il remplace l'élément `CapabilityTableEntry` précédemment reçu avec le même numéro `CapabilityTableEntryNumber`. Un élément `CapabilityTableEntry` sans capacité peut être utilisé pour éliminer l'élément `CapabilityTableEntry` précédemment reçu avec le même numéro `CapabilityTableEntryNumber`.

7.2.2.2 Descripteurs de capacités

Les descripteurs `CapabilityDescriptors` sont utilisés pour indiquer les capacités de transmission et de réception d'un terminal. Chaque descripteur `CapabilityDescriptor` correspond à une déclaration séparée sur les capacités du terminal.

Remplacée par une version plus récente

Le numéro `capabilityDescriptorNumber` est utilisé pour numéroter les descripteurs `CapabilityDescriptors`. Si un terminal a une préférence pour le mode dans lequel il voudrait fonctionner en émission ou en réception, et souhaite le faire savoir en transmettant ses capacités, il peut le faire en donnant de faibles valeurs au numéro `capabilityDescriptorNumber` pour les descripteurs `CapabilityDescriptors` qui se rapportent à son ou ses modes préférés.

La structure `simultaneousCapabilities` est un ensemble de capacités `AlternativeCapabilitySet`. Elle est utilisée pour énumérer les capacités simultanées du terminal.

Un ensemble `AlternativeCapabilitySet` est une séquence de numéros `CapabilityTableEntryNumbers`. Seuls les éléments `CapabilityTableEntrys` qui ont été définis devront être présents dans un ensemble `AlternativeCapabilitySet`, bien qu'il soit possible de définir des éléments `CapabilityTableEntrys` et de s'y référer dans le même ensemble `TerminalCapabilitySet`. Si un terminal a une préférence pour le mode dans lequel il voudrait fonctionner en émission ou en réception, et souhaite le faire savoir en transmettant ses capacités, il peut le faire en générant des éléments dans des ensembles `AlternativeCapabilitySets` dans un ordre de préférence décroissant.

Un terminal doit pouvoir prendre en compte simultanément n'importe quelle capacité 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` ne contenant que des capacités d'un seul type de support pour chaque type de support de communication assuré par le terminal. Ceci doit garantir que le terminal distant peut choisir un mode d'émission comprenant au moins une instance de chaque type de support de communication assuré par le récepteur.

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

NOTE 2 – Les terminaux qui ne peuvent pas modifier l'attribution des ressources peuvent indiquer complètement leurs capacités par l'utilisation d'un descripteur `CapabilityDescriptor` unique.

7.2.2.3 Capacités

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

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

Les choix `receiveAndTransmitVideoCapability`, `receiveAndTransmitAudioCapability` et `receiveAndTransmitDataApplicationCapability` indiquent la possibilité 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 de réception et d'émission 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 et H.234 [11] et [12], respectivement.

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 vecteur 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.

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 ensemble `TerminalCapability` envoyé. Sauf indication contraire, ces capacités sont des capacités de réception.

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

Le nombre `numberOfVCs` indique combien de voies virtuelles (VC) 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. Cela ne comprend pas la voie virtuelle (VC) utilisée pour la signalisation selon la Recommandation Q.2931 [20].

`vcCapability` est un ensemble de taille égale à la valeur `numberOfVCs`, contenant les capacités présentes pour chaque voie virtuelle (VC) disponible.

Remplacée par une version plus récente

La séquence aal1, quand elle est présente, indique la capacité de la couche 1 d'adaptation ATM ainsi que les options de cette couche, spécifiées dans la Recommandation I.363 [19], qui sont prises en charge. Les séquences de codage sont définies dans le Tableau 1.

TABLEAU 1/H.245

Codage de la couche d'adaptation ATM de codage 1

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, quand elle est présente, indique la capacité de la couche 5 d'adaptation ATM ainsi que les options de cette couche, spécifiées dans la Recommandation I.363 [19], qui sont prises en charge.

Les paramètres «forwardMaximumSDUSize» et «backwardMaximumSDUSize» indiquent la longueur maximale des unités CPCS-SDU dans les sens aller et retour, mesurées en octets. L'une des séquences aal1 ou aal5 (ou les deux) sera présente.

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].

availableBitRates indique les capacités en termes de débit de la voie virtuelle (VC). Il s'agit d'une séquence de différents débits admissibles, mesurés en unités de 64 kbit/s. Les débits sont indiqués par ordre décroissant, c'est-à-dire que le débit le plus élevé admis est indiqué en premier. Les débits admis peuvent être indiqués sous forme de valeurs individuelles au moyen du champ de débit unique singleBitRate, ou sous forme d'une gamme de débits rangeOfBitRates allant du débit le plus bas lowerBitRate au débit le plus élevé higherBitRate indiquant que toutes les valeurs comprises entre ces limites inférieure et supérieure incluses sont admises. Les débits indiqués sont mesurés au point 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 les trames LAPM I 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 respectivement 2 et 3.

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 de la délivrance du premier octet d'une trame audio dans le flux multiplexé et le moment où cet octet serait délivré sur un canal à débit constant sans multiplexage.

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

Remplacée par une version plus récente

«de base» indique que le multiplex ne peut recevoir que des descripteurs MultiplexEntryDescriptors de base tels qu'ils sont définis dans la Recommandation H.223 [8].

«étendu» indique que le multiplex peut recevoir des extensions dans les descripteurs MultiplexEntryDescriptors avec les paramètres supplémentaires définis ci-dessous.

La profondeur MaximumNesting indique la profondeur maximale d'emboîtement des champs subElementList référencés de façon récursive. Les descripteurs MultiplexEntryDescriptors n'utilisant pas le champ subElementList doivent être considérés comme ayant une profondeur d'emboîtement nulle.

La taille maximumElementListSize indique le nombre maximal de champs dans la SEQUENCE ASN.1.

La taille maximumSubElementListSize indique le nombre maximal de sous-éléments dans l'élément SubElementList.

V76Capability: indique les capacités spécifiques au multiplex V.76.

La structure suspendResumeCapabilitywAddress indique la capacité de prendre en charge les opérations de suspension/reprise V.76 avec un champ d'adresse. La structure suspendResumeCapabilitywoAddress indique la capacité de prendre en charge les opérations de suspension/reprise V.76 sans champ d'adresse.

La structure rejCapability indique la capacité de la fonction de protection contre les erreurs de multiplexage V.76 d'effectuer un rejet.

sREJCapability indique la capacité de la fonction de protection contre les erreurs du multiplex à effectuer une opération de rejet sélectif.

mREJCapability indique la capacité de la fonction de protection contre les erreurs du multiplex à effectuer de multiples opérations de rejet sélectif.

8bitCRCCapability indique la capacité du multiplex à utiliser le codage CRC sur 8-bits.

La structure crc16bitsCapability indique la capacité du multiplex d'utiliser le contrôle CRC à 16 bits.

32bitCRCCapability indique la capacité du multiplex à utiliser le codage CRC sur 32-bits.

La structure uihCapability indique la prise en charge des trames V.76 d'informations UIH.

La structure numOfDLCS indique le nombre de connexions DLC que le multiplex V.76 peut prendre en charge.

La structure twoOctetAddressFieldCapability indique la capacité du multiplex V.76 de prendre en charge un champ d'adresse de deux octets.

La structure loopBackTestCapability indique la prise en charge du rebouclage selon la Recommandation V.76.

La structure n401Capability indique la valeur maximale du compteur N401 décrit dans la Recommandation V.76.

La structure maxWindowSizeCapability indique la longueur maximale de fenêtre que le multiplex V.76 peut prendre en charge.

7.2.2.5 Capacités vidéo

Ceci indique des capacités de mise en œuvre de 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 traitement simultané. La possibilité de traitement simultané peut être indiquée par des instances de structures VideoCapability dans des ensembles AlternativeCapabilitySets avec un descripteur CapabilityDescriptor unique.

H.261VideoCapability: indique les capacités définies dans 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/ou 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/ou 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 de capacités de réception.

maxBitRate indique le débit maximal en unités de 100 bit/s qu'un émetteur peut émettre en mode vidéo ou qu'un récepteur peut recevoir en mode vidéo.

Remplacée par une version plus récente

stillImageTransmission indique la capacité d'émettre des images comme indiqué dans l'Annexe D/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 «vrai» indique qu'une telle opération est possible, alors qu'une valeur «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 les capacités, mais également à l'intérieur des limites imposées par les champs facultatifs (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 facultatifs. Les champs facultatifs 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	Echantillons par ligne
linesPerFrame	Lignes par image
framesPerSecond	Index, frame_rate_code, dans le Tableau 6-4/H.262
luminanceSampleRate	Echantillons 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/ou le décodage des images de format SQCIF et, dans le cas contraire, 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/ou le décodage des images de format QCIF et, dans le cas contraire, 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/ou le décodage des images de format CIF et, dans le cas contraire, 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/ou le décodage des images de format CIF et, dans le cas contraire, 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/ou le décodage des images de format 16CIF et, dans le cas contraire, aucune capacité n'est indiquée pour les images de format 16CIF.

maxBitRate indique le débit maximal en unités de 100 bit/s auquel un émetteur peut émettre en mode vidéo ou un récepteur recevoir en mode vidéo.

Les variables booléennes unrestrictedVector, arithmeticCoding, advancedPrediction et pbFrames, quand elles sont mises à «vrai», indiquent la capacité de transmettre et/ou de recevoir ces modes facultatifs définis dans les annexes de H.263.

La capacité temporalSpatialTradeOffCapability, quand elle est mise à «vrai», indique que le codeur peut faire varier son compromis entre les résolutions temporelle et spatiale comme l'exige le terminal à distance. Cela n'a pas de signification quand cela fait partie de capacités de réception.

Le nombre entier hrd-B, lorsqu'il est présent, indique le paramètre B de la mémoire tampon HRD; on le mesure en multiples de 128 bits. Lorsqu'il n'est pas présent, la valeur par défaut définie dans la Recommandation H.263 s'applique. Il s'agit d'une capacité de réception qui n'a pas de signification quand elle fait partie d'ensembles de capacités de transmission.

Remplacée par une version plus récente

Le nombre entier `bppMax`, lorsqu'il est présent, indique le nombre maximal de bits d'une image codée que le récepteur peut recevoir et décoder correctement; on le mesure en multiples de 1024 bits. Lorsqu'il n'est pas présent, la valeur par défaut définie dans la Recommandation H.263 s'applique. Il s'agit d'une capacité de réception qui n'a pas de signification quand elle fait partie d'ensembles de capacités de transmission.

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

IS11172 VideoCapability: indique les capacités IS11172-2 [33].

`constrainedBitstream` indique la capacité des trains de bits dans lesquels le fanion `constrained_parameters` est mis à «1»: une valeur «vrai» indique qu'une telle opération est possible, alors qu'une valeur «faux» indique qu'une telle opération n'est pas possible. Un codeur doit produire des trains de bits compris dans les limites imposées par les champs facultatifs (voir ci-dessous). Un décodeur doit pouvoir accepter tous les trains de bits compris dans les limites indiquées par les champs facultatifs. Les champs facultatifs sont des nombres entiers dont les unités sont définies dans le Tableau 3.

TABLEAU 3/H.245

Unités pour les codages selon IS11172-2

Codage ASN.1	Unités pour le paramètre référencé
<code>videoBitRate</code>	400 bits par seconde
<code>vbvBufferSize</code>	16 384 bits
<code>samplesPerLine</code>	Echantillons par ligne
<code>linesPerFrame</code>	Lignes par image
<code>pictureRate</code>	Voir 2.4.3.2 de IS11172-2
<code>luminanceSampleRate</code>	Echantillons par seconde

7.2.2.6 Capacités audio

Cela indique les capacités audio. L'indication de plus d'une capacité unique dans une structure `AudioCapability` unique n'indique pas une capacité de traitement simultanée. Une capacité de traitement simultanée peut être indiquée par des instances de capacité `AudioCapability` dans différents ensembles `AlternativeCapabilitySets` avec un descripteur `CapabilityDescriptor` unique.

La capacité de transmettre ou 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 de mémoire tampon STD disponible, comptée en multiples de 256 octets. Quand un multiplex H.223 est utilisé, ces nombres se réfèrent au nombre maximal de trames audio par paquets AL-SDU. La signification exacte des codages est indiquée dans le Tableau 4.

IS11172AudioCapability: indique la possibilité d'utiliser l'audio codé selon ISO/CEI 11172-3 [34].

Les variables booléennes qui sont mises à la valeur «vrai» indiquent que le mode particulier de fonctionnement est possible, alors que 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 `audioSampling32k`, `audioSampling44k1` et `audioSampling48k` indiquent quelles sont les fréquences d'échantillonnage audio à respectivement 32 kHz, 44,1 kHz et 48 kHz qui peuvent être traitées. Les variables booléennes `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 possibilité maximale de débit audio et se mesure en multiples de kbits par seconde.

IS13818AudioCapability: indique la possibilité d'utiliser l'audio codé selon ISO/CEI 13818-3 [35].

Les variables booléennes qui sont mises à la valeur «vrai» indiquent que le mode particulier de fonctionnement est possible, alors que 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 fréquences d'échantillonnage à respectivement 16 kHz, 22,05 kHz, 24 kHz, 32 kHz, 44,1 kHz et 48 kHz peuvent être traitées.

Remplacée par une version plus récente

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
g723	G.723.1 soit à 5,3 soit à 6,4 kbit/s
g728	audio G.728 à 16 kbit/s
g729	audio G.729 à 8 kbit/s
g-729AnnexA	audio G.729 Annexe A à 8 kbit/s

Les variables booléennes relatives au fonctionnement multicanaux indiquent la capacité de fonctionner dans les modes particuliers, comme cela est spécifié dans le Tableau 5.

TABLEAU 5/H.245

Codages multi-canaux 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

La variable booléenne lowFrequencyEnhancement indique la capacité d'un canal de renforcement des basses fréquences.

La variable booléenne multilingue, quand elle mise à «vrai», indique la capacité de disposer d'un nombre de canaux multilingues allant jusqu'à sept, et quand elle est mise 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 kbits par seconde.

Remplacée par une version plus récente

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 traitement simultané. La capacité de traitement simultané peut être indiquée par des instances de structure `DataApplicationCapability` dans différents ensembles `AlternativeCapabilitySets` d'un descripteur unique `CapabilityDescriptor`.

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 retransmission un protocole de retransmission. Cette exigence est implicitement incluse dans les codages de capacités appropriés.

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

`maxBitRate` indique le débit maximal en unités de 100 bit/s auquel un émetteur peut émettre en mode vidéo ou auquel un récepteur peut recevoir l'application de données indiquée.

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

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

`userData` indique la capacité d'utiliser des données d'utilisateur non spécifiées à partir de ports de données externes.

`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].

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

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

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

NOTE – L'usage de NLPID est largement décrit dans IETC RFC1490, «Interconnexion multiprotocole sur relais de trame».

`V76Control` indique la capacité du terminal V76 à utiliser un canal de commande hors bande.

`h222DataPartitioning` indique la capacité de recourir à l'usage modifié ou restreint du mode subdivision de données de la Recommandation H.262, spécifié dans la Recommandation H.222.1, dans lequel les données de mise en valeur sont transmises sous la forme d'un canal de données accompagné par la structure `DataProtocolCapability` indiquée.

DataProtocolCapability: contient une liste des protocoles de données.

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

`v42lapm` indique la capacité d'utiliser une application spécifiée de données en utilisant le protocole LAPM défini dans 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 au 4.5.2 de ISO/CEI 3309 [32].

`h310SeparateVCStack` indique la capacité d'utiliser une application spécifiée de données à l'aide de la pile de protocoles définie dans la Recommandation H.310 pour le transport de messages H.245 sur une voie virtuelle ATM différente de celle qui est utilisée pour les communications audiovisuelles.

`h310SingleVCStack` indique la capacité d'utiliser une application spécifiée de données à l'aide de la pile de protocoles définie dans la Recommandation H.310 pour le transport de messages H.245 sur la voie virtuelle ATM utilisée pour les communications audiovisuelles.

`transparent` indique la capacité de faire fonctionner une application de données spécifiée en utilisant un transfert de données en mode transparent.

ProfilT84: 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.

Remplacée par une version plus récente

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 utilisée 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 qu'une télécopie Gr. 3 MH (Huffman modifié) codée sur deux niveaux en séquentiel à résolution normale (200 × 100 ppi) peut être utilisée.

g3FacsMH200x200 indique qu'une télécopie Gr. 3 MH (Huffman modifié) codée sur deux niveaux en séquentiel à résolution élevée (200 × 200 ppi) peut être utilisée.

g4FacsMMR200x100 indique qu'une télécopie Gr. 4 MMR (Reed modifié modifié) codée sur deux niveaux en séquentiel à résolution normale (200 × 100 ppi) peut être utilisée.

g4FacsMMR200x200 indique qu'une télécopie Gr. 4 MMR (Reed modifié modifié) codée sur deux niveaux en séquentiel à résolution élevée (200 × 200 ppi) peut être utilisée.

jbig200x200Seq indique qu'une image à deux niveaux de gris codée en JBIG sur deux niveaux en séquentiel à la résolution 200 × 200 ppi peut être utilisée.

jbig200x200Prog indique qu'une image à deux niveaux de gris codée en JBIG sur deux niveaux en progressif à la résolution 200 × 200 ppi peut être utilisée.

jbig300x300Seq indique qu'une image à deux niveaux de gris codée en JBIG sur deux niveaux en séquentiel à la résolution 300 × 300 ppi peut être utilisée.

jbig300x300Prog indique qu'une image à deux niveaux de gris codée en JBIG sur deux niveaux en progressif à la résolution 300 × 300 ppi peut être utilisée.

digPhotoLow indique qu'une image couleur codée selon JPEG en séquentiel, de format maximal 720 × 576, peut être utilisée.

digPhotoMedSeq indique qu'une image couleur codée selon JPEG en séquentiel, de format maximal 1440 × 1152, peut être utilisée.

digPhotoMedProg indique qu'une image couleur codée selon JPEG en progressif, de format maximal 1440 × 1152, peut être utilisée.

digPhotoHighSeq indique qu'une image couleur codée selon JPEG en séquentiel, de format maximal 2880 × 2304, peut être utilisée.

digPhotoHighProg indique qu'une image couleur codée selon JPEG en progressif, de format maximal 2880 × 2304, peut être utilisée.

7.2.3 Acquittement de l'ensemble de capacités des terminaux

Ceci 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.

Remplacée par une version plus récente

Les motifs de l'envoi de ce message sont indiqués dans le Tableau 6.

TABLEAU 6/H.245

Motifs de refus d'un ensemble TerminalCapabilitySet

Codage ASN.1	Motif
unspecified	Aucun motif de refus n'est spécifié
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 plus d'éléments que ceux indiqués dans le numéro highestEntryNumberProcessed ou n'a pu en stocker aucun

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 sur voie logique

Cet ensemble de messages est destiné à la signalisation sur voie logique. Le même ensemble de messages est utilisé pour la signalisation sur voie logique unidirectionnelle et sur canal logique bidirectionnel; toutefois, certains paramètres ne sont présents que dans le cas de la signalisation sur canal logique bidirectionnel.

Le terme «forward» (voie directe) désigne la transmission depuis le terminal qui a fait la demande initiale de voie logique en direction de l'autre terminal; le terme «reverse» (voie inverse) désigne le sens de transmission inverse, dans le cas d'une demande de canal bidirectionnel.

7.3.1 Ouverture de la voie logique

Ce message est utilisé pour les tentatives d'établissement d'une connexion de voie logique unidirectionnelle entre une entité LCSE sortante et une entité LCSE homologue entrante ou d'établissement d'un canal logique bidirectionnel entre une entité B-LCSE sortante et une entité B-LCSE homologue entrante.

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

forwardLogicalChannelParameters: contient les paramètres associés à la voie logique dans le cas où l'on cherche à ouvrir un canal monodirectionnel, de même que les paramètres associés à la voie logique directe dans le cas où l'on cherche à ouvrir un canal bidirectionnel.

reverseLogicalChannelParameters: contient les paramètres associés à la voie logique inverse dans le cas où l'on tente d'ouvrir un canal bidirectionnel. La présence de cette structure indique que le canal demandé est un canal logique bidirectionnel présentant les paramètres indiqués; l'absence de cette structure indique que le canal demandé est un canal logique unidirectionnel.

NOTE – Les paramètres H.222 ne sont pas contenus dans les structures reverseLogicalChannelParameters car leurs valeurs ne sont pas connues du terminal qui est à l'origine de la demande.

PortNumber est un paramètre utilisable de bout en bout par un module utilisateur pour des besoins tels qu'associer un port d'entrée ou de sortie, ou un numéro de voie d'une couche supérieure, à la voie logique.

dataType indique le type de données devant être transporté sur la voie logique.

Si le paramètre est nullData, la voie logique ne sera pas utilisée pour le transport des données de trains de bits élémentaires, mais seulement pour les informations de la couche d'adaptation – si la vidéo doit être transmise dans un sens seulement, mais qu'un protocole de retransmission tel que le protocole AL3 défini dans la Recommandation H.223 doit être utilisé, une voie de retour est nécessaire pour transporter les demandes de retransmission – il peut également être utilisé pour décrire une voie logique ne contenant que les valeurs PCR dans le cas des flux de transport (TS) de H.222.1 [7].

Remplacée par une version plus récente

Les terminaux qui ne peuvent fonctionner qu'en mode unidirectionnel (émission ou réception) sur des types de support qui utilisent des voies bidirectionnelles doivent envoyer des capacités uniquement pour le sens de fonctionnement assuré. Pour le sens inverse, le type `nullData`, qui ne nécessite aucune capacité, sera utilisé. Les terminaux en mode émission uniquement devraient envoyer des capacités d'émission, mais les terminaux ne devraient pas interpréter l'absence de capacités d'émission comme une impossibilité du fonctionnement en mode émission uniquement.

Si le paramètre est du type `VideoCapability` ou `AudioCapability`, la voie logique peut être utilisée pour l'une quelconque des variations indiquées par chaque capacité individuelle; en outre, on doit pouvoir effectuer des commutations entre ces variantes en n'utilisant que la signalisation dans la bande de la voie logique – par exemple, dans le cas de la vidéo H.261, si les formats QCIF et CIF sont indiqués, on doit pouvoir effectuer une commutation entre ces formats image par image. Dans le cas de la structure `DataApplicationCapability`, on ne peut indiquer qu'une seule instance d'une capacité, faute d'une signalisation dans la bande autorisant une commutation entre les variations.

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

La structure **H222LogicalChannelParameters**: est utilisée pour indiquer des paramètres spécifiques à l'utilisation de la Recommandation H.222.1 [7]. Ils doivent être présents dans les paramètres `forwardLogicalChannelParameters` et ne doivent pas être présents dans la structure `reverseLogicalChannelParameters`.

L'identificateur `resourceID` indique dans quelle voie virtuelle ATM la voie logique doit être transportée. Les moyens par lesquels ce paramètre est associé à une voie virtuelle (VC) de l'ATM ne sont pas spécifiés dans cette 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 processus PID pour un flux de transport (TS) et à l'identificateur de flux `stream_id` pour un flux de programme (PS).

Le numéro `pcr-pid` indique l'identificateur de processus PID utilisé pour le transport des références temporelles de programme 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 devant être transportées 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 le programme dont les informations devant être transportées dans la voie logique font partie.

La structure **H223LogicalChannelParameters**: est 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`;
- `allFramed` (mode tramé AL1);
- `al1NotFramed` (mode non tramé AL1);
- `al2WithoutSequenceNumbers` (AL2 sans numéro de séquence présent);
- `al2WithSequenceNumbers` (AL2 avec numéros de séquence présents);
- `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: est utilisé pour indiquer les paramètres propres aux applications V.76.

`audioheader` est utilisé pour indiquer l'utilisation d'un en-tête audio sur la voie logique. Ce paramètre est valable pour les voies du type de données audio.

`suspendResume` est utilisé pour indiquer que la voie peut appliquer à d'autres voies logiques des procédures de suspension ou de reprise du service.

`eRM` indique que la voie logique doit mettre en œuvre les procédures de reprise sur erreur définies dans la Recommandation V.76.

Remplacée par une version plus récente

uNERM indique que la voie logique doit fonctionner dans le mode sans reprise sur erreur défini dans la Recommandation V.76.

Pour la description des paramètres n401, windowSize et loopbackTestProcedure, voir 12.2.1/V.42 et ses sous-paragraphes.

crcLength est un paramètre facultatif qui indique la longueur du code CRC utilisé dans le mode de reprise sur erreur. En l'absence de ce paramètre, il convient d'utiliser la longueur du code CRC par défaut. Le paramètre 8bitCRC indique qu'il convient d'utiliser un code CRC à 8 bits; le paramètre 16bitCRC indique qu'il convient d'utiliser le code CRC à 16 bits; le paramètre 32bitCRC indique qu'il convient d'utiliser le code CRC à 32 bits défini dans la Recommandation V.76.

Le paramètre recovery est un paramètre facultatif qui indique les procédures de reprise sur erreur définies dans la Recommandation V.76. En l'absence de ce paramètre, il convient d'utiliser la procédure de reprise sur erreur par défaut. Le paramètre sEJ indique qu'il convient d'utiliser la procédure de rejet sélectif de trame; le paramètre mEJ indique qu'il convient d'utiliser la procédure de rejet sélectif multiple définie dans la Recommandation V.76.

La structure wAddress indique que la voie de suspension/reprise doit utiliser le champ d'adresse défini dans la Recommandation V.76. La structure woAddress indique que la voie de suspension/reprise ne doit pas utiliser le champ d'adresse.

La structure uIH indique l'utilisation de trames d'informations UIH selon la Recommandation V.76.

La structure rej indique l'utilisation de la procédure de rejet selon la Recommandation V.76.

La structure V75Parameters est utilisée pour indiquer un paramètre spécifique selon la Recommandation V.75. La structure audioHeaderPresent indique la présence d'un en-tête audio V.75.

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 LCSE homologue ou de l'entité B-LCSE. Dans le cas d'une demande d'ouverture d'un canal logique unidirectionnel, ce message indique l'acceptation de ce canal logique unidirectionnel. Dans le cas d'une demande d'ouverture d'un canal logique bidirectionnel, il indique l'acceptation de ce canal logique bidirectionnel ainsi que les paramètres appropriés de la voie inverse.

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

Les paramètres reverseLogicalChannelParameters ne sont présents que dans le cas où il est répondu à une demande d'ouverture de canal bidirectionnel, et seulement dans ce cas.

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

Le numéro portNumber est un paramètre d'utilisateur à utilisateur qu'un utilisateur peut utiliser par exemple pour associer un port d'entrée ou de sortie, ou un numéro de voie de couche supérieure, à la voie logique inverse.

Les paramètres multiplexParameters indiquent les paramètres propres au multiplexage H.222 ou H.223 utilisés à des fins de transport sur la voie logique inverse.

NOTE – Les paramètres H.223 ne figurent pas dans les paramètres reverseLogicalChannelParameters, leurs valeurs ayant été spécifiées dans le message de demande d'ouverture de voie logique OpenLogicalChannel.

7.3.3 Rejet de l'ouverture de la voie logique

On utilise ce message pour rejeter la demande de connexion de la voie logique en provenance de l'entité LCSE homologue ou de l'entité B-LCSE.

NOTE – Dans le cas d'une demande d'ouverture de canal bidirectionnel, le rejet s'applique à la fois à la voie directe et à la voie inverse. Il n'est pas possible d'accepter l'une et de rejeter l'autre.

Le numéro forwardLogicalChannelNumber indique le numéro de la voie logique directe spécifiée dans la demande que l'on rejette.

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

7.3.4 Confirmation de l'ouverture de la voie logique

On utilise ce message en signalisation bidirectionnelle pour indiquer à l'entité B-LCSE entrante que la voie inverse est ouverte et qu'elle peut être utilisée à des fins de transmission.

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

Remplacée par une version plus récente

TABLEAU 7/H.245

Motifs de rejet du message d'ouverture d'une voie logique OpenLogicalChannel

Codage ASN.1	Motif
unspecified	Aucun motif de rejet n'est spécifié
unsuitableReverseParameters	Ces paramètres ne doivent être utilisés que pour rejeter une demande d'ouverture de canal logique bidirectionnel lorsque le seul motif du rejet réside dans l'inadéquation des paramètres reverseLogicalChannelParameters demandés. Un tel rejet doit être suivi immédiatement par le lancement de procédures visant à ouvrir un canal logique bidirectionnel analogue mais acceptable
dataTypeNotSupported	Le terminal n'a pas été capable d'utiliser le paramètre type de données dataType indiqué dans OpenLogicalChannel
dataTypeNotAvailable	Le terminal n'a pas été capable d'utiliser simultanément le paramètre type de données dataType indiqué dans OpenLogicalChannel et le paramètre dataTypes correspondant aux voies logiques qui sont déjà ouvertes
unknownDataType	Le terminal n'a pas compris le type de données dataType indiqué dans OpenLogicalChannel
dataTypeALCombinationNotSupported	Le terminal n'a pas été capable d'utiliser simultanément le paramètre type de données dataType indiqué dans OpenLogicalChannel et le type de la couche d'adaptation indiqué dans les paramètres H.233 LogicalChannelParameters

7.3.5 Fermeture de la voie logique

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

NOTE – Dans le cas d'un canal logique bidirectionnel, cette opération ferme à la fois la voie directe et la voie inverse. Il n'est pas possible de fermer l'une et pas l'autre.

Le numéro forwardLogicalChannel indique le numéro de la voie logique directe qui va être fermée.

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

TABLEAU 8/H.245

Origine de la libération sur temporisation de la voie logique

Codage ASN.1	Motif
user (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. Elle peut provenir d'une erreur de protocole

7.3.6 Acquiescement de la fermeture de la voie logique

On l'utilise pour confirmer la fermeture d'une connexion de voie logique.

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

7.3.7 Demande de fermeture de voie

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

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

Remplacée par une version plus récente

7.3.8 Acquittement de demande de fermeture de voie

Ce message 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 partie aller de la voie logique dont la fermeture a été demandée.

7.3.9 Refus de demande de fermeture de voie

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

Le numéro forwardLogicalChannelNumber indique le numéro de la partie aller de la voie logique dont la fermeture a été demandée.

Le champ de cause indique la raison du rejet de la demande de fermeture de la voie logique. La seule valeur de cause n'est pas spécifiée.

7.3.10 Libération de demande de fermeture de voie

Ce message est envoyé par l'entité CLCSE sortante en cas de fin de temporisation.

Le numéro forwardLogicalChannelNumber indique le numéro de la partie aller de la voie logique dont la fermeture a été demandée.

7.4 Messages de signalisation des tableaux de multiplexage

Cet ensemble de messages est destiné à garantir la sûreté des transmissions des entrées de tableaux de multiplexage de H.223 de l'émetteur vers le récepteur.

7.4.1 Envoi d'entrée de multiplexage

Ceci est utilisé pour envoyer des entrées de tableau de multiplexage de l'émetteur vers le récepteur. L'envoi est effectué à partir d'une entité MTSE sortante à une entité MTSE entrante homologue.

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

MultiplexEntryDescriptors est un ensemble de descripteurs MultiplexEntryDescriptors de 1 à 15.

MultiplexEntryDescriptor: décrit une entrée unique de tableau de multiplexage. Il inclut le numéro MultiplexTableEntryNumber et une liste d'éléments MultiplexElements. Une liste manquante de longueur nulle indique que l'entrée est inactivée.

MultiplexElement: est une structure récursive décrivant un élément unique et un compteur de répétition. Si l'élément est du type numéro logicalChannelNumber, il indique un intervalle unique d'une voie logique donnée, et le compteur de répétition indique la longueur de l'intervalle en octets. Si l'élément est du type subElementList, il indique une séquence d'éléments MultiplexElements imbriqués, et le compteur de répétition indique le nombre de fois où la séquence est répétée. Dans chacun des cas, si le champ repeatCount vaut untilClosingFlag, cela signifie que l'élément doit être répété indéfiniment jusqu'à l'apparition de l'indicateur de fermeture de l'élément de protocole MUX-PDU.

Dans chaque descripteur MultiplexEntryDescriptor, le compteur de répétition repeatCount de l'élément MultiplexElement final dans la liste elementList doit être mis à «untilClosingFlag», et le compteur repeatCount de tous les autres éléments MultiplexElements dans la liste elementList sera mis à «fini». Ceci assure que toutes les entrées de tableau de multiplexage définissent un schéma de séquence multiplex de longueur indéfinie, se répétant jusqu'à l'apparition de l'indicateur de fermeture de l'élément de protocole MUX-PDU. Un descripteur d'entrée «MultiplexEntryDescriptor» avec un champ «elementList» manquant doit indiquer une entrée désactivée.

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

7.4.2 Acquittement d'envoi d'entrée de multiplexage

Ceci est utilisé pour confirmer la réception d'au moins un descripteur multiplexEntryDescriptors dans un message MultiplexEntrySend provenant de l'entité MTSE homologue.

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

Le numéro multiplexTableEntryNumber indique quels numéros d'entrée de tableau de multiplexage sont confirmés.

Remplacée par une version plus récente

7.4.3 Refus d'envoi d'entrée de multiplexage

On l'utilise pour refuser au moins un descripteur `multiplexEntryDescriptors` d'un message `MultiplexEntrySend` provenant de l'entité MTSE homologue.

Le numéro `sequenceNumber` doit être égal au numéro `sequenceNumber` dans le message d'envoi `MultiplexEntrySend` pour lequel ceci est le refus.

La structure `MultiplexEntryRejectionDescriptions` spécifie les entrées de tableau qui sont refusées et le motif de leur refus. Les motifs de refus sont indiqués dans le Tableau 9.

TABLEAU 9/H.245

Motifs du refus du message `MultiplexEntrySend`

Codage ASN.1	Motif
unspecified	Aucun motif de refus spécifié
descriptorTooComplex	Le descripteur <code>MultiplexEntryDescriptor</code> a dépassé la capacité du terminal récepteur

7.4.4 Echec du message d'envoi d'entrée de multiplexage

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

Le numéro `multiplexTableEntryNumber` indique à quelles entrées de tableau de multiplexage une fin de temporisation est applicable.

7.5 Messages de signalisation de demande de tableau de multiplexage

Cet ensemble de messages a pour but de faire aboutir la demande de retransmission d'un ou de plusieurs descripteurs `MultiplexEntryDescriptors` de l'émetteur au récepteur.

7.5.1 Demande d'entrée de multiplexage

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

La structure `entryNumbers` est une liste de numéros `MultiplexTableEntryNumbers` des descripteurs `MultiplexEntryDescriptors` pour lesquels la retransmission est demandée.

7.5.2 Acquiescement de demande d'entrée de multiplexage

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

La structure `entryNumbers` est une liste de numéros `MultiplexTableEntryNumbers` des descripteurs `MultiplexEntryDescriptors` à transmettre.

7.5.3 Rejet de demande d'entrée de multiplexage

Ce message est envoyé par l'entité RMESE pour indiquer que l'entrée de multiplexage sera transmise. La structure «`entryNumbers`» est une liste des numéros d'entrée de table de multiplexage des descripteurs d'entrée de multiplexage qui ne seront pas transmis. Les valeurs des numéros d'entrée de table de multiplexage contenues dans la structure «`entryNumbers`» doivent correspondre aux valeurs de la structure «`MultiplexTableEntryNumber`» contenue dans les descriptions de rejet, faute de quoi des erreurs peuvent se produire en cours d'exploitation.

La structure «`RequestMultiplexEntryRejectionDescriptions`» spécifie quelles sont les entrées de table de multiplexage qui vont être rejetées et la cause de ce rejet. Les causes de rejet sont indiquées dans le Tableau 10.

TABLEAU 10/H.245

Motifs du rejet du message `MultiplexEntrySend`

Codage ASN.1	Motif
unspecified	Aucun motif de rejet spécifié

Remplacée par une version plus récente

7.5.4 Abandon de la demande d'entrée de multiplexage

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

La structure `entryNumbers` est une liste de numéros `MultiplexTableEntryNumbers` des descripteurs `MultiplexEntryDescriptors` pour lesquels une temporisation s'est produite.

7.6 Messages relatifs au mode demande

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

7.6.1 Mode demande

Ce message est utilisé pour demander des modes particuliers de transmission au terminal émetteur. C'est par ordre de préférence, une liste (la première de préférence) de modes que le terminal voudrait recevoir. Chaque mode est décrit en utilisant une description `ModeDescription`.

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

ModeDescription: est un ensemble constitué d'au moins un élément de mode `ModeElement`.

ModeElement: est utilisé pour décrire un élément de mode, c'est-à-dire une des parties constituantes d'une description complète de mode. Il indique le type de train de bits élémentaires qui est demandé et, facultativement, comment est présentée la demande de multiplexage.

`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`.

`h223ModeParameters` est utilisé pour indiquer des paramètres spécifiques à l'utilisation H.223 [8].

`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);
- `al2WithSequenceNumbers` (AL2 avec numéros de séquence inclus); et
- `al3` (AL3 en indiquant le nombre d'octets du champ de commande inclus, 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 qu'un multiplexage segmentable est demandé, et quand il est égal à «faux», il indique qu'un multiplexage non segmentable est demandé.

7.6.1.1 Mode vidéo

Permet un choix de modes vidéo `VideoModes`.

H261 VideoMode: indique la résolution d'image demandée (format QCIF ou CIF), le débit, en unités de 100 bit/s, et la transmission d'images fixes.

H262 VideoMode: indique le profil et le niveau demandés; les champs facultatifs, s'ils sont présents, indiquent les valeurs demandées des paramètres indiqués. Les champs facultatifs sont des nombres entiers dont les unités sont définies au Tableau 2.

H263 VideoMode: indique la résolution d'image demandée (formats SQCIF, QCIF, CIF, 4CIF et 16CIF) ainsi que le débit, en unités de 100 bit/s.

Les variables booléennes `unrestrictedVector`, `arithmeticCoding`, `advancedPrediction` et `pbFrames`, quand elles sont mises à «vrai», indiquent qu'il est demandé d'utiliser les modes facultatifs définis dans les annexes de la Recommandation H.263.

IS11172 VideoMode: indique la demande de train de bits `constrainedBitstream` et les champs facultatifs, lorsqu'ils sont présents, indiquent les valeurs demandées des paramètres indiqués. Les champs facultatifs sont des nombres entiers dont les unités sont définies au Tableau 3.

Remplacée par une version plus récente

7.6.1.2 Mode audio

Ceci est un choix de modes audio AudioModes.

La signification exacte des codages audio pour la série G est indiquée dans le Tableau 3. Il y a quatre options pour l'audio conformément à la Recommandation G.723.1 audio, pour permettre de demander chacun des débits (le faible débit de 5,3 kbit/s ou le débit élevé de 6,3 kbit/s) avec ou sans la suppression des pauses de parole.

IS11172AudioMode: est utilisé pour demander que l'audio soit codé selon l'ISO/CEI 11172-3 [34].

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

Le paramètre audioSampling indique quelle fréquence d'échantillonnage est demandée: audioSampling32k, audioSampling44k1 et audioSampling48k indiquent les fréquences d'échantillonnage de l'audio équivalant respectivement à 32 kHz, 44,1 kHz et 48 kHz.

Le paramètre multichannelType indique quel mode multivoie est demandé: les paramètres singleChannel, twoChannelStereo et twoChannelDual demandent respectivement les modes de fonctionnement sur voie mono, sur voie stéréo et sur voie double.

Le paramètre bitRate indique le débit audio demandé, et se mesure en multiples de kbits par seconde.

IS13818AudioMode: est utilisé pour demander que l'audio soit codé selon l'ISO/CEI 13818-3 [35].

Le paramètre audioLayer indique quelle couche de codage est utilisée: soit la couche audioLayer1, audioLayer2 ou audioLayer3.

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

Le paramètre multichannelType indique le mode multivoie demandé, conformément au Tableau 11.

La variable booléenne lowFrequencyEnhancement, quand elle est mise à la valeur «vrai», demande un canal de renforcement des basses fréquences.

La variable booléenne multilingue, quand elle est mise à la valeur «vrai», demande jusqu'à sept canaux multilingues.

Le paramètre bitRate indique le débit binaire audio demandé, et se mesure en multiples de kbits par seconde.

TABLEAU 11/H.245

Codages multivoie selon ISO/CEI 13818-3

Codage ASN.1	Signification sémantique du codage
singleChannel	Une voie, utilisant la configuration 1/0. Mode monovoie (selon l'ISO/CEI 11172-3)
twoChannelStereo	Deux voies, utilisant la configuration 2/0, mode à voies stéréo (selon l'ISO/CEI 11172-3)
twoChannelDual	Deux voies, utilisant la configuration 2/0, mode à deux voies (selon l'ISO/CEI 11172-3)
threeChannels2-1	Trois voies, utilisant la configuration 2/1. Voie gauche, voie droite et voie ambiophonique
threeChannels3-0	Trois voies, utilisant la configuration 3/0. Voie gauche, voie centrale et voie droite, sans voie ambiophonique
fourChannels2-0-2-0	Quatre voies, utilisant la configuration 2/0 + 2/0. Voies gauche et droite du premier programme et voies gauche et droite du second programme
fourChannels2-2	Quatre voies, utilisant la configuration 2/2. Voies gauche, droite, ambiophonique gauche et ambiophonique droite
fourChannels3-1	Quatre voies, utilisant la configuration 3/1. Voies gauche, centrale, droite et une voie ambiophonique
fiveChannels3-0-2-0	Cinq voies, utilisant la configuration 3/0 + 2/0. Voies gauche, centrale et droite du premier programme et voies gauche et droite du second programme
fiveChannels3-2	Cinq voies, utilisant la configuration 3/2. Voies gauche, centrale, droite, ambiophonique gauche et ambiophonique droite

Remplacée par une version plus récente

7.6.1.3 Mode données

Permet un choix d'applications de transmission de données et de débits.

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

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

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

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

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

t434 demande l'utilisation de la Recommandation T.434 [26] pour le transfert de fichiers binaires télématiques.

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

nlpid demande l'utilisation de l'application des données sur les couches Liaison et Réseau spécifiées.

V76Control demande l'utilisation du terminal V76 en plus d'une voie de commande hors bande.

h222DataPartitioning demande le recours à l'usage modifié ou restreint du mode subdivision de données de la Recommandation H.262, spécifié dans la Recommandation H.222.1, dans lequel les données de mise en valeur sont transmises sous la forme d'un canal de données accompagné par la structure DataProtocolCapability indiquée.

7.6.1.4 Mode chiffrement

Ceci est un choix de modes de chiffrement.

Le paramètre h233chiffrement demande l'utilisation du chiffrement selon les Recommandations H.233 et H.234 [11] [12].

7.6.2 Acquiescement de mode demande

Ceci est envoyé pour confirmer que le terminal de transmission a l'intention de transmettre dans un des modes demandés par le terminal de réception.

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

Le champ de réponse indique l'action issue du terminal distant.

Les réponses possibles sont indiquées dans le Tableau 12.

TABLEAU 12/H.245

Réponses de confirmation de mode demande

Codage ASN.1	Réponse
willTransmitMostPreferredMode	Le terminal émetteur sera modifié selon le mode préféré du récepteur
willTransmitLessPreferredMode	Le terminal émetteur sera modifié selon un des modes préférés du récepteur, mais pas selon le premier choix

7.6.3 Refus de mode demande

Ceci est envoyé pour refuser la demande du terminal récepteur.

Le numéro sequenceNumber sera le même que le numéro sequenceNumber dans la structure RequestMode pour laquelle ceci est la réponse.

Le champ de cause indique la raison du rejet du mode demandé.

Remplacée par une version plus récente

Les réponses possibles sont indiquées dans le Tableau 13.

TABLEAU 13/H.245

Réponses de refus de 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 de conférence multipoint
requestDenied	Le terminal émetteur ne modifiera pas son mode d'émission

7.6.4 Libération de mode demande

Ceci est utilisé par l'entité MRSE sortante 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 entre deux terminaux en cours de communication. Il permet également à un utilisateur de H.245 de déterminer si le protocole H.245 homologue existe toujours.

7.7.1 Demande de temps de propagation aller et retour

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

Le numéro sequenceNumber est utilisé pour numéroter des instances de 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 doit être le même que le numéro sequenceNumber dans la demande RoundTripDelayRequest pour laquelle ceci est la réponse.

7.8 Messages de boucle de maintenance

Cet ensemble de messages est utilisé par un terminal pour commander des fonctions de boucle de maintenance.

7.8.1 Demande de boucle de maintenance

Ceci est envoyé pour demander un type particulier de boucle vers le demandeur. Les types mediaLoop et logicalChannelLoop demandent que la boucle soit établie vers le demandeur sur une seule voie logique comme cela est indiqué par le numéro LogicalChannelNumber, alors que le type systemLoop s'applique à toutes les voies logiques. La définition exacte de ces types est spécifique au système et se trouve à l'extérieur du champ d'application de la présente Recommandation.

7.8.2 Acquiescement de boucle de maintenance

Ce message est utilisé pour confirmer que le terminal effectuera la mise en boucle demandée.

7.8.3 Rejet de boucle de maintenance

Ce message est utilisé pour indiquer que le terminal n'effectuera pas la mise en boucle demandée.

Un terminal peut utiliser le motif canNotPerformLoop pour indiquer qu'il n'est pas en mesure d'effectuer la mise en boucle demandée.

Remplacée par une version plus récente

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 Instructions

Un message de commande nécessite qu'une action particulière soit prise, mais sans exiger de réponse explicite.

7.9.1 Envoi de l'ensemble de capacités du terminal

Le message `specificRequest` donne l'ordre au terminal distant d'indiquer ses capacités d'émission et de réception en envoyant un ou plusieurs ensembles `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 toute autre cause d'incertitude; cependant, de tels messages ne devraient pas être envoyés répétitivement sans raison majeure.

Un terminal ne doit demander la transmission que des ensembles `capabilityTableEntryNumbers` et `capabilityDescriptorNumbers` qu'il a précédemment reçus. Il ne doit pas tenir compte des demandes de transmission des ensembles `capabilityTableEntryNumbers` et `capabilityDescriptorNumbers` qu'il n'a pas précédemment transmis et on ne saurait considérer qu'il s'est produit une défaillance quelconque.

La variable booléenne `multiplexCapability`, quand elle est mise à «vrai», demande la transmission du paramètre `MultiplexCapability`.

L'ensemble `capabilityTableEntryNumbers` est un ensemble de numéros `CapabilityTableEntryNumbers` indiquant les éléments de la structure `CapabilityTableEntrys` dont le terminal demande la transmission.

L'ensemble `capabilityDescriptorNumbers` est un ensemble de numéros `CapabilityDescriptorNumbers` indiquant les descripteurs de capacités `CapabilityDescriptors` dont le terminal demande la transmission.

Le message `genericRequest` donne l'ordre au terminal distant d'envoyer son ensemble de capacités entier.

7.9.2 Chiffrement

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

`encryptionSE` est un message d'échange de sessions (SE) H.233, sauf que les bits de protection d'erreur décrits dans la Recommandation H.233 ne doivent pas être utilisés.

`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 `encryptionAlgorithmIdentifier` indique au récepteur que le terminal émetteur associera la valeur d'identificateur de l'algorithme H.233 à l'algorithme de chiffrement non normalisé.

7.9.3 Contrôle de flux

Cette instruction 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 `virtualChannelID`, la limite s'applique à la voie virtuelle ATM donnée; 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 inclus dans cette Recommandation, mais devraient être spécifiés par des recommandations utilisant la présente Recommandation.

Remplacée par une version plus récente

Chaque transmission de ce message affecte une voie logique spécifique ou l'ensemble du multiplex. Plus d'une de ces instructions peut avoir cours en même temps, jusqu'à atteindre le nombre de voies logiques plus une, pour donner les limites 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 que 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.

7.9.4 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 cette 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 sera utilisé sur le RTGC.

Les options possibles sont indiquées dans le Tableau 14.

TABLEAU 14/H.245

Options après l'instruction EndSessionCommand en cas d'utilisation d'un modem de la série V sur le RTGC

Codage ASN.1	Option
telephonyMode	Le terminal doit lancer les procédures de libération définies dans la Recommandation pertinente de la série V relative au modem, sauf qu'il ne doit pas déconnecter physiquement la connexion RTPC
v8bis	Le terminal doit lancer les procédures de libération définies dans la Recommandation pertinente de la série V relative au modem et ouvrir une session V.8 bis
v34 V76	Le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour assurer le fonctionnement de ce modem conformément à la Recommandation V.76
v34DuplexFAX	Le terminal doit maintenir la connexion de mode V.34, mais l'utiliser pour assurer le fonctionnement de ce modem conformément à la Recommandation T.30 FAX [21]
v34H324	Le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour assurer le fonctionnement de ce modem conformément à la Recommandation H.324 [18]

7.9.5 Instructions diverses

On les emploie pour toute une série d'instructions, dont certaines figurent dans les Recommandations H.221 et H.230 [5] et [10], respectivement.

Le numéro 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 instructions sont du type videoFreezePicture, videoFastUpdatePicture, videoFastUpdateGOB, videoTemporalSpatialTradeOff, videoSendSyncEveryGOB et videoSendSyncEveryGOBCancel.

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

L'instruction multipointModeCommand donne l'ordre à un terminal de réception de se conformer à toutes les demandes requestMode émises par l'unité MCU. Le passage du codage audio G.711 au codage audio G.728 constitue un exemple de changement de mode.

L'instruction cancelMultipointModeCommand annule une instruction multipointModeCommand précédemment envoyée.

L'élément videoFreezePicture donne l'instruction 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 de commande de libération d'image gelée approprié.

Remplacée par une version plus récente

L'élément `videoFastUpdatePicture` donne l'instruction au décodeur vidéo de passer au mode de rafraîchissement accéléré à la première opportunité.

L'élément `videoFastUpdateGOB` donne l'instruction 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.

L'élément `videoTemporalSpatialTradeOff` donne les instructions 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.

L'élément `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 l'instruction `videoSendSyncEveryGOBCancel` soit reçue, moment à partir duquel le codeur vidéo distant peut décider de la fréquence de synchronisation des groupes de blocs (GOB). Ces instructions ne doivent être utilisées qu'avec le codage vidéo conforme à la Recommandation H.263.

7.10 Indications d'état

Une indication d'état contient des informations ne nécessitant pas qu'une action soit effectuée ou qu'une réponse soit apportée.

7.10.1 Fonction non disponible

Ceci est utilisé pour renvoyer des demandes, des réponses et des instructions qui n'ont pas été comprises par celui qui les transmet.

L'ensemble du paquet sous la forme de `RequestPDU`, `ResponsePDU` ou `CommandPDU` est renvoyé.

Si un terminal reçoit une demande, une réponse ou un ordre qu'il ne comprend pas, soit parce que ce message n'est pas normalisé soit parce que ce message a été défini dans une révision ultérieure de la présente Recommandation, il y a lieu que ce terminal réponde par l'envoi du message «`FunctionNotSupported`».

7.10.2 Indications d'état diverses

On les emploie pour diverses indications d'état dont certaines sont présentes dans les Recommandations H.221 et H.230 [5] et [10], respectivement.

Le numéro `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`.

L'élément `logicalChannelInactive` est utilisé pour indiquer que le contenu de la voie logique ne représente pas un signal normal. Son rôle est analogue à AIM et VIS définis dans H.230 [10].

L'élément `logicalChannelActive` est complémentaire de l'élément `logicalChannelInactive`. Son rôle est analogue à AIA et VIA définis dans la Recommandation H.230. Les éléments `MultipointZeroComm`, `cancelMultipointZeroComm`, `multipointSecondaryStatus` et `cancelMultipointSecondaryStatus` doivent avoir la même signification que respectivement MIZ, `cancelMIZ`, MIS et `cancelMIS`, comme cela a été défini dans la Recommandation H.230 [10].

L'élément `multipointConference` indique que le terminal participe à une conférence multipoint H.243 et qu'on attend de lui qu'il obéisse à la symétrisation du débit. Cependant, cette symétrisation du débit sera assurée à l'aide des messages de contrôle de flux `FlowControl`. Il est à noter que l'élément `multipointConference` a exactement la même signification que le MCC de la Recommandation H.230 [10]. Il est également à noter que l'élément `multipointConference`, comme le MCC, n'exige pas de symétrie de mode.

L'élément `videoIndicateReadyToActivate` doit avoir la même signification que VIR comme cela a été défini dans la Recommandation H.230, c'est-à-dire qu'il est transmis par un terminal dont l'utilisateur a décidé de ne pas envoyer de signaux vidéo, à moins qu'il reçoive également la vidéo provenant de l'autre extrémité.

L'élément `videoTemporalSpatialTradeOff` indique au décodeur vidéo distant son compromis en cours entre la définition temporelle et spatiale. Une valeur de 0 indique une définition spatiale élevée et une valeur de 31 indique une fréquence de trame élevée. Les valeurs s'échelonnant entre 0 et 31 indiquent de façon monotone une fréquence de trame supérieure.

Remplacée par une version plus récente

Les valeurs réelles ne correspondent pas à des valeurs précises de la définition spatiale ou de la fréquence de trame. Un terminal ayant indiqué une capacité temporalTradeOffCapability doit transmettre cette indication à chaque fois que le compromis est modifié et au moment de l'ouverture d'une voie logique pour la vidéo.

7.10.3 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 ou le surremplissage et le sous-remplissage de la mémoire tampon du décodeur, étant donné 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 l'objet est du type logicalChannelNumber, les informations s'appliquent à une voie logique donnée; quand l'objet est du type virtualChannelID, les informations s'appliquent au canal virtuel (VC) de l'ATM donné; et quand l'objet est du type wholeMultiplex, les informations s'appliquent à l'ensemble du multiplex.

Les éléments estimatedReceivedJitterMantissa et estimatedReceivedJitterExponent permettent d'évaluer la gigue qui a été 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

estimatedReceivedJitterMantissa	Mantisse
0	1
1	2,5
2	5
3	7,5

estimatedReceivedJitterExponent indique l'exposant de l'évaluation de gigue tel qu'il est donné 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.

Remplacée par une version plus récente

Le compteur `skippedFrameCount` indique le nombre de trames qui a été omis par le décodeur depuis la dernière réception de `JitterControlPDU`. Etant donné que la valeur maximale de codage est 15, si cette option est implémentée, 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` selon H. 262 [14].

7.10.4 Indication de décalage temporel selon H.223

Ceci 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 dans le multiplex, pour réaliser la synchronisation avec la voie logique `logicalChannelNumber1`, observée dans le multiplex également. Le retard effectif nécessaire à la synchronisation dépend de l'implémentation du décodeur et est du ressort du récepteur.

7.10.5 Indication d'état de nouvelle voie virtuelle ATM

Cette indication est utilisée pour indiquer les paramètres d'une voie virtuelle ATM que le terminal se propose d'ouvrir.

`resourceID` est utilisé pour identifier la voie virtuelle ATM. Les moyens par lesquels ce paramètre est associé à une voie virtuelle ATM ne sont pas spécifiés dans la présente Recommandation.

`bitRate` indique le débit, mesuré au point AAL-SAP de la voie virtuelle, en unités de 64 kbit/s.

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

`bitRateLockedToNetworkClock` indique que le débit de la voie virtuelle est synchronisé sur l'horloge du réseau local. Cela ne garantit pas que l'horloge de débit sera synchronisée sur le réseau local au niveau du récepteur, des horloges de réseau communes n'étant pas forcément disponibles.

`aal` indique la couche d'adaptation ATM qui sera utilisée ainsi que les paramètres de cette couche.

La séquence `aal1` indique les options de la couche d'adaptation 1 du mode ATM, spécifiée dans la Recommandation I.363 [19], mises en œuvre. Les codages sont définis dans le Tableau 1.

La séquence `aal5` indique les options de la couche d'adaptation 5 du mode ATM, spécifiée dans la Recommandation I.363 [19], mises en œuvre. `forwardMaximumSDUSize` et `backwardMaximumSDUSize` indiquent la taille maximale des unités CPCS-SDU dans le sens direct et dans le sens inverse, mesurée en octets.

`multiplex` indique le type de multiplexage qui sera utilisé sur la voie virtuelle ATM. Les options sont `noMultiplex` (pas de multiplexage H.222.0), `H.222.0 TransportStream` et `H.222.0 ProgramStream`.

7.10.6 Données de l'utilisateur

Ceci est utilisé pour les données 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.

NOTE – Toutes les données transmises dans la Recommandation H.245, y compris les données de l'utilisateur ne seront pas chiffrées.

8 Procédures

8.1 Introduction

Ce 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.

Remplacée par une version plus récente

Les procédures destinées à effectuer les fonctions suivantes sont décrites dans ce paragraphe:

- choix du mode maître ou esclave;
- échange de capacités du terminal;
- signalisation de voie logique;
- signalisation de canal logique bidirectionnel;
- demande de fermeture de voie logique du terminal de réception;
- demande d'ouverture de canal logique bidirectionnel du terminal esclave;
- modification d'entrée de tableau de multiplexage selon H.223;
- mode demande de transmission de récepteur à émetteur;
- définition du temps de propagation aller et retour.

8.1.1 Méthode de spécification

Les procédures sont généralement spécifiées dans ce 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 de protocole. Les primitives doivent avoir pour objectif de définir des procédures de protocole et non pas de spécifier ou de rendre obligatoire une implémentation 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. Ces états 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 au moyen d'un diagramme de transition d'états. 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 des entités de protocole décrites possèdent également des temporisateurs, qui leur sont associés.

Un temporisateur est identifié par la notation T_n , où n est un nombre. Dans les diagrammes SDL, le réglage d'un temporisateur signifie que celui-ci est chargé avec une valeur spécifiée et qu'il est lancé. Le réarmement d'un temporisateur signifie que celui-ci est arrêté avec conservation de sa valeur au moment de l'arrêt. L'expiration d'un temporisateur signifie que celui-ci a fonctionné pendant le temps qui lui a été imparti et qu'il a atteint la valeur zéro.

Une entité de protocole peut également avoir des paramètres associés. Un paramètre est identifié par la notation N_n , où n est un nombre.

Ces temporisateurs et compteurs sont énumérés dans l'Appendice III.

Certaines entités de protocole définissent une primitive d'erreur pour signaler des conditions d'erreur protocolaire à une entité de gestion.

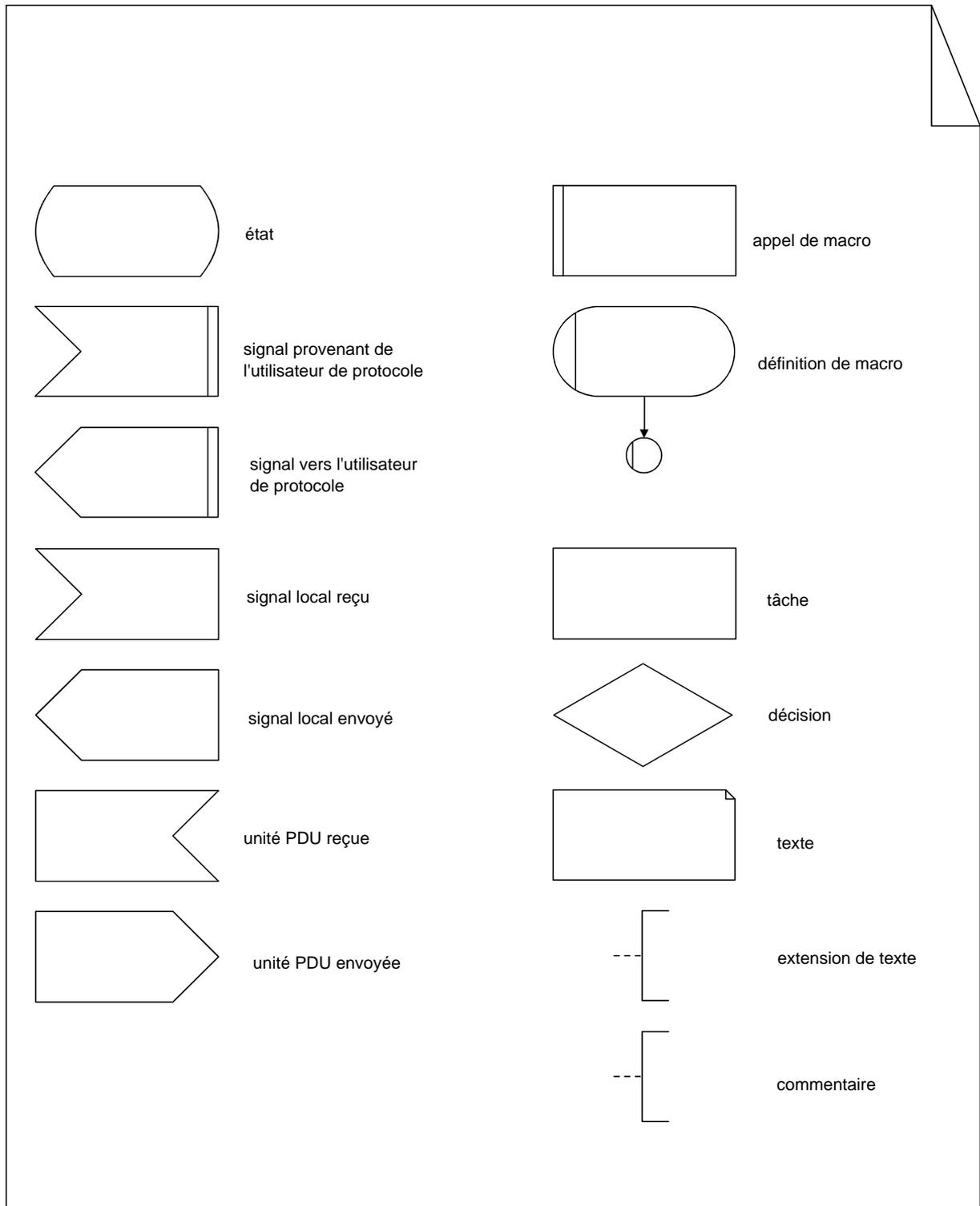
8.1.4 Diagrammes SDL

Les diagrammes SDL indiquent les actions dues aux 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, tel que spécifié par les diagrammes de transition d'état, ne sont pas indiquées dans les diagrammes SDL. Les réponses à la réception de messages non pertinents sont cependant décrites dans les diagrammes SDL.

Remplacée par une version plus récente

8.1.5 Langage SDL clé

Le langage SDL clé est décrit dans la Figure 1.



T1519120-95/d001

FIGURE 1/H.245

Langage SDL clé

Remplacée par une version plus récente

8.2 Procédures de choix du mode maître ou esclave

8.2.1 Introduction

Des conflits peuvent surgir lorsqu'au moins deux terminaux en communication déclenchent simultanément des événements semblables, pour lesquels les ressources disponibles n'autorisent qu'une seule occurrence de l'événement, par exemple: l'ouverture de voies logiques. Pour résoudre de tels conflits, un terminal pourra agir en tant que maître cependant que l'autre terminal ou les autres terminaux agira(ont) en tant que terminal(aux) esclave(s). Les procédures décrites ici permettent aux terminaux en communication de définir quel sera le terminal maître et quel(s) sera(seront) le(s) terminal(aux) esclave(s).

Le protocole décrit ici est désigné comme l'entité de signalisation permettant le choix du mode maître ou esclave (MSDSE). Il y a une instance d'entité MSDSE dans chaque terminal mis en communication.

Chacun des terminaux peut déclencher le processus de choix du mode maître ou esclave en transmettant la primitive de demande DETERMINE à son entité MSDSE. Le résultat de la procédure est renvoyé par les primitives d'indication et de confirmation DETERMINE. La primitive d'indication DETERMINE indique le résultat, sans préciser si le terminal distant en a connaissance. La primitive de confirmation DETERMINE indique le résultat et confirme que le terminal distant en a lui aussi connaissance.

Un terminal doit répondre à des procédures subordonnées à la connaissance du résultat et lancées par le terminal distant à un moment quelconque après que le résultat indiquant le choix de l'état soit connu du terminal local. Cette opération pourra intervenir avant que le terminal local ait reçu confirmation que le terminal distant a lui aussi connaissance du résultat. Un terminal ne doit pas lancer de procédures subordonnées à la connaissance du résultat avant d'avoir reçu confirmation que le terminal distant a lui aussi 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 a priorité.

8.2.1.1 Aperçu général du protocole – Lancement par l'utilisateur local

Une procédure de choix du mode maître ou esclave est lancée lorsque la primitive de demande DETERMINE est émise par l'utilisateur de l'entité MSDSE. Un message «MasterSlaveDetermination» est envoyé à l'entité MSDSE homologue et le temporisateur T106 est lancé. Si un message «MasterSlaveDeterminationAck» est reçu en réponse au message «MasterSlaveDetermination», 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 été appliquée et un message «MasterSlaveDeterminationAck» est envoyé à l'entité MSDSE homologue. Si cependant un message «MasterSlaveDeterminationReject» est reçu en réponse au message «MasterSlaveDetermination», un nouveau numéro de détermination d'état est produit, le temporisateur T106 est relancé et un autre message «MasterSlaveDetermination» est envoyé. Si, après l'envoi N100 fois du message «MasterSlaveDetermination» aucun message «MasterSlaveDeterminationAck» n'est toujours pas reçu, 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 pu donner de résultat.

Si le temporisateur T106 expire, l'utilisateur de l'entité MSDSE en 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 – Lancement par l'utilisateur distant

Lorsqu'un message «MasterSlaveDetermination» est reçu à l'entité MSDSE, une procédure de détermination du statut est lancée. Si cette procédure renvoie un résultat déterminé, l'utilisateur en est informé par la primitive d'indication DETERMINE, un message «MasterSlaveDeterminationAck» est envoyé à l'entité MSDSE homologue et le temporisateur T106 est armé. Si un message «MasterSlaveDeterminationAck» est reçu en réponse au message «MasterSlaveDeterminationAck», 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 été appliquée.

Si le temporisateur T106 expire, l'utilisateur de l'entité MSDSE en est informé par la primitive d'indication REJECT.

Si toutefois la procédure de détermination du statut renvoie un résultat indéterminé, le message «MasterSlaveDeterminationReject» est envoyé à l'entité MSDSE homologue.

8.2.1.3 Aperçu général du protocole – Lancement simultané

Si un message «MasterSlaveDetermination» est reçu à l'entité MSDSE qui elle-même a déjà lancé une procédure de détermination de statut et qui attend un message «MasterSlaveDeterminationAck» ou «MasterSlaveDeterminationReject»,

Remplacée par une version plus récente

une procédure de détermination de statut est lancée. Si cette procédure renvoie un résultat déterminé, l'entité MSDSE répond comme si la procédure avait été lancée par l'utilisateur distant et les procédures décrites ci-dessus pour cette condition s'appliquent.

Si toutefois la procédure de détermination du statut renvoie un résultat indéterminé, un nouveau numéro de détermination de statut est produit et l'entité MSDSE répond comme si la procédure avait de nouveau été lancée par l'utilisateur de l'entité MSDSE locale, comme décrit ci-dessus.

8.2.1.4 Procédure de détermination du statut

La procédure suivante est utilisée pour déterminer quel est le terminal maître, à partir des valeurs des champs «terminalType» et «statusDeterminationNumber». On compare d'abord les valeurs de «terminalType» et le terminal ayant le plus grand numéro de type de terminal est choisi comme étant le maître. Si les numéros de type de terminal sont identiques, les numéros de détermination de statut sont comparés par arithmétique modulo, afin de déterminer quel est le maître.

Si les deux terminaux ont des valeurs égales dans le champ «terminalType» et si la différence entre les valeurs du 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 établies au moyen des primitives indiquées dans le Tableau 17.

TABLEAU 17/H.245

Primitives et paramètres

Nom générique	Type			
	demande	indication	réponse	confirmation
DETERMINE	– (Note 1)	TYPE	Non défini (Note 2)	TYPE
REJECT	Non défini	–	Non défini	Non défini
ERROR	Non défini	ERRCODE	Non défini	Non défini

NOTE 1 – «–» signifie qu'il n'y a pas de paramètres.
NOTE 2 – « non défini » signifie que cette primitive n'est pas définie.

8.2.2.2 Définition des primitives

Ces primitives sont définies comme suit:

- La primitive DETERMINE est utilisée pour lancer et pour retourner le résultat de la procédure de choix du mode maître ou esclave.

La primitive de demande DETERMINE est utilisée pour lancer la procédure de choix du mode maître ou esclave.

La primitive d'indication DETERMINE est utilisée pour indiquer le résultat de la procédure de choix du mode maître ou esclave. Le résultat de cette procédure pouvant ne pas être connu du terminal distant, celui-ci ne doit pas lancer les procédures subordonnées à la connaissance du résultat, bien qu'il soit tenu de répondre à de telles procédures.

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 le résultat de cette procédure est connu des deux terminaux. Le terminal peut lancer des procédures subordonnées à la connaissance du résultat et doit y répondre.

- La primitive REJECT indique que la procédure de choix du mode maître ou esclave n'a pas abouti.
- La primitive ERROR signale les erreurs de l'entité MSDSE à une entité de gestion.

Remplacée par une version plus récente

8.2.2.3 Définition des paramètres

Les paramètres de primitives indiqués dans le Tableau 17 sont définis comme suit:

- a) le paramètre TYPE indique l'état du terminal. Il a pour valeur 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 peut prendre le paramètre ERRCODE.

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:

état 0: IDLE

Aucune procédure de choix du mode maître ou esclave n'a été lancée.

état 1: OUTGOING AWAITING RESPONSE

L'utilisateur de l'entité MSDSE local a demandé le lancement de la procédure de choix du mode maître ou esclave. Une réponse de l'entité MSDSE distante est attendue.

état 2: INCOMING AWAITING RESPONSE

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 de l'entité MSDSE distante est attendue.

8.2.2.5 Diagramme de transition 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.

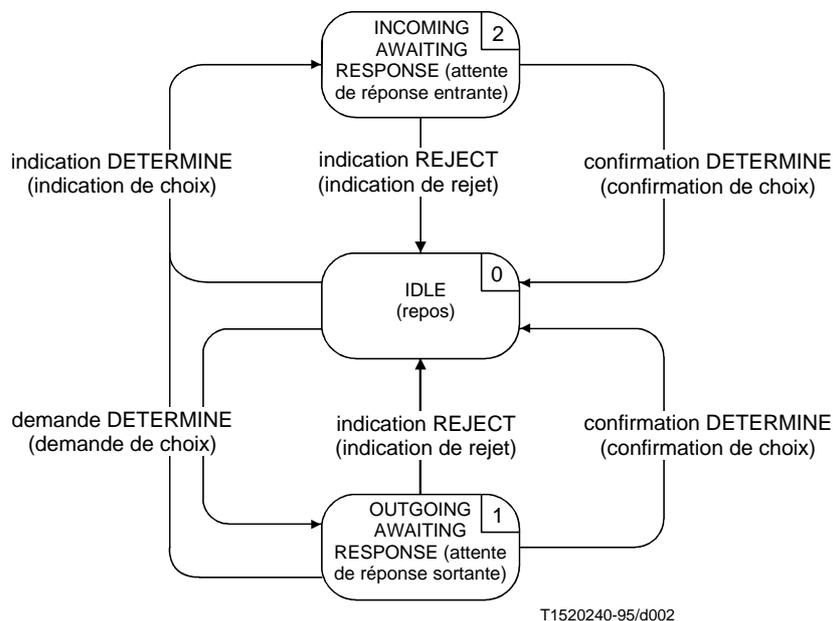


FIGURE 2/H.245

Diagramme de transition d'états pour la séquence de primitives d'une 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 MSDSE tels qu'ils ont été définis au paragraphe 6 et qui relèvent du protocole MSDSE.

Remplacée par une version plus récente

TABLEAU 18/H.245

Noms et champs des messages d'une entité MSDSE

Fonction	Message	Champ
Determination (choix)	MasterSlaveDetermination	terminalType statusDeterminationNumber
	MasterSlaveDeterminationAck	Decision
	MasterSlaveDeterminationReject	Cause (motif)
Error recovery	MasterSlaveDeterminationRelease	–

8.2.3.2 Variables d'état d'une entité MSDSE

Les variables d'état suivantes de l'entité MSDSE sont définies comme suit:

sv_TT

Cette variable d'état contient le numéro de type de terminal pour ce terminal.

sv_SDNUM

Cette variable d'état contient le numéro du choix de l'état 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 prend pour valeurs «master», «slave» et «indeterminate».

sv_NCOUNT

Cette variable d'état est utilisée pour compter le nombre de messages MasterSlaveDetermination envoyés pendant l'état OUTGOING AWAITING RESPONSE.

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 (attente de réponse sortante) et pendant l'état INCOMING AWAITING RESPONSE (attente de réponse entrante). Il spécifie le temps maximal autorisé pendant lequel aucun message d'accusé de réception 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 la variable d'état sv_NCOUNT.

8.2.4 Procédures de l'entité MSDSE

8.2.4.1 Introduction

La Figure 3 récapitule les primitives MSDSE et leurs paramètres, ainsi que les messages.

8.2.4.2 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs par défaut indiquées dans le Tableau 19, sauf lorsque leurs valeurs sont expressément mentionnées dans les diagrammes SDL.

Remplacée par une version plus récente

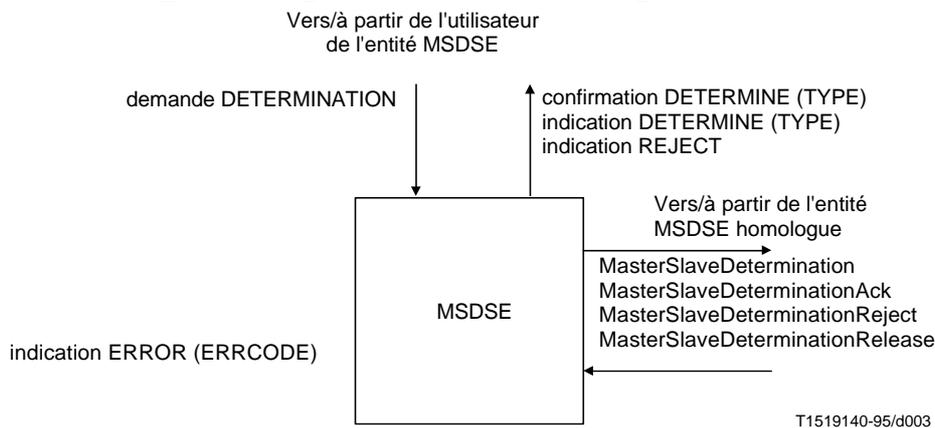


FIGURE 3/H.245

Primitives et messages dans l'entité MSDSE

TABLEAU 19/H.245

Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut
confirmation DETERMINE	TYPE	MasterSlaveDeterminationAck.decision
indication DETERMINE	TYPE	sv_STATUS

8.2.4.3 Valeurs par défaut des champs de messages

Les champs de messages prennent les valeurs par défaut indiquées dans le Tableau 20, sauf lorsque leurs valeurs sont expressément mentionnées dans les diagrammes SDL.

TABLEAU 20/H.245

Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut
MasterSlaveDetermination	terminalType statusDeterminationNumber	sv_TT sv_SDNUM
MasterSlaveDeterminationAck	decision	Contraire de sv_STATUS, c'est-à-dire si (sv_STATUS = master) decision = slave si (sv_STATUS = slave) decision = master
MasterSlaveDeterminationReject	cause	identicalNumbers

8.2.4.4 Valeurs du paramètre ERRCODE

Le Tableau 21 indique les valeurs que le paramètre ERRCODE de la primitive d'indication ERROR peut prendre pour l'entité MSDSE.

Remplacée par une version plus récente

TABLEAU 21/H.245

Valeurs du paramètre ERRCODE dans l'entité MSDSE

Type d'erreur	Code d'erreur	Situation d'erreur	Etat
Pas de réponse de l'entité MSDSE distante	A	expiration du temporisateur local T106	OUTGOING AWAITING RESPONSE INCOMING AWAITING RESPONSE
L'entité MSDSE distante constate qu'aucune réponse ne lui parvient de l'entité MSDSE locale	B	expiration du temporisateur distant T106	OUTGOING AWAITING RESPONSE INCOMING AWAITING RESPONSE
Message incorrect	C	MasterSlaveDetermination	INCOMING AWAITING RESPONSE
	D	MasterSlaveDeterminationReject	INCOMING AWAITING RESPONSE
Valeur de champ incompatible	E	MasterSlaveDeterminationAck. decision != sv_STATUS	INCOMING AWAITING RESPONSE
Nombre maximal de nouvelles tentatives	F	sv_NCOUNT = N100	OUTGOING AWAITING RESPONSE

8.2.4.5 Description SDL

Les procédures MSDSE sont décrites en langage SDL dans la Figure 4.

L'opération terminalTypeProcess permet de renvoyer un numéro identifiant des types de terminaux différents – terminaux, unités MCU ou centres tête de ligne, par exemple.

L'opération randomNumber permet de renvoyer un nombre aléatoire compris dans la gamme 0..2²⁴-1.

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). 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 sortante et une entité CESE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité CESE pour chaque appel.

Tous les terminaux devant être utilisés dans les applications de communication point à point ou ceux connectés à un pont de conférence MCU doivent pouvoir reconnaître un ensemble TerminalCapabilitySet et sa structure, de telles valeurs de capacités étant à cet égard obligatoires pour ces applications, toutes les valeurs de capacité non reconnues doivent être omises, et cela ne doit pas produire de défaillance.

L'échange de capacités peut être effectué à n'importe quel moment. L'échange de capacités peut signaler à la fois des capacités modifiées et des capacités non modifiées. Des capacités non modifiées 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 par rapport à la spécification formelle suivante du protocole, c'est celle-ci qui a priorité.

8.3.1.1 Aperçu général du protocole – entité CESE sortante

Un échange de capacités est lancé lorsque la primitive de demande TRANSFER est émise par l'utilisateur à l'entité CESE sortante. Un message TerminalCapabilitySet est envoyé à l'entité CESE homologue entrante et le temporisateur T101 est lancé. Si un message TerminalCapabilitySetAck est reçu en réponse au message TerminalCapabilitySet, le temporisateur T101 est arrêté et l'utilisateur est informé, par la primitive de confirmation TRANSFER, que l'échange de capacités a été réalisé. Si toutefois un message TerminalCapabilitySetReject est reçu en réponse au message TerminalCapabilitySet, le temporisateur T101 est arrêté et l'utilisateur est informé, par la primitive d'indication REJECT, que l'entité CESE homologue a refusé l'échange de capacités.

Remplacée par une version plus récente

Si le temporisateur T101 expire, l'utilisateur de l'entité CESE sortante en est informé par la primitive d'indication REJECT et un message TerminalCapabilitySetRelease est envoyé.

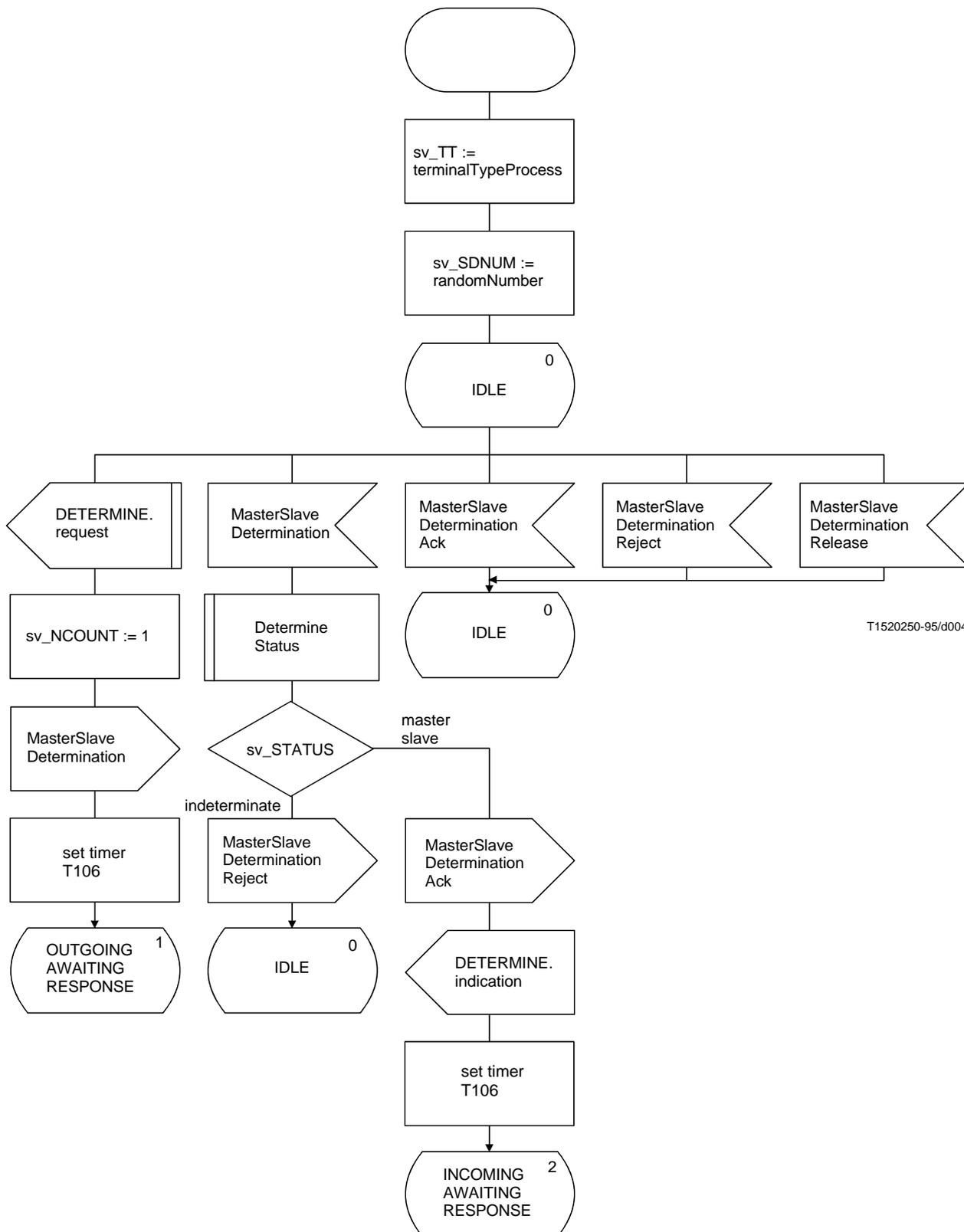


FIGURE 4 i)/H.245

Diagramme SDL de l'entité MSDSE

Remplacée par une version plus récente

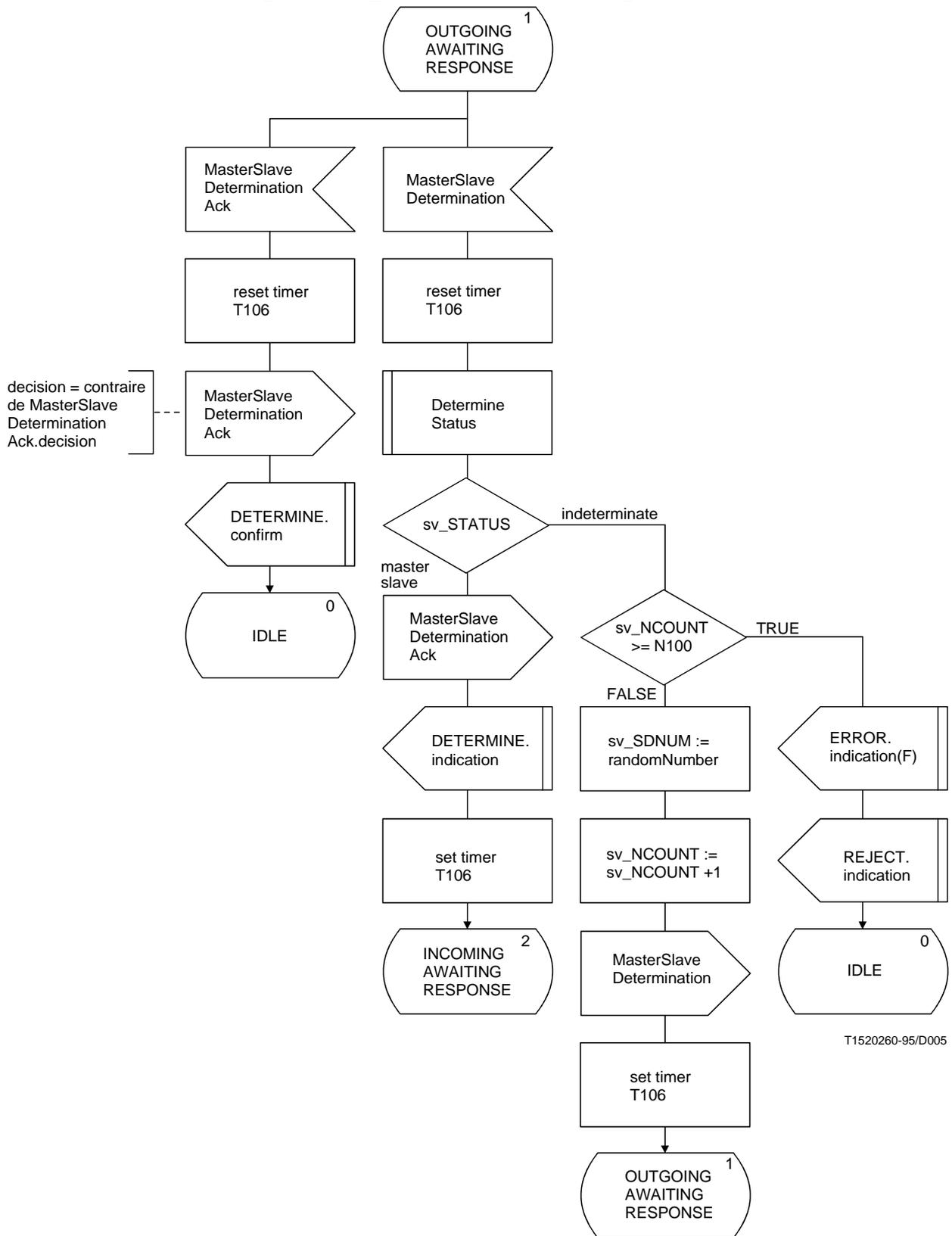
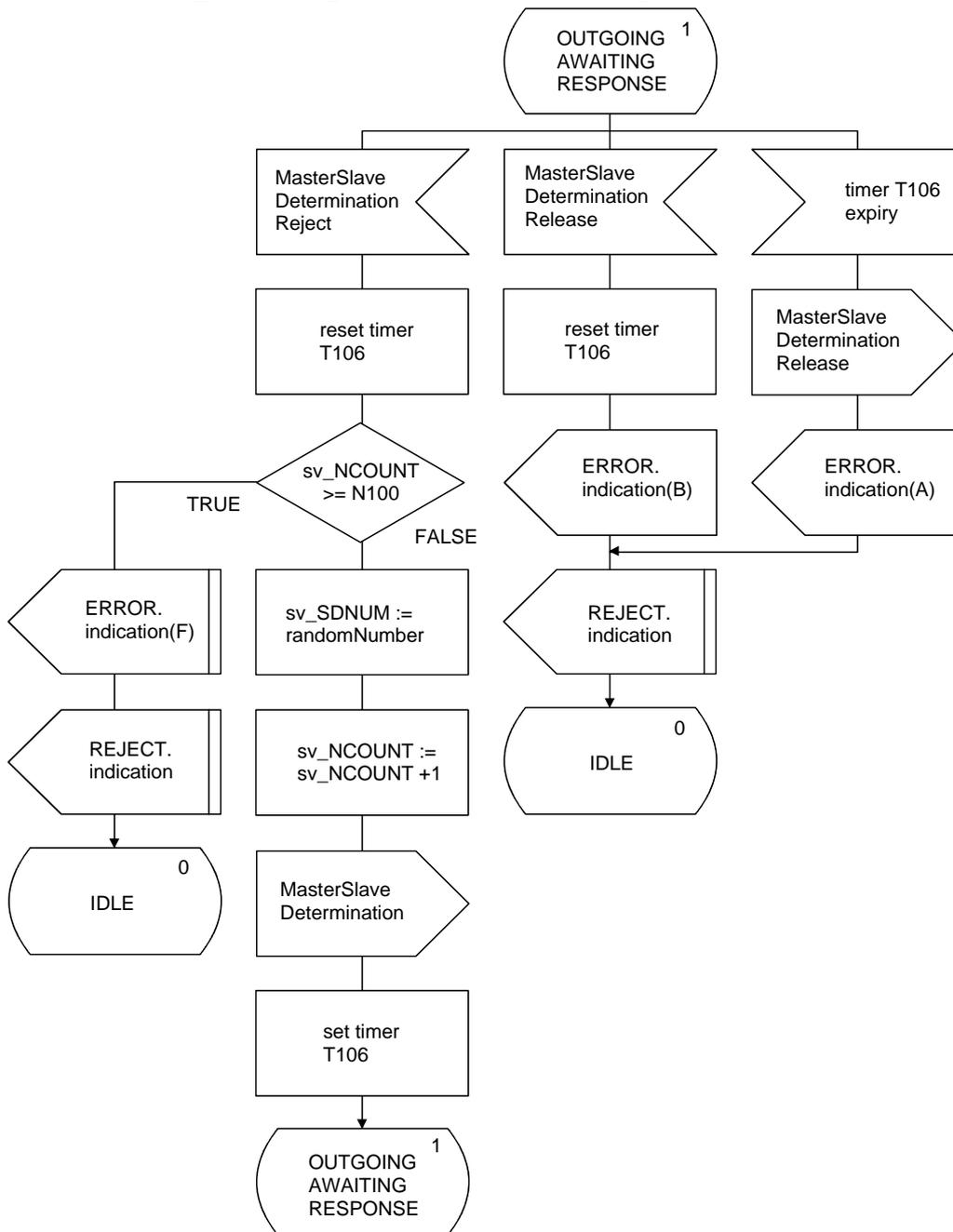


FIGURE 4 ii)/H.245

Diagramme SDL de l'entité MSDSE

Remplacée par une version plus récente

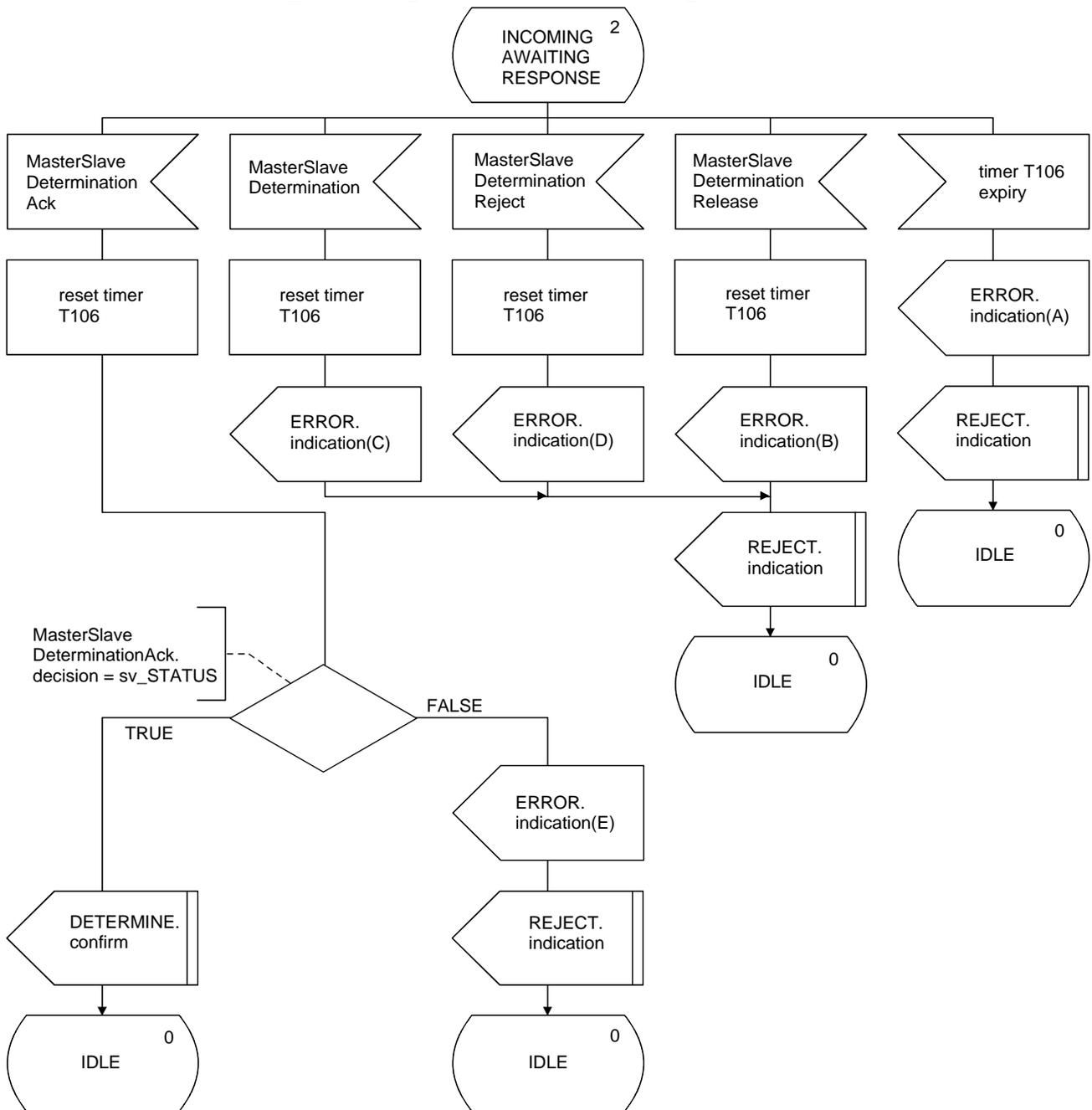


T1520270-95/d006

FIGURE 4 iii)/H.245

Diagramme SDL de l'entité MSDSE

Remplacée par une version plus récente



T1520280-95/d007

FIGURE 4 iv)/H.245

Diagramme SDL de l'entité MSDSE

Remplacée par une version plus récente

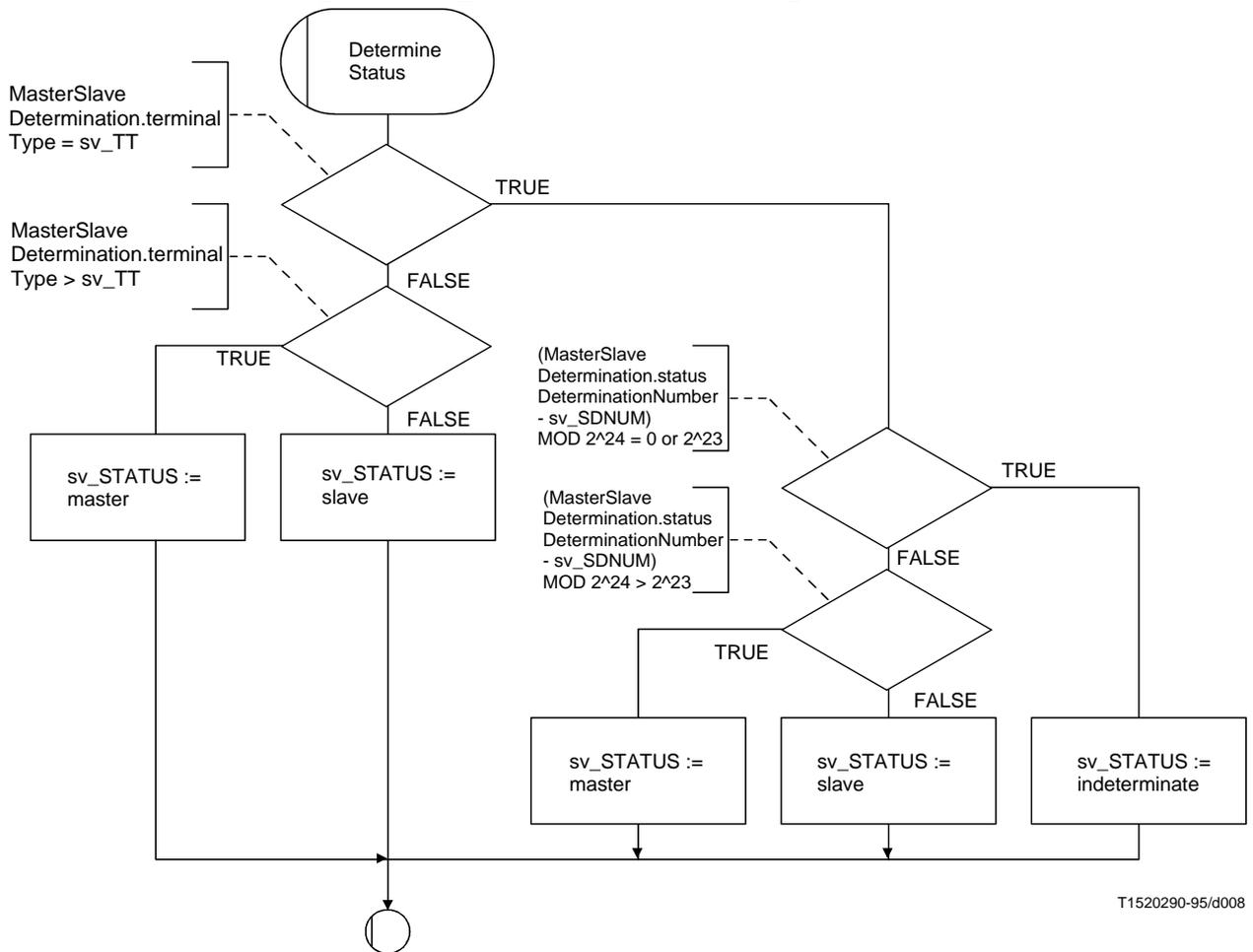


FIGURE 4 v)/H.245

Diagramme SDL de l'entité MSDSE

8.3.1.2 Aperçu général du protocole – entité CESE entrante

Lorsqu'un message TerminalCapabilitySet est reçu à 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 émettant la primitive de réponse TRANSFER et un message TerminalCapabilitySetAck est envoyé à l'entité CESE homologue sortante.

8.3.2 Communication entre l'entité CESE et l'utilisateur de l'entité CESE

8.3.2.1 Primitive 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.

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.

Remplacée par une version plus récente

TABLEAU 22/H.245

Primitives et paramètres

Nom générique	Type			
	demande	indication	réponse	confirmation
TRANSFER	PROTOID	PROTOID	– (Note 1)	–
	MUXCAP CAPTABLE CAPDESCRIPTORS	MUXCAP CAPTABLE CAPDESCRIPTORS		
REJECT (refus)	CAUSE (motif)	SOURCE (origine) CAUSE	Non défini (Note 2)	Non défini

NOTE 1 – «–» signifie absence de paramètres.
NOTE 2 – «Non défini» signifie que cette primitive n'est pas définie.

8.3.2.3 Définitions des paramètres

La définition des paramètres des primitives indiquées au Tableau 22 est la suivante:

- le paramètre PROTOID sert à identifier le protocole. Il est appliqué sur le champ «protocollIdentifieur» du message «TerminalCapabilitySet» et transporté en transparence jusqu'à l'utilisateur de l'entité CESE homologue. Ce paramètre est obligatoire;
- le paramètre MUXCAP est le paramètre des capacités du multiplex. Il est appliqué sur le champ «multiplexCapability» du message «TerminalCapabilitySet» et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- le paramètre CAPTABLE est le paramètre du tableau de capacités. Il peut y avoir un nombre d'entrées de tableau de capacités, décrites à l'aide de ce paramètre, égal ou supérieur à zéro. Ce paramètre est appliqué sur le champ «CapabilityTable» du message «TerminalCapabilitySet» et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- le paramètre CAPDESCRIPTORS est le paramètre des descripteurs de capacités. Il peut y avoir un ou plusieurs descripteurs de capacités, décrits à l'aide de ce paramètre. Ce paramètre est appliqué au champ «capabilityDescriptors» du message «TerminalCapabilitySet» et transmis de façon transparente vers l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- 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;
- 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 pour une entité CESE sortante sont:

état 0: IDLE (repos)

L'entité CESE est à l'état «idle».

état 1: AWAITING RESPONSE (attente de la réponse)

L'entité CESE attend une réponse de l'entité CESE distante.

Remplacée par une version plus récente

Les états pour une entité CESE entrante sont:

état 0: IDLE

L'entité CESE est à l'état «idle».

état 1: AWAITING RESPONSE

L'entité CESE attend une réponse de la part de l'utilisateur de l'entité CESE.

8.3.2.5 Diagramme de transition 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.

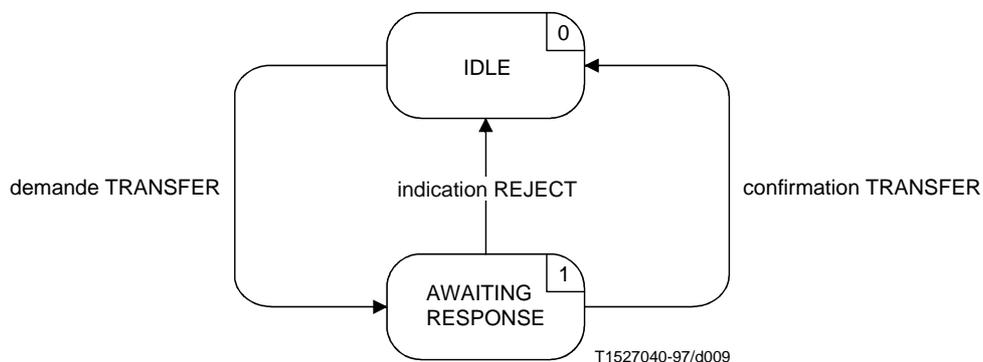


FIGURE 5/H.245

Diagramme de transition d'états pour la séquence de primitives d'une entité CESE sortante

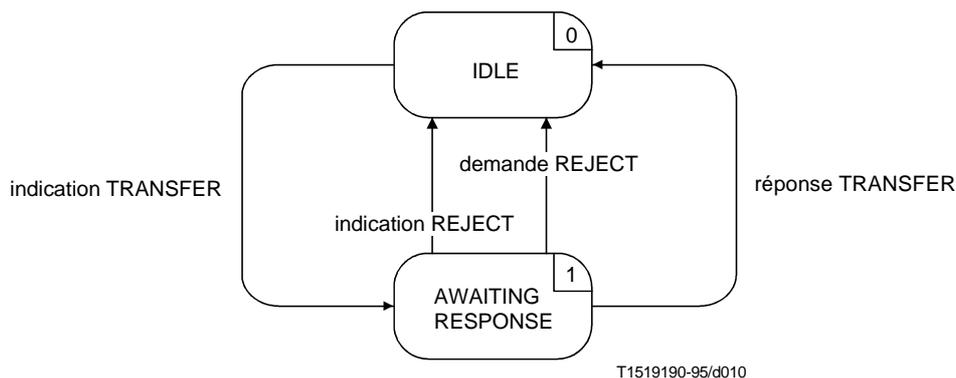


FIGURE 6/H.245

Diagramme de transition 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 de l'entité CESE, tels qu'ils sont définis dans le paragraphe 6 et se rapportent au protocole des entités CESE.

Remplacée par une version plus récente

TABLEAU 23/H.245

Noms et champs de messages de l'entité CESE

Fonction	Message	Sens	Champ
Transfer	TerminalCapabilitySet	S → E	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors
	TerminalCapabilitySetAck	S ← E	sequenceNumber
Reject	TerminalCapabilitySetReject	S ← E	sequenceNumber cause (motif)
Reset	TerminalCapabilitySetRelease	S → E	–
S	Sortante		
E	Entrante		

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 appliquée au champ sequenceNumber du message TerminalCapabilitySet avant la transmission de ce 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 au cours duquel aucun message TerminalCapabilitySetAck ou TerminalCapabilitySetReject ne peut être reçu.

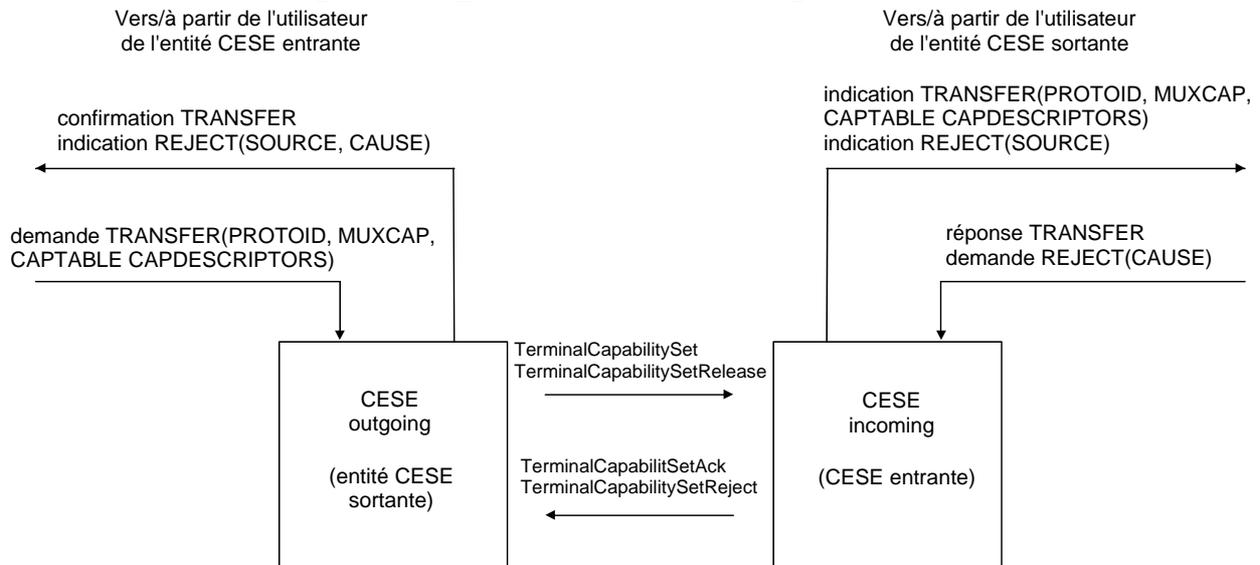
8.3.4 Procédures de l'entité CESE

La Figure 7 récapitule les primitives de l'entité CESE et leurs paramètres, de même que les messages, pour chacune des entités CESE sortante et entrante.

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.

Remplacée par une version plus récente



T1519200-95/d011

FIGURE 7/H.245

Primitives et messages dans l'entité de signalisation de l'échange de capacités (CESE)

TABLEAU 24/H.245

Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
indication TRANSFER	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	TerminalCapabilitySet.protocolIdentifier TerminalCapabilitySet.multiplexCapability TerminalCapabilitySet.capabilityTable TerminalCapabilitySet.capabilityDescriptors
indication REJECT	SOURCE CAUSE	USER 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.

Remplacée par une version plus récente

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 demande TRANSFER(PROTOID) demande TRANSFER(MUXCAP) demande TRANSFER(CAPTABLE) demande TRANSFER(CAPDESCRIPTORS)
TerminalCapabilitySetAck	sequenceNumber	in_SQ
TerminalCapabilitySetReject	sequenceNumber cause	in_SQ demande 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.

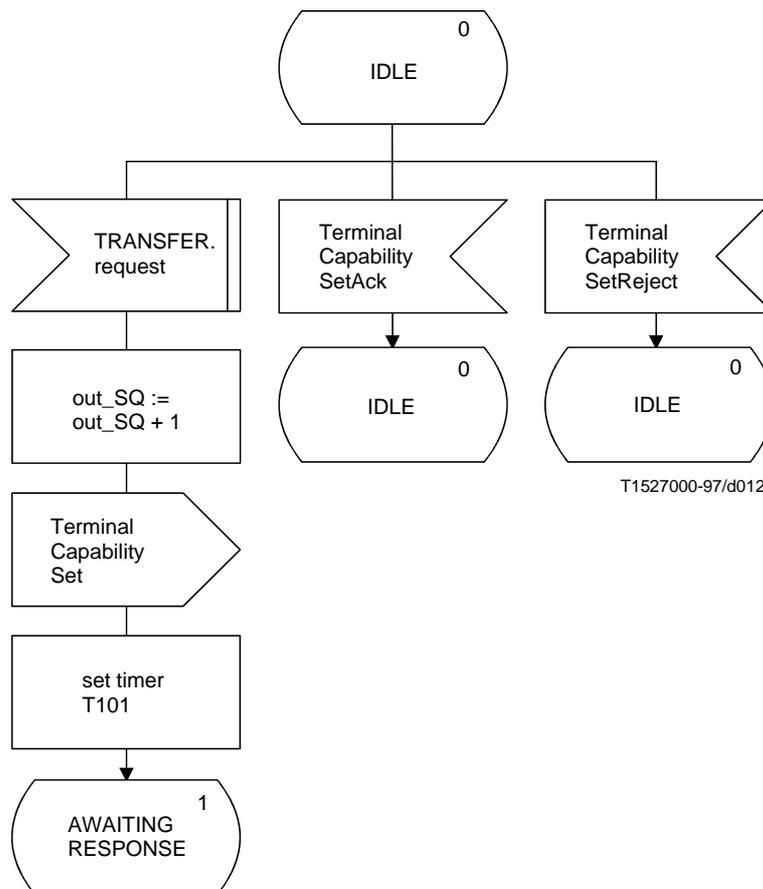
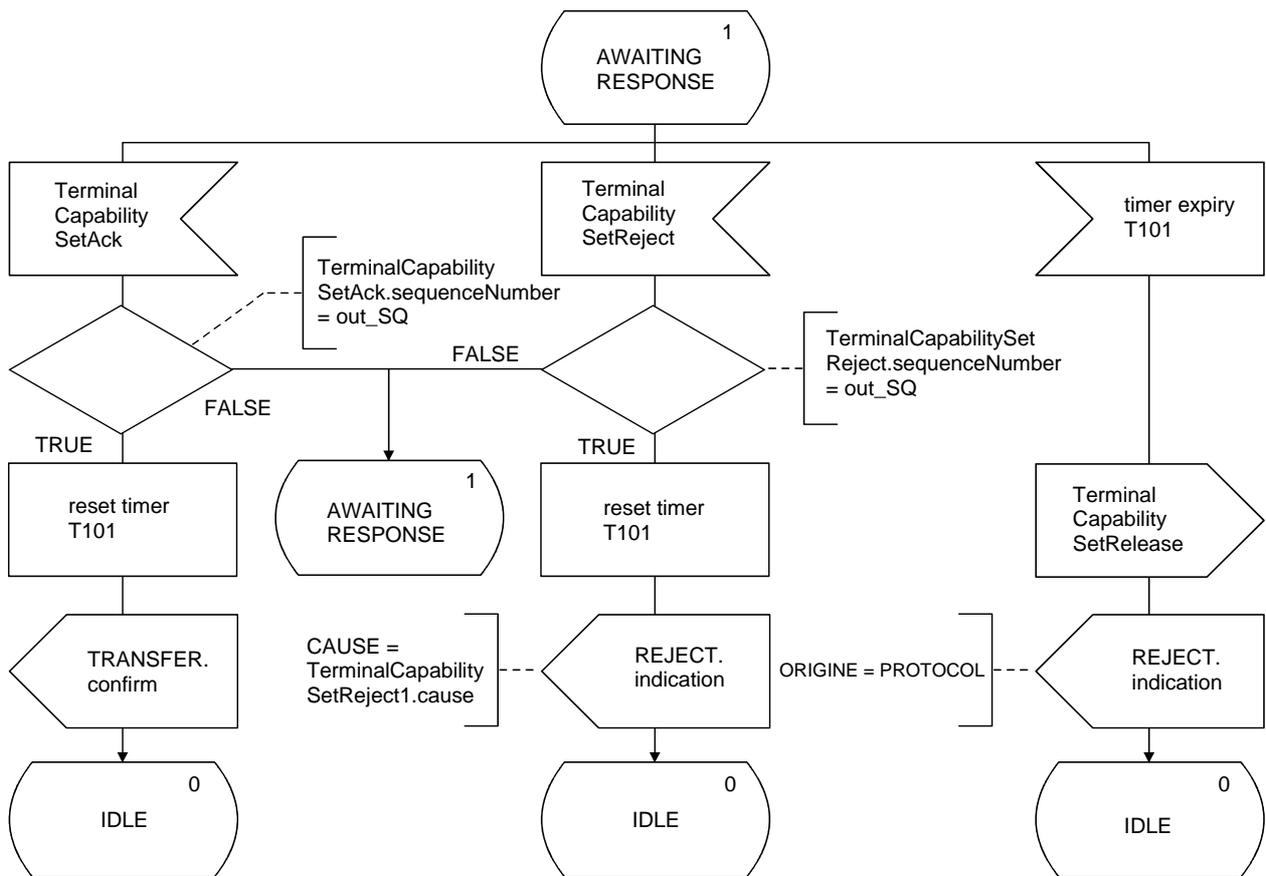


FIGURE 8 i)/H.245

Description SDL de l'entité CESE sortante

Remplacée par une version plus récente



T1520300-95/d013

FIGURE 8 ii)/H.245

Description SDL de l'entité CESE sortante

Remplacée par une version plus récente

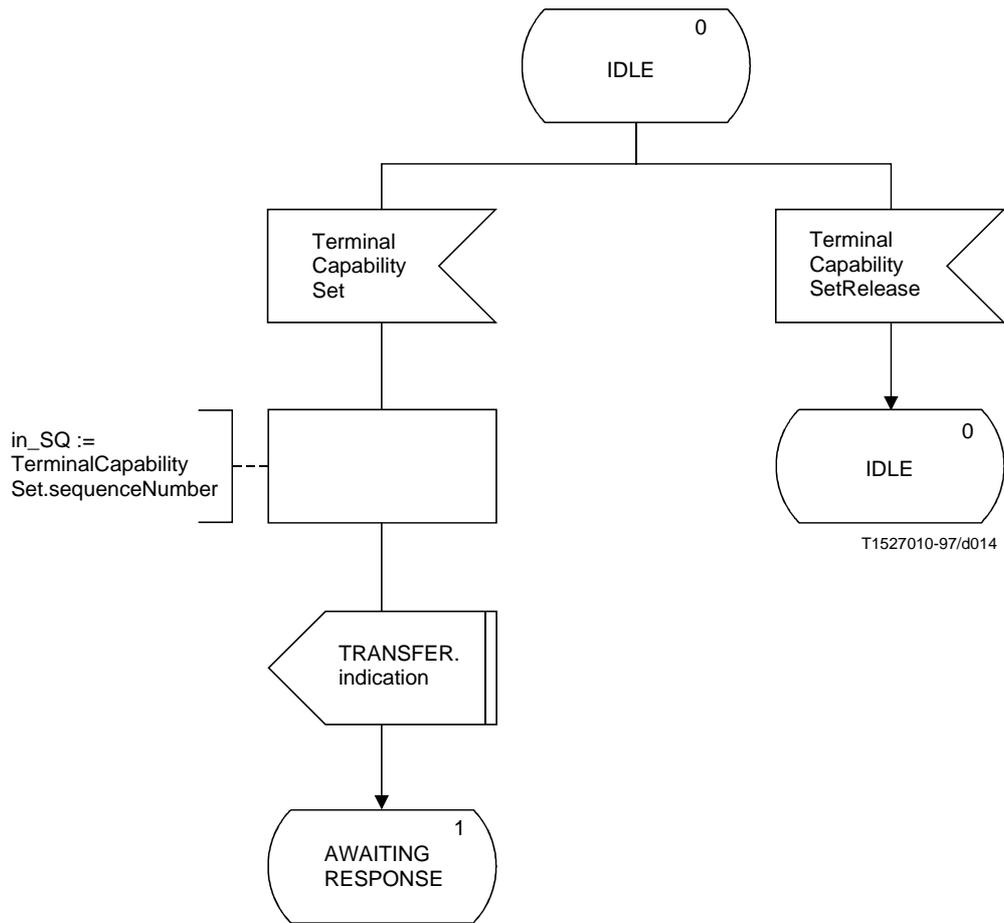
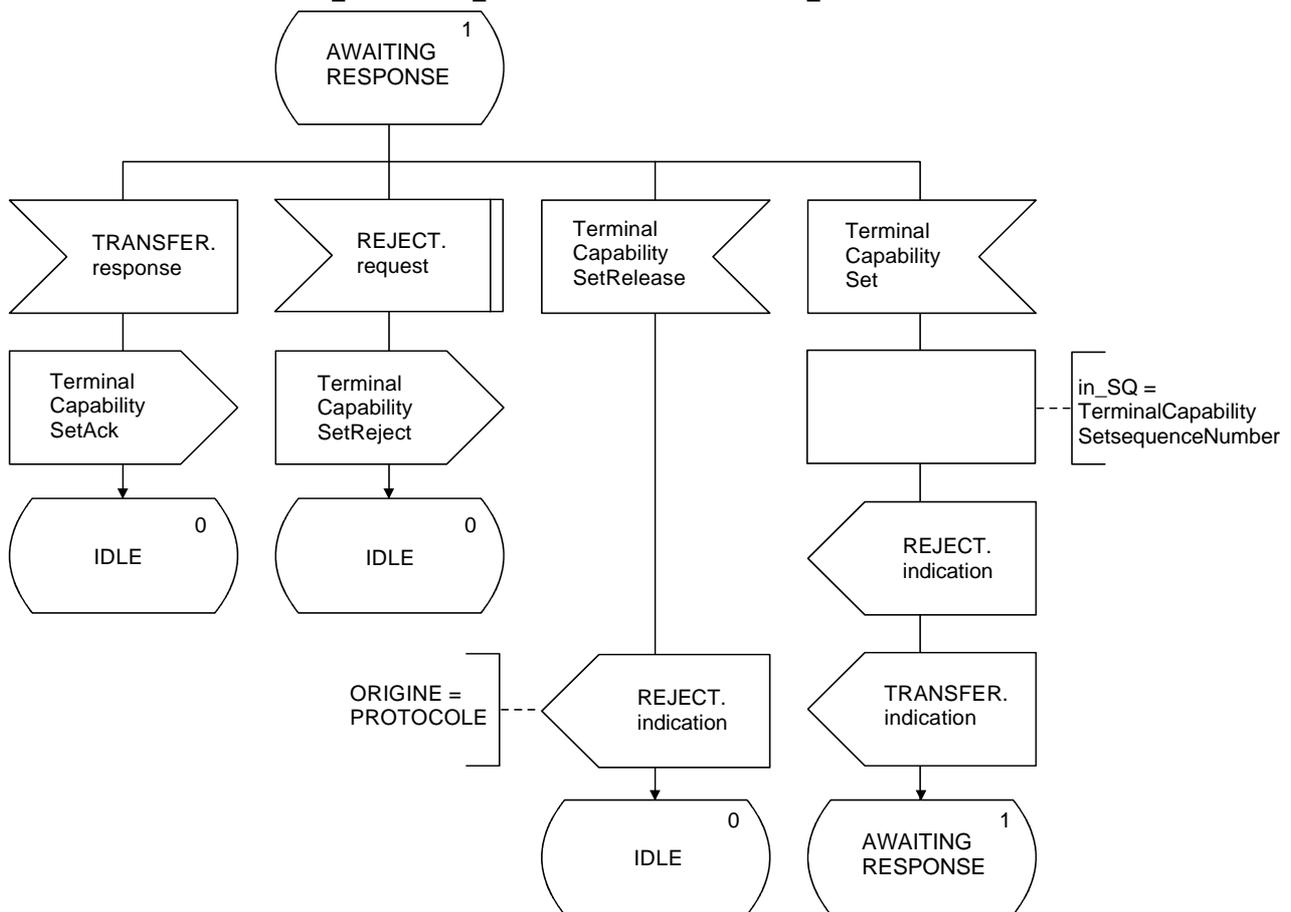


FIGURE 9 i)/H.245

Description SDL de l'entité CESE entrante

Remplacée par une version plus récente



T1519250-95/d015

FIGURE 9 ii)/H.245

Description SDL de l'entité CESE entrante

8.4 Procédures de signalisation sur voies logiques unidirectionnelles

8.4.1 Introduction

Le protocole spécifié ici garantit l'ouverture et la fermeture des voies logiques unidirectionnelles par les procédures d'acquiescement.

Ce protocole est désigné sous le nom d'entité de signalisation de voie logique (LCSE). 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 à l'entité LCSE homologue par les messages appropriés définis au 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 unidirectionnelle. Il n'y a pas de connexion entre une entité LCSE entrante et une entité LCSE sortante situées d'un même côté, hormis celle qui est établie par l'intermédiaire des primitives vers l'utilisateur de l'entité LCSE et à partir de ce même utilisateur. Les situations d'erreur de l'entité LCSE font l'objet d'un compte rendu.

Les données ne doivent être envoyées que sur une voie logique à l'état ESTABLISHED. Les données reçues sur une voie logique qui n'est pas à l'état ESTABLISHED doivent être ignorées et on ne saurait considérer qu'une défaillance s'est produite.

La commutation de mode doit être effectuée par la fermeture et l'ouverture des voies logiques existantes ou par l'ouverture de nouvelles voies logiques.

Remplacée par une version plus récente

NOTE – Les recommandations qui se réfèrent à la présente Recommandation peuvent définir des voies logiques par défaut. Ces voies doivent être considérées comme étant à l'état ESTABLISHED dès le début de la communication et ne doivent pas être ouvertes au moyen de ces procédures. Elles peuvent cependant être fermées par ces procédures et ultérieurement réouvertes en vue du même usage ou d'un usage différent.

Un terminal qui n'est plus à même de traiter les signaux sur une voie logique doit prendre des mesures appropriées: fermeture de la voie logique et transmission des informations de capacité (modifiée) pertinentes au terminal distant.

Le texte suivant donne un aperçu général du fonctionnement du protocole d'entité LCSE. En cas de divergence entre cette spécification et la spécification formelle, c'est celle-ci qui a priorité.

8.4.1.1 Aperçu général du protocole

L'ouverture d'une voie logique est déclenchée lorsque la primitive de demande ESTABLISH est émise par l'utilisateur situé à l'entité LCSE sortante. Un message OpenLogicalChannel contenant les paramètres de voie logique descendante mais ne contenant pas les paramètres de voie logique montante, est envoyé à l'entité LCSE homologue entrante et le temporisateur T103 est lancé. Si un message OpenLogicalChannelAck est reçu en réponse au message OpenLogicalChannel, le temporisateur T103 est arrêté et l'utilisateur est informé, par la primitive de confirmation ESTABLISH, que la voie logique a bien été ouverte. Celle-ci peut désormais être utilisée pour transmettre des informations d'utilisateur. Si toutefois 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 d'entité LCSE homologue a refusé l'établissement de la voie logique.

Si le temporisateur T103 expire dans cette période, l'utilisateur en est informé par la primitive d'indication RELEASE et un message CloseLogicalChannel est envoyé à l'entité LCSE homologue entrante.

Une voie logique qui a été normalement établie peut être fermée lorsque la primitive de demande RELEASE est émise par l'utilisateur situé à l'entité LCSE sortante. Un message CloseLogicalChannel est envoyé à l'entité LCSE homologue entrante et le temporisateur T103 est lancé. Lorsqu'un message CloseLogicalChannelAck est reçu, le temporisateur T103 est arrêté et l'utilisateur est informé, par la primitive de confirmation RELEASE, que la voie logique a été normalement fermée.

Si le temporisateur T.103 expire dans cette période, l'utilisateur en est informé par la primitive d'indication RELEASE.

Avant que le message OpenLogicalChannelAck ou OpenLogicalChannelReject ait été reçu en réponse à un message OpenLogicalChannel préalablement envoyé, l'utilisateur situé à l'entité LCSE sortante peut fermer la voie logique au moyen de la primitive de demande RELEASE.

Avant que le message CloseLogicalChannelAck ait été reçu en réponse à un message CloseLogicalChannel préalablement envoyé, l'utilisateur situé à l'entité LCSE sortante peut établir une nouvelle voie logique en émettant la primitive de demande ESTABLISH.

8.4.1.2 Aperçu général du protocole – Entité LCSE entrante

Lorsqu'un message OpenLogicalChannel est reçu à l'entité LCSE entrante, l'utilisateur est informé, par la primitive d'indication ESTABLISH, de la demande d'ouverture d'une nouvelle voie logique. L'utilisateur de l'entité LCSE entrante signale l'acceptation de la demande d'établissement de la voie logique en émettant la primitive de réponse ESTABLISH et un message OpenLogicalChannelAck est envoyé à l'entité LCSE homologue sortante. La voie logique peut ensuite être utilisée pour recevoir des informations d'utilisateur. L'utilisateur de l'entité LCSE entrante signale le rejet de la demande d'établissement de la voie logique en émettant la primitive de demande RELEASE et un message OpenLogicalChannelReject est envoyé à l'entité LCSE homologue sortante.

Une voie logique qui a été normalement établie peut être fermée lorsque le message CloseLogicalChannel est reçu à 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 à l'entité LCSE homologue sortante.

8.4.1.3 Résolution des conflits

Des conflits peuvent se produire lorsque plusieurs demandes d'ouverture de voies logiques sont émises en même temps. C'est d'après la connaissance des capacités échangées qu'il est possible de déduire l'existence d'un conflit.

Les terminaux doivent être capables de détecter l'apparition ou la possibilité d'apparition d'un conflit et doivent agir comme suit.

Avant que des voies logiques puissent être ouvertes, un certain terminal doit être désigné comme étant le terminal maître et l'autre comme le terminal esclave. Le protocole défini en 8.2 offre un moyen de prendre cette décision. Le terminal maître doit rejeter immédiatement toute demande issue du terminal esclave qu'il reconnaît comme étant une demande conflictuelle. Le terminal esclave peut identifier de tels conflits mais doit répondre à la demande issue du terminal maître, avec la connaissance que sa demande antérieure sera rejetée.

Remplacée par une version plus récente

NOTE – De tels conflits peuvent être causés par des ressources limitées en termes de terminaux, par exemple lorsque des capacités de réception et de transmission dépendent l'une de l'autre, comme dans le cas d'un terminal qui peut gérer un certain nombre d'algorithmes audio, mais qui ne peut décoder que le même algorithme pendant que celui-ci code.

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 s'effectuent 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 absence de paramètres.
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:

- 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 rend compte des erreurs de l'entité LCSE à une entité de gestion.

8.4.2.3 Définition des paramètres

La définition des paramètres de primitives indiquées dans le Tableau 26 est la suivante:

- le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie logique. Ce paramètre est appliqué sur le champ «forwardLogicalChannelParameters» du message OpenLogicalChannel et est transporté en transparence jusqu'à l'utilisateur de l'entité LCSE homologue;
- le paramètre SOURCE indique à l'utilisateur de l'entité LCSE l'origine de la libération de voie logique. Le paramètre SOURCE est mis à la valeur «USER» ou «LCSE», indiquant soit l'utilisateur de l'entité LCSE, soit l'entité LCSE elle-même. Ce dernier cas peut être le résultat d'une erreur de protocole;
- 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»;
- 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 distinctement pour une entité LCSE sortante et une entité LCSE entrante. Les états pour une entité LCSE sortante sont:

état 0: RELEASED (voie logique libérée)

Remplacée par une version plus récente

La voie logique est libérée. La voie logique ne doit pas être utilisée pour envoyer des données sortantes.

état 1: AWAITING ESTABLISHMENT (attente de l'établissement de la voie logique)

L'entité LCSE sortante attend l'établissement d'une voie logique avec une entité LCSE homologue entrante. La voie logique ne doit pas être utilisée pour envoyer des données sortantes.

état 2: ESTABLISHED (voie logique établie)

La connexion de voie logique entre des entités LCSE homologues a été établie. La voie logique peut être utilisée pour envoyer des données sortantes.

état 3: AWAITING RELEASE (attente de libération)

L'entité LCSE sortante attend la libération d'une voie logique avec l'entité LCSE homologue entrante. La voie logique ne doit pas être utilisée pour envoyer des données sortantes.

Les états correspondant à une entité LCSE entrante sont:

état 0: RELEASED (voie logique libérée)

La voie logique est libérée. La voie logique ne doit pas être utilisée pour recevoir des données entrantes.

état 1: AWAITING ESTABLISHMENT (attente de l'établissement)

L'entité LCSE entrante attend l'établissement d'une voie logique avec une entité LCSE sortante homologue. La voie logique ne doit pas être utilisée pour recevoir des données entrantes.

état 2: ESTABLISHED (voie logique établie)

Une connexion de voie logique entre entités LCSE homologues a été établie. La voie logique peut être utilisée pour recevoir des données entrantes.

8.4.2.5 Diagramme de transition d'états

La séquence autorisée de primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE est définie ci-après. La séquence autorisée de primitives concerne des états de l'entité LCSE tels qu'ils sont perçus par l'utilisateur de l'entité LCSE. Les séquences autorisées sont spécifiées pour désigner successivement une entité LCSE sortante et une entité LCSE entrante, comme cela est décrit dans les Figures 10 et 11 respectivement.

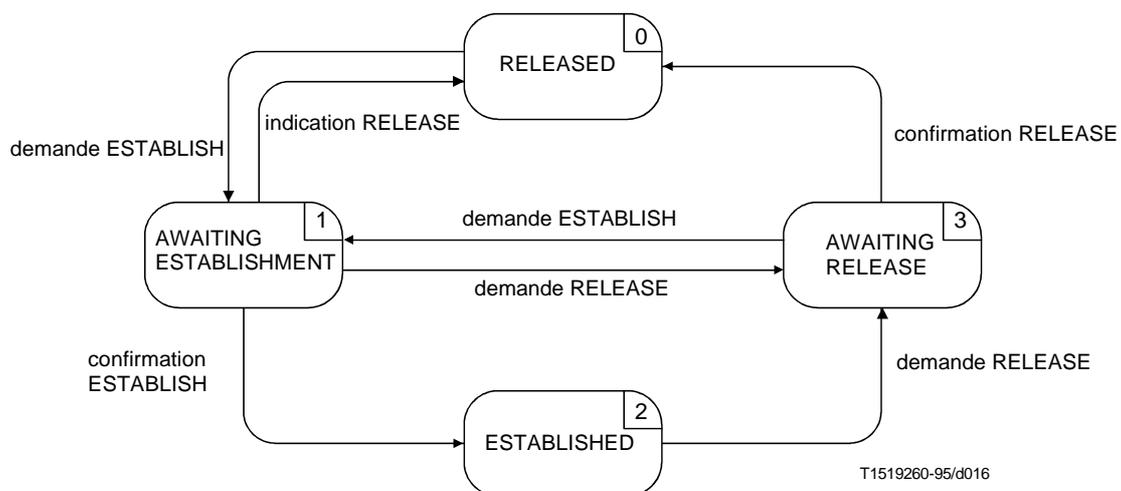


FIGURE 10/H.245

Diagramme de transition d'états pour la séquence de primitives d'une entité LCSE sortante

Remplacée par une version plus récente

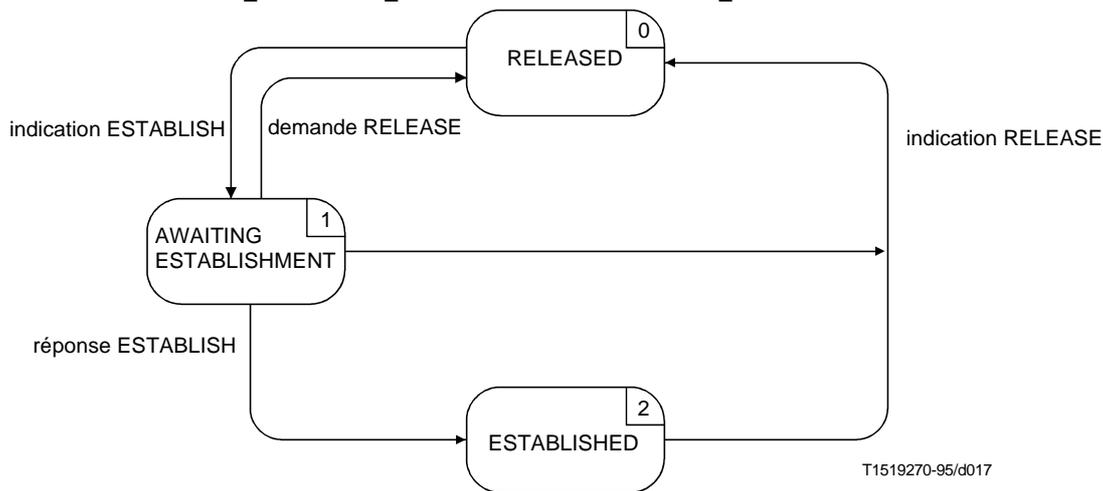


FIGURE 11/H.245

Diagramme de transition d'états pour la séquence de primitives d'une entité LCSE entrante

8.4.3 Communications entre entités LCSE homologues

8.4.3.1 Messages de l'entité 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	Sortante		
E	Entrante		

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 de messages d'entités LCSE envoyés par une entité LCSE sortante. Pour des 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.

Remplacée par une version plus récente

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. Cette variable est initialisée lors du lancement de l'entité LCSE entrante. La valeur de la variable in_LCN est utilisée pour remplir le champ forwardLogicalChannelNumber de messages d'entités LCSE envoyés par une entité LCSE entrante. Pour des messages LCSE reçus par une entité LCSE entrante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de la variable in_LCN.

8.4.3.3 Temporiseurs LCSE

Le temporisateur suivant est spécifié pour l'entité LCSE sortante:

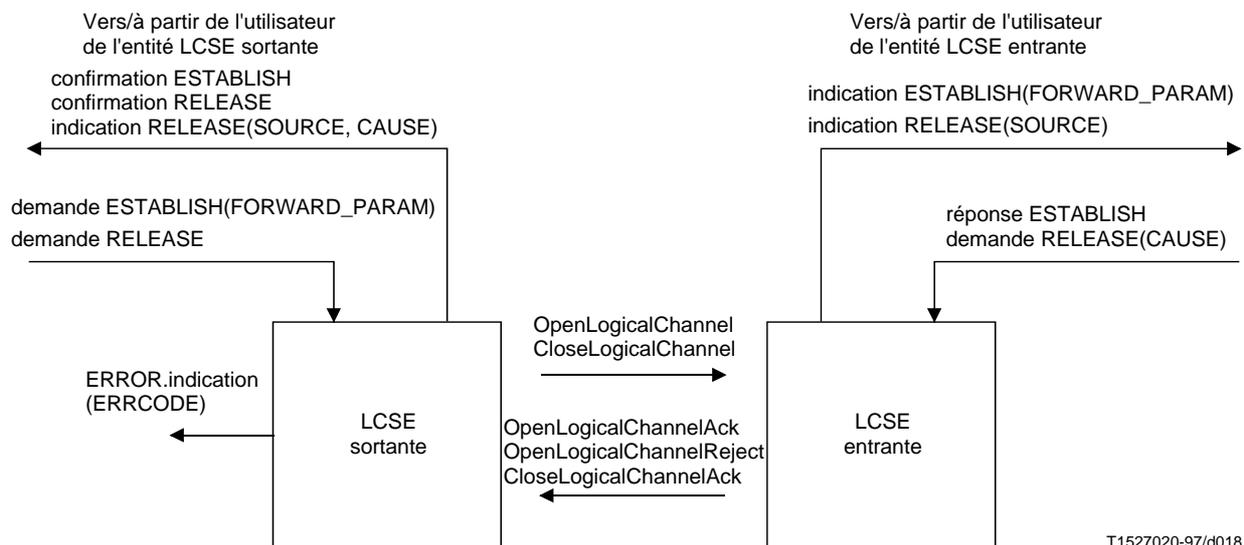
T103

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 de l'entité 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 sortante et entrante.



T1527020-97/d018

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.

Remplacée par une version plus récente

TABLEAU 28/H.245

Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut (Note)
indication ESTABLISH	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
indication 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 qui sont indiquées dans le Tableau 29.

TABLEAU 29/H.245

Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut
OpenLogicalChannel (Note 2)	forwardLogicalChannelNumber forwardLogicalChannelParameters	out_LCN demande ESTABLISH(FORWARD_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
OpenLogicalChannelReject	forwardLogicalChannelNumber cause	in_LCN demande 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 de voie logique montante ne sont pas codés dans les procédures de signalisation de voie logique unidirective.

8.4.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique un cas d'erreur particulier. 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	Cas d'erreur	Etat
Message non pertinent	A	OpenLogicalChannelAck	RELEASED
	B	OpenLogicalChannelReject	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
Aucune réponse provenant de l'entité LCSE homologue	D	Fin de temporisation T 103	AWAITING ESTABLISHMENT AWAITING RELEASE

Remplacée par une version plus récente

8.4.4.5 Description SDL

Les procédures de l'entité LCSE sortante et de l'entité LCSE entrante sont décrites en langage SDL dans les Figures 13 et 14 respectivement.

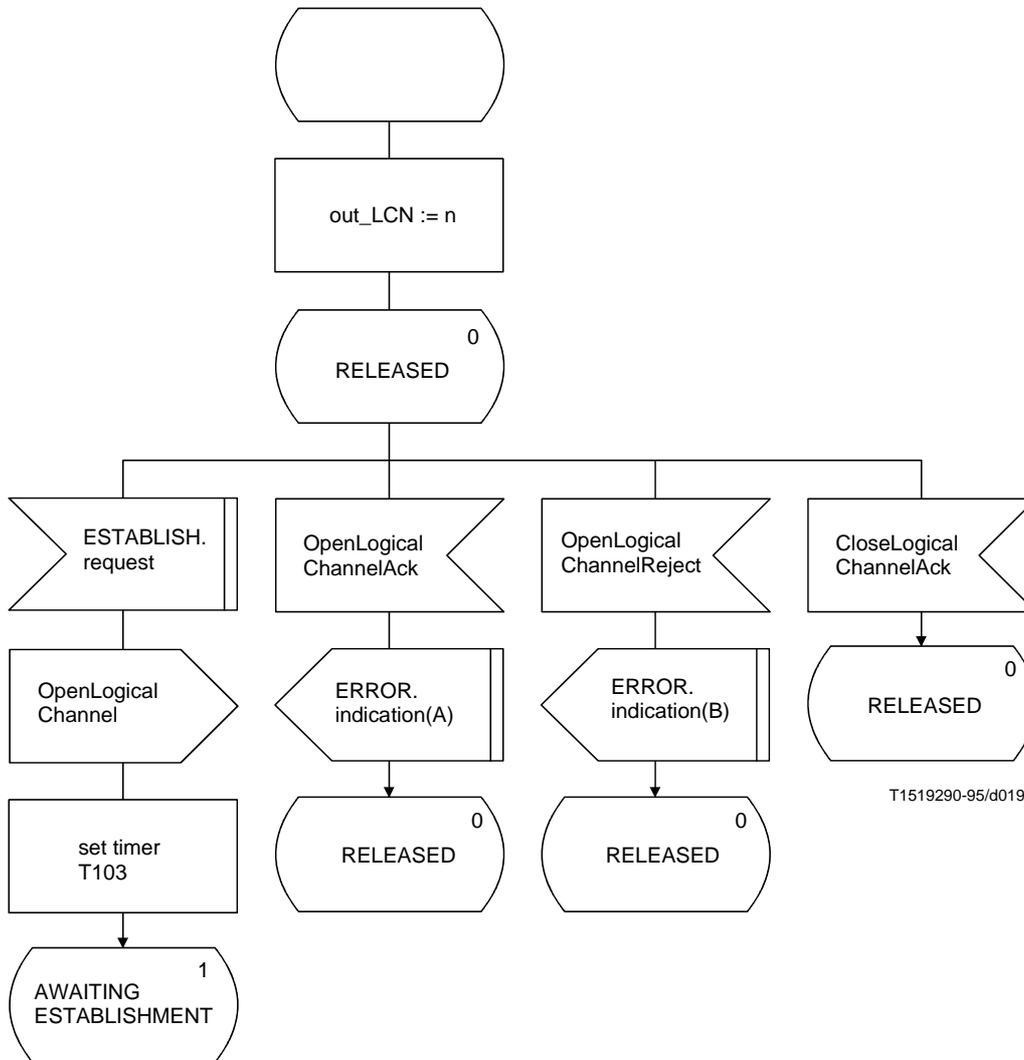
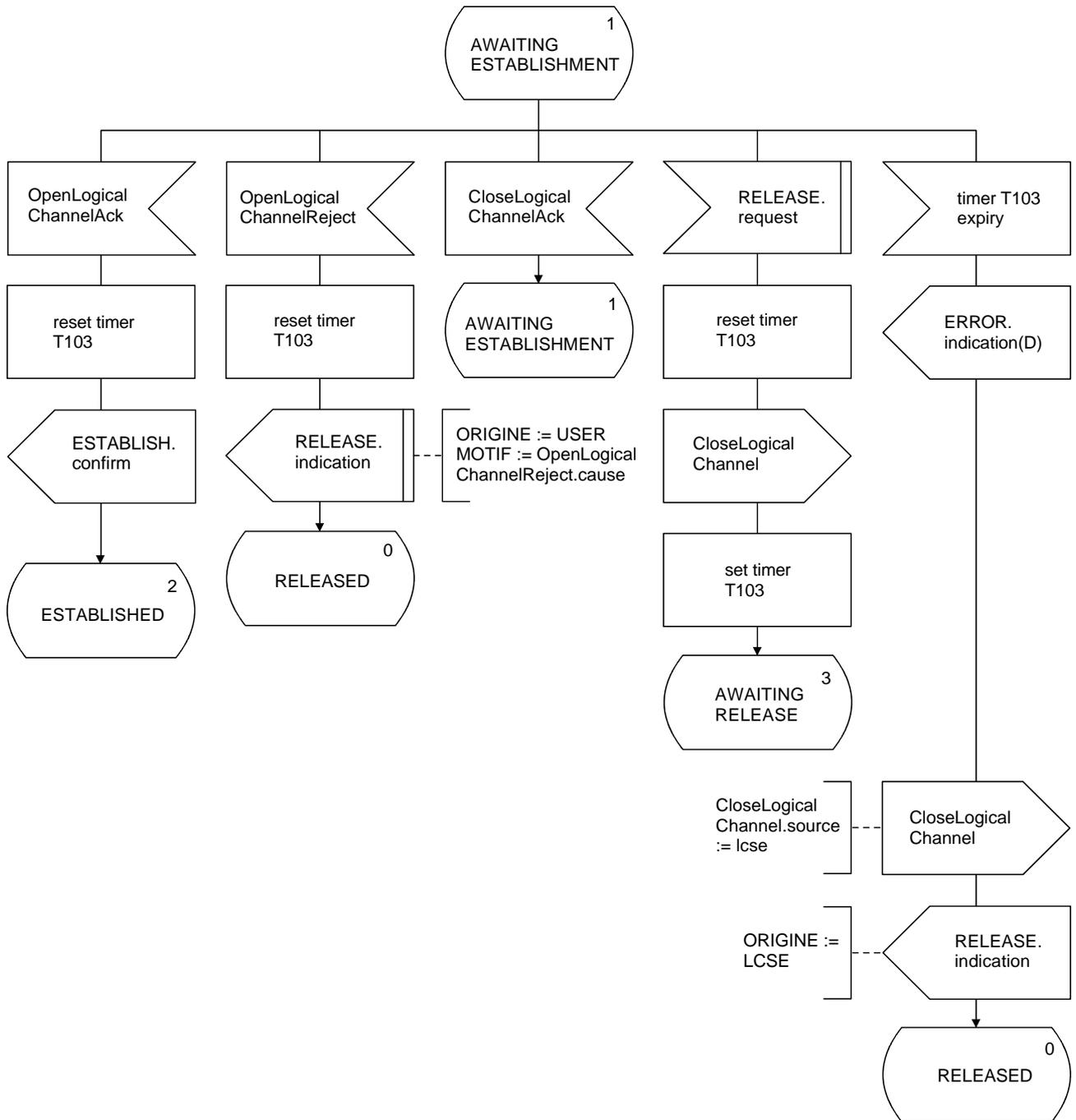


FIGURE 13 i)/H.245

Description SDL de l'entité LCSE sortante

Remplacée par une version plus récente

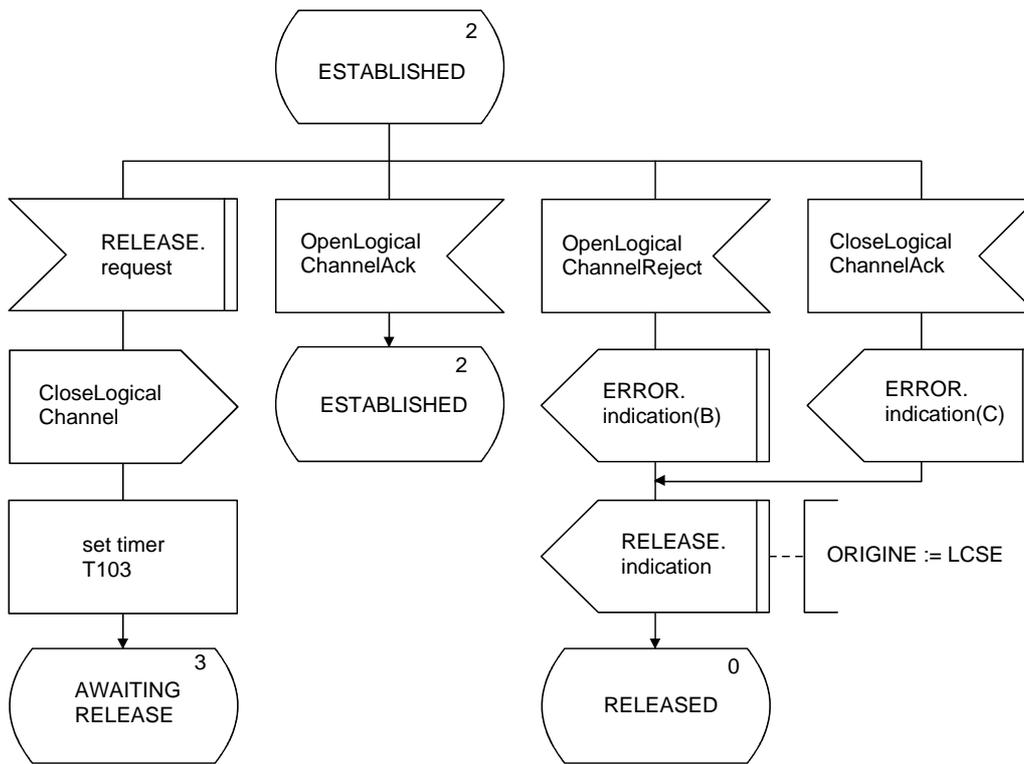


T1519300-95/d020

FIGURE 13 ii)/H.245

Description SDL de l'entité LCSE sortante

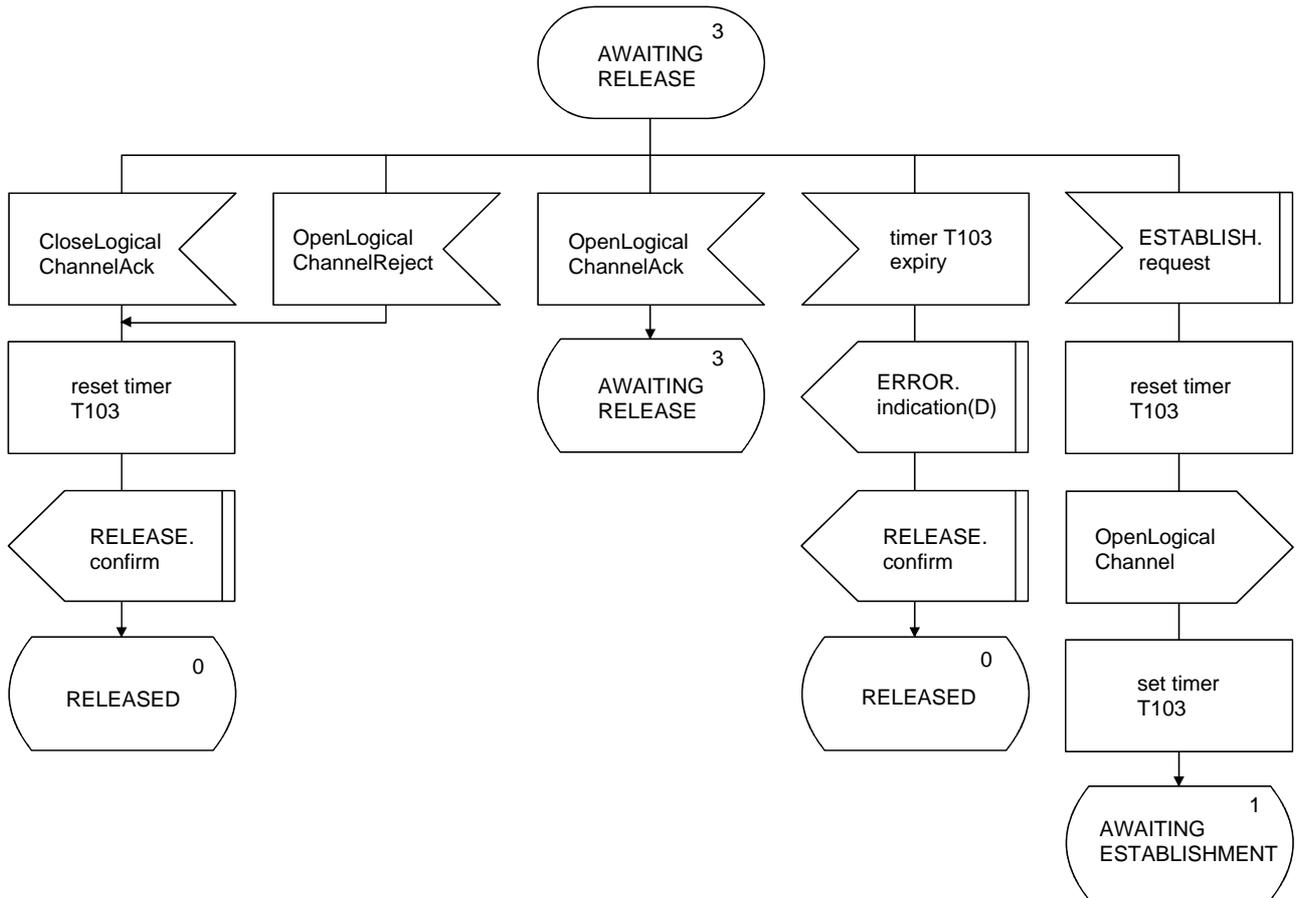
Remplacée par une version plus récente



T1519310-95/d021

FIGURE 13 iii)/H.245
Description SDL de l'entité LCSE sortante

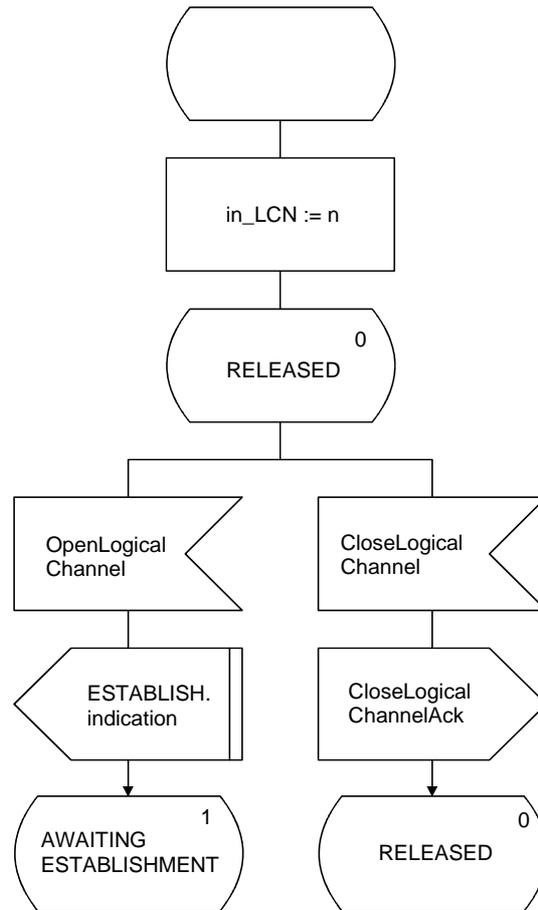
Remplacée par une version plus récente



T1519320-95/d022

FIGURE 13 iv)/H.245
Description SDL de l'entité LCSE sortante

Remplacée par une version plus récente



T1519330-95/d023

FIGURE 14 i)/H.245
Description SDL de l'entité LCSE entrante

Remplacée par une version plus récente

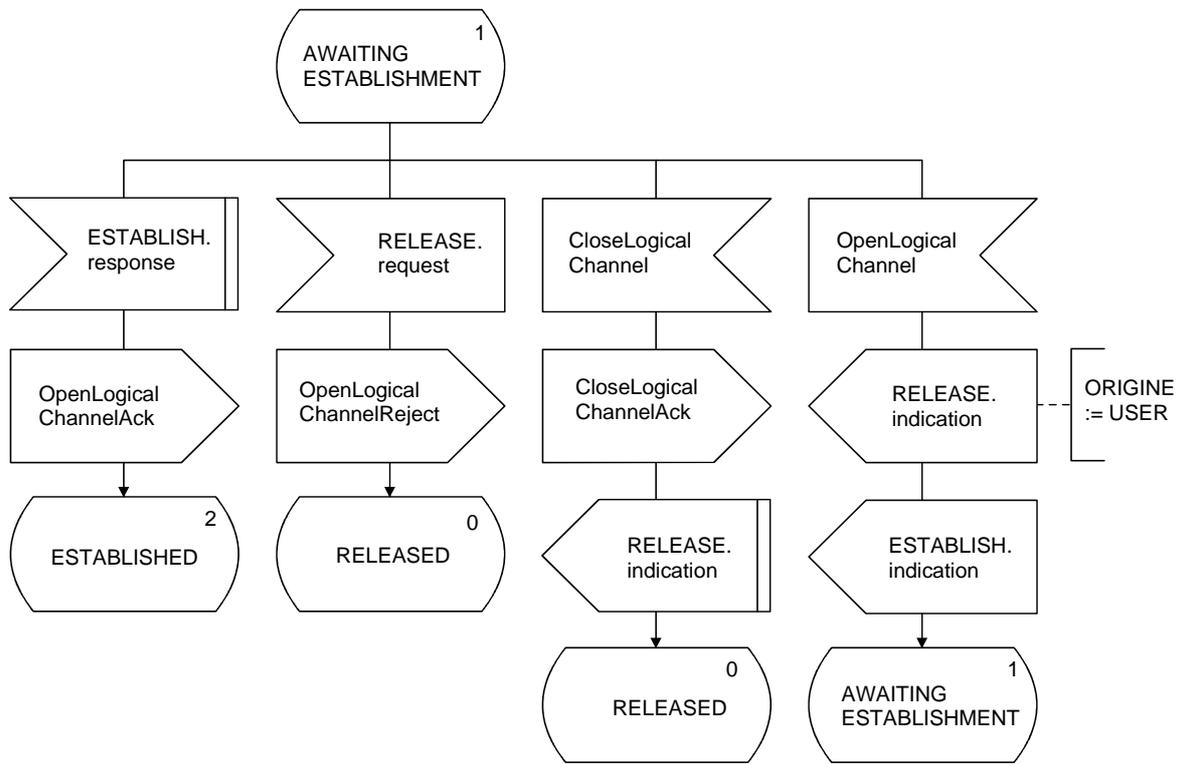


FIGURE 14 ii)/H.245
Description SDL de l'entité LCSE entrante

Remplacée par une version plus récente

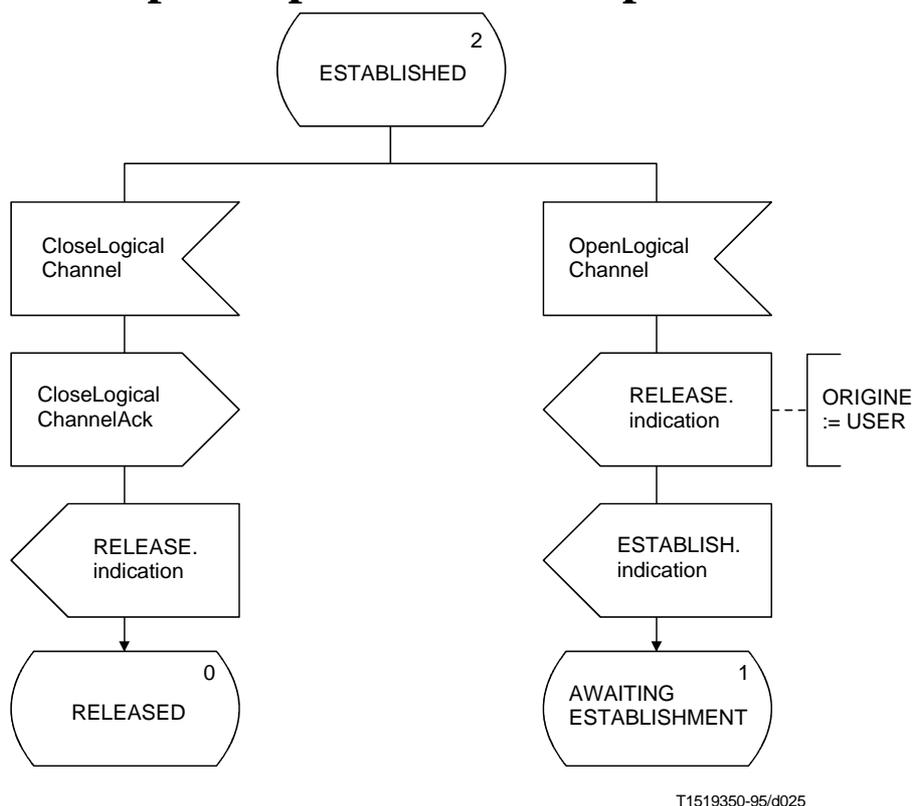


FIGURE 14 iii)/H.245

Description SDL de l'entité LCSE entrante

8.5 Procédures de signalisation de canal logique bidirectionnel

8.5.1 Introduction

Le protocole spécifié ici garantit l'ouverture et la fermeture des canaux logiques bidirectionnels par les procédures d'acquiescement.

On appelle ce protocole l'entité de signalisation de canal logique bidirectionnel (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, et entre les états B-LCSE. Les informations de protocole sont transmises à l'entité B-LCSE homologue par les messages appropriés définis au paragraphe 6.

Il y a une entité B-LCSE sortante et une entité B-LCSE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité B-LCSE correspondant à chaque canal logique bidirectionnel.

Un canal logique bidirectionnel est constitué de deux voies unidirectionnelles associées. Le terme «forward» (voie directe) désigne la transmission depuis le terminal dont émane la demande de canal logique bidirectionnel en direction de l'autre terminal; le terme «reverse» (voie inverse) désigne le sens de transmission inverse.

Les données ne doivent être envoyées que sur un canal logique bidirectionnel à l'état ESTABLISHED. Elles peuvent toutefois être reçues sur la voie inverse lorsque l'entité B-LCSE entrante est à l'état AWAITING CONFIRMATION. Les données reçues durant d'autres états que l'état ESTABLISHED et l'état AWAITING CONFIRMATION doivent être rejetées et on ne saurait considérer qu'une défaillance s'est produite.

Un terminal peut rejeter une demande d'ouverture d'un canal logique bidirectionnel au seul motif qu'il ne peut recevoir les paramètres de la voie inverse demandés. Dans ce cas, il doit rejeter la demande en invoquant un motif analogue à `unsuitableReverseParameters` et doit immédiatement lancer les procédures d'établissement d'un canal logique bidirectionnel, ainsi que le lui a demandé le terminal distant, dont les paramètres de voie inverse sont identiques aux paramètres de voie directe de la demande du terminal distant rejetée et présentant des paramètres de voie directe que le terminal puisse prendre en charge et que le terminal distant soit réputé pouvoir prendre en charge.

Remplacée par une version plus récente

La commutation de mode doit être effectuée par la fermeture et l'ouverture des voies logiques existantes ou par l'ouverture de nouvelles voies logiques.

NOTE – Les recommandations qui se réfèrent à la présente Recommandation peuvent définir des voies logiques par défaut. Ces voies doivent être considérées comme étant à l'état ESTABLISHED dès le début de la communication et ne doivent pas être ouvertes au moyen de ces procédures. Elles peuvent cependant être fermées par ces procédures et ultérieurement rouvertes en vue du même usage ou d'un usage différent.

Un terminal qui n'est plus à même de traiter les signaux sur une voie logique doit prendre des mesures appropriées: fermeture de la voie logique et transmission des informations de capacité (modifiée) pertinentes au terminal distant.

Le texte suivant donne un aperçu du fonctionnement du protocole B-LCSE. En cas de divergence entre ce texte et la spécification formelle, cette dernière prévaut.

8.5.1.1 Aperçu général du protocole

L'émission de la primitive de demande ESTABLISH par l'utilisateur dans l'entité B-LCSE sortante déclenche l'ouverture d'une voie logique. Un message OpenLogicalChannel contenant les paramètres des voies logiques directe et inverse, est envoyé à l'entité B-LCSE entrante homologue et le temporisateur T103 est mis en marche. A la réception d'un message OpenLogicalChannelAck en réponse à un message OpenLogicalChannel, le temporisateur T103 s'arrête, un message OpenLogicalChannelConfirm est envoyé à l'entité B-LCSE entrante homologue et l'utilisateur est informé par une primitive de confirmation ESTABLISH que la voie logique a bien été ouverte. Celle-ci peut dès lors être utilisée pour transmettre et recevoir des informations d'utilisateur. Toutefois, à la réception d'un message OpenLogicalChannelReject en réponse au message OpenLogicalChannel, le temporisateur T103 s'arrête et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité B-LCSE homologue a refusé d'établir la voie logique.

En cas d'expiration du temporisateur T103 pendant cette période, l'utilisateur en est informé par la primitive d'indication RELEASE et un message CloseLogicalChannel est envoyé à l'entité B-LCSE entrante homologue.

Une voie logique qui a été établie avec succès peut être fermée par l'émission de la primitive de demande RELEASE par l'utilisateur dans l'entité B-LCSE sortante. Un message CloseLogicalChannel est envoyé à l'entité B-LCSE entrante homologue et le temporisateur T103 est mis en marche. A la réception d'un message CloseLogicalChannelAck, le temporisateur T103 s'arrête et l'utilisateur est informé du succès de la fermeture de la voie logique par la primitive de confirmation RELEASE.

En cas d'expiration du temporisateur T103 pendant cette période, l'utilisateur en est informé par la primitive d'indication RELEASE.

Avant réception des messages OpenLogicalChannelAck ou OpenLogicalChannelReject en réponse à un message OpenLogicalChannel précédemment envoyé, l'utilisateur dans l'entité B-LCSE sortante peut fermer la voie logique au moyen de la primitive de demande RELEASE.

Avant réception du message CloseLogicalChannelAck en réponse à un message CloseLogicalChannel précédemment envoyé, l'utilisateur dans l'entité B-LCSE sortante peut établir une nouvelle voie logique en émettant la primitive de demande ESTABLISH.

8.5.1.2 Aperçu général du protocole – Entité B-LCSE

A la réception d'un message OpenLogicalChannel 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'établissement de la voie logique en émettant la primitive de réponse ESTABLISH et un message OpenLogicalChannelAck est envoyé à l'entité B-LCSE sortante homologue. La voie directe du canal logique bidirectionnel peut dès lors être utilisée pour recevoir les informations d'utilisateur. L'utilisateur de l'entité B-LCSE entrante signale le rejet de la demande d'établissement de la voie logique en émettant la primitive de demande RELEASE et un message OpenLogicalChannelReject est envoyé à l'entité B-LCSE sortante homologue.

A la réception d'un message OpenLogicalChannelConfirm dans l'entité B-LCSE entrante, l'utilisateur est informé de l'établissement du canal logique bidirectionnel par la primitive de confirmation ESTABLISH. La voie inverse du canal logique bidirectionnel peut dès lors être utilisée pour transmettre les informations d'utilisateur.

Une voie logique qui a été établie avec succès peut être fermée à la réception d'un message CloseLogicalChannel dans l'entité B-LCSE entrante. L'utilisateur de l'entité B-LCSE entrante est informé de la fermeture de la voie logique par la primitive d'indication RELEASE et le message CloseLogicalChannelAck est envoyé à l'entité B-LCSE sortante homologue.

Remplacée par une version plus récente

8.5.1.3 Résolution des conflits

Le lancement simultané de demandes d'ouverture de voies logiques peut donner lieu à des conflits. La connaissance des capacités échangées peut permettre de déterminer l'existence d'un conflit. Dans d'autres occasions, les deux terminaux peuvent déclencher l'ouverture d'un canal logique bidirectionnel pour le même usage, bien que les paramètres exacts demandés puissent être différents et que les deux terminaux aient la capacité suffisante pour les deux demandes. Les terminaux doivent être en mesure de détecter la présence de ces deux situations et doivent agir comme suit.

Pour pouvoir ouvrir les voies logiques, il faut au préalable choisir un des deux terminaux comme terminal maître et l'autre comme terminal esclave. Le protocole défini au 8.2 indique une manière de procéder à ce choix. Le terminal maître doit rejeter immédiatement toute demande en provenance du terminal esclave qu'il identifie comme étant source de conflits. Le terminal esclave peut identifier ces conflits mais il doit répondre à la demande en provenance du terminal maître, en sachant que sa demande précédente sera rejetée.

Dans le second type de conflit défini ci-dessus, il est impossible de distinguer les cas dans lesquels deux canaux bidirectionnels sont effectivement utiles de ceux dans lesquels un seul de ces canaux suffit. Dans leur réponse, les terminaux supposeront qu'un seul canal bidirectionnel est utile, mais un terminal peut ultérieurement renouveler sa demande si cette supposition s'avère inexacte.

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

L'entité B-LCSE et l'utilisateur de l'entité B-LCSE communiquent à l'aide des primitives indiquées dans le Tableau 31.

TABLEAU 31/H.245

Primitives et paramètres

Nom générique	Type			
	demande	indication	réponse	confirmation
ESTABLISH	FORWARD_PARAM REVERSE_PARAM	FORWARD_PARAM REVERSE_PARAM	REVERSE_DATA	REVERSE_DATA
RELEASE	CAUSE	SOURCE CAUSE	Non défini (Note 2)	– (Note 1)
ERROR	Non défini	ERRCODE	Non défini	Non défini

NOTE 1 – «–» signifie absence de paramètres.
NOTE 2 – «Non défini» signifie que cette primitive n'existe pas.

8.5.2.2 Définition des primitives

Ces primitives sont définies comme suit:

- 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 signale les erreurs de l'entité B-LCSE à une entité de gestion.

8.5.2.3 Définition des paramètres

Les paramètres des primitives indiqués au Tableau 31 sont définis comme suit:

- le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie directe, c'est-à-dire celle qui va du terminal contenant l'entité B-LCSE sortante jusqu'au terminal contenant l'entité B-LCSE entrante. Ce paramètre correspond au champ forwardLogicalChannelParameters du message OpenLogicalChannel et est acheminé en transparence vers l'utilisateur de l'entité LCSE homologue;

Remplacée par une version plus récente

- b) le paramètre REVERSE_PARAM spécifie les paramètres associés à la voie inverse, c'est-à-dire celle qui va du terminal contenant l'entité B-LCSE entrante jusqu'au terminal contenant l'entité B-LCSE sortante. Ce paramètre correspond au champ reverseLogicalChannelParameters du message OpenLogicalChannel et est acheminé en transparence vers l'utilisateur de l'entité LCSE homologue;
- c) le paramètre REVERSE_DATA spécifie quelques uns des paramètres associés à la voie inverse, c'est-à-dire celle qui va du terminal contenant l'entité B-LCSE entrante jusqu'au terminal contenant l'entité B-LCSE sortante. Ce paramètre est appliqué au champ reverseLogicalChannelParameters du message OpenLogicalChannelAck et est acheminé en transparence vers l'utilisateur de l'entité B-LCSE homologue;
- d) le paramètre SOURCE indique à l'utilisateur de l'entité B-LCSE l'origine de la libération de la voie logique. Ce paramètre prend la valeur «USER» ou «B-LCSE», désignant l'utilisateur de l'entité B-LCSE ou l'entité B-LCSE elle-même. Ce dernier cas peut découler d'une erreur de protocole;
- e) le paramètre CAUSE indique le motif pour lequel l'utilisateur de l'entité B-LCSE homologue a rejeté une demande d'établissement d'une 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 du 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 entités B-LCSE homologues. Les états sont spécifiés séparément pour chacune des entités B-LCSE sortante et B-LCSE entrante. Les états d'une entité B-LCSE sortante sont les suivants:

état 0: RELEASED (libre)

La voie logique est libérée. Elle ne doit pas être utilisée pour envoyer ou recevoir des données.

état 1: AWAITING ESTABLISHMENT (attente d'établissement)

L'entité B-LCSE sortante attend de pouvoir établir une voie logique avec une entité B-LCSE entrante homologue. La voie logique ne doit pas être utilisée pour envoyer ou recevoir des données.

état 2: ESTABLISHED (voie établie)

La connexion de la voie logique entre entités B-LCSE homologues a été établie. La voie logique peut être utilisée pour envoyer et recevoir des données.

état 3: AWAITING RELEASE (attente de libération)

L'entité B-LCSE sortante attend de pouvoir libérer une voie logique avec l'entité B-LCSE entrante homologue. La voie logique ne doit pas être utilisée pour envoyer des données, mais elle continue de pouvoir être utilisée pour la réception de données.

Les états d'une entité B-LCSE entrante sont les suivants:

état 0: RELEASED (libre)

La voie logique est libérée. Elle ne doit pas être utilisée pour recevoir ou envoyer des données.

état 1: AWAITING ESTABLISHMENT (attente d'établissement)

L'entité B-LCSE entrante attend de pouvoir établir une voie logique avec une entité B-LCSE sortante homologue. La voie logique ne doit pas être utilisée pour recevoir ou envoyer des données.

état 2: AWAITING CONFIRMATION (attente de confirmation)

L'entité B-LCSE entrante attend confirmation de l'établissement de la voie logique avec une entité B-LCSE sortante homologue. La voie logique ne doit pas être utilisée pour envoyer des données, mais des données peuvent y être reçues.

état 3: ESTABLISHED (voie établie)

Une connexion de voie logique entre entités B-LCSE homologues a été établie. La voie logique peut être utilisée pour recevoir et envoyer des données.

8.5.2.5 Diagramme de transition d'états

La séquence autorisée de primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE est définie ici. Cette séquence autorisée de primitives se rapporte aux états de l'entité B-LCSE considérés du point de vue de l'utilisateur de l'entité B-LCSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités B-LCSE sortante et B-LCSE entrante, comme le montrent respectivement la Figure 15 et la Figure 16.

Remplacée par une version plus récente

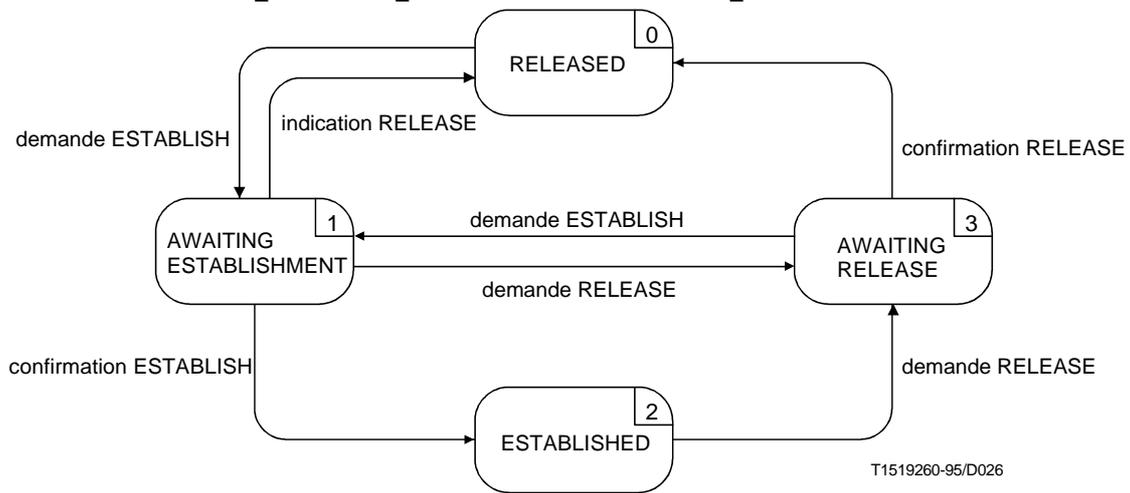


FIGURE 15/H.245

Diagramme de transition d'états pour la séquence de primitives dans l'entité B-LCSE sortante

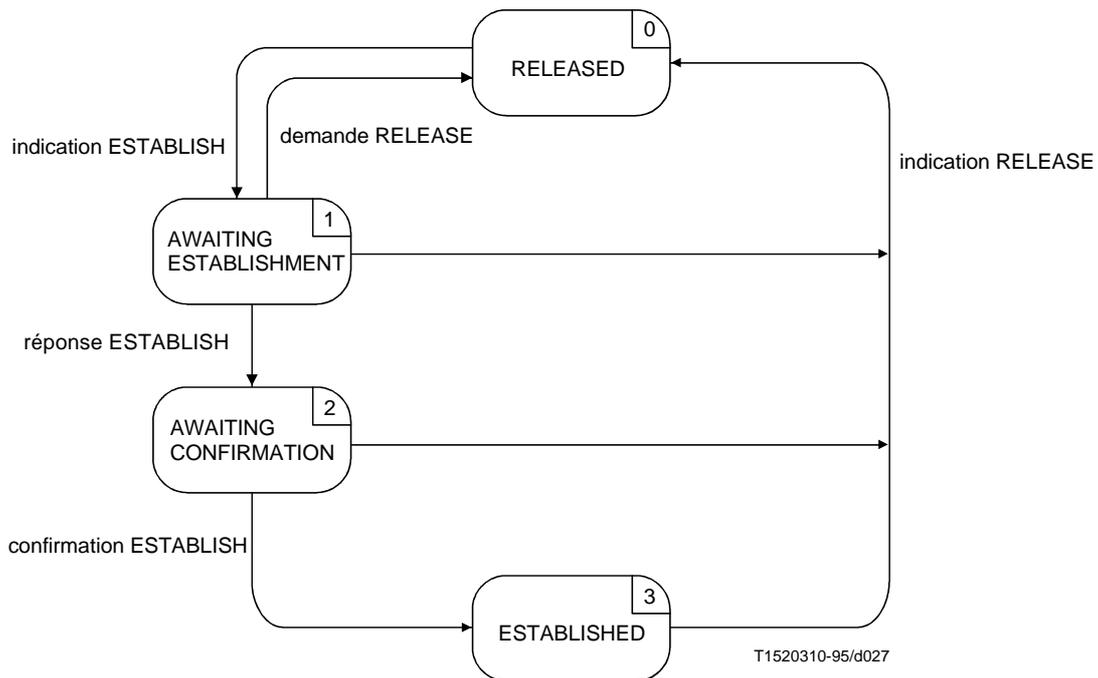


FIGURE 16/H.245

Diagramme de transition d'états pour la séquence de primitives dans l'entité B-LCSE entrante

Remplacée par une version plus récente

8.5.3 Communication entre entités B-LCSE homologues

8.5.3.1 Messages de l'entité B-LCSE

Le Tableau 32 montre les messages et les champs de l'entité B-LCSE définis au paragraphe 6 et se rapportant au protocole des entités B-LCSE.

TABLEAU 32/H.245

Noms et champs des messages de l'entité B-LCSE

Fonction	Message	Sens	Champ
Etablissement	OpenLogicalChannel	S → E	forwardLogicalChannelNumber forwardLogicalChannelParameters reverseLogicalChannelParameters
	OpenLogicalChannelAck	S ← E	forwardLogicalChannelNumber reverseLogicalChannelParameters
	OpenLogicalChannelReject	S ← E	forwardLogicalChannelNumber cause
	OpenLogicalChannelConfirm	S ← E	forwardLogicalChannelNumber
Libération	CloseLogicalChannel	S → E	forwardLogicalChannelNumber
	CloseLogicalChannelAck	S ← E	source forwardLogicalChannelNumber
S Sortante E Entrante			

8.5.3.2 Variables d'état de l'entité B-LCSE

La variable d'état suivante est définie pour l'entité B-LCSE sortante:

out_LCN

Cette variable d'état permet de distinguer les diverses entités B-LCSE sortantes. Elle est initialisée lors du lancement de l'entité B-LCSE sortante. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages B-LCSE envoyés depuis une entité B-LCSE sortante. Pour les messages B-LCSE reçus dans une entité B-LCSE sortante, la valeur du champ forwardLogicalChannelNumber de ces messages est identique à la valeur de out_LCN.

La variable d'état suivante est définie pour l'entité B-LCSE entrante:

in_LCN

Cette variable d'état permet de distinguer les entités B-LCSE entrantes. Elle est initialisée lors du lancement de l'entité B-LCSE entrante. La valeur de in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages B-LCSE envoyés depuis une entité B-LCSE entrante. Pour les messages B-LCSE reçus dans une entité B-LCSE entrante, la valeur du champ forwardLogicalChannelNumber de ces messages est identique à la valeur de in_LCN.

8.5.3.3 Temporiseurs B-LCSE

Le temporisateur suivant est spécifié pour les entités B-LCSE sortante et entrante:

T103

Dans l'entité B-LCSE sortante, ce temporisateur est utilisé aux états AWAITING ESTABLISHMENT (attente d'établissement) et AWAITING RELEASE (attente de libération). Il spécifie le temps maximal pendant lequel aucun message OpenLogicalChannelAck, OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu.

Dans l'entité B-LCSE entrante, ce temporisateur est utilisé à l'état AWAITING CONFIRMATION (attente de confirmation). Il spécifie le temps maximal pendant lequel aucun message OpenLogicalChannelConfirm ne peut être reçu.

Remplacée par une version plus récente

8.5.4 Procédures de l'entité 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 sortante et entrante.

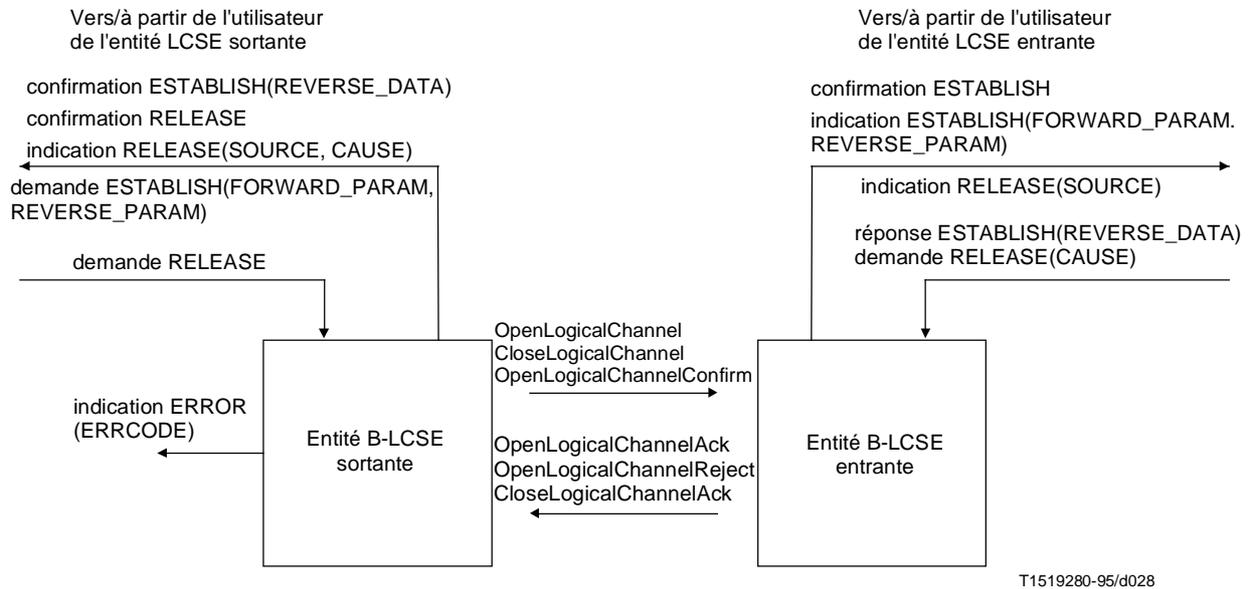


FIGURE 17/H.245

Primitives et messages dans l'entité de signalisation de canal logique bidirectionnel

8.5.4.2 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs par défaut indiquées dans le Tableau 33, sauf lorsque leurs valeurs sont expressément mentionnées dans les diagrammes SDL.

TABLEAU 33/H.245

Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut (Note)
indication ESTABLISH	FORWARD_PARAM REVERSE_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters OpenLogicalChannel.reverseLogicalChannelParameters
indication RELEASE	SOURCE CAUSE	CloseLogicalChannel.source null

NOTE – Lorsque le message ne comporte aucune indication de champ, la valeur de codage d'un paramètre de primitive doit être nulle.

Remplacée par une version plus récente

8.5.4.3 Valeurs par défaut des champs des messages

Les champs de message prennent les valeurs par défaut indiquées dans le Tableau 34, sauf lorsque leurs valeurs sont expressément mentionnées dans les diagrammes SDL.

TABLEAU 34/H.245

Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut (Note)
OpenLogicalChannel	forwardLogicalChannelNumber	out_LCN
	forwardLogicalChannelParameters	demande ESTABLISH(FORWARD_PARAM)
	reverseLogicalChannelParameters	demande ESTABLISH(REVERSE_PARAM)
OpenLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
	reverseLogicalChannelParameters	réponse ESTABLISH(REVERSE_DATA)
OpenLogicalChannelReject	forwardLogicalChannelNumber cause	in_LCN demande RELEASE(CAUSE)
OpenLogicalChannelConfirm	forwardLogicalChannelNumber	out_LCN
CloseLogicalChannel	forwardLogicalChannelNumber	out_LCN
	source	user
CloseLogicalChannelAck	forwardLogicalChannelNumber	in_LCN
NOTE – Un champ de message ne doit pas être codé si le paramètre correspondant de la primitive est nul, c'est-à-dire s'il est absent.		

8.5.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR.indication indique une situation d'erreur particulière. Le Tableau 35 montre les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE sortante et le Tableau 36 montre les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE entrante.

TABLEAU 35/H.245

Valeurs du paramètre ERRCODE dans l'entité B-LCSE sortante

Type d'erreur	Code d'erreur	Situation d'erreur	Etat
Message incorrect	A	OpenLogicalChannelAck	RELEASED
	B	OpenLogicalChannelReject	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
Pas de réponse de l'entité B-LCSE homologue	D	expiration du temporisateur T103	AWAITING ESTABLISHMENT AWAITING RELEASE

Remplacée par une version plus récente

TABLEAU 36/H.245

Valeurs du paramètre ERRCODE dans l'entité B-LCSE entrante

Type d'erreur	Code d'erreur	Situation d'erreur	Etat
Message incorrect	E	OpenLogicalChannelConfirm	AWAITING ESTABLISHMENT
Pas de réponse 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 sortante et de l'entité B-LCSE entrante sont décrites en langage SDL aux Figures 18 et 19 respectivement.

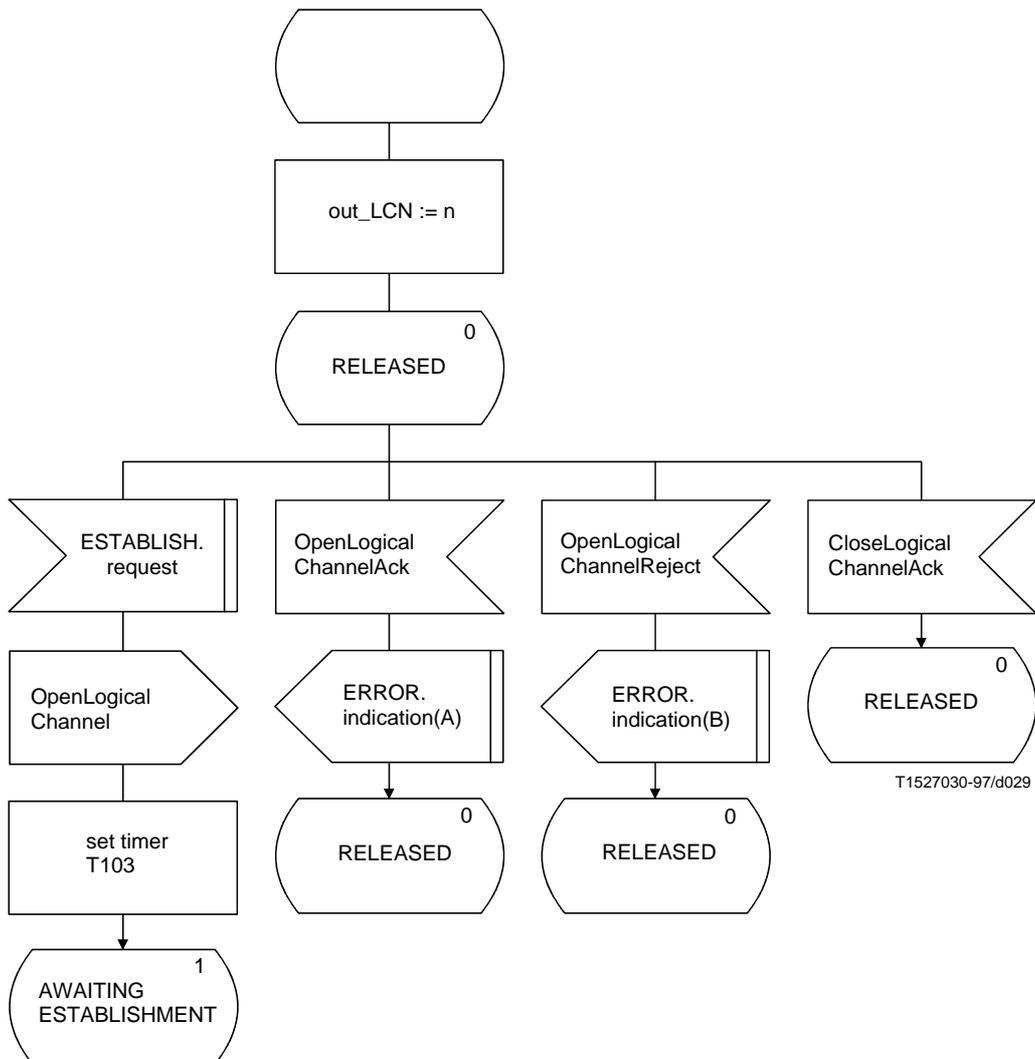
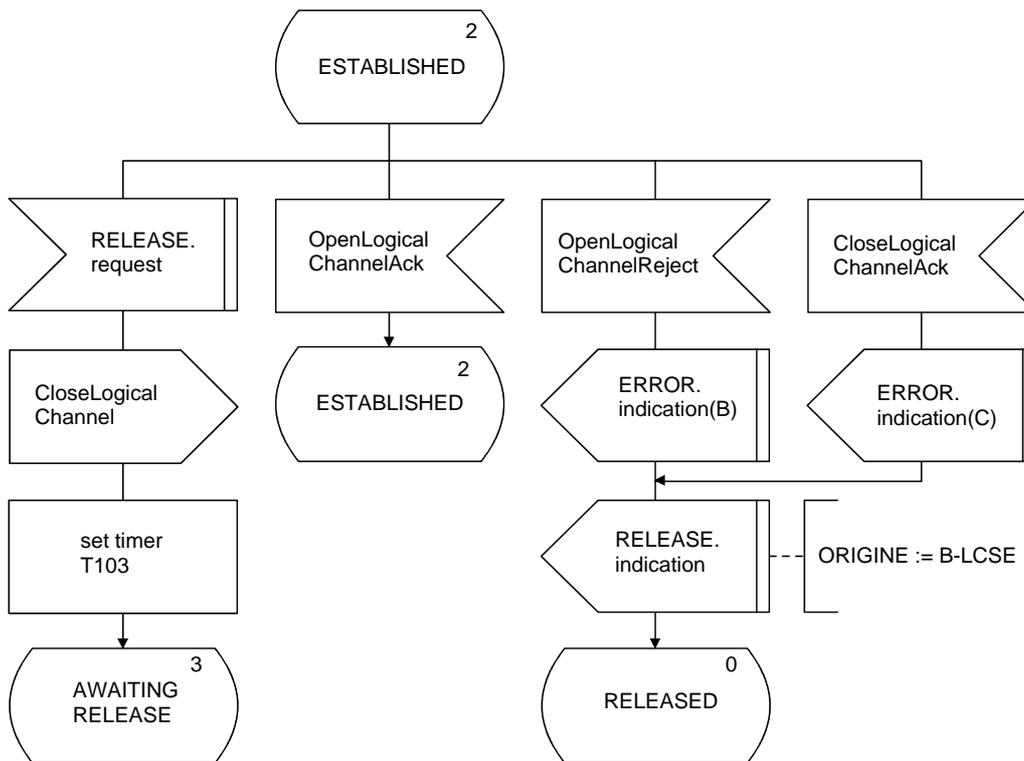


FIGURE 18 i)/H.245

Description SDL de l'entité B-LCSE sortante

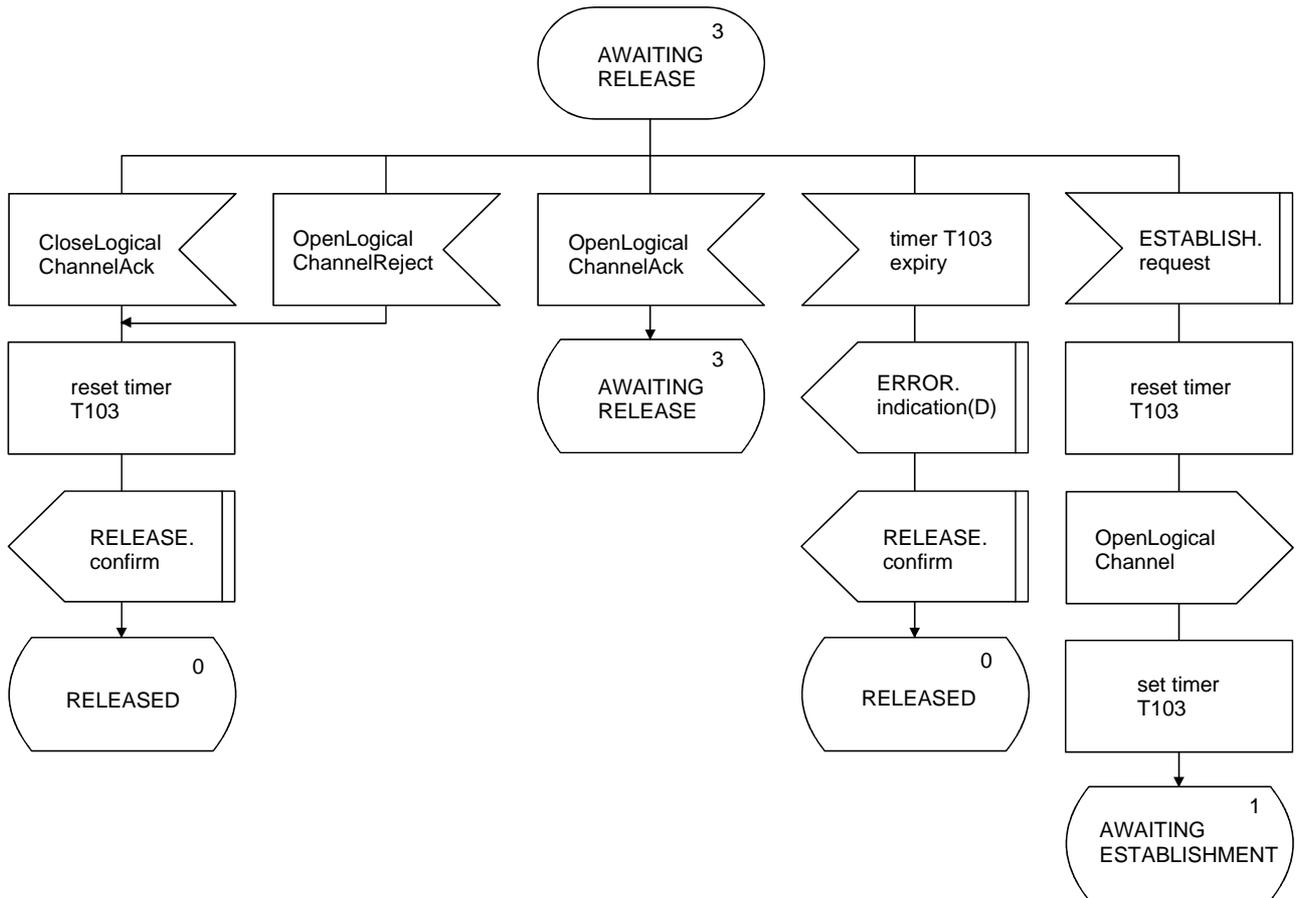
Remplacée par une version plus récente



T1519310-95/d031

FIGURE 18 iii)/H.245
Description SDL de l'entité B-LCSE sortante

Remplacée par une version plus récente



T1519320-95/d032

FIGURE 18 iv)/H.245
Description SDL de l'entité B-LCSE sortante

Remplacée par une version plus récente

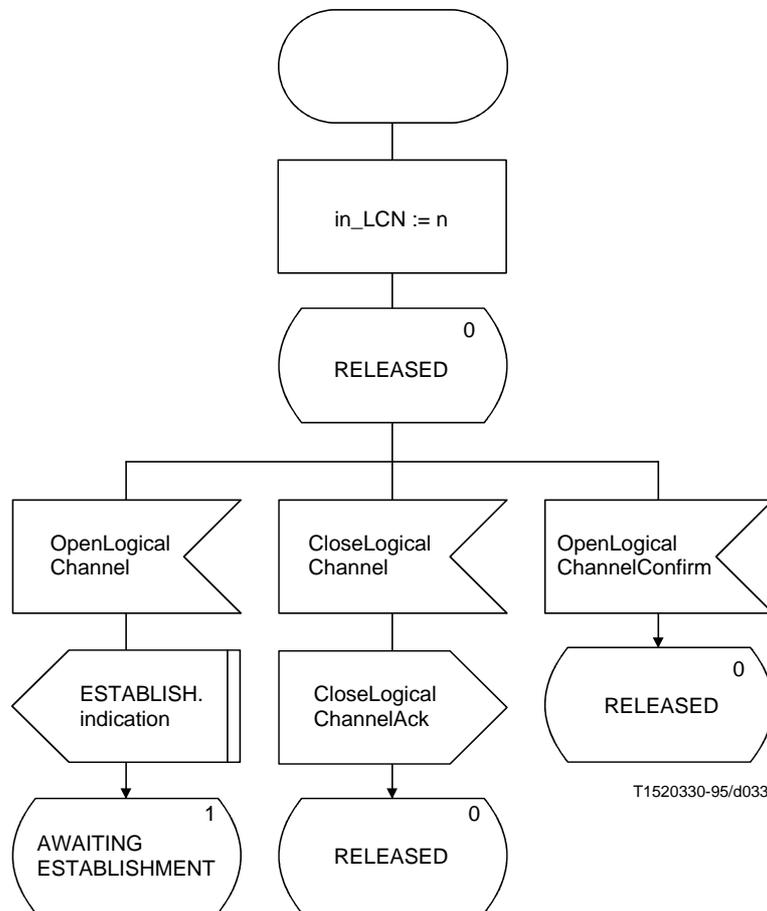
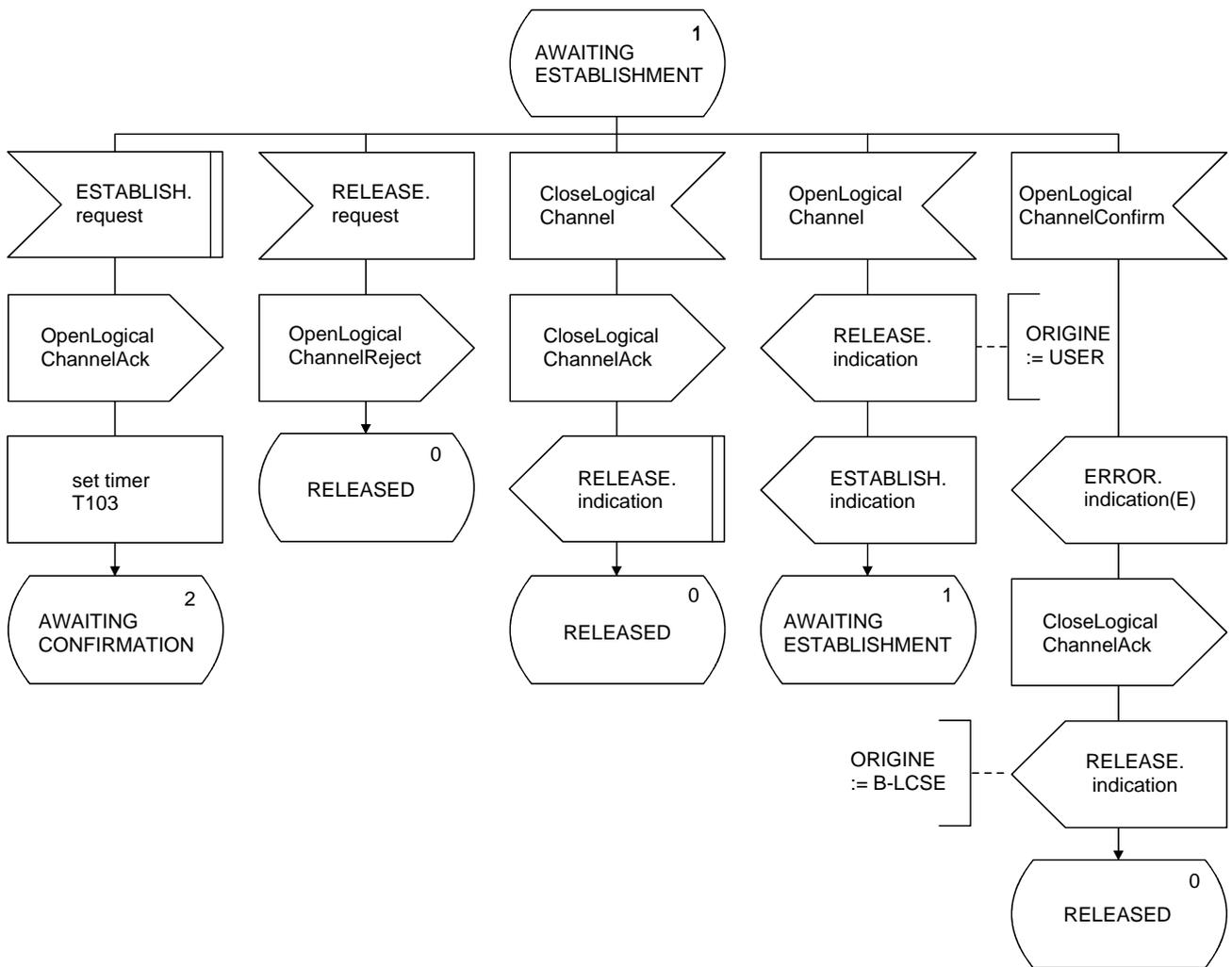


FIGURE 19 i)/H.245

Description SDL de l'entité B-LCSE entrante

Remplacée par une version plus récente



T1520340-95/d034

FIGURE 19 ii)/H.245

Description SDL de l'entité B-LCSE entrante

Remplacée par une version plus récente

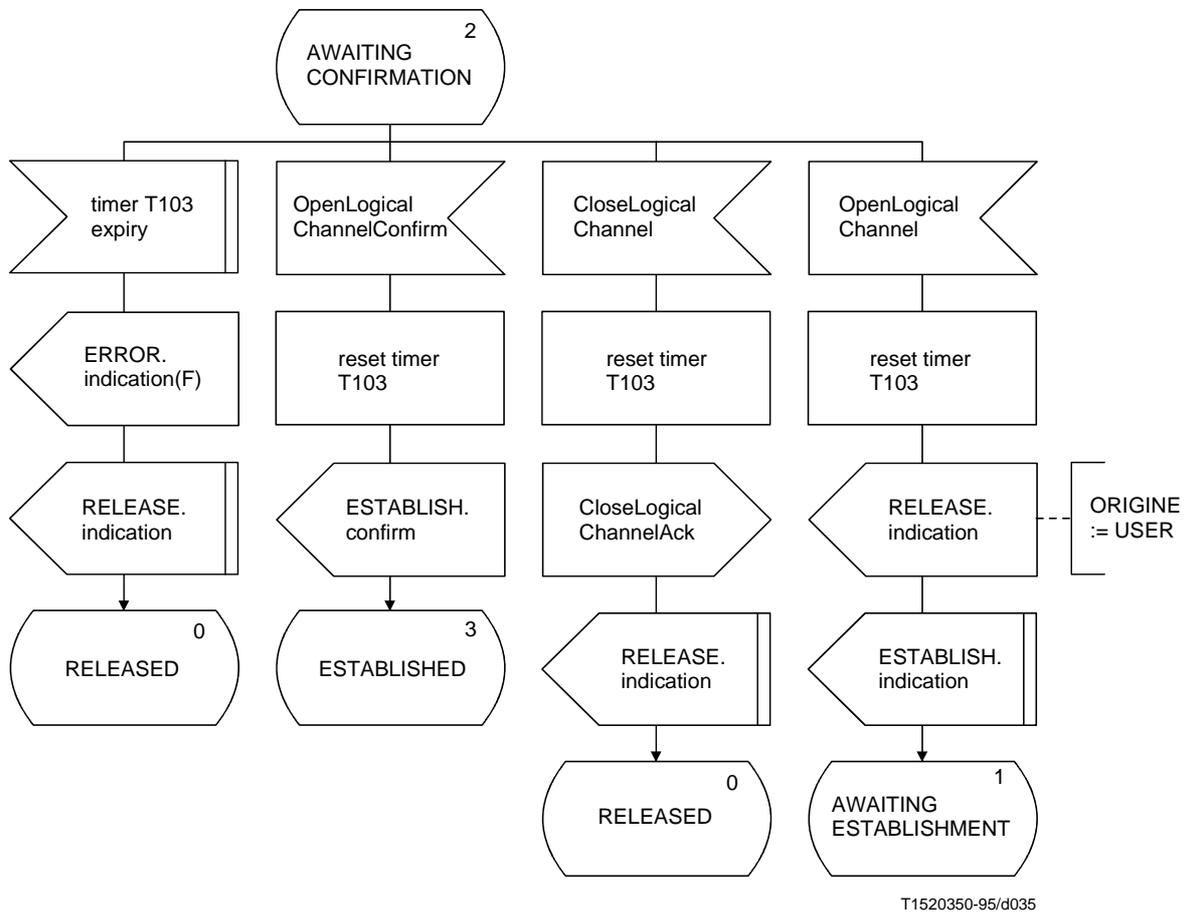


FIGURE 19 iii)/H.245

Description SDL de l'entité B-LCSE entrante

Remplacée par une version plus récente

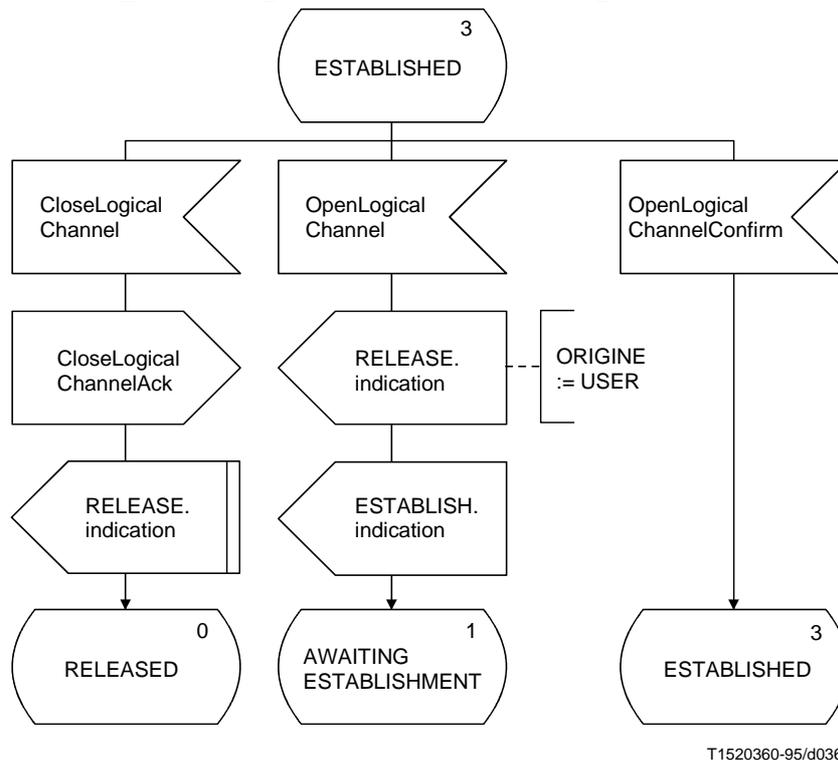


FIGURE 19 iv)/H.245

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 au terminal distant de fermer une voie logique. On notera que ces procédures ne concernent que les demandes de fermeture; l'opération de fermeture de voie logique s'effectue au moyen des procédures d'entités LCSE et B-LCSE. Ces procédures sont ici désignées sous le terme *entité de signalisation de fermeture de voies logiques (CLCSE)*. 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 transmises vers l'entité CLCSE homologue par des messages appropriés, définis dans le paragraphe 6. Il existe une entité CLCSE sortante et une entité CLCSE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité CLCSE pour chaque voie logique.

Si un terminal se trouve dans l'impossibilité de traiter les signaux entrants, il peut utiliser ces procédures pour demander la fermeture des voies logiques appropriées. Le texte suivant donne un aperçu du fonctionnement du protocole. En cas de divergence entre ce texte et la spécification formelle du protocole qui suit, cette dernière prévaut.

Un terminal qui répond à une telle action positivement, c'est-à-dire en émettant la primitive de réponse CLOSE, doit déclencher la fermeture de la voie logique en envoyant la primitive de demande RELEASE à l'entité LCSE ou B-LCSE appropriée le plus tôt possible.

8.6.1.1 Aperçu général du protocole – Entité CLCSE sortante

Une procédure de demande de fermeture de voie logique est lancée lorsque la primitive de demande CLOSE est émise par l'utilisateur situé dans l'entité CLCSE sortante. Un message RequestChannelClose est envoyé à l'entité CLCSE homologue entrante et le temporisateur T108 est lancé. Si un message RequestChannelCloseAck est reçu en réponse au message RequestChannelClose, le temporisateur T108 est arrêté et l'utilisateur est informé, par la primitive de confirmation CLOSE, que la procédure de demande de fermeture de voie logique a été appliquée. Si toutefois un

Remplacée par une version plus récente

message RequestChannelCloseReject est reçu en réponse au message RequestChannelClose, le temporisateur T108 est arrêté et l'utilisateur est informé, par la primitive d'indication REJECT, que l'utilisateur de l'entité CLCSE homologue a refusé de fermer la voie logique.

Si le temporisateur T108 expire, l'utilisateur de l'entité CLCSE sortante en 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

Lorsqu'un message RequestChannelClose est reçu à 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 la demande de fermeture de voie logique en émettant la primitive de réponse CLOSE et un message RequestChannelCloseAck est envoyé à l'entité CLCSE homologue sortante. L'utilisateur de l'entité CLCSE entrante signale le rejet de la demande de fermeture de voie logique en émettant la primitive de demande REJECT et un message RequestChannelCloseReject est envoyé à l'entité CLCSE homologue sortante.

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

Les communications entre l'entité CLCSE et l'utilisateur de l'entité CLCSE s'effectuent 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 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:

- les primitives de type CLOSE sont utilisées pour demander la fermeture d'une voie logique;
- 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:

- 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 provient d'une fin de temporisation.
- 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 de l'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».

Remplacée par une version plus récente

Etat 1: AWAITING RESPONSE

L'entité CLCSE attend une réponse provenant de l'entité CLCSE distante.

Les états pour 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 transition d'états

Ce sous-paragraphe définit la séquence autorisée de primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités CLCSE sortante et entrante, comme cela est indiqué dans les Figures 20 et 21 respectivement.

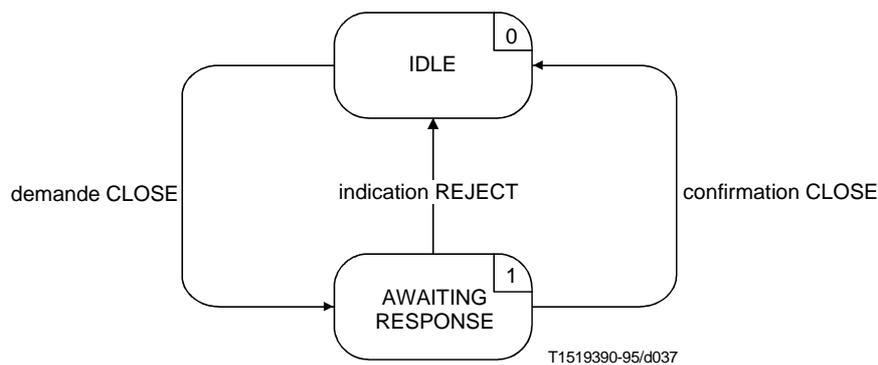


FIGURE 20/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité CLCSE sortante

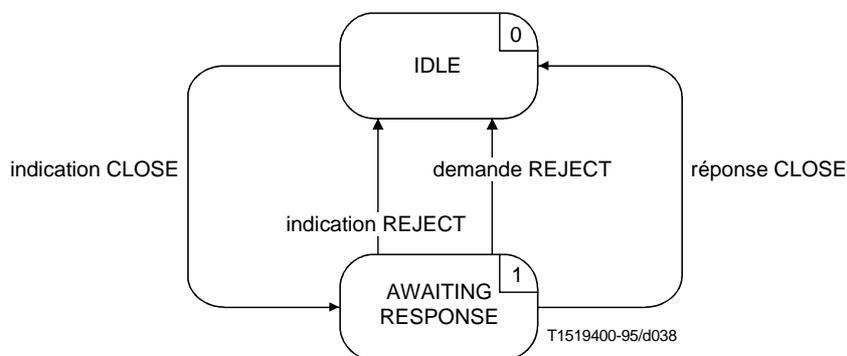


FIGURE 21/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité CLCSE entrante

Remplacée par une version plus récente

8.6.3 Communications 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 sont 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	Sortante		
E	Entrante		

8.6.3.2 Variables d'état de l'entité CLCSE

La variable d'état suivante est définie pour l'entité CLCSE sortante:

out_LCN

Cette variable d'état permet de distinguer les entités CLCSE sortantes. Elle est initialisée lors du lancement de l'entité CLCSE sortante. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages CLCSE envoyés depuis une entité CLCSE sortante. Pour les messages CLCSE reçus dans une entité CLCSE sortante, la valeur de champ forwardLogicalChannelNumber de ces messages est identique à la valeur de out_LCN.

La variable d'état suivante est définie pour l'entité CLCSE entrante:

in_LCN

Cette variable d'état permet de distinguer les entités CLCSE entrantes. Elle est initialisée lors du lancement de l'entité CLCSE entrante. La valeur de in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages CLCSE envoyés depuis une entité CLCSE entrante. Pour les messages CLCSE reçus dans une entité CLCSE entrante, la valeur de champ forwardLogicalChannelNumber de ces messages est identique à la valeur de in_LCN.

8.6.3.3 Temporisateurs CLCSE

Le temporisateur suivant est spécifié pour l'entité CLCSE sortante:

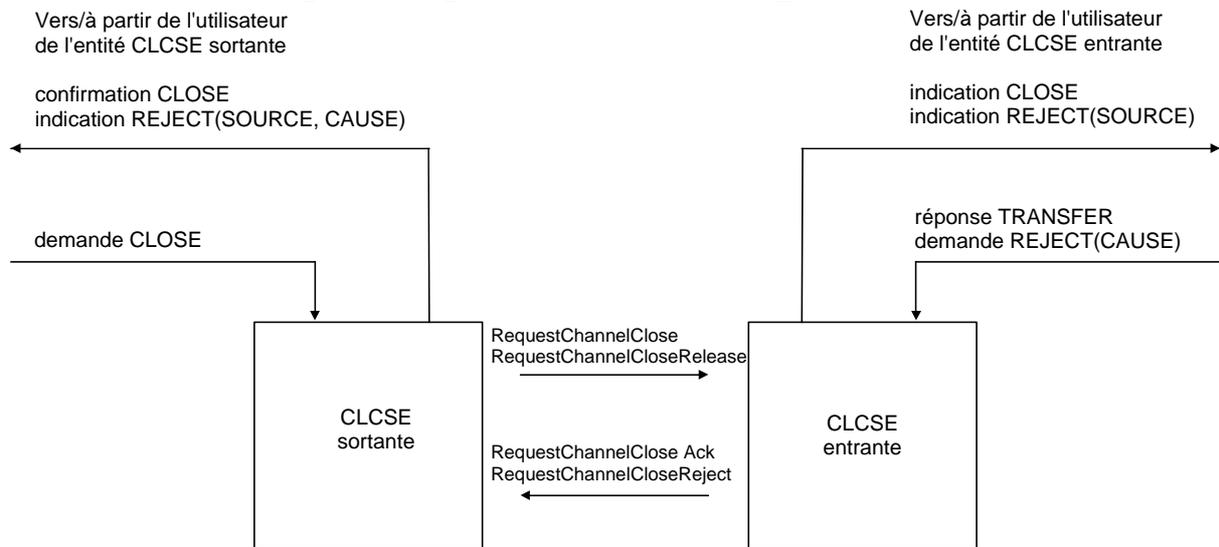
T108

Ce temporisateur est utilisé à l'état AWAITING RESPONSE (attente de la réponse). Il spécifie le temps maximal écoulé avant qu'un message d'erreur ne soit produit, pendant lequel aucun message RequestChannelCloseAck ou RequestChannelCloseReject ne peut être reçu.

8.6.4 Procédures de l'entité CLCSE

La Figure 22 récapitule les primitives CLCSE et leurs paramètres, de même que les messages, pour chacune des entités CLCSE sortante et entrante.

Remplacée par une version plus récente



T1519410-95/d039

FIGURE 22/H.245

Primitives et messages dans l'entité de signalisation de fermeture de la voie logique (CLCSE)

8.6.4.1 Valeurs par défaut des paramètres des 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 des primitives par défaut

Primitive	Paramètre	Valeur par défaut
indication REJECT	SOURCE	USER
	CAUSE	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 messages

Message	Champ	Valeur par défaut
RequestChannelClose	forwardLogicalChannelNumber	out_LCN
RequestChannelCloseAck	forwardLogicalChannelNumber	in_LCN
RequestChannelCloseReject	forwardLogicalChannelNumber	in_LCN
	cause	demande REJECT(CAUSE)
RequestChannelCloseRelease	forwardLogicalChannelNumber	out_LCN

Remplacée par une version plus récente

8.6.4.3 Description SDL

Les procédures des entités CLCSE sortantes et entrantes sont exprimées en SDL, respectivement dans les Figures 23 et 24.

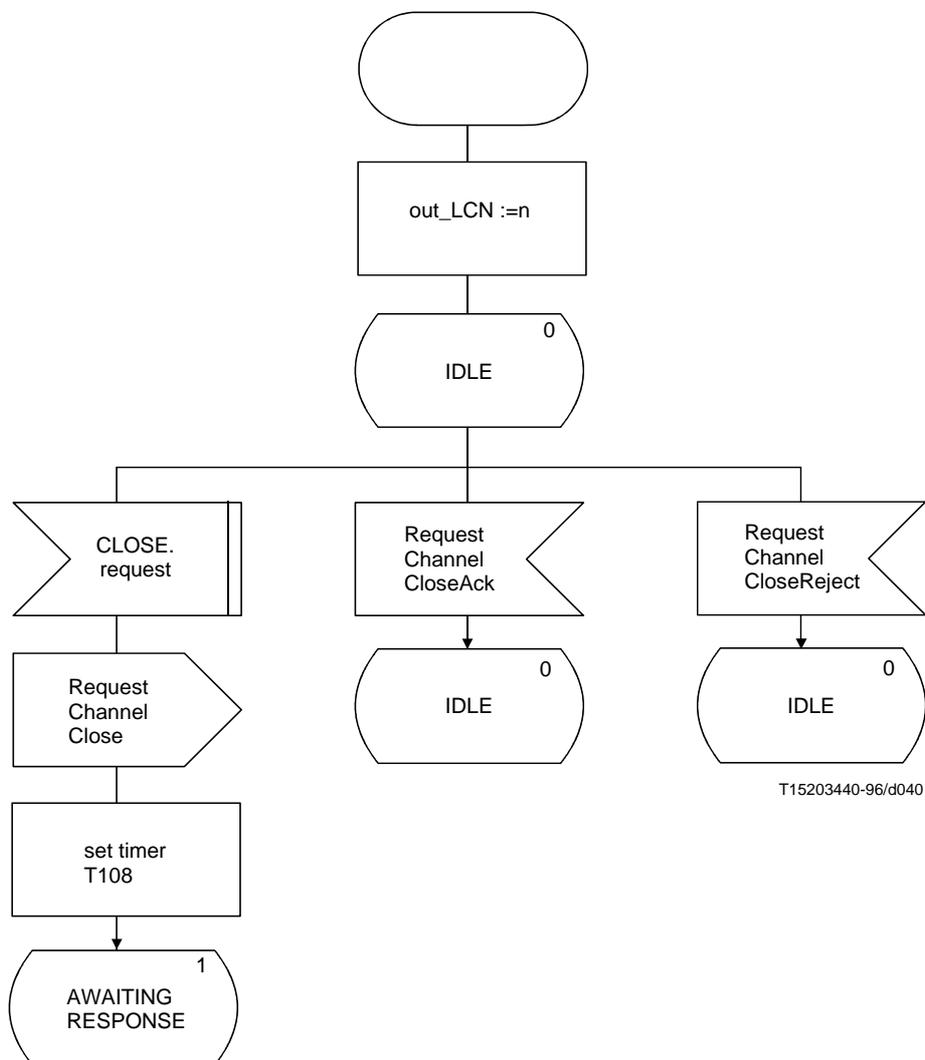
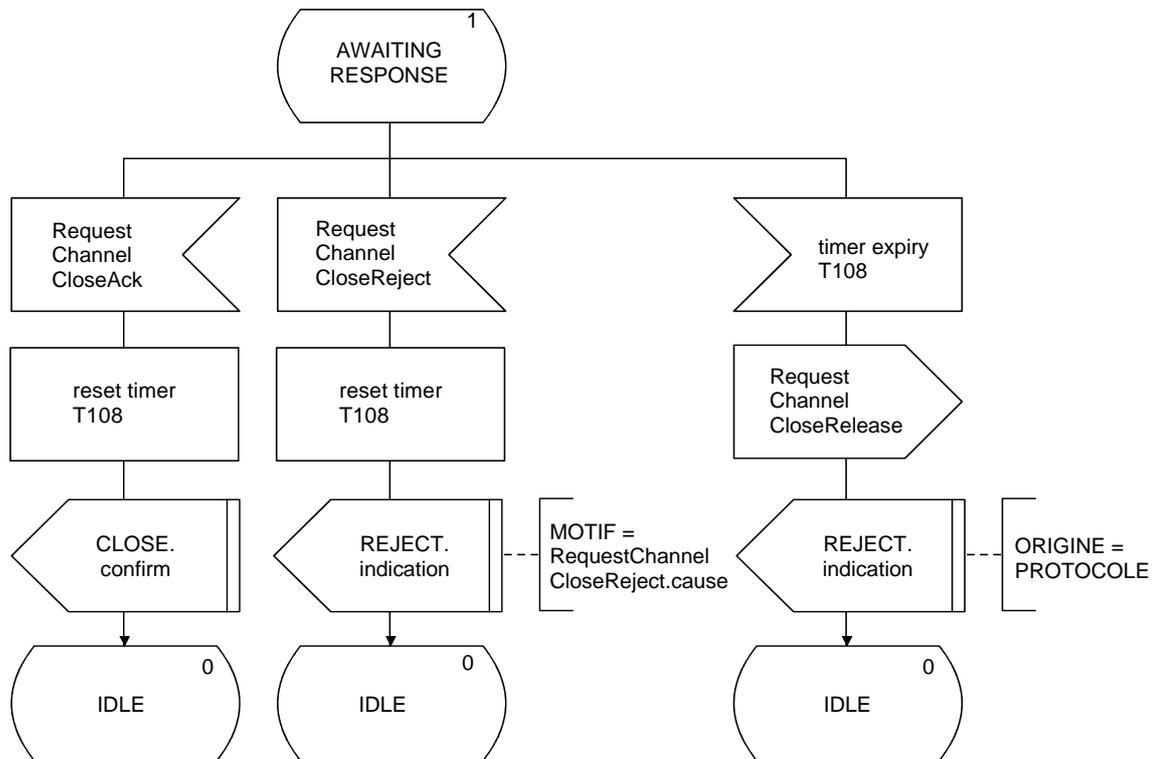


FIGURE 23 i)/H.245

Description SDL de l'entité CLCSE sortante

Remplacée par une version plus récente



T1523450-96/d041

FIGURE 23 ii)/H.245

Description SDL de l'entité CLCSE sortante

Remplacée par une version plus récente

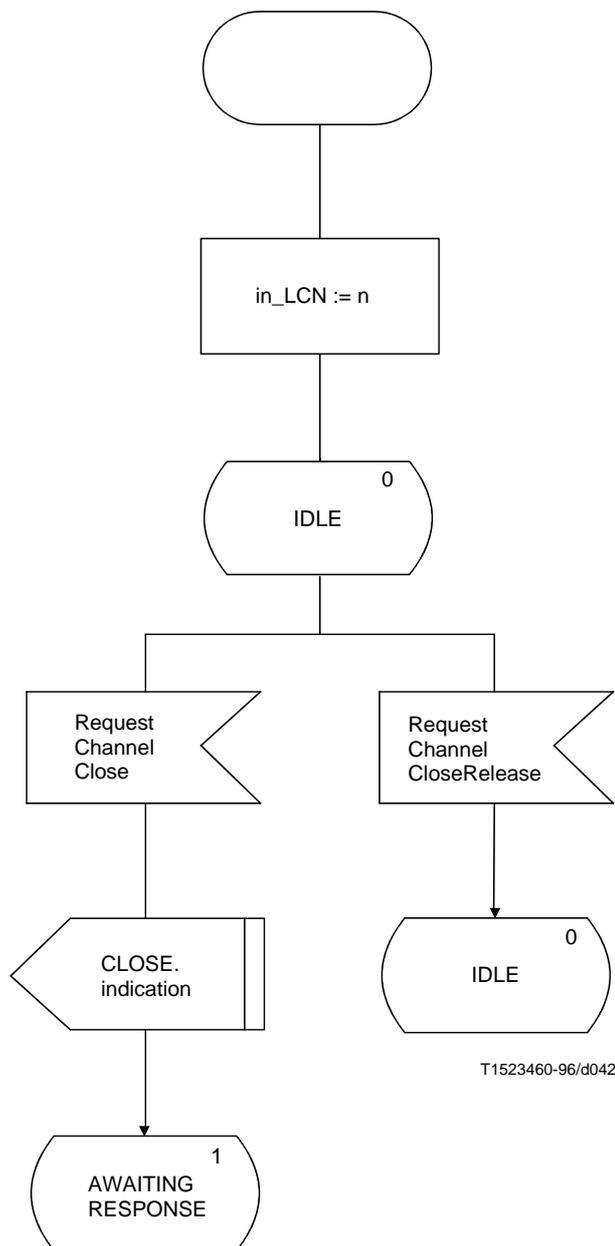


FIGURE 24 i)/H.245

Description SDL de l'entité CLCSE entrante

Remplacée par une version plus récente

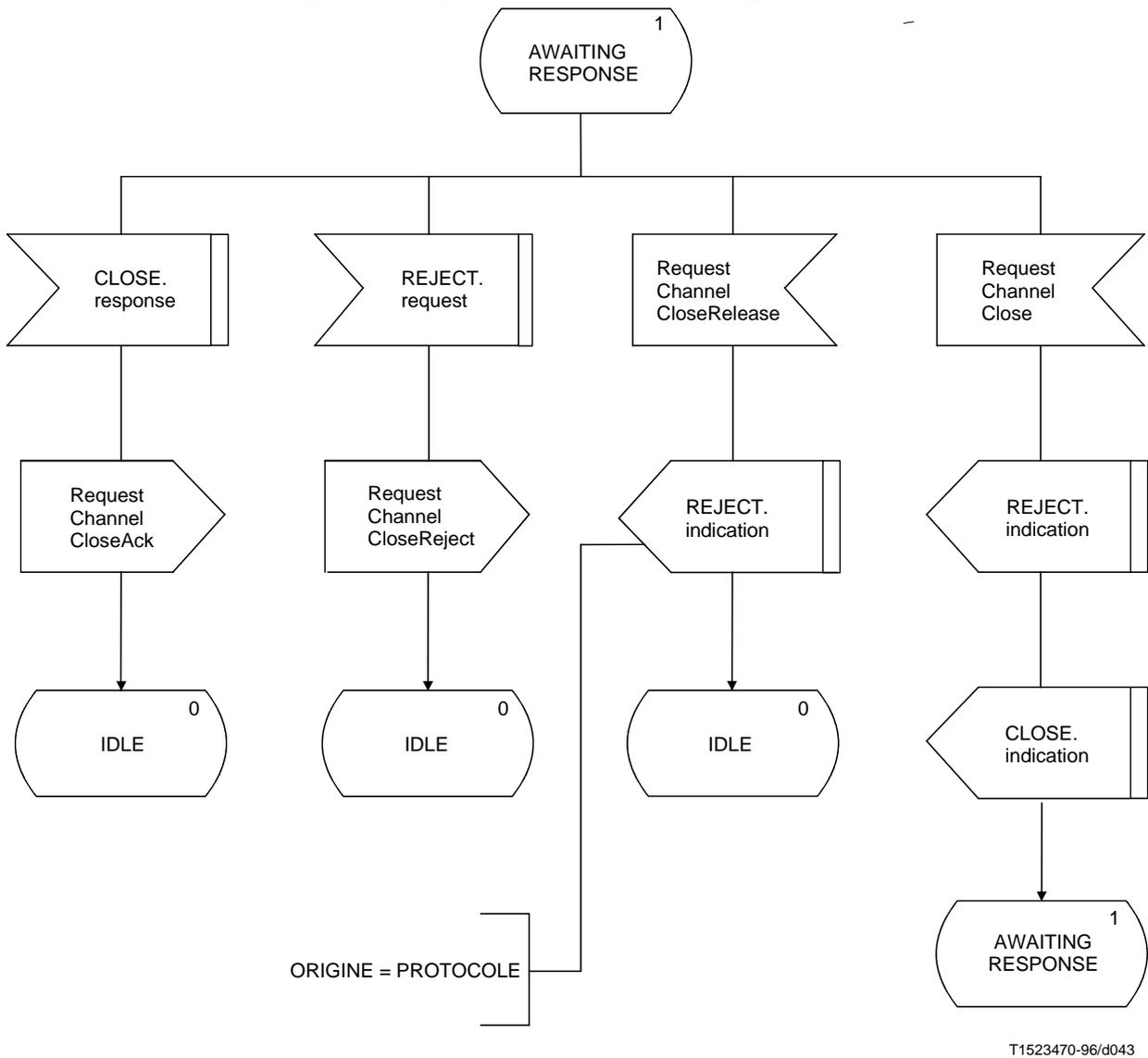


FIGURE 24 ii)/H.245

Description SDL de l'entité CLCSE entrante

8.7 Procédures relatives au tableau de multiplexage selon la Recommandation H.223

8.7.1 Introduction

Le tableau de multiplexage permet d'associer chaque octet à l'intérieur d'un message H.223 MUX-message [8] à un numéro particulier de voie logique. Le tableau de multiplexage H.223 peut comporter jusqu'à 16 entrées numérotées de 0 à 15. Les entrées de tableau 1 à 15 doivent être envoyées par des émetteurs vers des récepteurs comme l'indiquent les procédures suivantes.

Les procédures décrites ici sont désignées sous le nom d'entité de signalisation du tableau de multiplexage (MTSE). Ces procédures 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 à l'entité MTSE homologue par les messages appropriés définis au paragraphe 6.

Il y a une entité MTSE sortante et une entité MTSE entrante. Il y a une instance de l'entité MTSE pour chaque entrée de tableau de multiplexage.

Remplacée par une version plus récente

Un terminal émetteur utilise ce protocole pour signaler à un terminal distant une ou plusieurs nouvelles entrées de tableau de multiplexage. Ce terminal distant peut accepter ou rejeter ces nouvelles entrées. Si le terminal distant accepte une entrée de tableau de multiplexage, l'entrée précédente au numéro d'entrée donné est remplacée par la nouvelle entrée.

L'émetteur peut désactiver une entrée de tableau de multiplexage en envoyant un descripteur MultiplexEntryDescriptor sans paramètre elementList. L'émetteur ne doit en aucun cas utiliser une entrée de tableau de multiplexage désactivée. Avant de transmettre un message MultiplexEntrySend, l'émetteur doit cesser d'utiliser les entrées décrites dans ce message. Il ne doit pas recommencer à utiliser ces entrées avant d'avoir reçu un message MultiplexEntrySendAck. On procède ainsi car si on ne cesse pas d'utiliser ces entrées de tableau de multiplexage avant d'envoyer le message MultiplexEntrySend, des erreurs pouvant prêter à confusion risquent d'être introduites dans le récepteur.

L'émetteur doit cesser d'utiliser des entrées désactivées avant d'envoyer le message MultiplexEntrySend indiquant que ces entrées ont été désactivées. Les entrées désactivées peuvent être réutilisées à tout moment par la transmission d'un message MultiplexEntrySend destiné à activer ces entrées. La désactivation d'entrées qui ne sont plus nécessaires à l'émetteur peut augmenter la probabilité de détection d'erreurs dans le champ du code de multiplexage H.223.

NOTE – Au cours de la mise à jour de certaines entrées de tableau de multiplexage, on peut continuer à utiliser d'autres entrées (actives). En outre, on peut supprimer une entrée de tableau de multiplexage du message MultiplexEntry envoyé pour modifier d'autres entrées de tableau de multiplexage.

Au début de la communication, sauf indication contraire dans une recommandation appropriée, seule l'entrée 0 du tableau est disponible pour les transmissions et les entrées 1 à 15 sont désactivées.

Le message RequestMultiplexEntry peut être utilisé à tout moment pour obtenir la retransmission d'entrées de tableau de multiplexage spécifiques à partir du terminal distant, par exemple par suite d'une interruption ou pour toute autre cause d'incertitude.

Le texte suivant donne une vue d'ensemble du fonctionnement du protocole. En cas de divergence entre la présente spécification et la spécification formelle, c'est celle-ci qui a priorité.

8.7.1.1 Aperçu général du protocole – Entité MTSE sortante

Une procédure de demande d'envoi d'entrée de table de multiplexage est lancée lorsque la primitive de demande TRANSFER est émise par l'utilisateur situé dans l'entité MTSE sortante. Un message MultiplexEntrySend est envoyé à l'entité MTSE homologue entrante et le temporisateur T104 est lancé. Si un message MultiplexEntrySendAck est reçu en réponse au message MultiplexEntrySend, le temporisateur T104 est arrêté et l'utilisateur est informé, par la primitive de confirmation TRANSFER, que la demande d'envoi d'entrée de table de multiplexage a été suivie d'effet. Si toutefois un message MultiplexEntrySendReject est reçu en réponse au message MultiplexEntrySend, le temporisateur T104 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 de table de multiplexage.

Si le temporisateur T104 expire, l'utilisateur de l'entité MTSE sortante en est informé par la primitive d'indication REJECT et un message MultiplexEntrySendRelease est envoyé.

Seuls sont acceptés les messages MultiplexEntrySendAck et MultiplexEntrySendReject qui sont des réponses au plus récent message MultiplexEntrySend. Les réponses à des messages MultiplexEntrySend antérieurs sont ignorées.

Une nouvelle procédure de demande d'envoi d'entrée de table de multiplexage peut être lancée par l'utilisateur de l'entité MTSE sortante au moyen de la primitive de demande TRANSFER, avant la réception d'un message MultiplexEntrySendAck ou MultiplexEntrySendReject.

8.7.1.2 Aperçu général du protocole – Entité MTSE entrante

Lorsqu'un message MultiplexEntrySend est reçu à l'entité MTSE entrante, l'utilisateur est informé, par la primitive d'indication TRANSFER, de la demande d'envoi d'entrée de table de multiplexage. L'utilisateur de l'entité MTSE entrante signale son acceptation de l'entrée de table de multiplexage en émettant la primitive de réponse TRANSFER et un message MultiplexEntrySendAck est envoyé à l'entité MTSE homologue sortante. L'utilisateur de l'entité MTSE entrante signale, par l'envoi de la primitive de demande REJECT, le rejet de l'entrée de table de multiplexage et un message MultiplexEntrySendReject est envoyé à l'entité MTSE homologue sortante.

Un nouveau message MultiplexEntrySend peut être reçu avant que l'utilisateur de l'entité MTSE entrante ait répondu à un message MultiplexEntrySend antérieur. 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 au message de nouvelle entrée de table de multiplexage.

Si un message MultiplexEntrySendRelease est reçu avant que l'utilisateur de l'entité MTSE entrante ait répondu à un message MultiplexEntrySend antérieur, cet utilisateur de l'entité MTSE entrante en est informé par la primitive d'indication REJECT et l'entrée précédente de la table de multiplexage est ignorée.

Remplacée par une version plus récente

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 absence de paramètres.
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 transférer les entrées de table de multiplexage;
- les primitives REJECT sont utilisées pour refuser des entrées de table de multiplexage et pour mettre fin au transfert de telles entrées.

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 de table de multiplexage. Ce paramètre est appliqué sur le champ «MultiplexEntryDescriptor» du message «MultiplexEntrySend» et transporté en transparence depuis l'utilisateur de l'entité MTSE sortante jusqu'à l'utilisateur de l'entité MTSE entrante. Plusieurs paramètres MUX-DESCRIPTOR peuvent être 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;
- le paramètre CAUSE indique le motif de rejet d'une entrée de table de multiplexage. Il 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 séparément pour chaque entité MTSE sortante et pour chaque entité MTSE entrante.

Les états pour une entité MTSE sortante sont:

Etat 0: IDLE

Il n'y a aucun transfert entre entités MTSE en cours. L'entrée multiplex peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'utilisateur de l'entité MTSE a demandé le transfert d'une entrée de table de multiplexage et une réponse est attendue de l'entité MTSE homologue. L'entrée de table de multiplexage ne doit pas être utilisée par l'émetteur.

Remplacée par une version plus récente

Les états pour une entité MTSE entrante sont:

Etat 0: IDLE

Il n'y a aucun transfert entre entités MTSE en cours. L'entrée multiplex peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'entité MTSE homologue a transféré une entrée de table de multiplexage et une réponse est attendue de l'utilisateur de l'entité MTSE. Cette entrée de table de multiplexage ne peut pas être en cours d'utilisation par l'émetteur.

8.7.2.5 Diagramme de transition d'états

Ce 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.

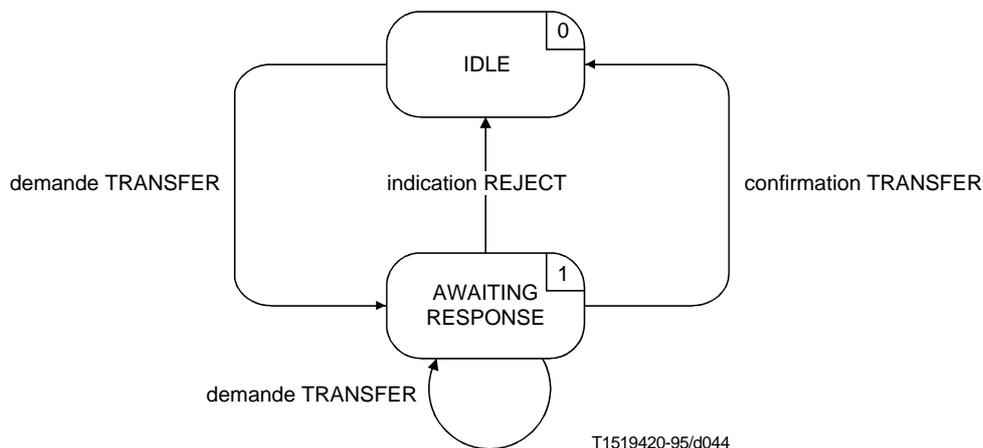


FIGURE 25/H.245

Diagramme de transition d'états pour une séquence de primitives de l'entité MTSE sortante

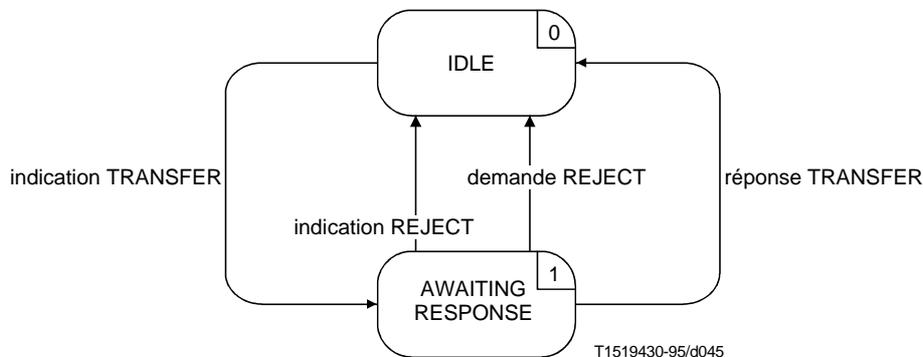


FIGURE 26/H.245

Diagramme de transition d'états pour une séquence de primitives de l'entité MTSE entrante

Remplacée par une version plus récente

8.7.3 Communications 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 messages de l'entité MTSE

Fonction	Message	Sens	Champ
Transfer	MultiplexEntrySend	S → E	sequenceNumber multiplexTableEntryNumber MultiplexElement
	MultiplexEntrySendAck	S ← E	sequenceNumber multiplexTableEntryNumber
Reject	MultiplexEntrySendReject	S ← E	sequenceNumber multiplexTableEntryNumber rejectionDescriptions.Cause
Reset	MultiplexEntrySendRelease	S → E	MultiplexTableEntryNumber
S	Sortante		
E	Entrante		

8.7.3.2 Variables d'état de l'entité MTSE

Les variables d'état suivantes sont définies à l'entité MTSE sortante:

out_ENUM

Cette variable d'état établit une distinction entre plusieurs entités MTSE sortantes. Elle est initialisée en même temps que l'entité MTSE sortante. La valeur de la variable out_ENUM est utilisée pour fixer le champ multiplexTableEntryNumber des messages envoyés par une entité MTSE sortante. Pour les messages d'entité MTSE reçus dans une entité MTSE sortante, la valeur du champ multiplexTableEntryNumber contenue dans ces messages est identique à la valeur de la variable out_ENUM:

out_SQ

Cette variable d'état est utilisée pour indiquer le message MultiplexEntrySend envoyé le plus récemment. Elle est augmentée de 1 et appliquée sur le champ sequenceNumber du message MultiplexEntrySend avant la transmission de ce message. L'opération arithmétique effectuée sur la variable out_SQ est modulo 256.

Les variables d'état suivantes sont définies à l'entité MTSE entrante:

in_ENUM

Cette variable d'état établit une distinction entre plusieurs entités MTSE entrantes. Elle est initialisée en même temps que l'entité MTSE entrante. La valeur de la variable in_ENUM est utilisée pour fixer le champ multiplexTableEntryNumber des messages envoyés par une entité MTSE entrante. Pour les messages d'entité MTSE reçus dans une entité MTSE entrante, la valeur du champ multiplexTableEntryNumber contenue dans ces messages est identique à la valeur de la variable in_ENUM:

in_SQ

Cette variable d'état est utilisée pour mémoriser la valeur du champ sequenceNumber du message MultiplexEntrySend reçu le plus récemment. Les messages MultiplexEntrySendAck et MultiplexEntrySendReject ont leur champ sequenceNumber mis à la valeur de la variable in_SQ, avant d'être envoyés à l'entité MTSE homologue.

Remplacée par une version plus récente

8.7.3.3 Temporisateurs MTSE

Le temporisateur suivant est spécifié pour l'entité MTSE sortante:

T107

Ce temporisateur est utilisé à l'état AWAITING RESPONSE. Il spécifie le temps maximal pendant lequel aucun message MultiplexEntrySendAck ou MultiplexEntrySendReject ne peut être reçu.

8.7.4 Procédures de l'entité MTSE

8.7.4.1 Introduction

La Figure 27 récapitule les primitives MTSE et leurs paramètres, de même que les messages, pour chacune des entités MTSE sortante et entrante.

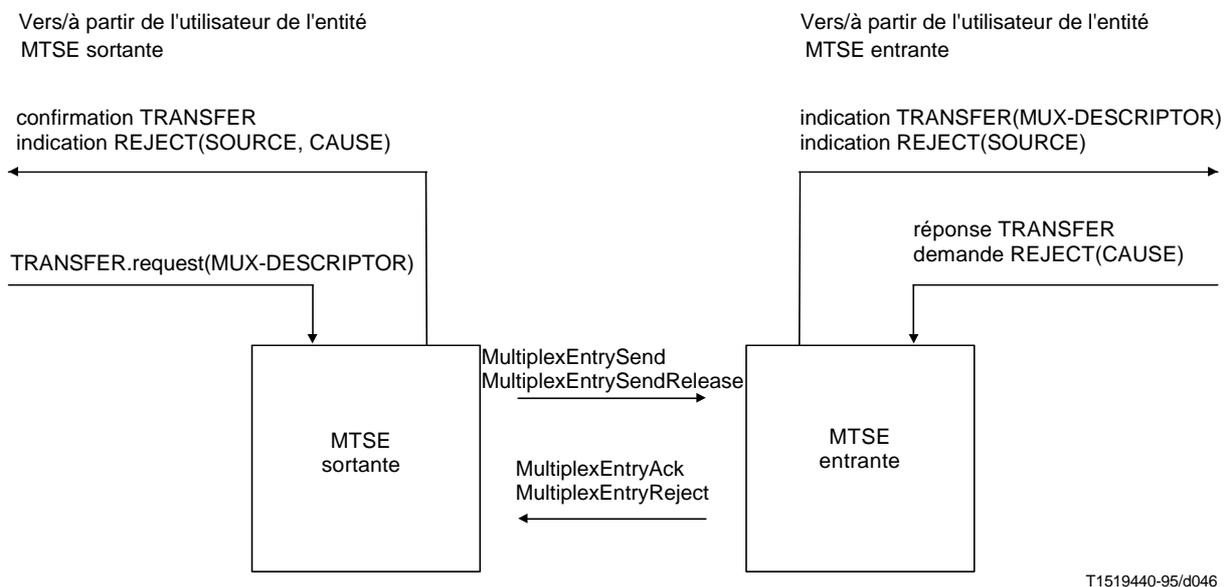


FIGURE 27/H.245

Primitives et messages dans l'entité MultiplexTableSignalling

8.7.4.2 Valeurs par défaut des paramètres des 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
indication TRANSFER	MUX-DESCRIPTOR	MultiplexEntrySend.MultiplexEntryDescriptor
indication REJECT	SOURCE	USER
	CAUSE	null

Remplacée par une version plus récente

8.7.4.3 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 44.

TABLEAU 44/H.245

Valeurs par défaut des champs de message

Message	Champ	Valeur par défaut (Note)
MultiplexEntrySend	sequenceNumber	out_SQ
	multiplexTableEntryNumber	out_ENUM
	MultiplexElement	demande TRANSFER (MUX-DESCRIPTOR)
MultiplexEntrySendAck	sequenceNumber	in_SQ
	multiplexTableEntryNumber	in_ENUM
MultiplexEntrySendReject	sequenceNumber	in_SQ
	multiplexTableEntryNumber	in_ENUM
	cause	demande 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 SDL

Les procédures des entités MTSE sortantes et entrantes sont exprimées en SDL, respectivement dans les Figures 28 et 29.

8.8 Procédures de demande d'entrée de multiplexage

8.8.1 Introduction

Ces procédures sont utilisées par un terminal pour demander la retransmission d'un ou de plusieurs descripteurs MultiplexEntryDescriptors. Ces procédures sont désignées ici sous le nom d'entités de signalisation de demande d'entrée de multiplexage (RMESE). Elles 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 par les messages appropriés définis au paragraphe 6. Il y a une entité RMESE sortante et une entité RMESE entrante. Il y a une instance de l'entité RMESE pour chaque entrée de tableau de multiplexage.

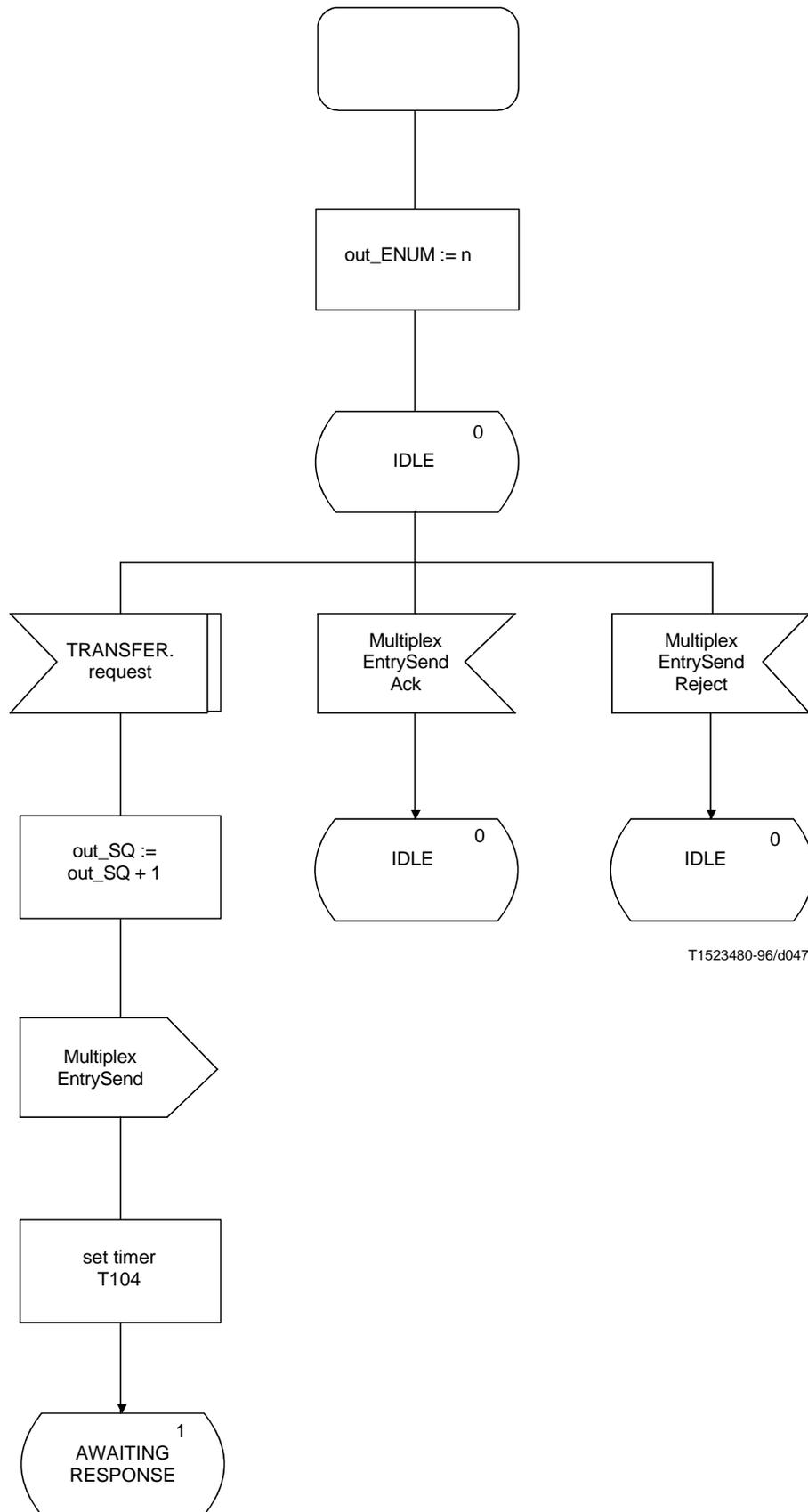
Un terminal qui répond à une telle action positivement, c'est-à-dire en émettant la primitive de réponse SEND, doit lancer les procédures de table de multiplexage pour envoyer l'entrée de table de multiplexage le plus tôt possible.

8.8.1.1 Aperçu général du protocole – Entité RMESE sortante

L'émission de la primitive de demande SEND par l'utilisateur dans l'entité RMESE sortante lance la procédure de demande d'entrée de multiplexage. Un message RequestMultiplexEntry est envoyé à l'entité RMESE entrante homologue et le temporisateur T107 est mis en marche. A la réception d'un message RequestMultiplexEntryAck en réponse au message RequestMultiplexEntry, le temporisateur T107 s'arrête et l'utilisateur est informé par la primitive de confirmation SEND que la procédure de demande d'entrée de multiplexage a été menée à bien. Toutefois, à la réception d'un message RequestMultiplexEntryReject en réponse au message RequestMultiplexEntry, le temporisateur T107 s'arrête 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.

A l'expiration du temporisateur T107, l'utilisateur de l'entité RMESE sortante en est informé par la primitive d'indication REJECT et un message RequestMultiplexEntryRelease est envoyé.

Remplacée par une version plus récente



T1523480-96/d047

FIGURE 28 i)/H.245

Description SDL de l'entité MTSE sortante

Remplacée par une version plus récente

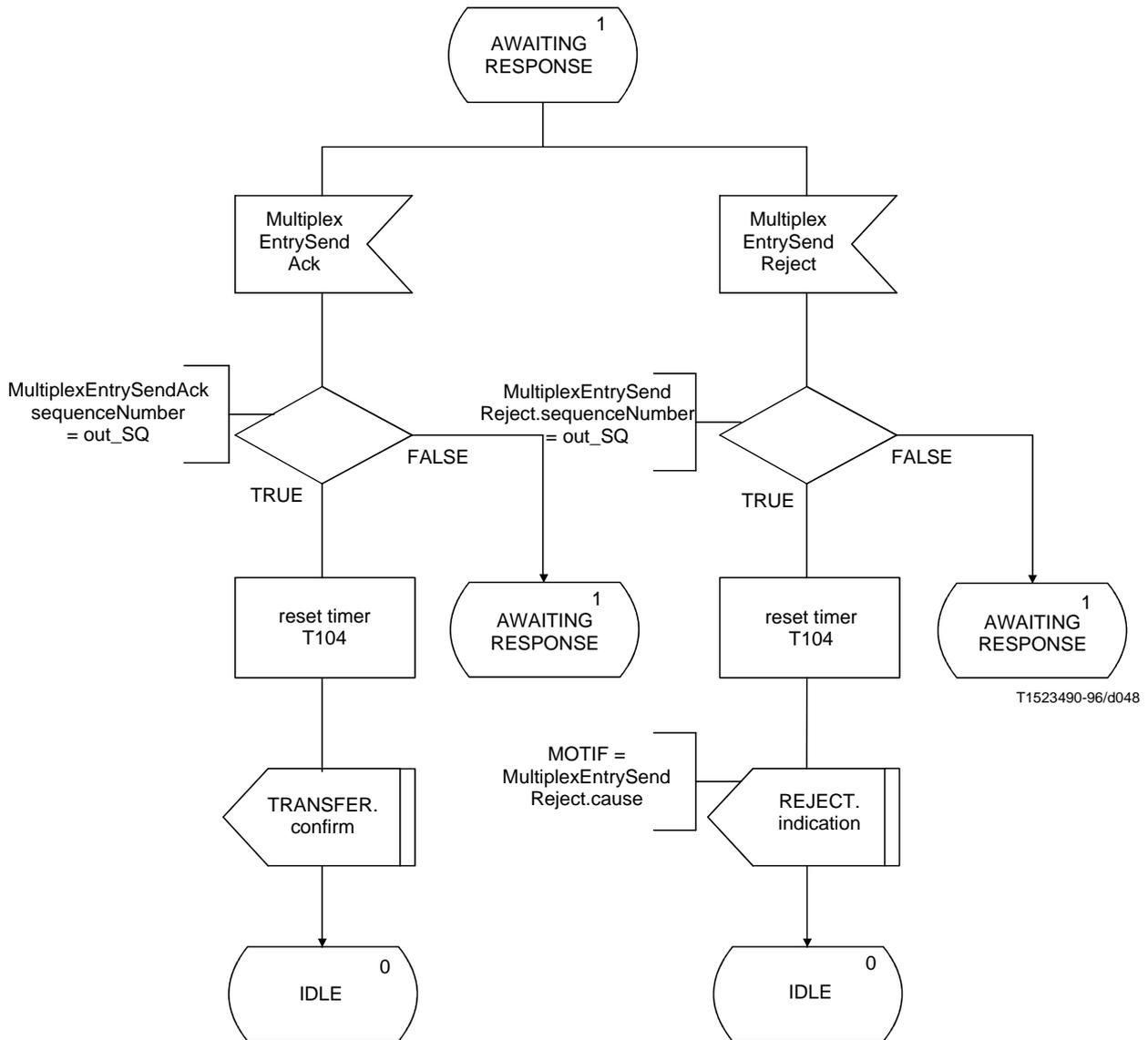
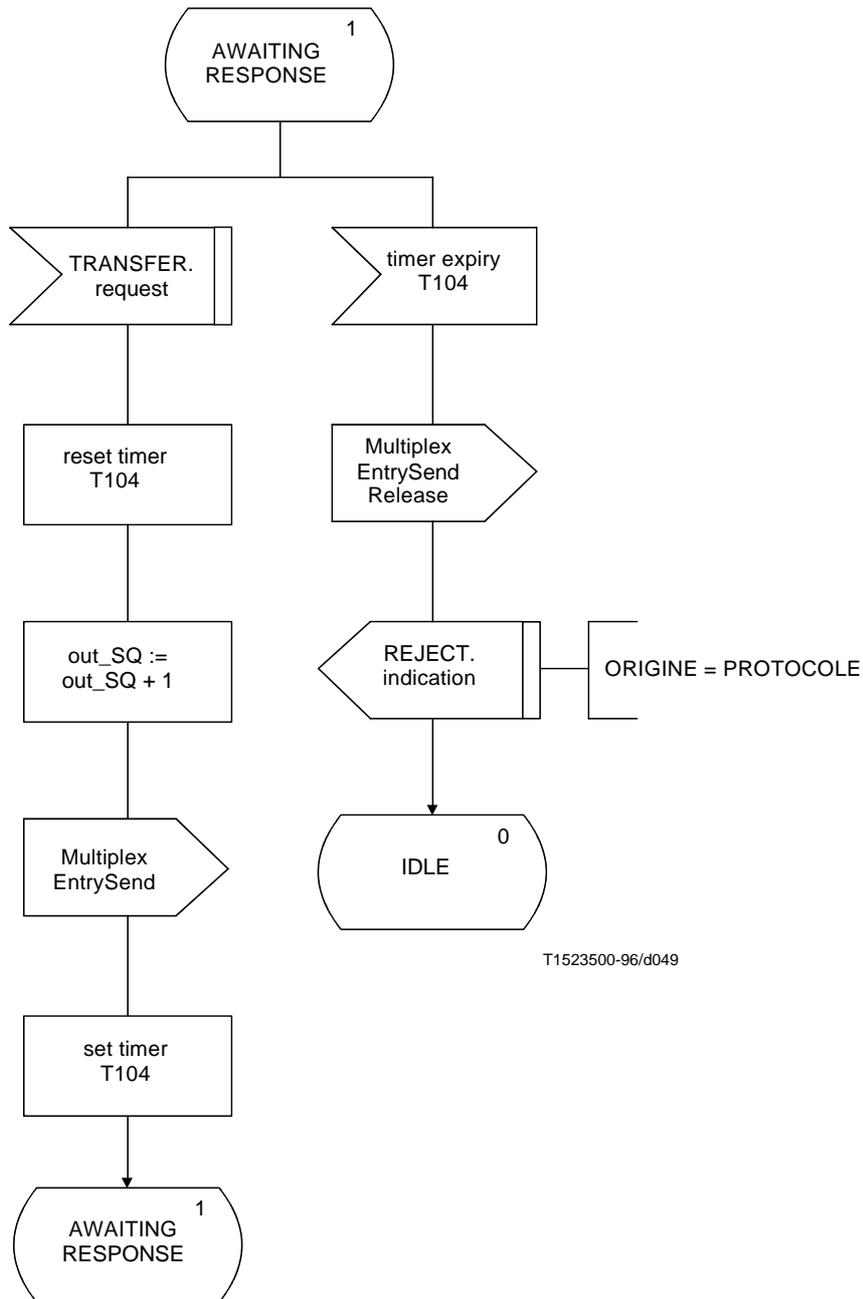


FIGURE 28 ii)/H.245

Description SDL de l'entité MTSE sortante

Remplacée par une version plus récente



T1523500-96/d049

FIGURE 28 iii)/H.245

Description SDL de l'entité MTSE sortante

Remplacée par une version plus récente

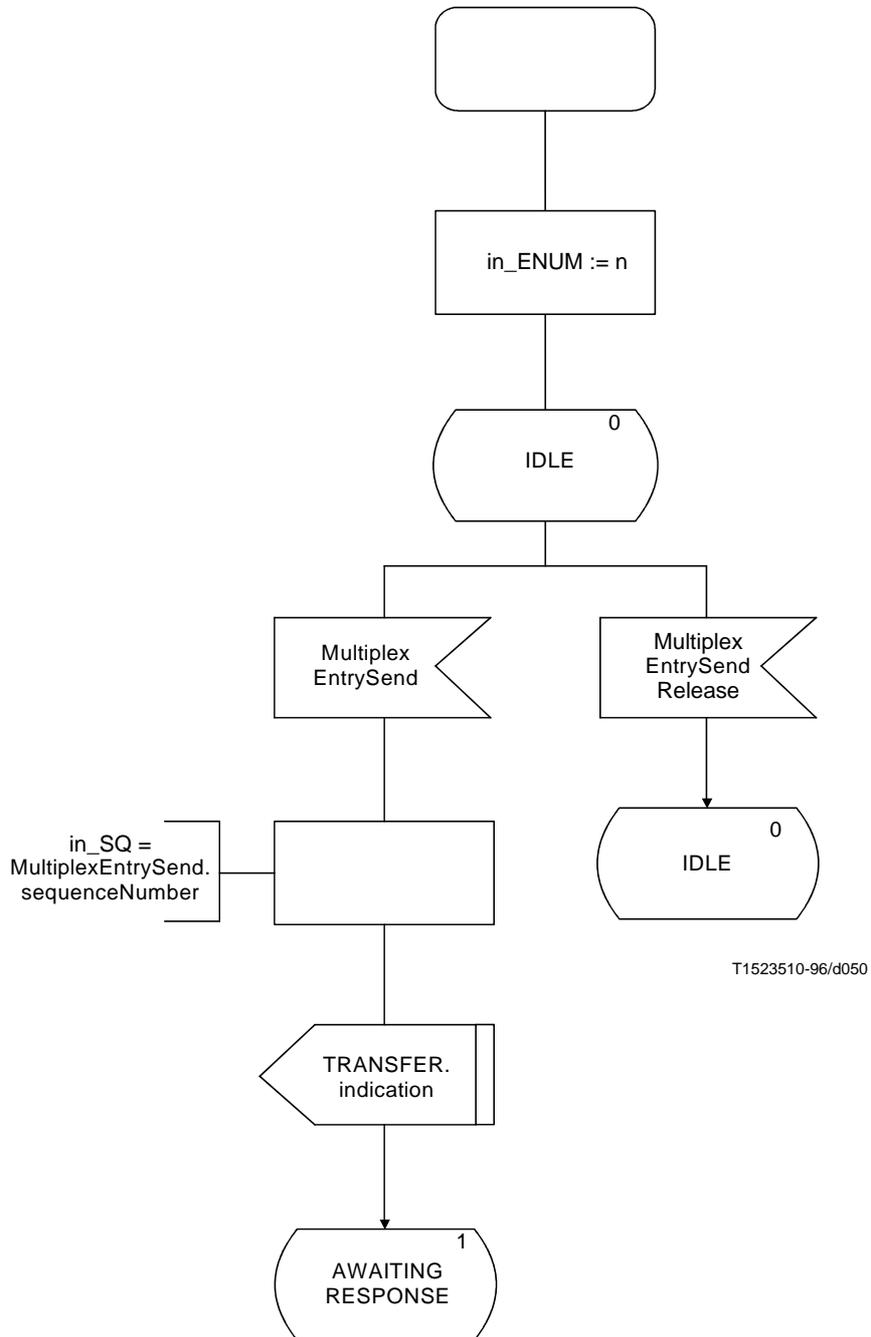


FIGURE 29 i)/H.245

Description SDL de l'entité MTSE entrante

Remplacée par une version plus récente

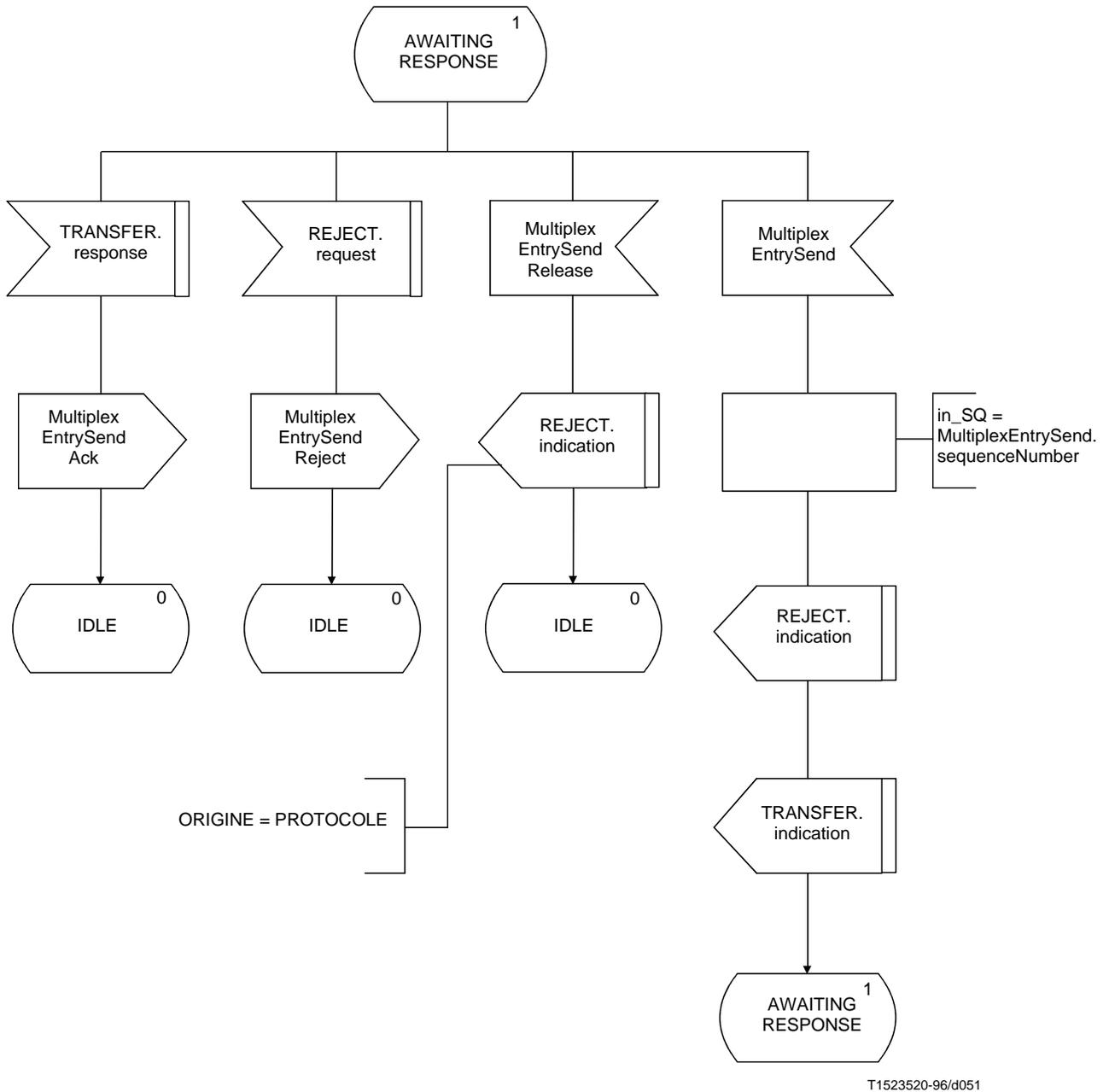


FIGURE 29 ii)/H.245
Description SDL de l'entité MTSE entrante

Remplacée par une version plus récente

8.8.1.2 Aperçu général du protocole – Entité RMESE entrante

A la réception d'un message RequestMultiplexEntry 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 entrante signale l'acceptation de la demande d'entrée de multiplexage en émettant la primitive de réponse SEND et un message RequestMultiplexEntryAck est envoyé à l'entité RMESE sortante homologue. L'utilisateur de l'entité RMESE entrante signale le rejet de la demande d'entrée de multiplexage en émettant la primitive de demande REJECT et un message RequestMultiplexEntryReject est envoyé à 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

Les communications entre l'entité RMESE et l'utilisateur de l'entité RMESE sont établies à l'aide des 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 absence de paramètres.
NOTE 2 – «Non défini» signifie que cette primitive n'est pas définie.

8.8.2.2 Définition des primitives

Ces primitives sont définies comme suit:

- les primitives SEND sont utilisées pour demander la transmission d'une entrée de multiplexage;
- les primitives REJECT sont utilisées pour rejeter la demande de transmission d'une entrée de multiplexage.

8.8.2.3 Définition des paramètres

Les paramètres de primitives indiqués au Tableau 45 sont définis comme suit:

- 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 découler d'une fin de temporisation.
- le paramètre CAUSE indique le motif du refus d'envoyer une entrée de tableau de multiplexage. Le paramètre CAUSE est absent 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 d'une entité RMESE sortante sont les suivants:

Etat 0: IDLE

Remplacée par une version plus récente

L'entité RMESE est au repos.

Etat 1: AWAITING RESPONSE

L'entité RMESE attend une réponse de l'entité RMESE distante.

Les états d'une entité RMESE entrante sont les suivants:

Etat 0: IDLE

L'entité RMESE est au repos.

Etat 1: AWAITING RESPONSE

L'entité RMESE attend une réponse de l'utilisateur de l'entité RMESE.

8.8.2.5 Diagramme de transition 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 entités RMESE sortante et entrante, comme l'indiquent respectivement les Figures 30 et 31.

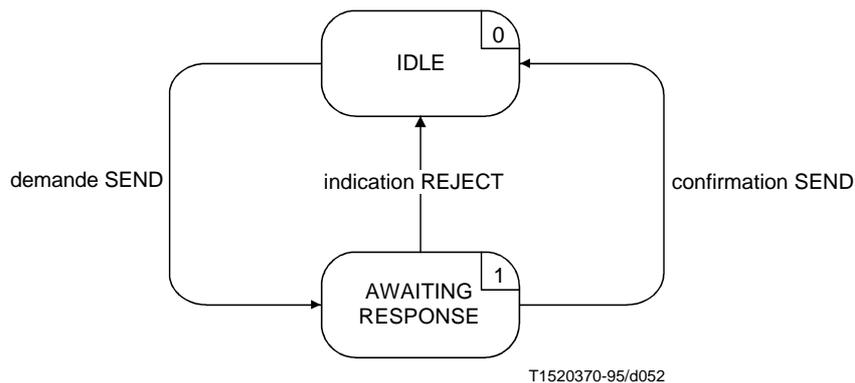


FIGURE 30/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité RMESE sortante

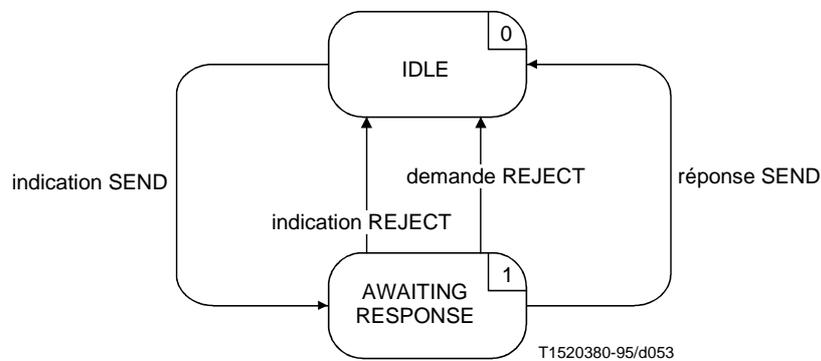


FIGURE 31/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité RMESE entrante

Remplacée par une version plus récente

8.8.3 Communication entre entités RMESE homologues

8.8.3.1 Messages

Le Tableau 46 indique les messages et les champs de l'entité RMESE définis au paragraphe 6 et se rapportant 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
	RequestMultiplexEntryReject	S ← E	multiplexTableEntryNumber rejectionDescriptions.cause
Reset	RequestMultiplexEntryRelease	S → E	–
S	Sortante		
E	Entrante		

8.8.3.2 Variables d'état de l'entité RMESE

Les variables d'état suivantes sont définies pour l'entité RMESE sortante:

out_ENUM

Cette variable d'état permet de distinguer les entités RMESE sortantes. Elle est initialisée lors du lancement de l'entité RMESE sortante. La valeur de la variable out_ENUM est utilisée pour remplir le champ multiplexTableEntryNumber de messages d'entité RMESE envoyés à partir d'une entité RMESE sortante. Pour les messages RMESE reçus par une entité RMESE sortante, la valeur du champ multiplexTableEntryNumber de ces messages est identique à la valeur de out_ENUM.

La variable d'état suivante est définie pour l'entité RMESE entrante:

in_ENUM

Cette variable d'état permet de distinguer les entités RMESE entrantes. Elle est initialisée lors du lancement 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. Pour les messages RMESE reçus par une entité RMESE entrante, la valeur du champ multiplexTableEntryNumber de ces messages est identique à la valeur de in_ENUM.

8.8.3.3 Temporiseurs RMESE

Le temporisateur suivant est spécifié pour l'entité RMESE sortante:

T107

Ce temporisateur est utilisé à 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 de l'entité RMESE

La Figure 32 récapitule les primitives de l'entité RMESE et leurs paramètres, de même que les messages, pour chacune des entités RMESE sortante et entrante.

Remplacée par une version plus récente

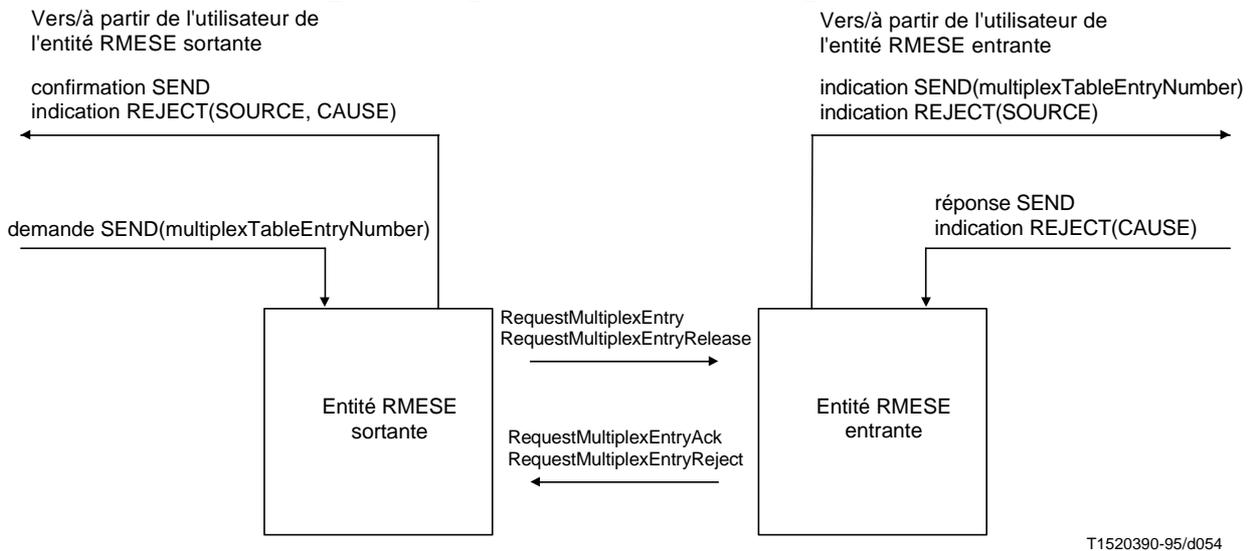


FIGURE 32/H.245

Primitives et messages dans l'entité de signalisation de demande d'entrée de multiplexage

8.8.4.1 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs par défaut indiquées dans le Tableau 47, lorsque leurs valeurs ne sont pas expressément mentionnées dans les diagrammes SDL.

TABLEAU 47/h.245

Valeurs par défaut des paramètres de primitives

Primitive	Paramètre	Valeur par défaut
indication REJECT	SOURCE	USER
	CAUSE	null

8.8.4.2 Valeurs par défaut des champs de messages

Les champs de messages prennent les valeurs par défaut indiquées dans le Tableau 48, lorsque leurs valeurs ne sont pas expressément mentionnées dans les diagrammes SDL.

TABLEAU 48/h.245

Valeurs par défaut des champs de messages

Message	Champ	Valeur par défaut
RequestMultiplexEntry	multiplexTableEntryNumber	out_ENUM
RequestMultiplexEntryAck	multiplexTableEntryNumber	in_ENUM
RequestMultiplexEntryReject	multiplexTableEntryNumber	in_ENUM
	cause	demande REJECT (CAUSE)
RequestMultiplexEntryRelease	multiplexTableEntryNumber	out_ENUM

Remplacée par une version plus récente

8.8.4.3 Description SDL

Les procédures de l'entité RMESE sortante et de l'entité RMESE entrante sont exprimées en langage SDL sur les Figures 33 et 34 respectivement.

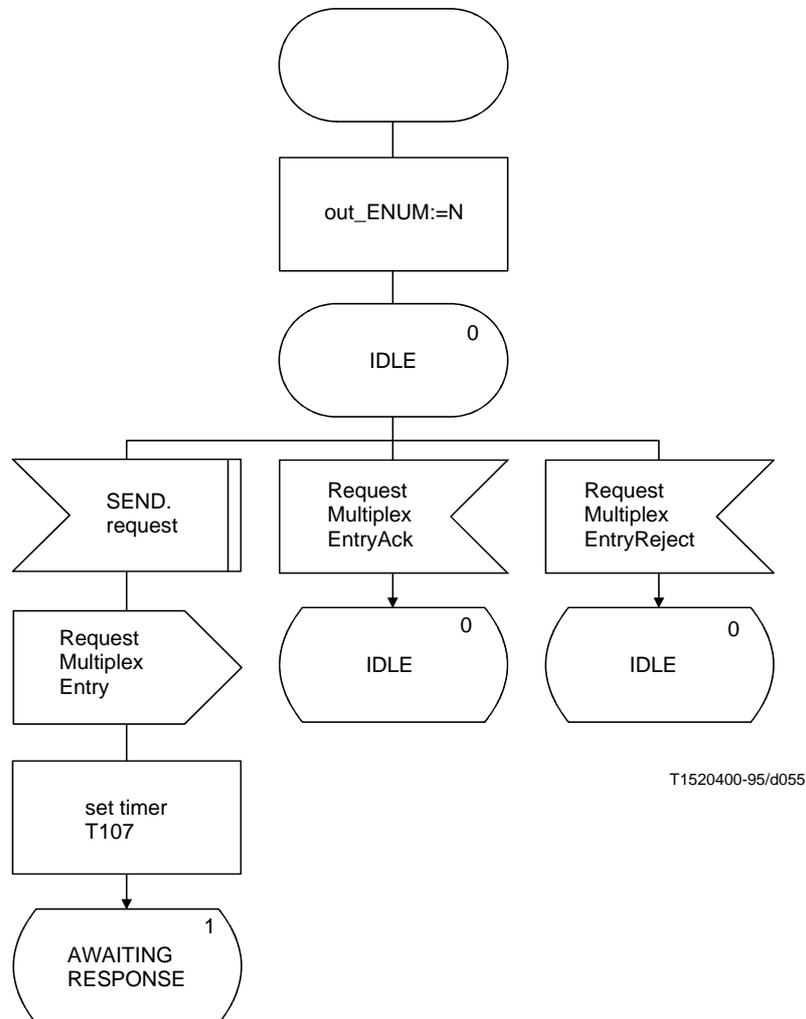


FIGURE 33 i)/H.245

Description SDL de l'entité RMESE sortante

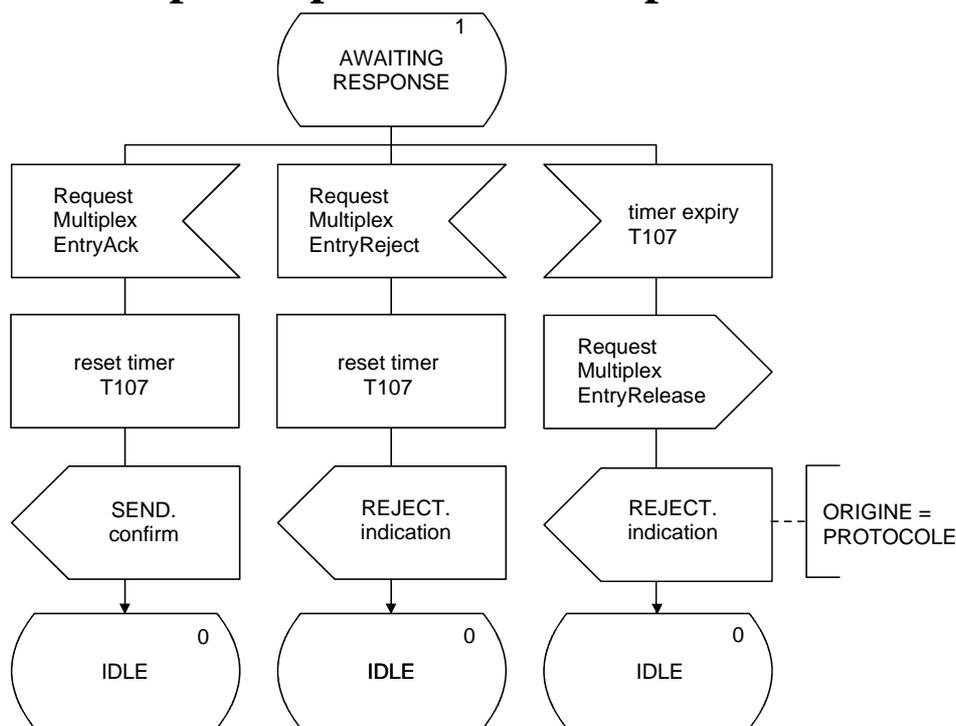
8.9 Procédures de mode demande

8.9.1 Introduction

Les procédures décrites dans les paragraphes suivants permettent à un terminal de demander à un terminal distant d'utiliser un mode particulier de fonctionnement dans son sens d'émission. Il est fait référence ici à ces procédures sous le nom de l'entité de signalisation de mode demande (MRSE). 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 donne une telle réponse favorable, c'est-à-dire qui envoie la primitive de réponse TRANSFER, doit lancer les procédures de signalisation de voie logique afin d'établir le mode de transmission approprié dès que possible.

Remplacée par une version plus récente



T1520410-95/d056

FIGURE 33 ii)/H.245

Diagramme SDL de l'entité RMESE sortante

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 Mode Request. Un terminal dont les capacités actuellement disponibles contiennent une ou plusieurs capacités d'émission et qui reçoit une telle demande devrait satisfaire la demande.

Un mode demandé ne doit pas être envoyé 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 un mode demandé, il peut satisfaire cette demande.

Un terminal qui reçoit le message multipointModeCommand doit donner suite à toutes les demandes de mode reçues, jusqu'à ce que la commande soit annulée par la réception d'un message cancelMultipointModeCommand. Une demande de mode peut être envoyée à un terminal dont le message multipointModeCommand a déjà été envoyé.

Le mode demandé peut inclure des canaux qui sont déjà ouverts. Par exemple, dans le cas où un canal pour G.723 serait ouvert et un terminal souhaiterait recevoir un canal G.728 supplémentaire, il enverrait une demande de mode contenant à la fois les canaux G.723 et G.728. L'absence de la demande de canal G.723 indiquerait que G.723 n'est plus souhaité.

NOTE – La description du mode de demande spécifie un mode complet. Si, par exemple, le mode vidéo est en cours de transmission et qu'un terminal reçoit une demande de mode ne comportant pas de spécification pour la vidéo, il s'agit d'une demande d'arrêt de la transmission vidéo.

Si une source alimente plusieurs récepteurs, il 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 d'entité MRSE. En cas de divergence entre la présente spécification et la spécification formelle, c'est celle-ci qui a priorité.

Remplacée par une version plus récente

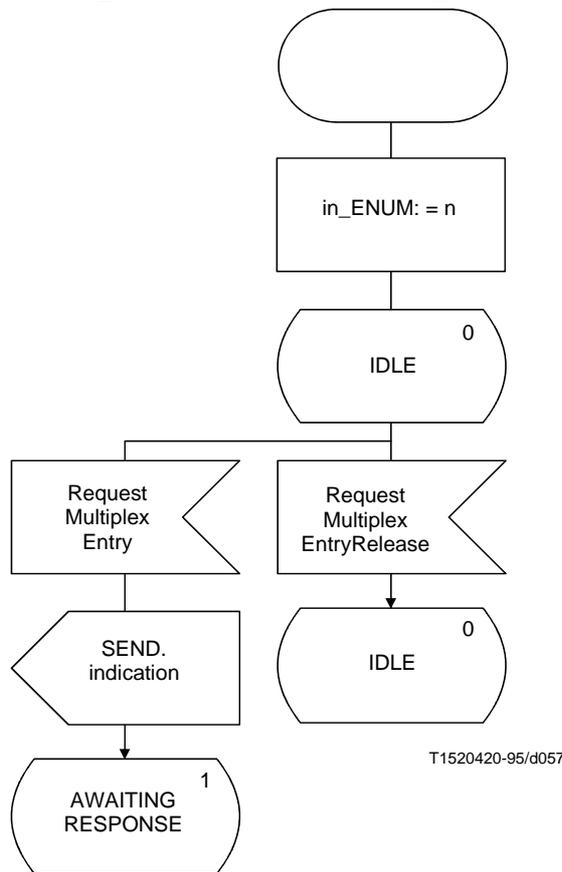


FIGURE 34 i)/H.245

Diagramme SDL de l'entité RMESE entrante

8.9.1.1 Aperçu général du protocole – Entité MRSE sortante

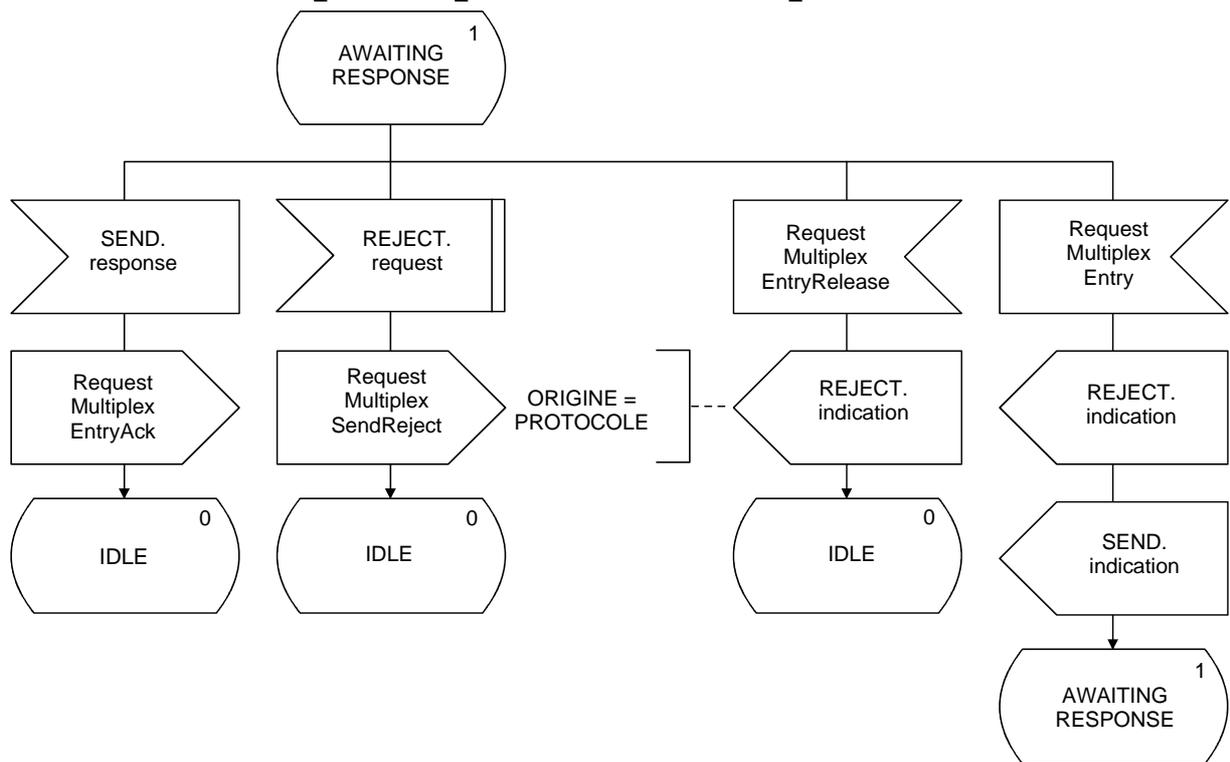
Une procédure de demande de mode est lancée lorsque la primitive de demande TRANSFER est émise par l'utilisateur situé dans l'entité MRSE sortante. Un message RequestMode est envoyé à l'entité MRSE homologue entrante et le temporisateur T109 est lancé. Si un message RequestModeAck est reçu en réponse au message RequestMode, le temporisateur T109 est arrêté et l'utilisateur est informé, par la primitive de confirmation TRANSFER, que la demande de mode a été suivie d'effet. Si toutefois 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 le mode.

Si le temporisateur T109 expire, l'utilisateur de l'entité MRSE sortante en est informé par la primitive d'indication REJECT et un message RequestModeRelease est envoyé.

Seuls sont acceptés les messages RequestModeAck et RequestModeReject qui sont des réponses au plus récent message RequestMode. Les réponses à des messages RequestMode antérieurs sont ignorées.

Une nouvelle procédure de demande de mode peut être lancée par l'utilisateur de l'entité MRSE sortante au moyen de la primitive de demande TRANSFER, avant la réception d'un message RequestModeAck ou RequestModeReject.

Remplacée par une version plus récente



T1520430-95/d058

FIGURE 34 ii)/H.245

Description SDL de l'entité RMESE entrante

8.9.1.2 Aperçu général du protocole – Entité MRSE entrante

Lorsqu'un message RequestMode est reçu à l'entité MRSE entrante, l'utilisateur est informé, par la primitive d'indication TRANSFER, de la demande de mode. L'utilisateur de l'entité MRSE entrante signale son acceptation du mode en émettant la primitive de réponse TRANSFER et un message RequestModeAck est envoyé à l'entité MRSE homologue sortante. L'utilisateur de l'entité MRSE entrante signale, par l'envoi de la primitive de demande REJECT, le rejet du mode et un message RequestModeReject est envoyé à l'entité MRSE homologue sortante.

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 au message de nouvelle demande de mode.

Si un message RequestModeRelease est reçu avant que l'utilisateur de l'entité MRSE entrante ait répondu à un message RequestMode antérieur, cet utilisateur de l'entité MRSE entrante en est informé par la primitive d'indication REJECT et la demande de mode précédente est ignorée.

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.

Remplacée par une version plus récente

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:

- les primitives TRANSFER sont utilisées pour le transfert du mode demandé;
- les primitives REJECT sont utilisées pour refuser un mode demandé.

8.9.2.3 Définition des paramètres

La définition des paramètres de primitives indiqués dans le Tableau 49 est la suivante:

- le paramètre MODE-ELEMENT spécifie un élément de mode. Ce paramètre est inséré dans le champ requestedModes du message RequestMode et est transporté en transparence depuis l'utilisateur de l'entité MRSE sortante jusqu'à l'utilisateur MRSE entrante. Ce paramètre est obligatoire. Plusieurs paramètres MODE-ELEMENT peuvent être associés aux primitives TRANSFER;
- le paramètre MODE-PREF signale à l'utilisateur si la demande du mode préféré en premier sera acceptée ou non. Ce paramètre est inséré dans le champ response du message RequestModeAck et transporté en transparence depuis l'utilisateur de l'entité RMSE entrante jusqu'à l'utilisateur RMSE sortante. Ses deux valeurs possibles sont: «MOST-PREFERRED» (première préférence) et «LESS-PREFERRED» (deuxième préférence);
- 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;
- le paramètre CAUSE indique le motif du refus de fermeture d'une voie logique. 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 pour une entité MRSE entrante sont:

Etat 0: IDLE

L'entité MRSE est à l'état «idle».

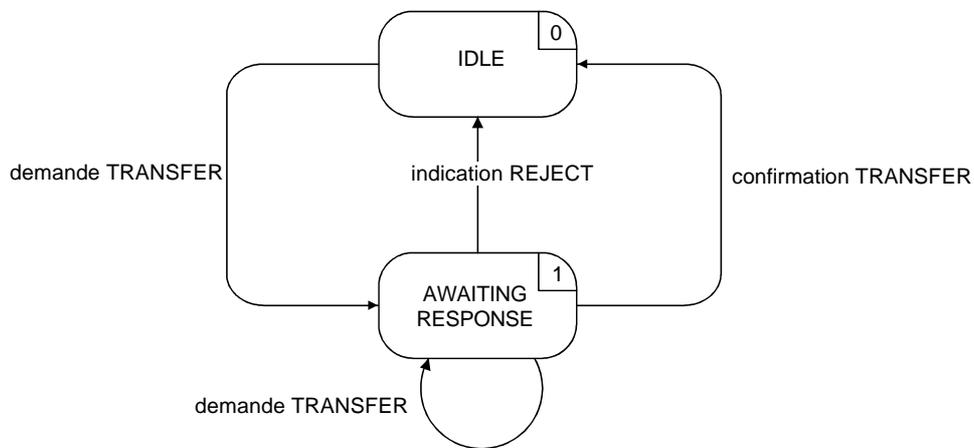
Etat 1: AWAITING RESPONSE

L'entité MRSE attend une réponse de l'utilisateur de l'entité MRSE.

Remplacée par une version plus récente

8.9.2.5 Diagramme de transition d'états

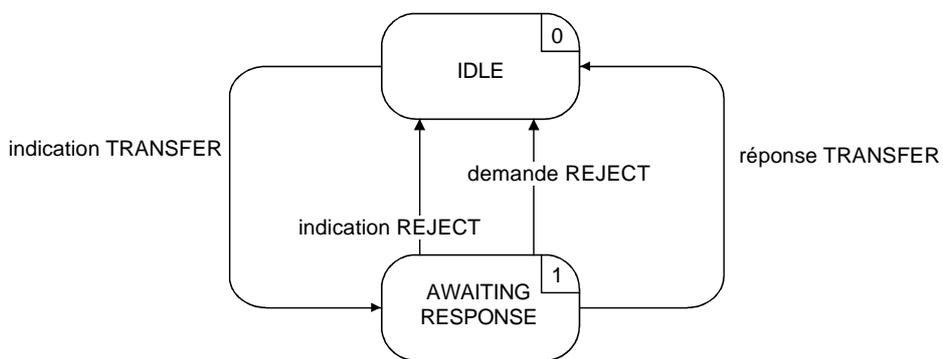
La séquence autorisée de primitives entre l'entité MRSE et l'utilisateur de l'entité MRSE est définie dans ce sous-paragraphe. Les séquences autorisées sont spécifiées séparément pour chacune des entités MRSE sortante et entrante, comme l'indiquent respectivement les Figures 35 et 36.



T1519480-95/d059

FIGURE 35/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité MRSE sortante



T1519490-95/d060

FIGURE 36/H.245

Diagramme de transition d'états pour la séquence de primitives de l'entité MRSE entrante

Remplacée par une version plus récente

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 Sortante
E Entrante

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 appliquée au champ sequenceNumber du message RequestMode avant la transmission de ce 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 Temporiseurs MRSE

Le temporisateur suivant est spécifié pour l'entité MRSE sortante:

T109

Ce temporisateur est utilisé pendant l'état AWAITING RESPONSE. Il spécifie la durée maximale au cours de laquelle aucun message RequestModeACK ou RequestModeReject ne peut être reçu.

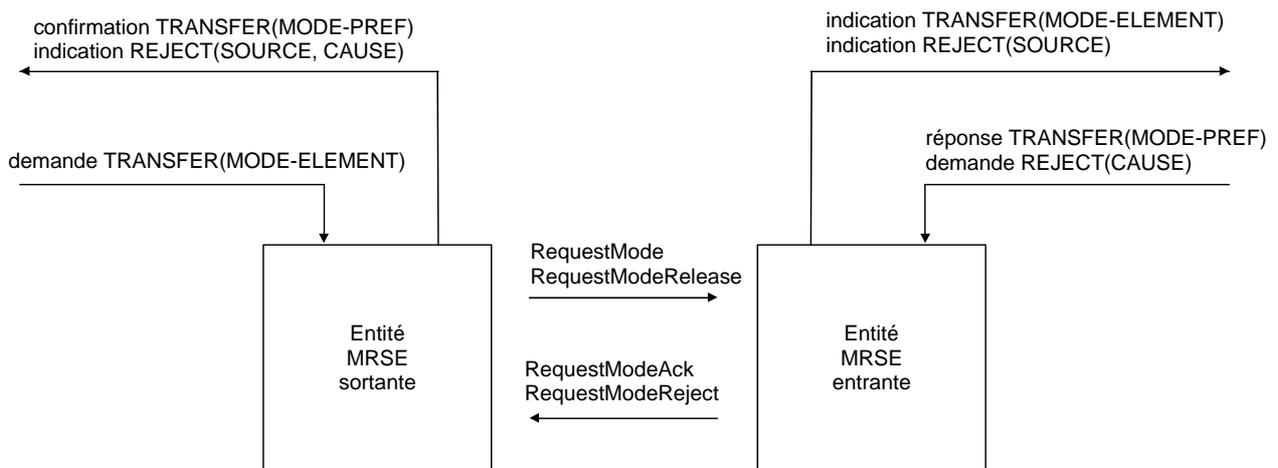
8.9.4 Procédures de l'entité MRSE

La Figure 37 récapitule les primitives de l'entité MRSE et leurs paramètres, de même que les messages, pour chacune des entités MRSE sortante et entrante.

Remplacée par une version plus récente

Vers/à partir de l'utilisateur de MRSE sortante

Vers/à partir de l'utilisateur de MRSE entrante



T1519500-95/d061

FIGURE 37/H.245

Primitives et messages dans l'entité de signalisation du mode demande (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	Valeur par défaut
indication TRANSFER	MODE-ELEMENT	RequestMode.requestedModes
confirmation TRANSFER	MODE-PREF	RequestModeAck.response
indication REJECT	SOURCE CAUSE	USER 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	Valeur par défaut
RequestMode	sequenceNumber	out_SQ
	requestedModes	demande TRANSFER(MODE-ELEMENT)
RequestModeAck	sequenceNumber	in_SQ
	response	réponse TRANSFER(MODE-PREF)
RequestModeReject	sequenceNumber cause	in_SQ demande REJECT(CAUSE)
RequestModeRelease	-	-

Remplacée par une version plus récente

8.9.4.3 Diagrammes SDL

Les procédures des entités MRSE sortantes et entrantes sont exprimées en SDL, respectivement dans les Figures 38 et 39.

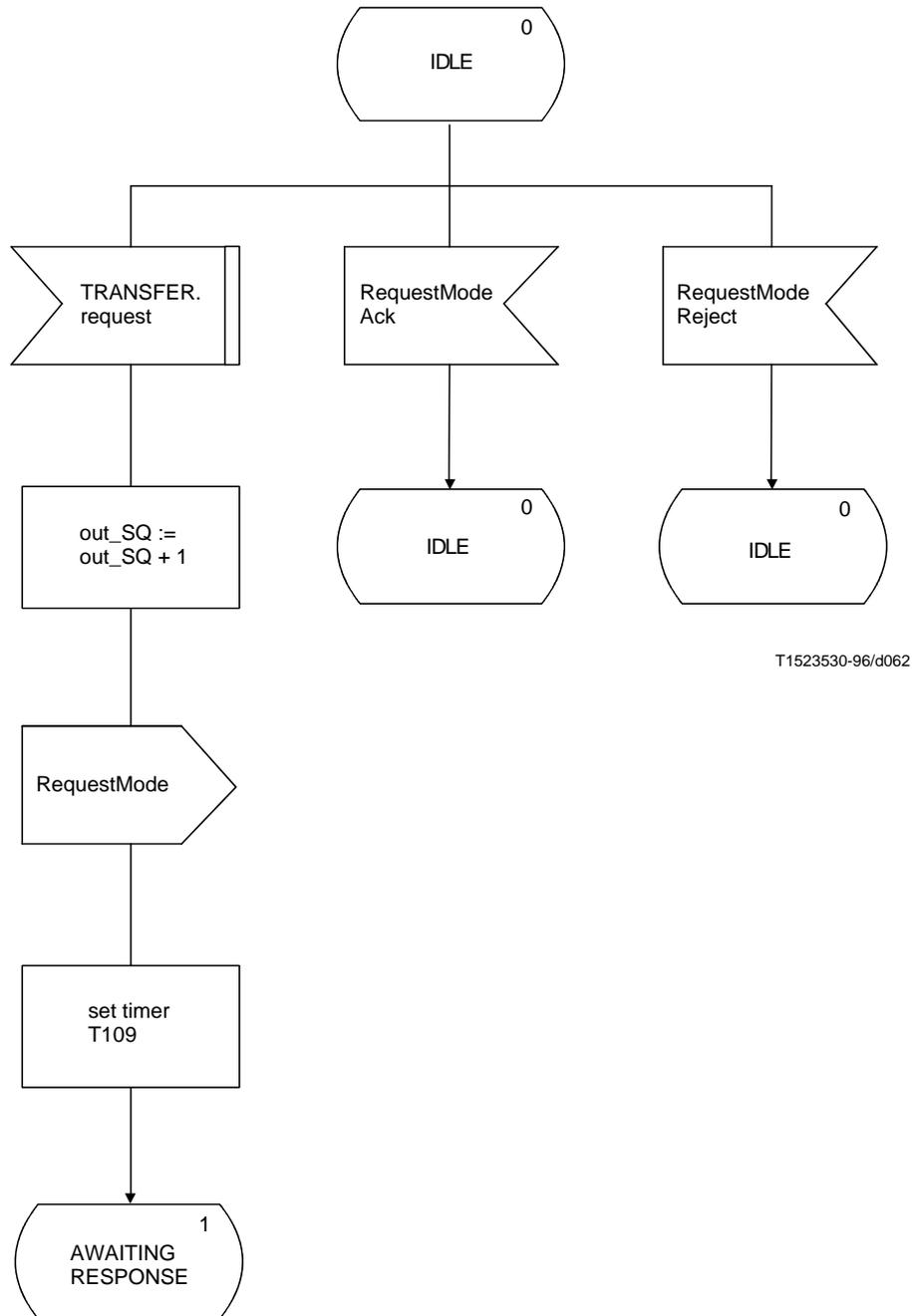


FIGURE 38 i)/H.245

Description SDL de l'entité MRSE sortante

Remplacée par une version plus récente

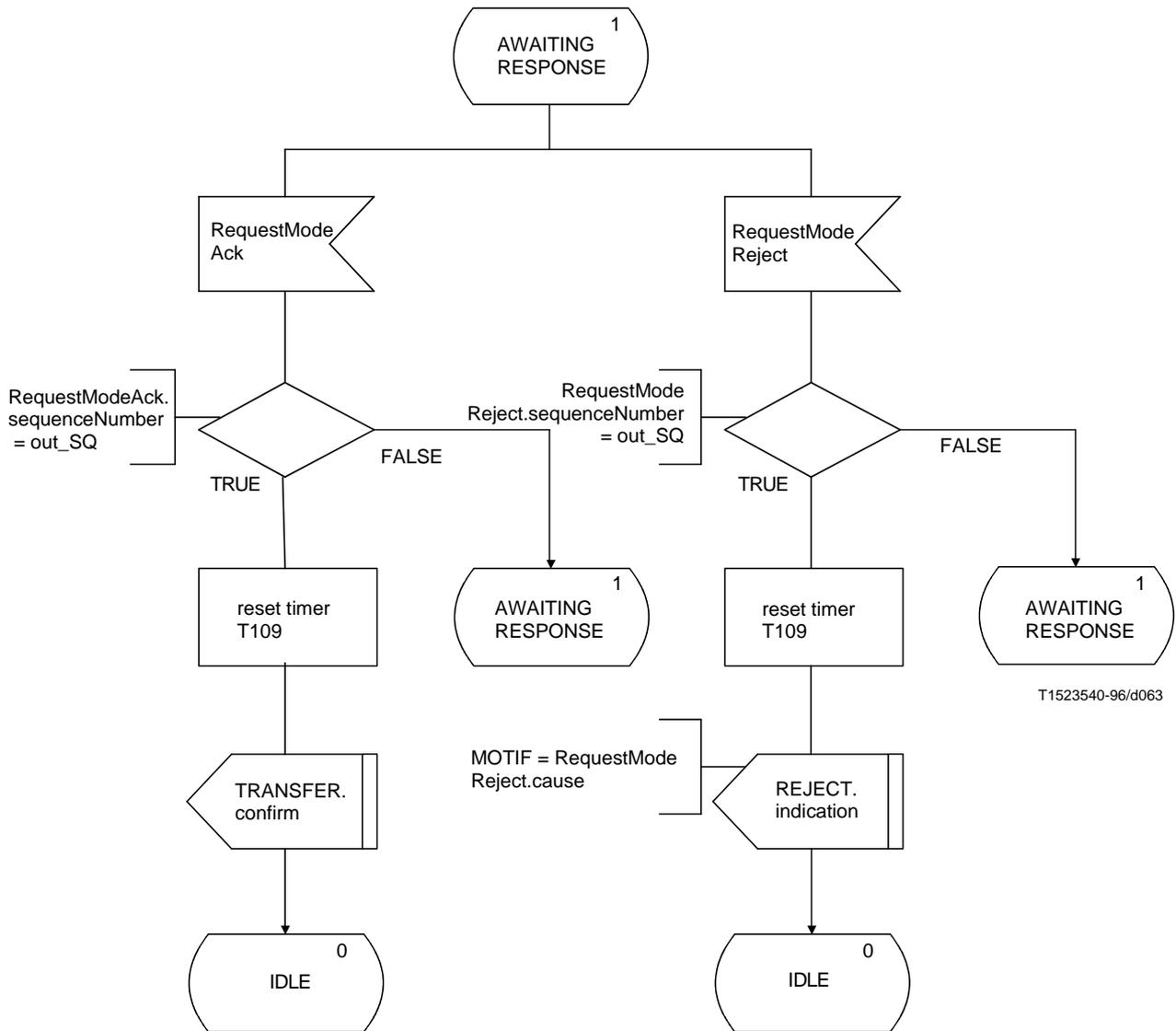


FIGURE 38 ii)/H.245

Description SDL de l'entité MRSE sortante

Remplacée par une version plus récente

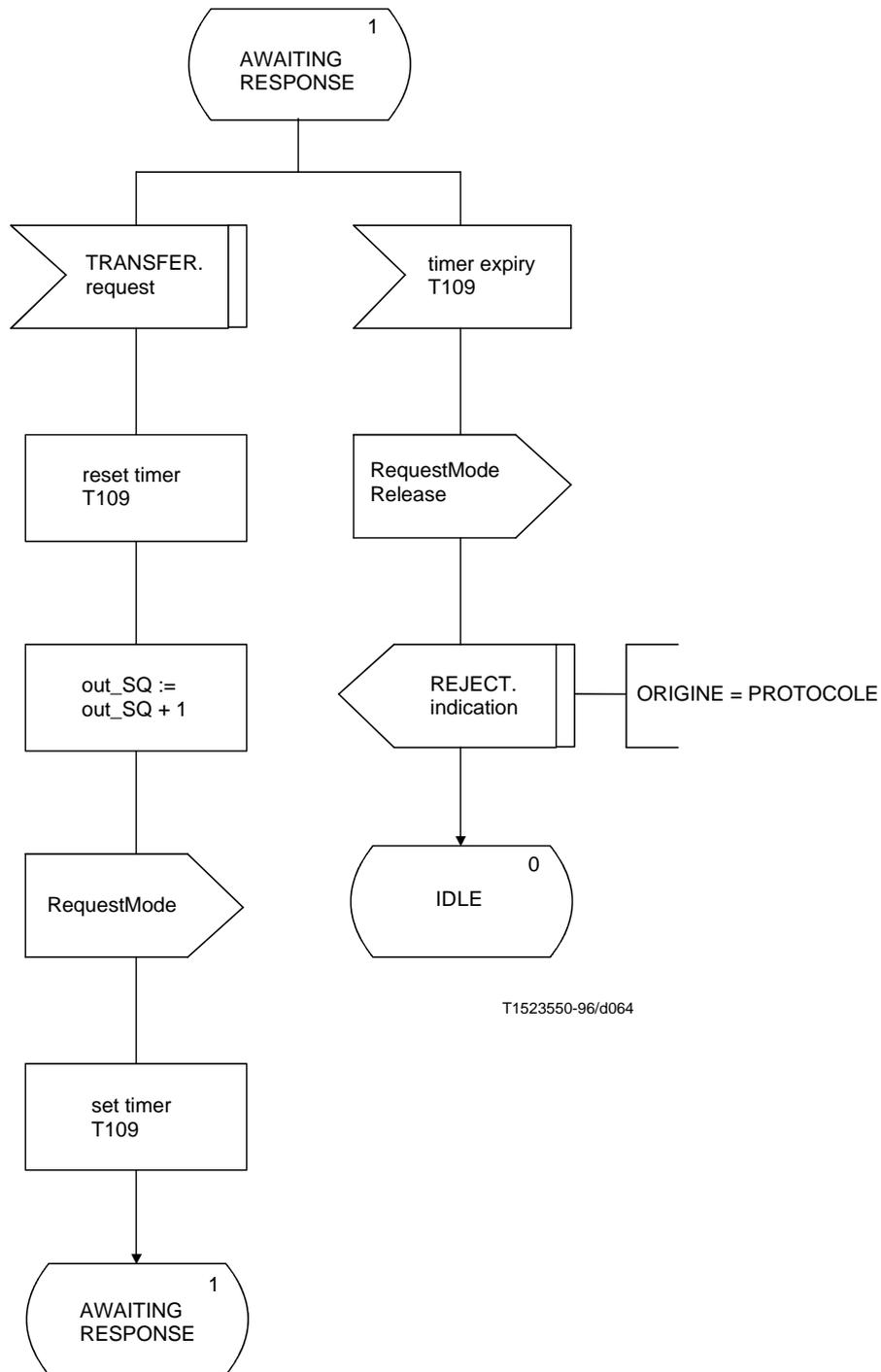


FIGURE 38 iii)/H.245

Description SDL de l'entité MRSE sortante

Remplacée par une version plus récente

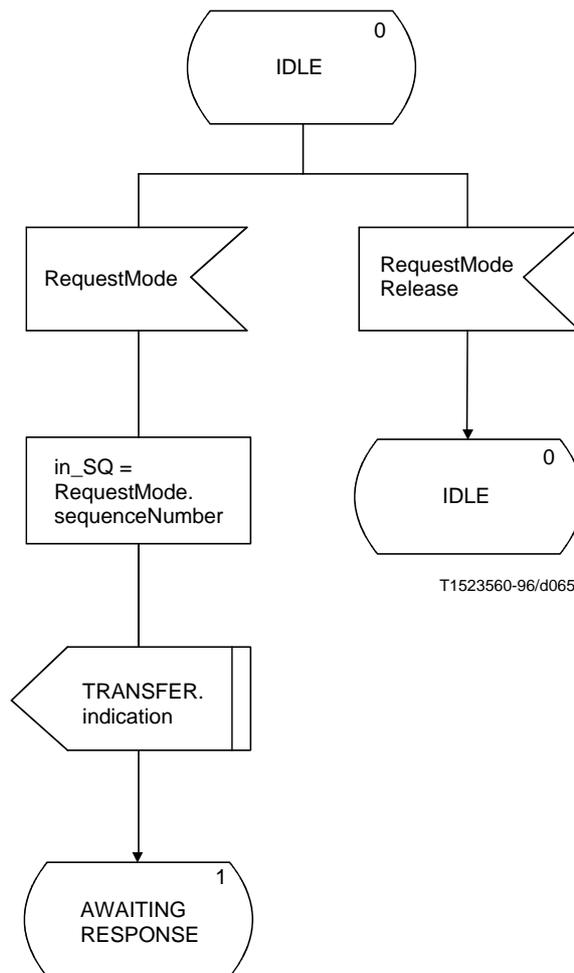
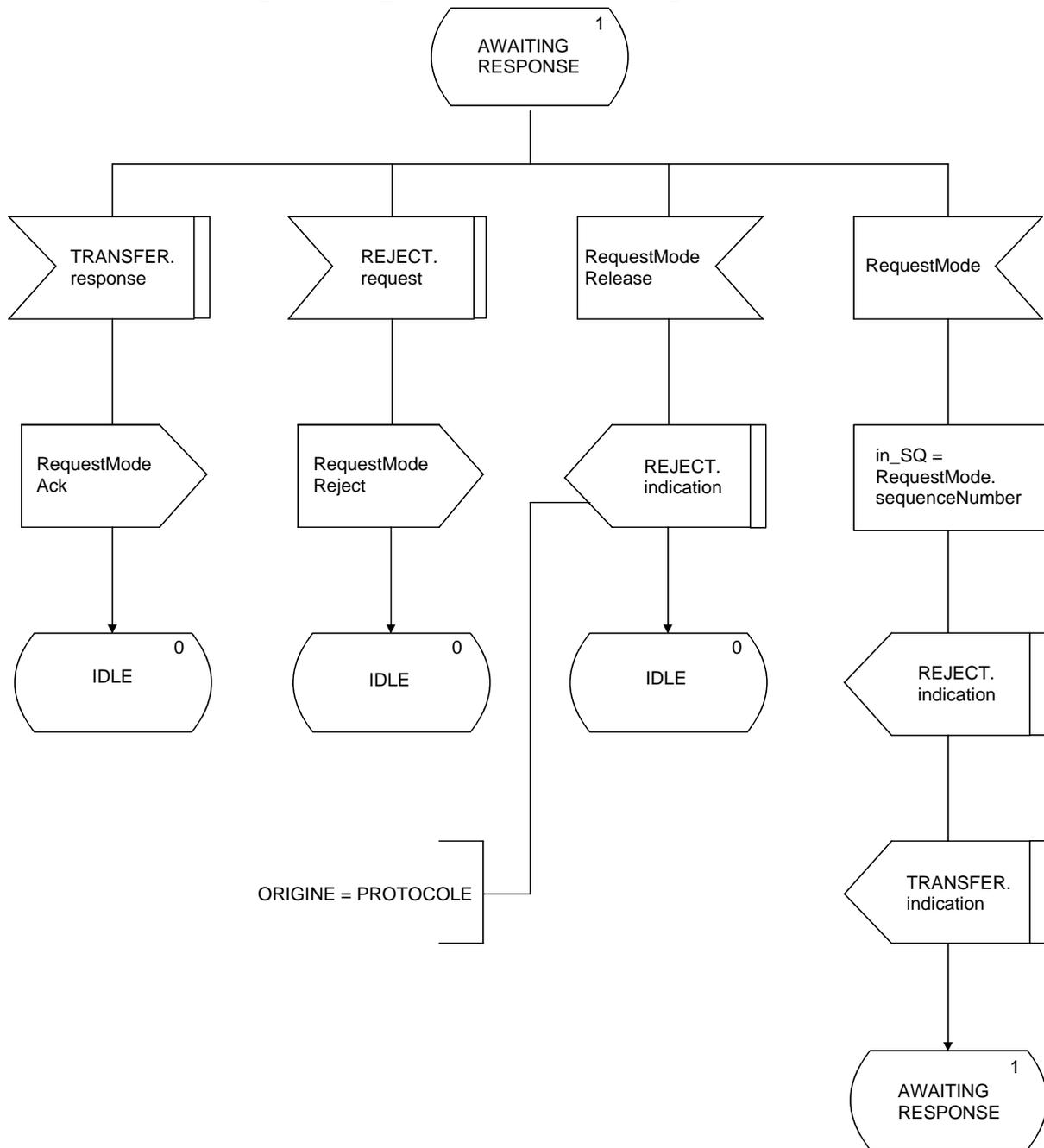


FIGURE 39 i)/H.245

Description SDL de l'entité MRSE entrante

Remplacée par une version plus récente



T1523570-96/D066

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 dans ces sous-paragraphes 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 existe toujours.

Remplacée par une version plus récente

Il est fait référence à la fonction décrite ici sous le nom d'entité de signalisation du temps de propagation aller et retour (RTDSE). 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. Tout terminal peut procéder à la détermination du temps de propagation aller et retour.

Le texte suivant donne un aperçu général du fonctionnement du protocole d'entité RTDSE. En cas de divergence entre la présente spécification et la spécification formelle, c'est celle-ci qui a priorité.

8.10.1.1 Aperçu général du protocole – Entité RTDSE

Une procédure de détermination du temps de propagation aller-retour est lancée lorsque la primitive de demande TRANSFER est émise par l'utilisateur situé dans l'entité RTDSE. Un message RoundTripDelayRequest est envoyé à l'entité RTDSE homologue et le temporisateur T105 est lancé. 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, de la valeur du temps de propagation aller-retour, qui est la valeur du temporisateur T105.

Si un message RoundTripDelayRequest est, à un moment quelconque, reçu de l'entité RTDSE homologue, un message RoundTripDelayResponse est immédiatement envoyé à l'entité RTDSE homologue.

Si le temporisateur T105 expire, l'utilisateur de l'entité RTDSE en est informé par la primitive d'indication EXPIRY.

Seul est accepté le message RoundTripDelayResponse qui est une réponse au plus récent message RoundTripDelayRequest. Les réponses à des messages RoundTripDelayRequest antérieurs sont ignorées.

Une nouvelle procédure de détermination du temps de propagation aller-retour peut être lancée par l'utilisateur de l'entité RTDSE au moyen de la primitive de demande TRANSFER, avant la réception d'un message RoundTripDelayResponse.

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 implémentation ou y 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 absence de paramètres.
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:

- la primitive TRANSFER est utilisée pour demander la détermination du temps de propagation aller et retour et pour en donner un compte rendu;
- 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:

- le paramètre DELAY indique le résultat du temps de propagation aller et retour.

Remplacée par une version plus récente

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:

état 0: IDLE

Il n'y a pas de requête RTDSE transfer en cours.

état 1: AWAITING RESPONSE

L'utilisateur de l'entité RTDSE locale 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 transition d'états

Ce 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.

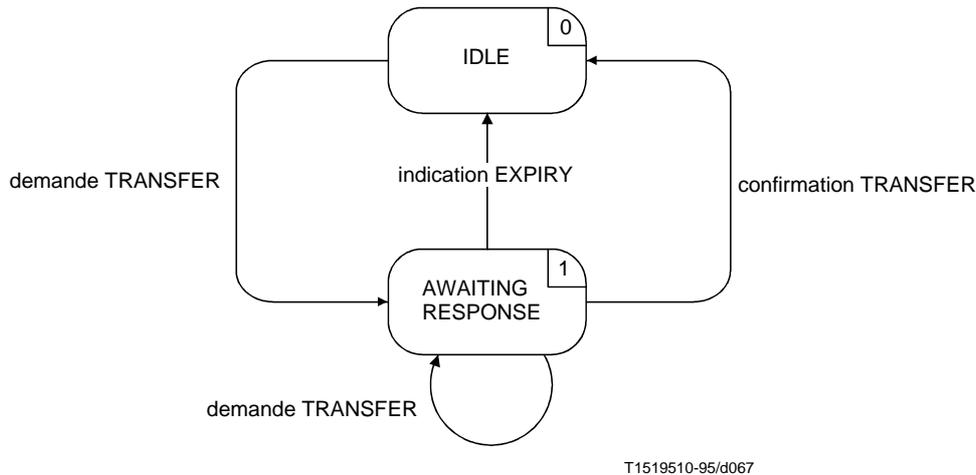


FIGURE 40/H.245

Diagramme de transition 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 de l'entité 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 appliquée 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.

Remplacée par une version plus récente

8.10.3.3 Temporisateurs RTDSE

Le temporisateur suivant est spécifié pour l'entité RTDSE sortante:

T105

Ce temporisateur est utilisé à l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RoundTripDelayResponse ne peut être reçu.

8.10.4 Procédures de l'entité RTDSE

8.10.4.1 Introduction

La Figure 41 récapitule les primitives RTDSE et leurs paramètres, ainsi que les messages.

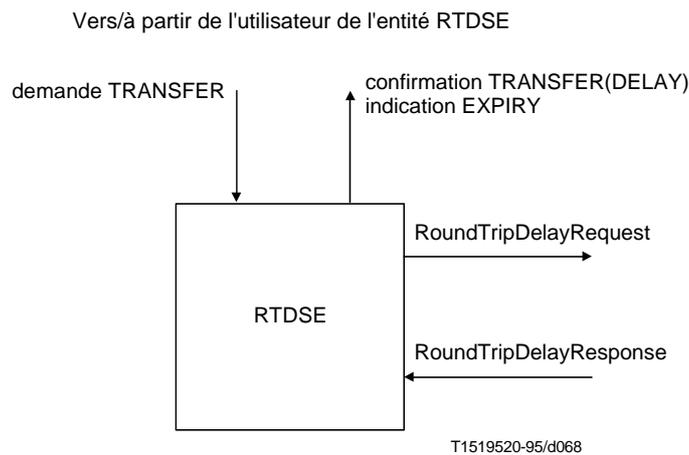


FIGURE 41/H.245

Primitives et messages de l'entité RTDSE

8.10.4.2 Valeurs par défaut des paramètres des 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 des primitives

Primitive	Paramètre	Valeur par défaut
confirmation TRANSFER	DELAY	Valeur initiale du temporisateur T105 moins la valeur du temporisateur T105
indication EXPIRY	–	–

NOTE – Les temporisateurs sont définis comme décomptant jusqu'à zéro. Le paramètre DELAY indique la durée pendant laquelle le temporisateur a fonctionné. Il a donc la valeur de la différence entre le réglage initial et la valeur conservée du temporisateur.

8.10.4.3 Valeurs par défaut des champs message

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs messages prennent des valeurs identiques à celles indiquées dans le Tableau 56.

Remplacée par une version plus récente

TABLEAU 56/H.245

Valeurs par défaut des champs 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.

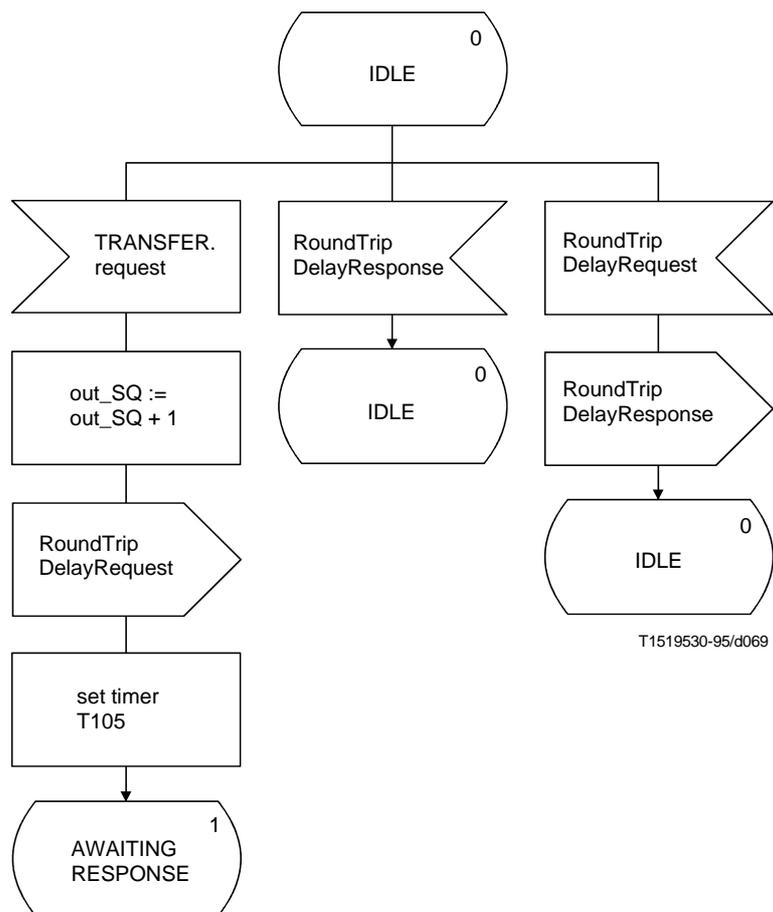


FIGURE 42 i)/H.245

Description SDL de l'entité RTDSE

Remplacée par une version plus récente

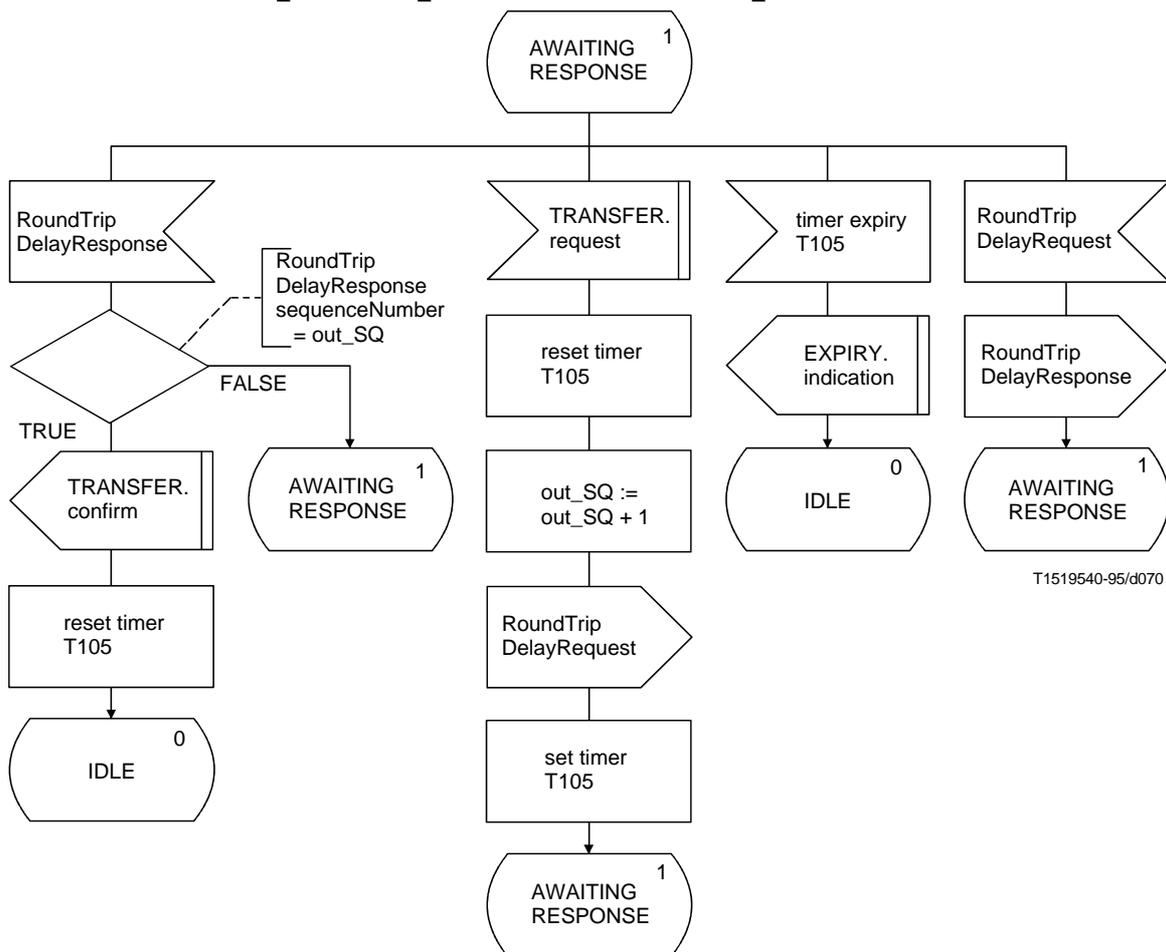


FIGURE 42 ii)/H.245

Description SDL de l'entité RTDSE

8.11 Procédures de boucle de maintenance

8.11.1 Introduction

Le protocole spécifié ici garantit le fonctionnement des boucles de maintenance par les procédures d'acquiescement.

Ce protocole est désigné ici sous le nom d'entité de signalisation de boucle de maintenance (MLSE). Les procédures sont spécifiées en termes de primitives, à l'interface entre l'entité MLSE et l'utilisateur de l'entité MLSE, et entre les états MLSE. Les informations de protocole sont transmises à l'entité MLSE homologue par les messages appropriés 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 de l'entité MLSE correspondant à chaque canal logique bidirectionnel, et une pour la boucle du système. Il n'y a pas de connexion entre une entité MLSE entrante et une entité MLSE sortante situées d'un même côté, hormis celle qui a été établie par l'intermédiaire des primitives vers l'utilisateur de l'entité MLSE et à partir de ce même utilisateur. Les situations d'erreur de l'entité MLSE font l'objet d'un compte rendu.

Le terminal qui contient l'entité MLSE entrante doit boucler les données appropriées lorsqu'il se trouve à l'état LOOPED, et à aucun autre moment. Le terminal qui contient l'entité MLSE sortante doit pouvoir recevoir des données bouclées, quel que soit l'état dans lequel il se trouve, mais ne doit recevoir que des données bouclées lorsqu'il se trouve à l'état LOOPED.

NOTE – Le message MaintenanceLoopOffCommand s'applique à toutes les entités MLSE. On l'utilise toujours pour stopper toutes les boucles de maintenance.

Remplacée par une version plus récente

Le texte suivant donne un aperçu du fonctionnement du protocole MLSE. En cas de divergence entre ce texte et la spécification formelle, cette dernière prévaut.

8.11.1.1 Aperçu général du protocole

L'émission de la primitive de demande LOOP par l'utilisateur dans l'entité MLSE sortante déclenche l'établissement d'une boucle de maintenance. Un message MaintenanceLoopRequest est envoyé à l'entité MLSE entrante homologue et le temporisateur T102 est mis en marche. A la réception d'un message MaintenanceLoopAck en réponse au message MaintenanceLoopRequest, le temporisateur T102 s'arrête et l'utilisateur est informé par une primitive de confirmation LOOP que la boucle de maintenance a bien été établie. Toutefois, à la réception d'un message MaintenanceLoopReject en réponse au message MaintenanceLoopRequest, le temporisateur T102 s'arrête et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité MLSE homologue a refusé d'établir la boucle de maintenance.

En cas d'expiration du temporisateur T102 pendant cette période, l'utilisateur en est informé par la primitive d'indication RELEASE et un message MaintenanceLoopOffCommand est envoyé à l'entité MLSE entrante homologue. Cette opération annulera toutes les boucles de maintenance et pas seulement celles se rapportant à l'entité MLSE concernée.

Une boucle de maintenance qui a été établie avec succès peut être annulée au moment où la primitive de demande RELEASE est émise par l'utilisateur dans l'entité MLSE sortante. Un message MaintenanceLoopOffCommand est envoyé à l'entité MLSE entrante homologue.

Avant réception des messages MaintenanceLoopAck ou MaintenanceLoopReject en réponse à un message MaintenanceLoopRequest précédemment envoyé, l'utilisateur dans l'entité MLSE sortante peut annuler la boucle de maintenance au moyen de la primitive de demande RELEASE.

8.11.1.2 Aperçu général du protocole – Entité entrante

A la réception d'un message MaintenanceLoopRequest 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 émettant la primitive de réponse LOOP et un message MaintenanceLoopAck est envoyé à l'entité MLSE sortante homologue. La boucle de maintenance doit dès lors être établie. L'utilisateur de l'entité MLSE entrante signale le rejet de la demande d'établissement de la boucle de maintenance en émettant 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 à la réception du message MaintenanceLoopOffCommand dans l'entité MLSE entrante. L'utilisateur de l'entité MLSE entrante en 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

L'entité MLSE et l'utilisateur de l'entité MLSE communiquent à l'aide 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 absence de paramètres.
NOTE 2 – «Non défini» signifie que cette primitive n'existe pas.

Remplacée par une version plus récente

8.11.2.2 Définition des primitives

Ces primitives sont définies comme suit:

- a) les primitives LOOP sont utilisées pour établir une boucle de maintenance;
- b) les primitives RELEASE sont utilisées pour supprimer une boucle de maintenance;
- c) la primitive ERROR signale les erreurs de l'entité MLSE à une entité de gestion.

8.11.2.3 Définition de paramètres

Les paramètres des primitives indiqués dans le Tableau 57 sont définis comme suit:

- a) le paramètre LOOP_TYPE spécifie les paramètres associés à la boucle de maintenance. Il possède les valeurs suivantes: «SYSTEM», «MEDIA», et «LOGICAL CHANNEL». Ce paramètre, associé au numéro de voie logique, détermine la valeur du champ «type» du message MaintenanceLoopRequest qui est ensuite transporté en transparence jusqu'à l'utilisateur de l'entité MLSE homologue;
- b) le paramètre SOURCE indique à l'utilisateur de l'entité MLSE l'origine de la libération de la boucle de maintenance. Le paramètre SOURCE prend la valeur «USER» ou «MLSE», désignant l'utilisateur de l'entité MLSE ou l'entité MLSE. Ce dernier cas peut découler d'une erreur de protocole;
- c) le paramètre CAUSE indique le motif pour lequel l'utilisateur de l'entité MLSE homologue a rejeté une demande d'établissement d'une boucle de maintenance. Le paramètre CAUSE est absent quand le paramètre SOURCE indique «MLSE»;
- d) le paramètre ERRCODE indique le type d'erreur de l'entité MLSE. Le Tableau 61 indique les valeurs autorisées du paramètre ERRCODE.

8.11.2.4 Etats de l'entité 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 l'échange de messages entre entités MLSE homologues. Les états sont spécifiés séparément pour chacune des entités MLSE sortante et entrante. Les états d'une entité MLSE sortante sont les suivants:

état 0: NOT LOOPED

Pas de boucle de maintenance.

état 1: AWAITING RESPONSE

L'entité MLSE sortante attend de pouvoir établir une boucle de maintenance avec une entité MLSE entrante homologue.

état 2: LOOPED

La boucle de maintenance entre les entités MLSE homologues a été établie. Toutes les données reçues sur la voie appropriée doivent être des données bouclées.

Les états d'une entité MLSE entrante sont les suivants:

état 0: NOT LOOPED

Pas de boucle de maintenance.

état 1: AWAITING RESPONSE

L'entité MLSE entrante attend de pouvoir établir une boucle de maintenance avec une entité MLSE sortante homologue. Les données appropriées ne doivent pas être bouclées.

état 2: LOOPED

Une boucle de maintenance entre les entités MLSE homologues a été établie. Toutes les données reçues sur la voie appropriée doivent être bouclées.

8.11.2.5 Diagramme de transition d'états

La séquence autorisée de primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE est définie ici. Elle se rapporte aux états de l'entité MLSE considérés du point de vue de l'utilisateur de l'entité MLSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités MLSE sortante et entrante, comme l'indiquent respectivement les Figures 43 et 44.

Remplacée par une version plus récente

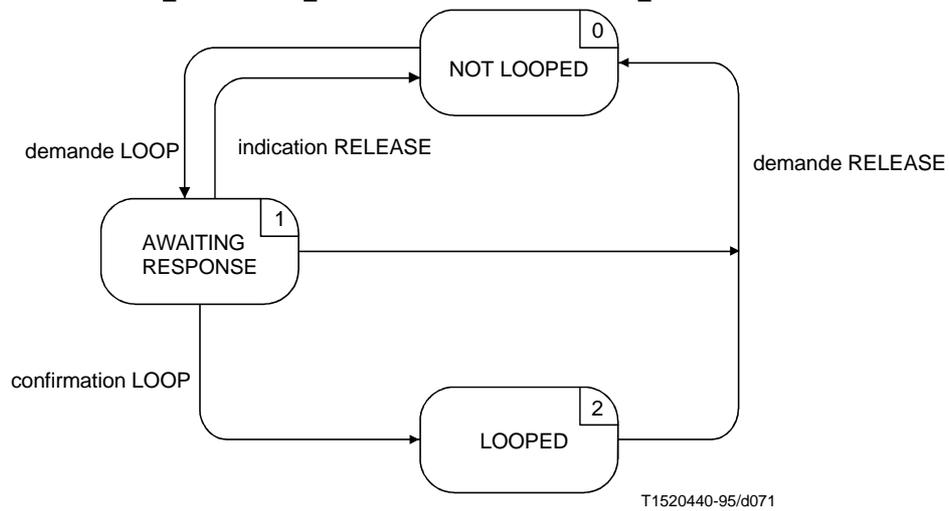


FIGURE 43/H.245

Diagramme de transition d'états pour une séquence de primitives de l'entité MLSE sortante

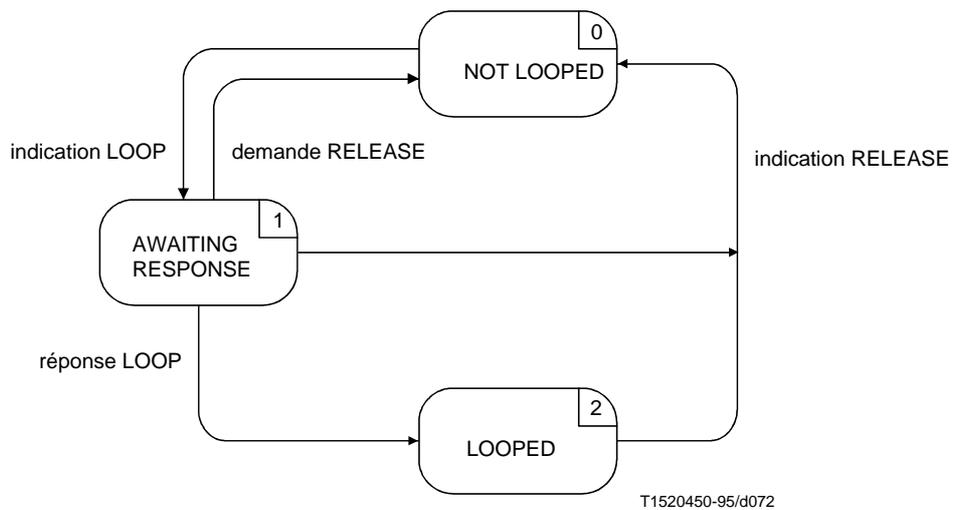


FIGURE 44/H.245

Diagramme de transition d'états pour une séquence de primitives de l'entité MLSE entrante

8.11.3 Communication entre entités MLSE homologues

8.11.3.1 Messages de l'entité MLSE

Le Tableau 58 indique les messages et les champs de l'entité MLSE, définis dans le paragraphe 6 et se rapportant au protocole MLSE.

Remplacée par une version plus récente

TABLEAU 58/H.245

Noms et champs de messages de l'entité MLSE

Fonction	Message	Sens	Champ
Etablissement	MaintenanceLoopRequest	S → E	type
	MaintenanceLoopAck	S ← E	type
	MaintenanceLoopReject	S ← E	type cause
Libération	MaintenanceLoopOffCommand	S → E	–
S Sortante E Entrante			

8.11.3.2

La variable d'état suivante est définie à l'entité MLSE sortante:

out_MLN

Cette variable d'état établit une distinction entre plusieurs entités MLSE sortantes. Elle est initialisée en même temps que l'entité MLSE sortante. La valeur de la variable out_MLN est utilisée pour fixer le champ «type» des messages «MaintenanceLoopRequest» envoyés par une entité MLSE sortante.

La variable d'état suivante est définie à l'entité MLSE entrante:

in_MLN

Cette variable d'état établit une distinction entre plusieurs entités MLSE entrantes. Elle est initialisée en même temps que l'entité MLSE entrante. Pour les messages MaintenanceLoopRequest reçus dans une entité MLSE entrante, la valeur du champ «type» contenue dans ces messages correspond à la valeur de la variable in_MLN.

in_TYPE

Cette variable d'état est utilisée pour mémoriser la valeur du paramètre LOOP_TYPE lorsque la demande de boucle de maintenance est reçue. Cette variable d'état aide à fixer le champ «type» du message MaintenanceLoopAck.

8.11.3.3 Temporisateurs 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 autorisé pendant lequel aucun message MaintenanceLoopAck ou MaintenanceLoopReject ne peut être reçu.

8.11.4 Procédures de l'entité 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 sortante et entrante.

8.11.4.2 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs par défaut indiquées dans le Tableau 59, lorsque leurs valeurs ne sont pas expressément mentionnées dans les diagrammes SDL.

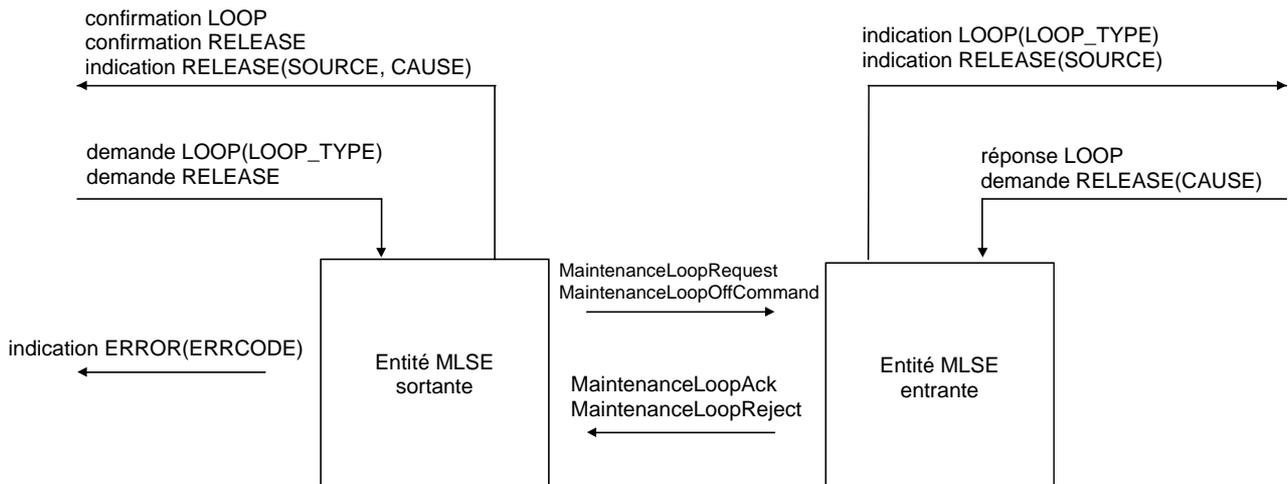
8.11.4.3 Valeurs par défaut des champs de messages

Les champs de messages prennent les valeurs par défaut indiquées dans le Tableau 60, lorsque leurs valeurs ne sont pas expressément mentionnées dans les diagrammes SDL.

Remplacée par une version plus récente

Vers/à partir de l'utilisateur
de l'entité MLSE sortante

Vers/à partir de l'utilisateur
de l'entité MLSE entrante



T1520460-95/d073

FIGURE 45/H.245

Primitives et messages dans l'entité de signalisation de boucle de maintenance

TABLEAU 59/H.245

Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut (Note)
indication LOOP	LOOP_TYPE	MaintenanceLoopRequest.type
indication RELEASE	SOURCE CAUSE	USER MaintenanceLoopReject.cause

NOTE – Lorsque le message ne comporte aucune indication de champ, la valeur de codage d'un paramètre de primitive doit être nulle.

TABLEAU 60/H.245

Valeurs par défaut du champ de message

Message	Champ	Valeur par défaut (Note 1)
MaintenanceLoopRequest	type	demande LOOP(LOOP_TYPE) et out_MLN (Note 2)
MaintenanceLoopAck	type	in_LOOP et in_MLN (Note 3)
MaintenanceLoopReject	type cause	in_LOOP et in_MLN (Note 3) demande RELEASE(CAUSE)
MaintenanceLoopOffCommand	–	–

NOTE 1 – Un champ de message ne doit pas être codé si le paramètre de la primitive correspondante 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.

Remplacée par une version plus récente

8.11.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une situation d'erreur particulière. Le Tableau 61 montre les valeurs que le paramètre ERRCODE peut prendre dans l'entité MLSE sortante. Aucune primitive d'indication ERROR n'est associée à l'entité MLSE entrante.

TABLEAU 61/H.245

Valeurs du paramètre ERRCODE dans l'entité MLSE sortante

Type d'erreur	Code d'erreur	Situation d'erreur	Etat
Message incorrect	A	MaintenanceLoopAck	LOOPED
Pas de réponse de l'entité MLSE homologue	B	expiration du temporisateur T102	AWAITING RESPONSE

8.11.4.5 Description SDL

Les procédures des entités MLSE sortante et entrante sont exprimées en langage SDL sur les Figures 46 et 47 respectivement.

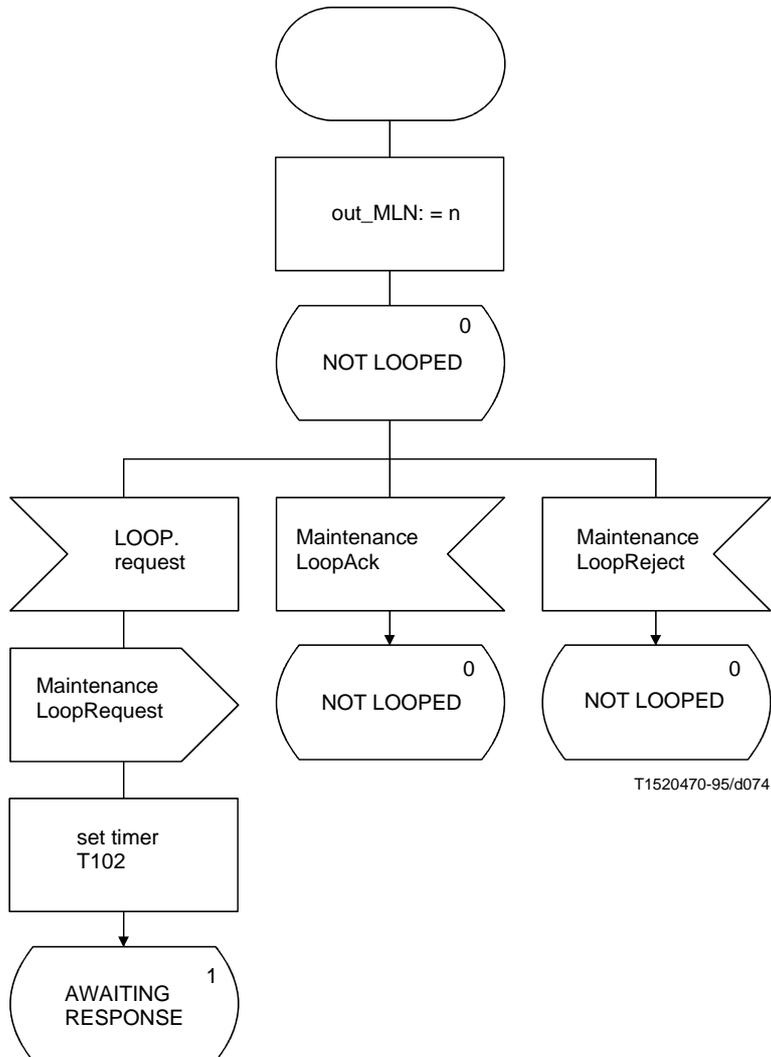
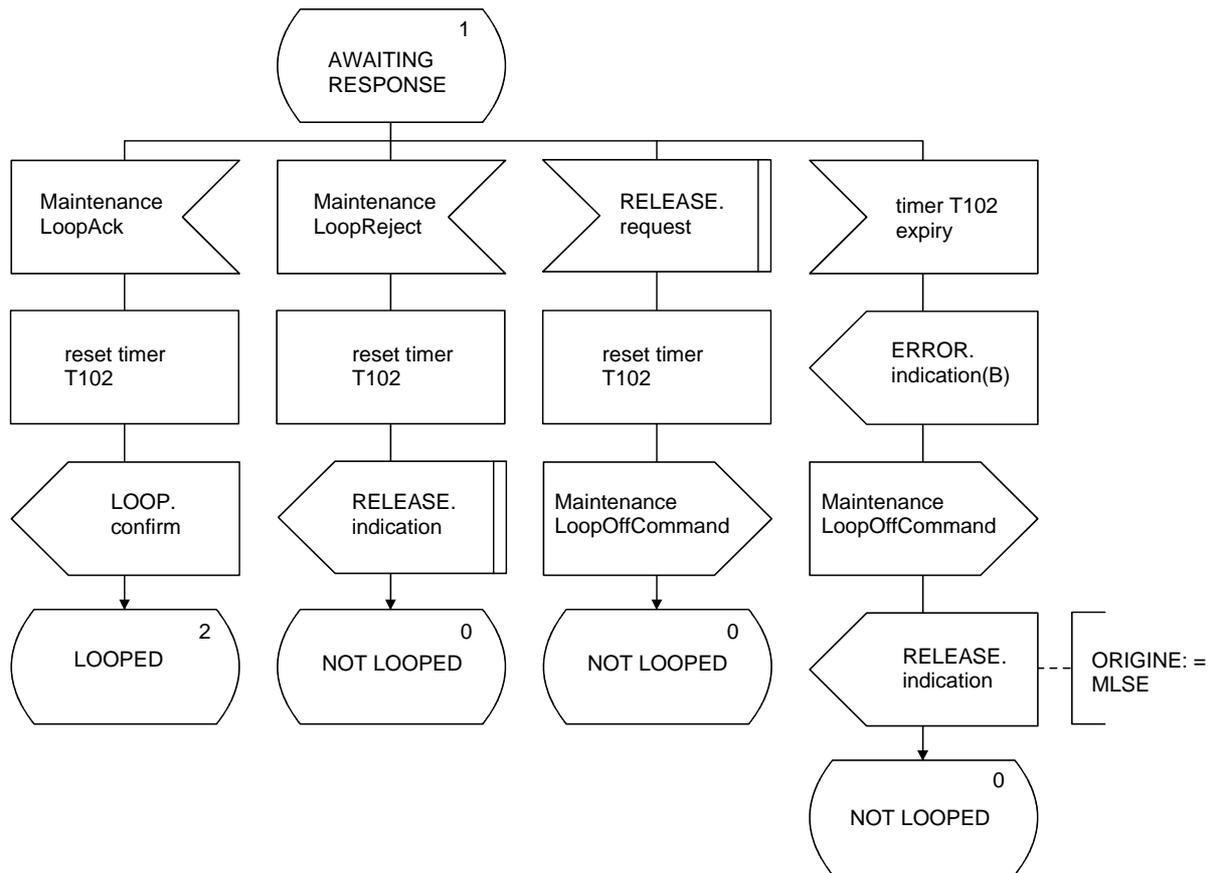


FIGURE 46 i)/H.245

Description SDL de l'entité MLSE sortante

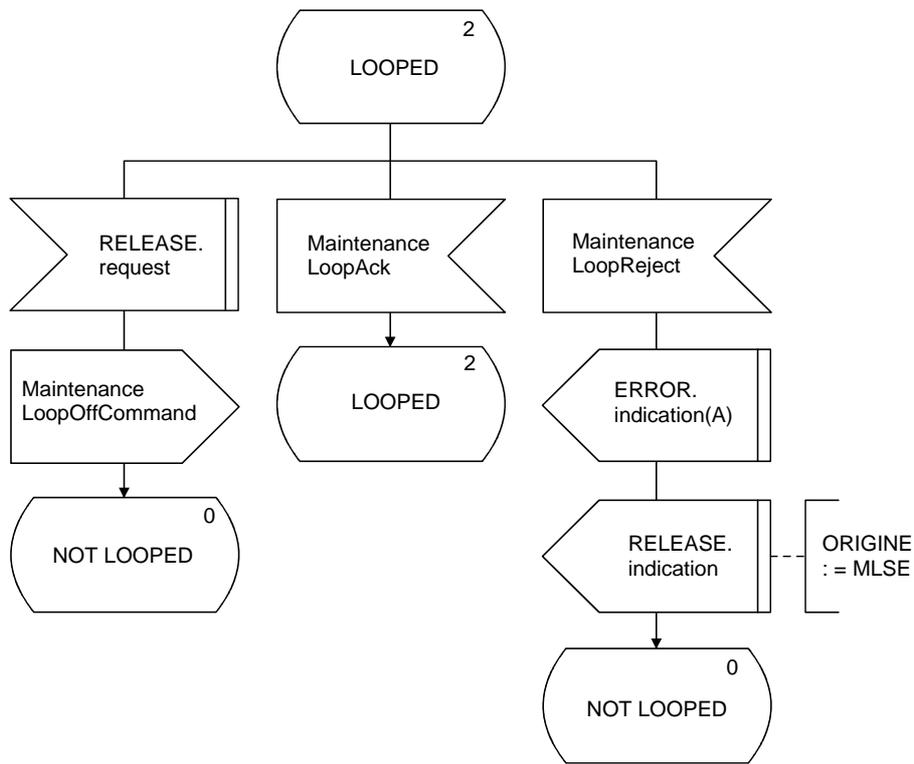
Remplacée par une version plus récente



T1520480-95/d075

FIGURE 46 ii)/H.245
Description SDL de l'entité MLSE sortante

Remplacée par une version plus récente



T1520490-95/d076

FIGURE 46 iii)/H.245
Description SDL de l'entité MLSE sortante

Remplacée par une version plus récente

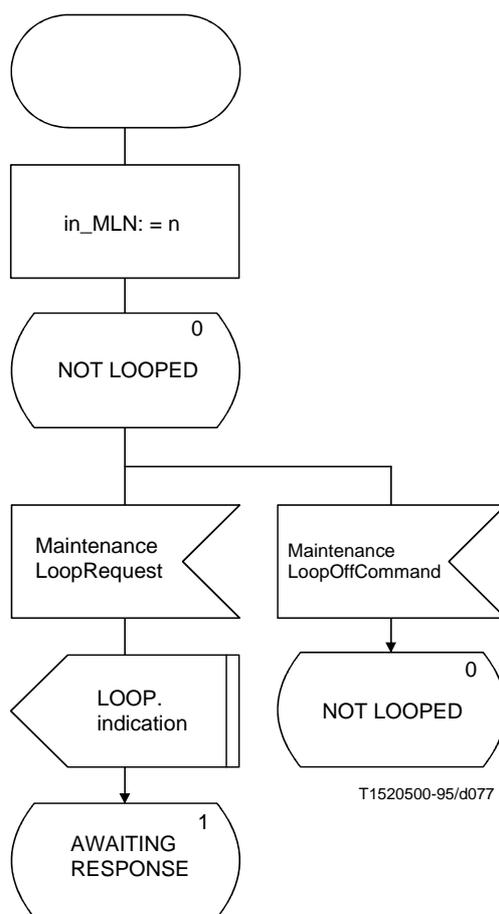
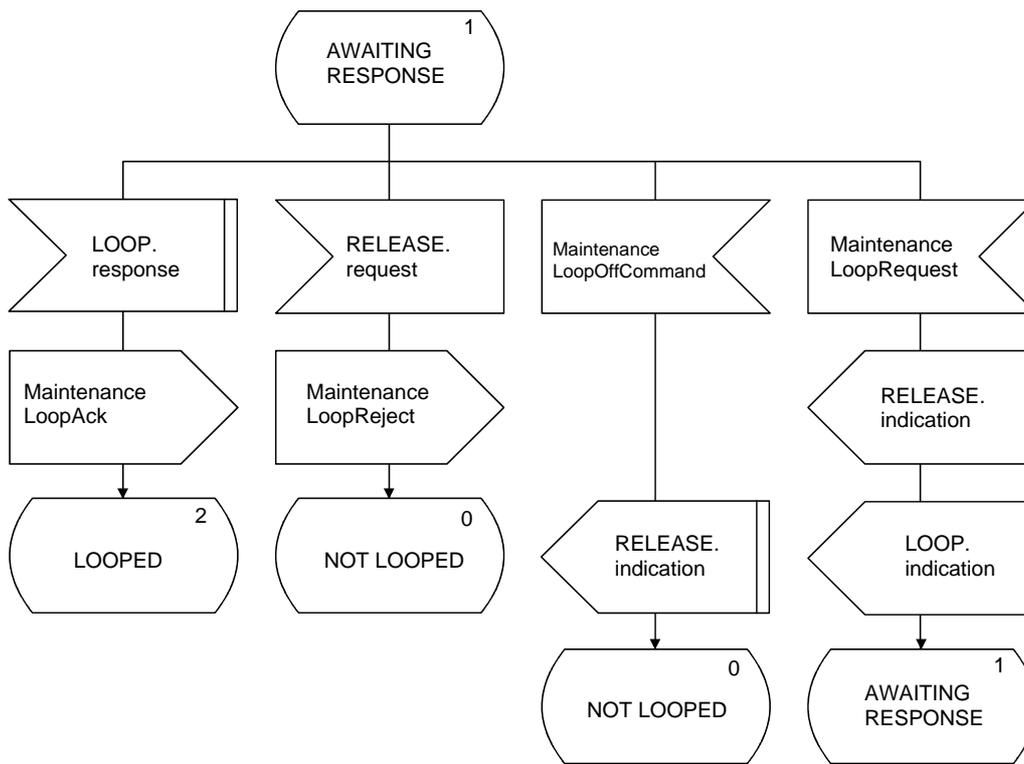


FIGURE 47 i)/H.245

Description SDL de l'entité MLSE entrante

Remplacée par une version plus récente

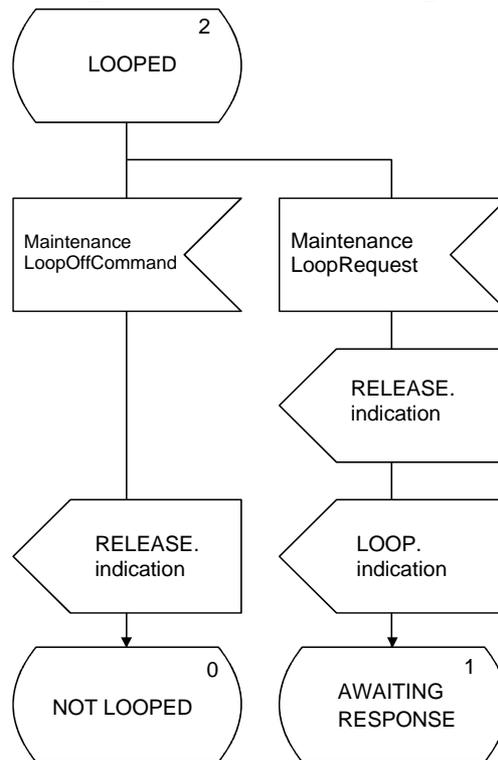


T1520510-95/d078

FIGURE 47 ii)/H.245

Description SDL de l'entité MLSE entrante

Remplacée par une version plus récente



T1520520-95/d079

FIGURE 47 iii)/H.245

Description SDL de l'entité MLSE entrante

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/H.245

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 à utiliser comme protocole de commande d'un système multimédia. A l'heure actuelle, une seule version normalisée est définie

Remplacée par une version plus récente

Appendice I

Aperçu général de la notation ASN.1

I.1 Introduction à la notation ASN.1

La notation de syntaxe abstraite un (ASN.1) est un langage de spécification de données. Ce langage a été initialement normalisé sous la référence de 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 notation 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 qu'une récurrence.

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 leurs valeurs. Les règles de codage compact (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 implémentations 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 des 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 en notation 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:

maximumAI2SDUSize

INTEGER (0..65535)

Les nombres entiers limités sont codés différemment selon la taille de la gamme. Supposons que N soit le nombre d'entiers 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 aligné sur octets
supérieur	comme le nombre minimal d'octets alignés précédé par le codage du nombre d'octets selon la règle ci-dessus.

Remplacée par une version plus récente

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» prennent de 0 à 7 bits qui sont ajoutés à la chaîne codée de sorte que le champ suivant puisse commencer à la limite 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.

Remplacée par une version plus récente

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 limite de TAILLE 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.

I.4 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 **NonStandardIdentifieur** contient:

```
object          OBJECT IDENTIFIER (identificateur d'objet)
```

Ce paramètre est codé comme les données codées avec les règles de codage de base (BER) (voir la Recommandation X.690) précédées par la longueur de ce codage en octets. La longueur est codée comme un nombre entier semi-limité (voir l'exemple OCTET STRING ci-dessus).

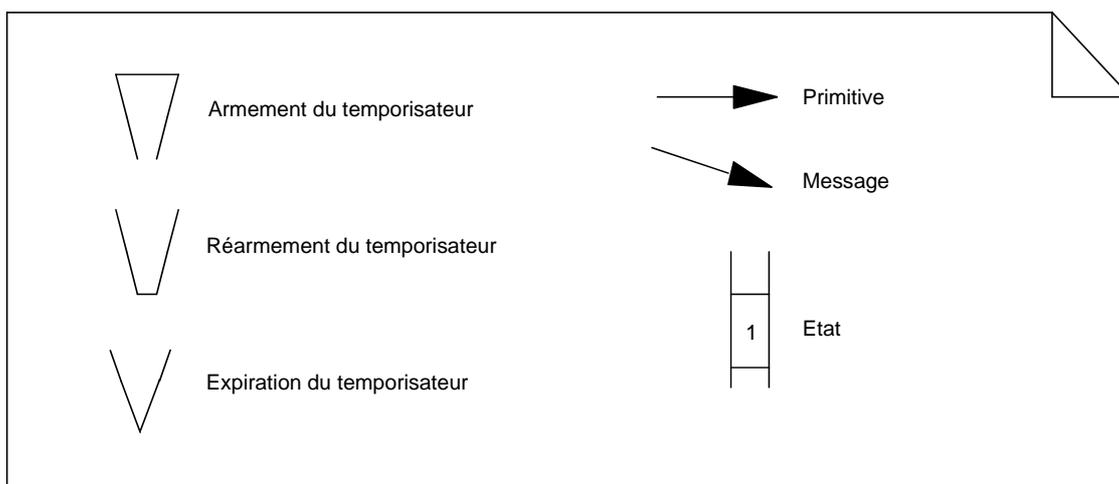
Remplacée par une version plus récente

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 de cet appendice.



T1523580-96/d080

FIGURE II.1-1/H.245

Symboles utilisés dans les figures

II.2 Entité de signalisation de choix entre mode maître ou esclave

Dans les Figures II.2-1 à II.2-5, les messages sont représentés par les noms abrégés indiqués dans le Tableau II.2-1.

TABLEAU II.2-1/H.245

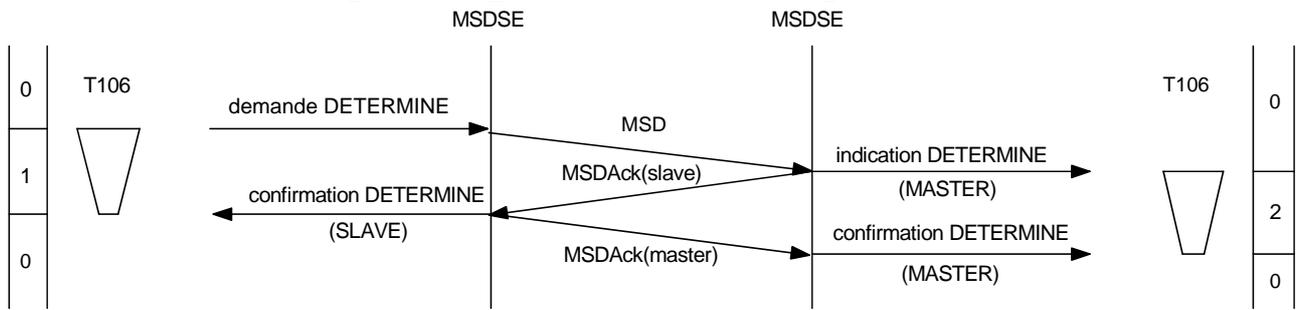
Noms abrégés du choix maître/esclave

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 et de confirmation DETERMINE est celle du paramètre TYPE. La valeur de champ associée au message MasterSlaveDeterminationAck est celle du champ «decision».

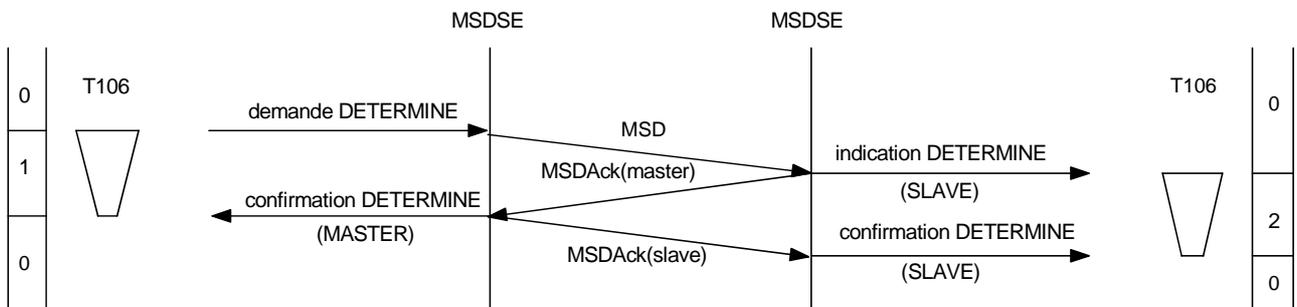
Remplacée par une version plus récente



T1523590-96/d081

FIGURE II.2-1/H.245

Choix du mode maître/esclave – Terminal maître à l'entité MSDSE distante

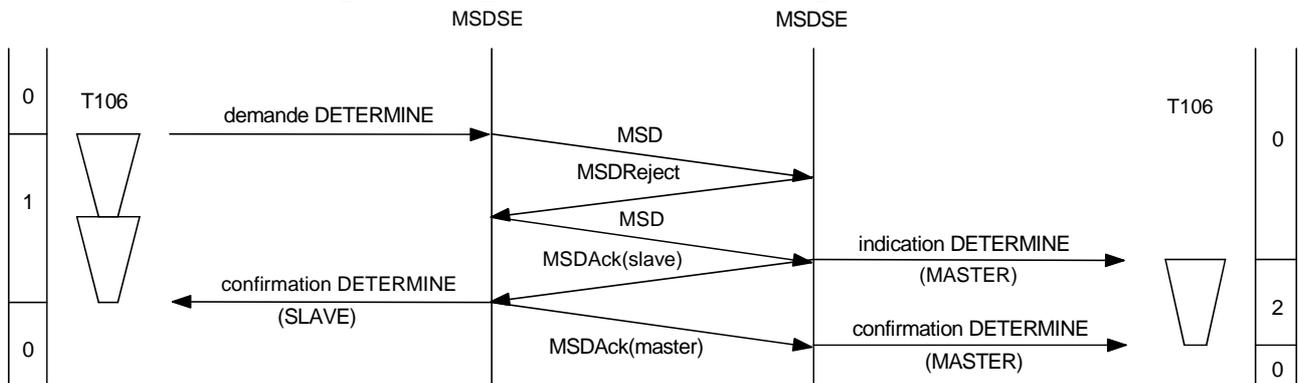


T1523600-96/d082

FIGURE II.2-2/H.245

Choix du mode maître/esclave – Esclave à l'entité MSDSE distante

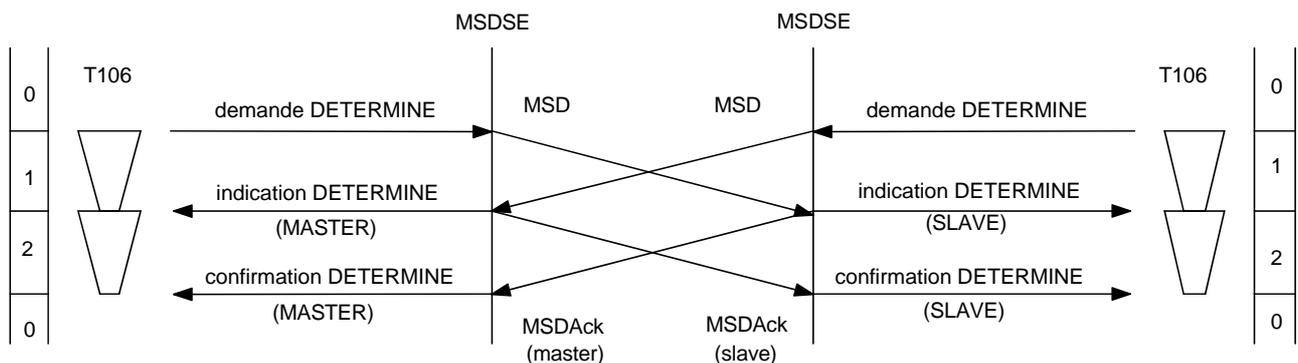
Remplacée par une version plus récente



T1523610-96/d083

FIGURE II.2-3/H.245

Choix du mode maître/esclave – La première tentative a donné un résultat indéterminé – La deuxième tentative a été efficace



T1523620-96/d084

FIGURE II.2-4/H.245

Choix du mode maître/esclave – Choix simultané

Remplacée par une version plus récente

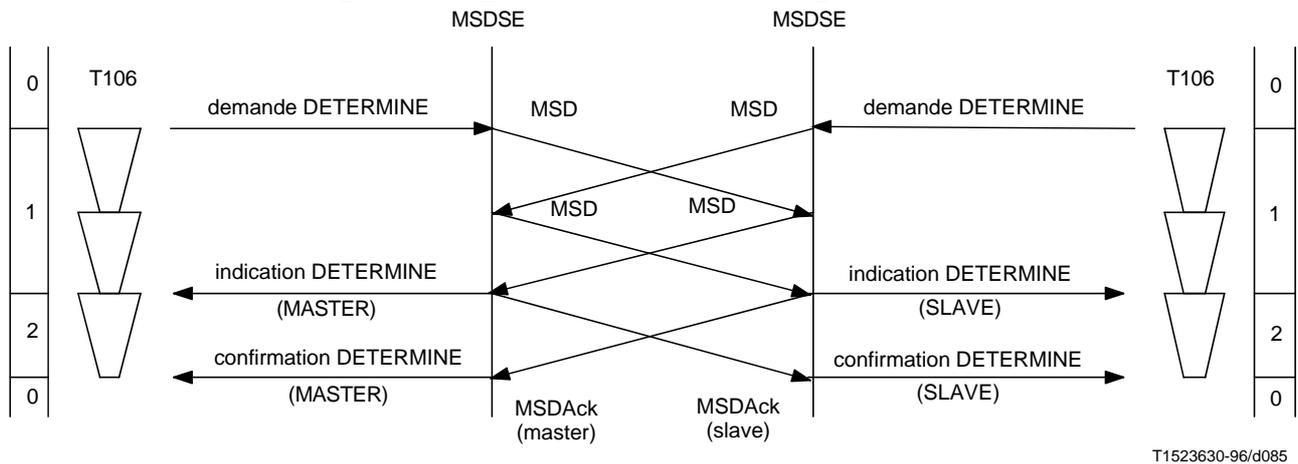


FIGURE II.2-5/H.245

Choix du mode maître/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 nouveaux ordres mais ne peut rien demander à l'autre terminal qui dépend de la connaissance du résultat de la détermination de statut. Le terminal de gauche ne peut ni accepter ni lancer de nouvelles procédures. Une deuxième procédure de détermination de statut doit être lancée

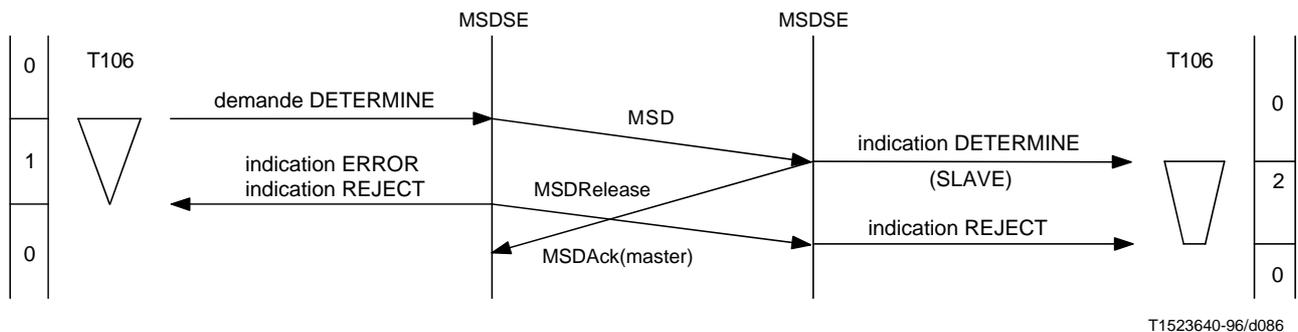


FIGURE II.2-6/H.245

Choix du mode maître/esclave – Temporisateur local T106 à expiration avec esclave à l'entité distante

Dans la Figure II.2-7, le temporisateur distant T106 a expiré au cours de l'état INCOMING AWAITING ACKNOWLEDGEMENT. Les deux terminaux connaissent leur statut. Le terminal de gauche peut recevoir et émettre des ordres. Le terminal distant ne sait cependant pas si le terminal local est prêt à recevoir, et il ne peut émettre d'ordres qui dépendent de la connaissance du résultat de la détermination du statut. Une deuxième procédure de détermination du statut doit être lancée.

Remplacée par une version plus récente

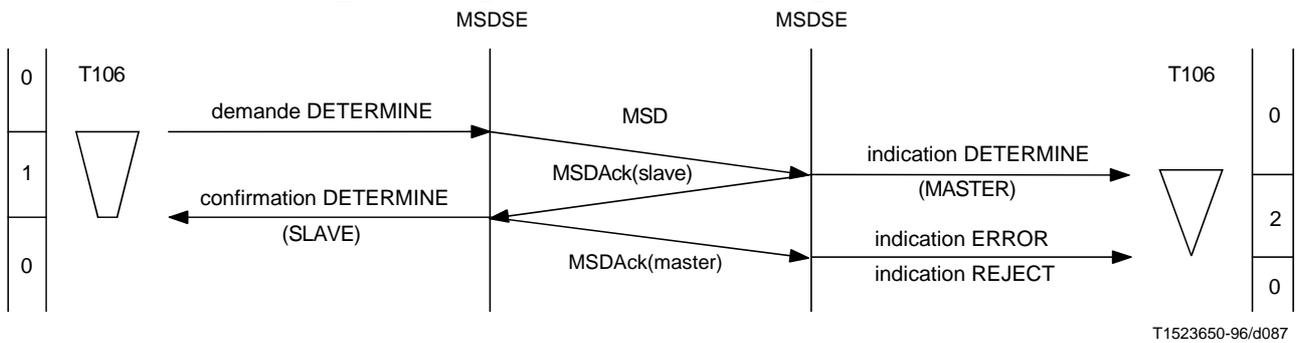


FIGURE II.2-7/H.245

Choix du mode maître/esclave – Temporisateur distant T106 à expiration avec terminal maître à l'entité distante

Dans la Figure II.2-8, le temporisateur distant T106 est arrivé à expiration au cours de 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 ordres. Cependant, le terminal de gauche ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre d'ordres qui dépendent de la connaissance du résultat de la détermination de statut. Le terminal de gauche peut recevoir de tels ordres. Une deuxième procédure de détermination de statut doit être lancée.

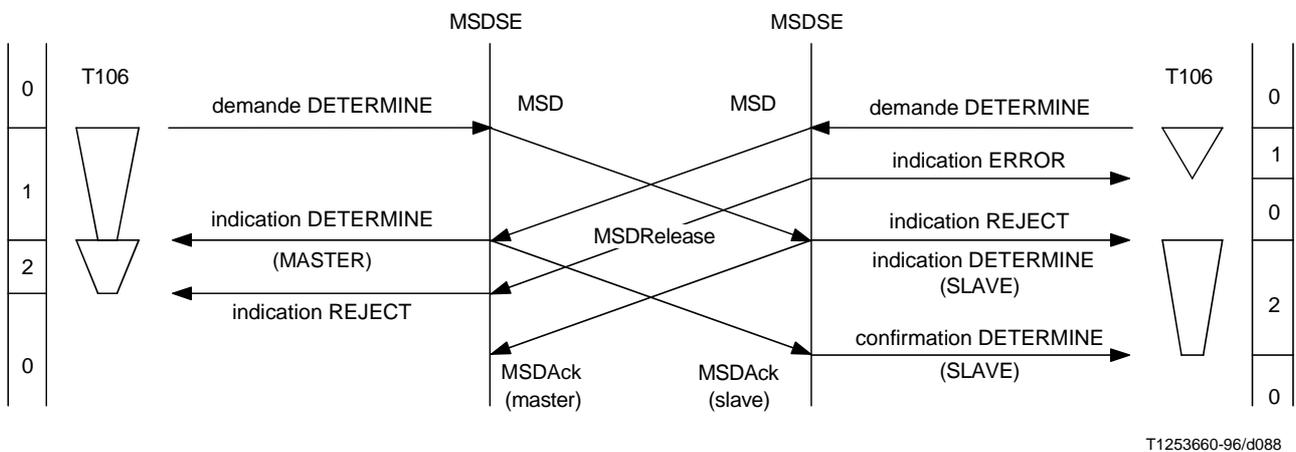


FIGURE II.2-8/H.245

Choix du mode maître/esclave – Procédures de détermination simultanée avec expiration du temporisateur T106 dans l'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 ordres. Cependant, le terminal de droite ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre d'ordres qui dépendent de la connaissance du résultat de la détermination de statut. Le terminal de droite peut recevoir de tels ordres. Une deuxième procédure de détermination de statut doit être lancée.

Remplacée par une version plus récente

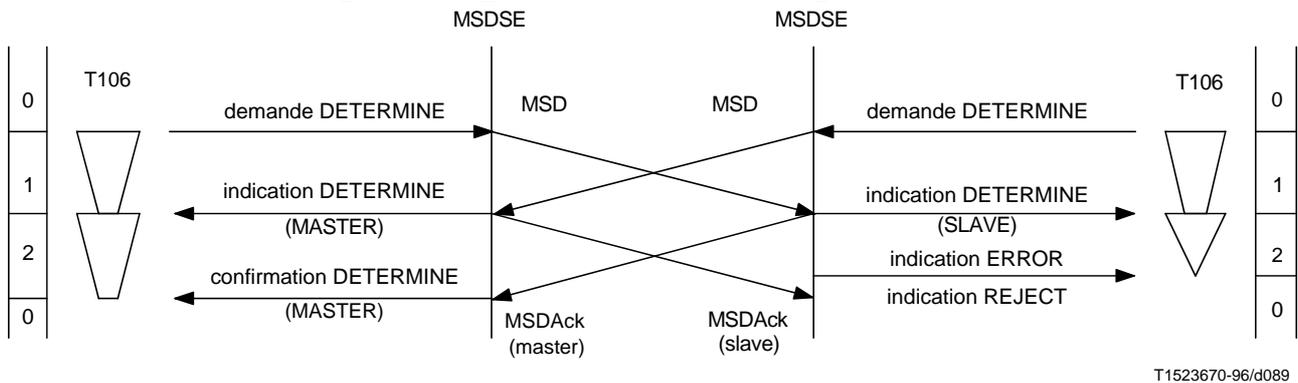


FIGURE II.2-9/H.245

Choix du mode maître/esclave – Procédure 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.

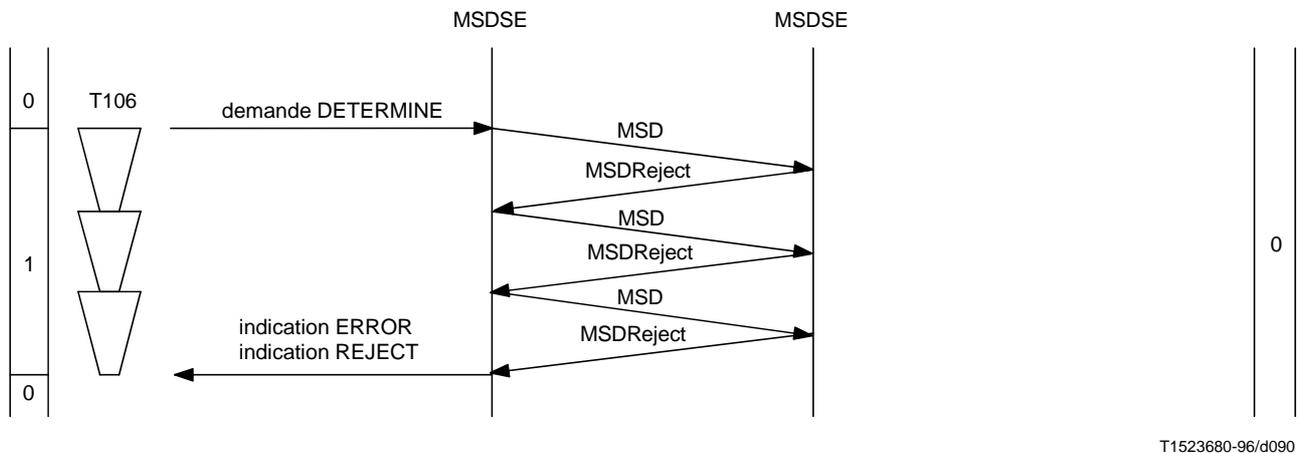


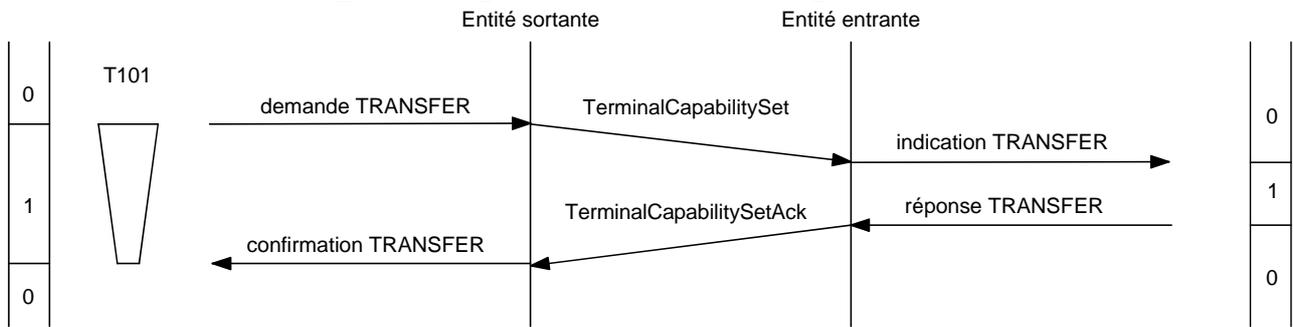
FIGURE II.2-10/H.245

Choix du mode maître/esclave – Résultat indéterminé avec N100 = 3

II.3 Entité de signalisation d'échange de capacités

Les Figures II.3-1 à II.3-4 illustrent les procédures d'entité CESE. Les états IDLE et AWAITING RESPONSE sont respectivement désignés par «0» et «1».

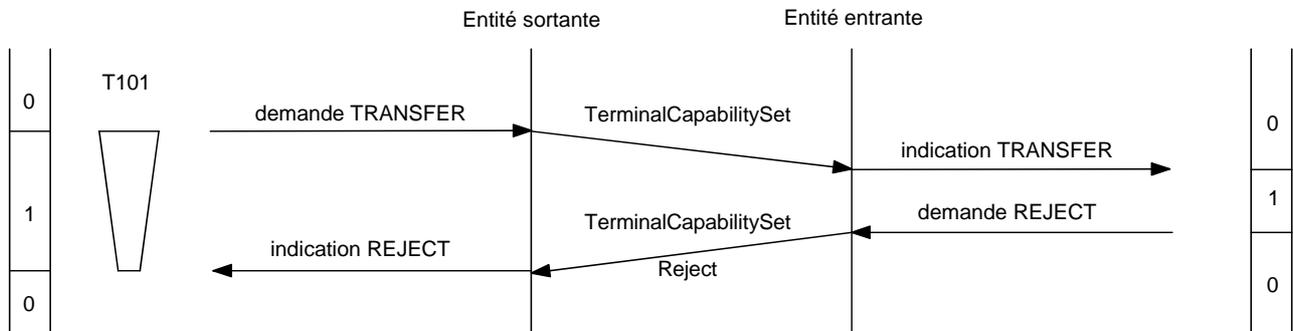
Remplacée par une version plus récente



T1523690-96/d091

FIGURE II.3-1/H.245

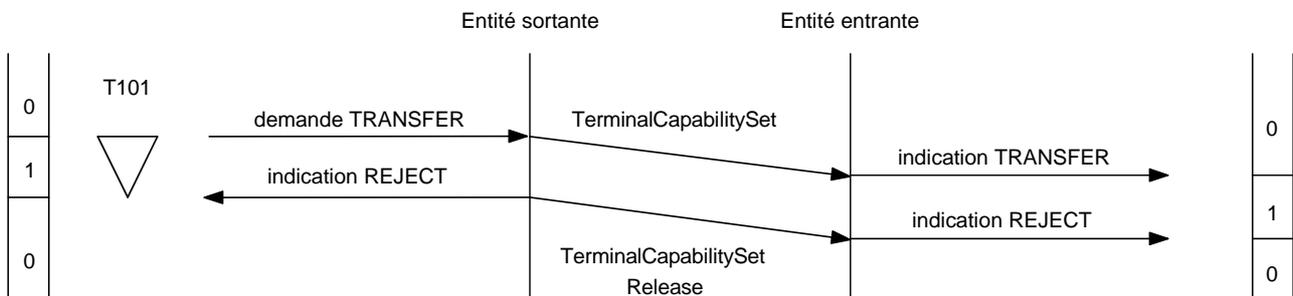
Echange de capacités avec acceptation issue de l'utilisateur de l'entité CESE homologue entrante



T1523700-96/d092

FIGURE II.3-2/H.245

Echange de capacités avec rejet issu de l'utilisateur de l'entité CESE homologue entrante

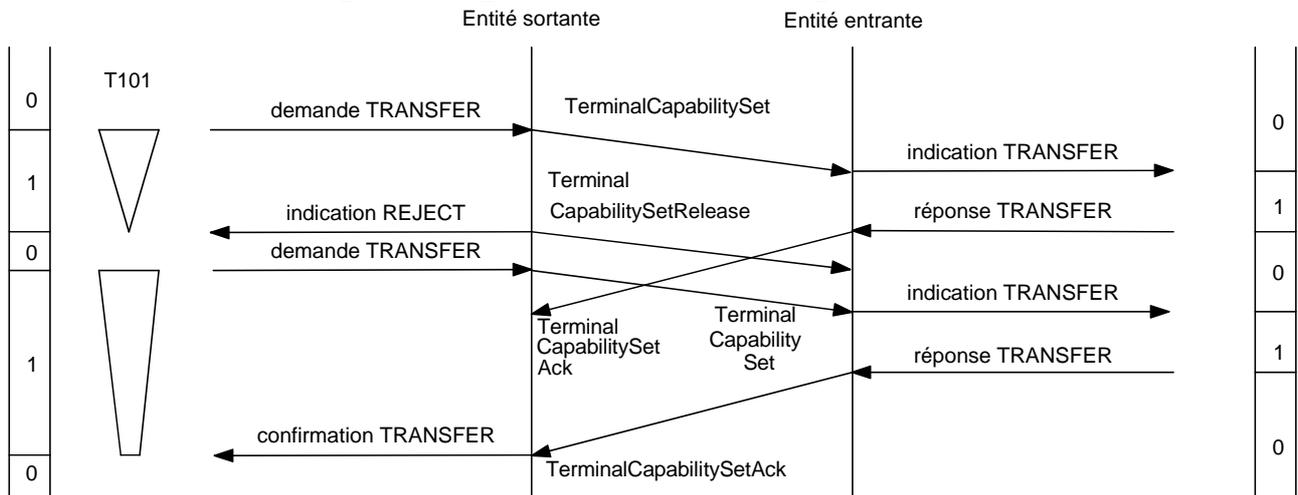


T1523710-96/d093

FIGURE II.3-3/H.245

Echange de capacités avec expiration du temporisateur T101 – Le message TerminalCapabilitySetRelease message arrive à l'entité CESE entrante avant la réponse issue de l'utilisateur de l'entité CESE entrante

Remplacée par une version plus récente



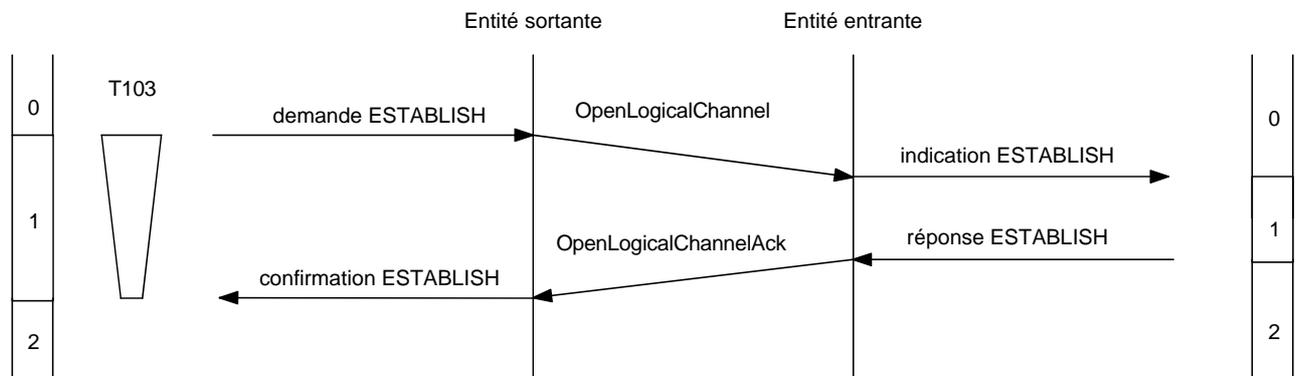
T1523720-96/d094

FIGURE II.3-4/H.245

Echange de capacités avec expiration du temporisateur T101 suivi d'un deuxième échange de capacité – Le message TerminalCapabilitySetRelease arrive à l'entité CESE entrante après la réponse issue 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 ignoré – Seul le second échange de capacités est efficace

II.4 Entité de signalisation de voie logique

Les Figures II.4-1 à II.4-7 illustrent les procédures d'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».

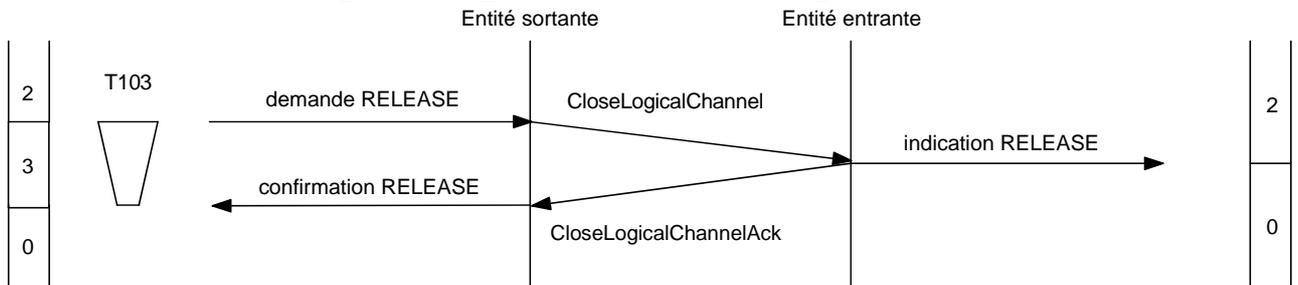


T1523730-96/d095

FIGURE II.4-1/H.245

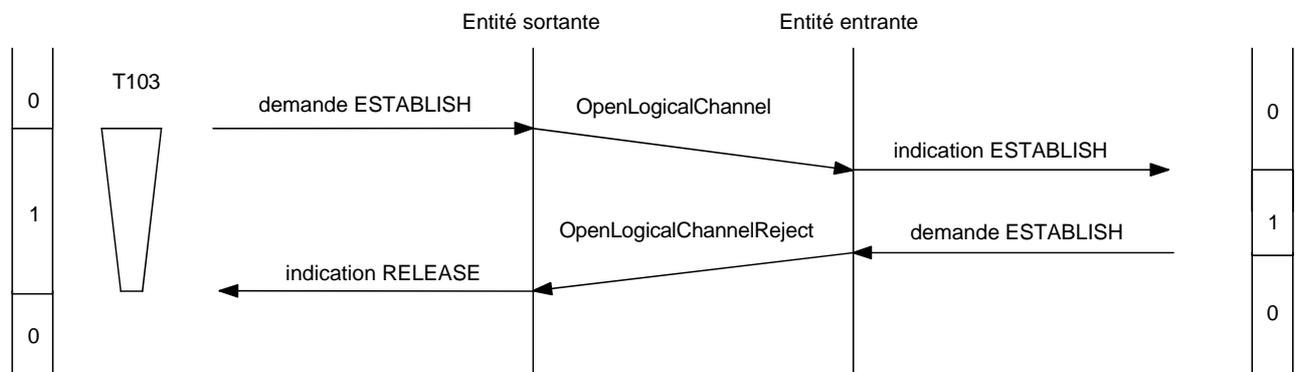
Etablissement d'une voie logique

Remplacée par une version plus récente



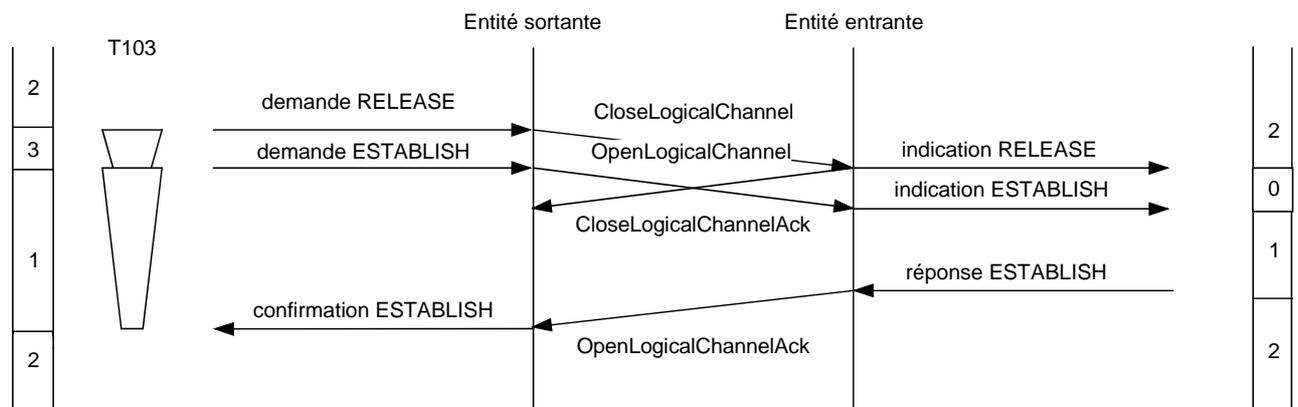
T1523740-96/d096

FIGURE II.4-2/H.245
Libération de voie logique



T1523750-96/d097

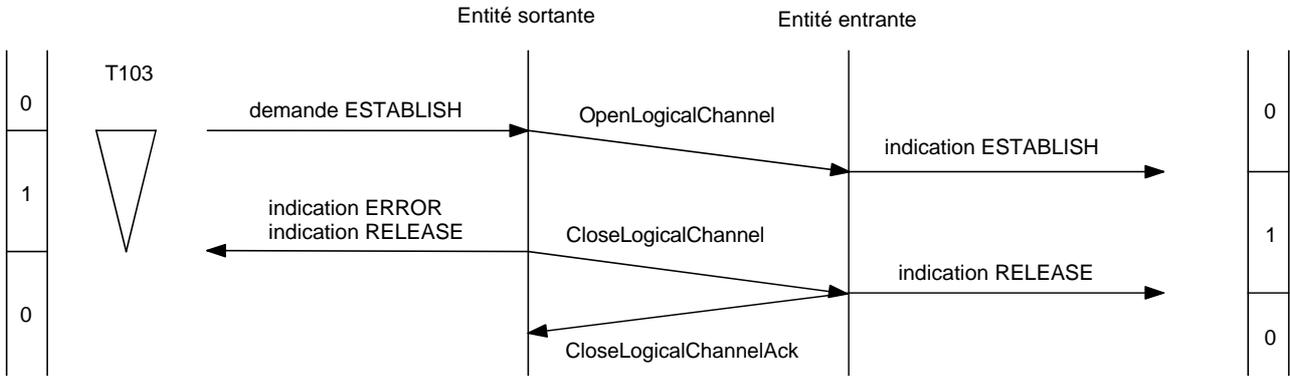
FIGURE II.4-3/H.245
Rejet d'établissement de voie logique par utilisateur d'entité LCSE homologue



T1523760-96/d098

FIGURE II.4-4/H.245
Libération de voie logique suivie d'un rétablissement immédiat

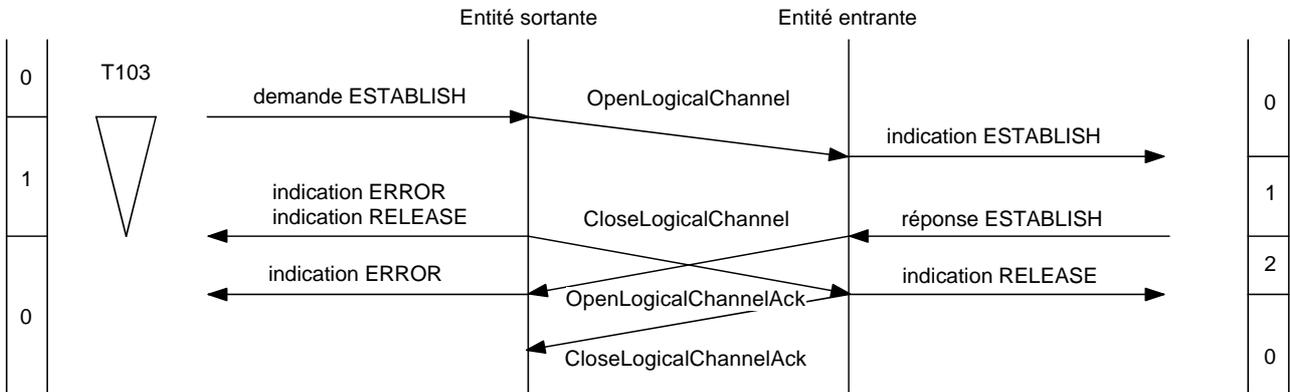
Remplacée par une version plus récente



T1523770-96/d099

FIGURE II.4-5/H.245

Demande d'établissement de voie logique avec expiration du temporisateur T103 en raison d'un retard de la réponse issue de l'utilisateur d'entité LCSE homologue entrante

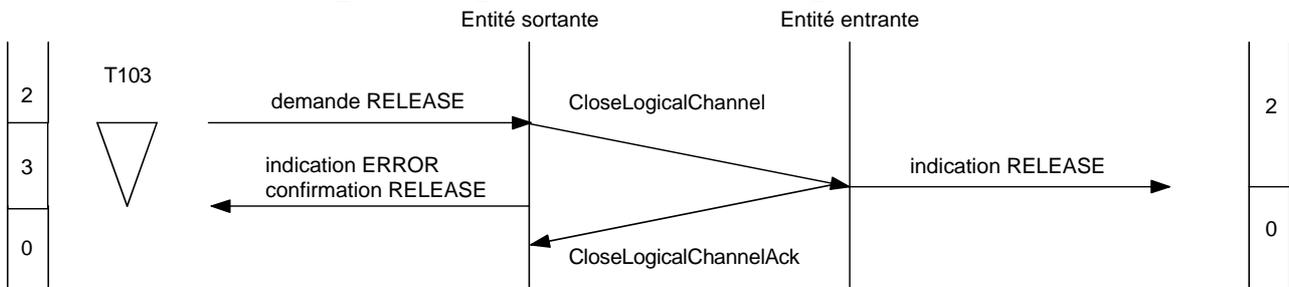


T1523780-96/d100

FIGURE II.4-6/H.245

Demande d'établissement de voie logique avec expiration du temporisateur T103 – Le temporisateur T103 a expiré après transmission du message OpenLogicalChannelAck à l'entité LCSE entrante, mais avant réception du message OpenLogicalChannelAck à l'entité LCSE sortante

Remplacée par une version plus récente



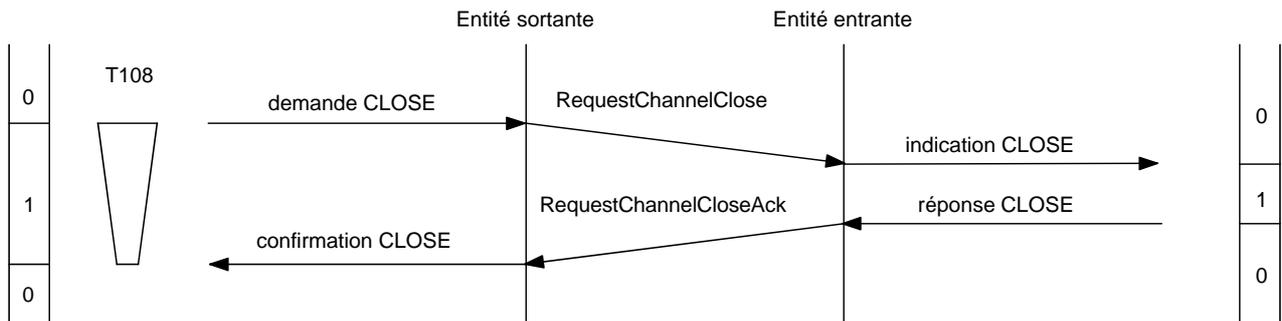
T1523790-96/d101

FIGURE II.4-7/H.245

Demande de libération de voie logique avec expiration du temporisateur T103

II.5 Entité de signalisation de clôture de voie logique

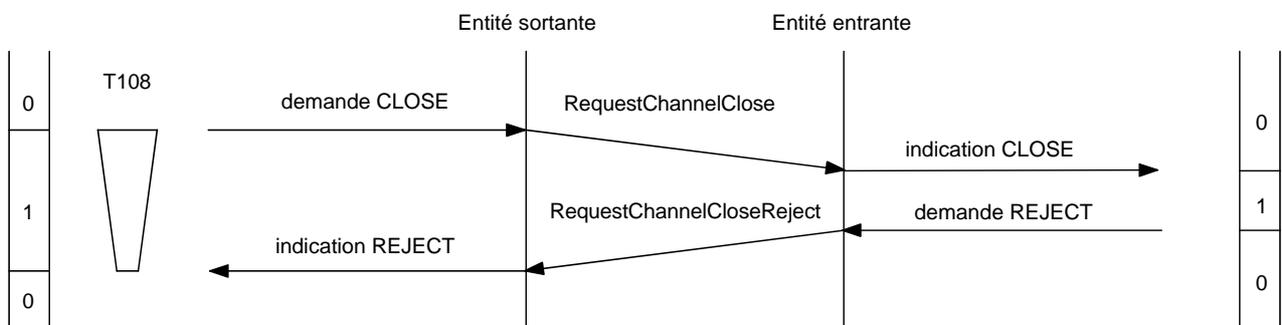
Les Figures II.5-1 à II.5-4 illustrent les procédures d'entité CLCSE. Les états IDLE et AWAITING RESPONSE sont désignés par «0» et «1» respectivement.



T1523800-96/d102

FIGURE II.5-1/H.245

Demande de fermeture de voie logique



T1523810-96/d103

FIGURE II.5-2/H.245

Demande de fermeture de voie logique avec rejet issu de l'utilisateur de l'entité CLCSE homologue entrante

Remplacée par une version plus récente

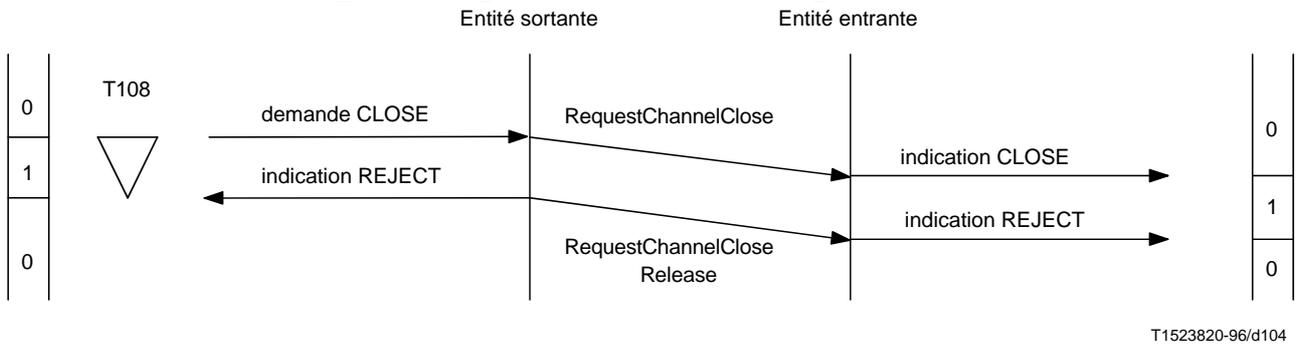


FIGURE II.5-3/H.245

Demande de fermeture de voie logique avec expiration du temporisateur T108 – Le message RequestChannelCloseRelease arrive à l'entité CLCSE entrante avant la réponse issue de l'utilisateur de l'entité CLCSE entrante

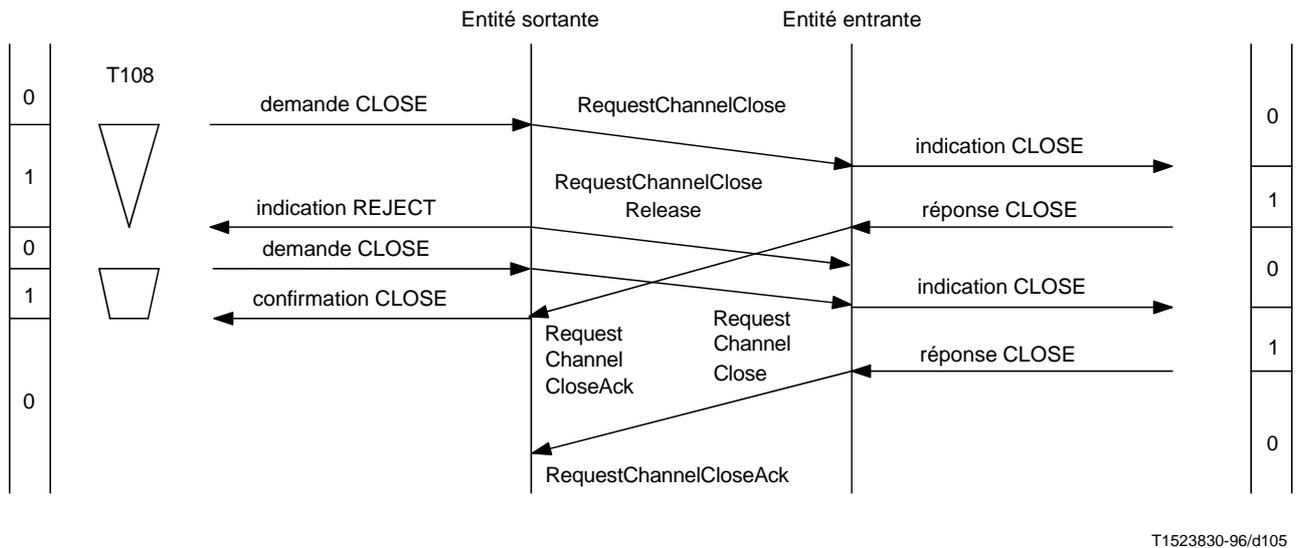


FIGURE II.5-4/H.245

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 réception du premier message RequestChannelClose

II.6 Entité de signalisation de table de multiplexage

Les Figures II.6-1 à II.6-5 illustrent les procédures d'entité MTSE. Les états IDLE et AWAITING RESPONSE states sont respectivement désignés par «0» et «1».

Remplacée par une version plus récente

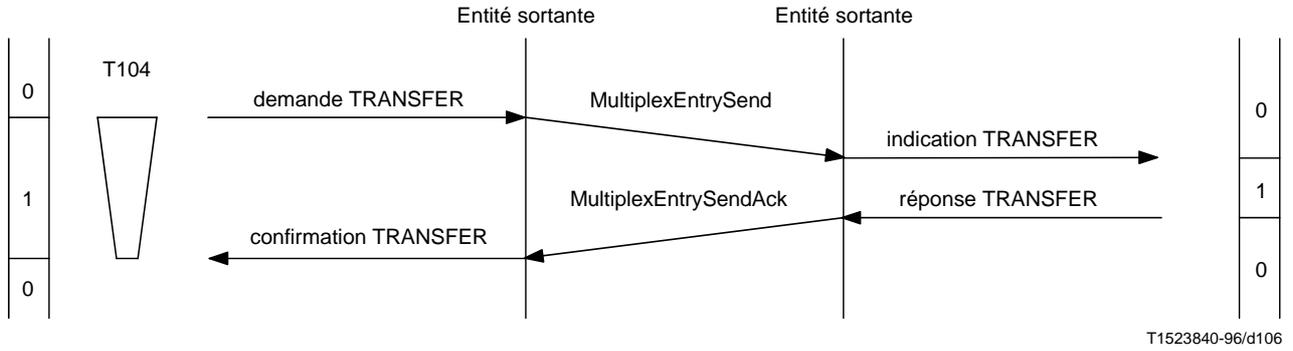


FIGURE II.6-1/H.245

Demande efficace d'envoi de table de multiplexage

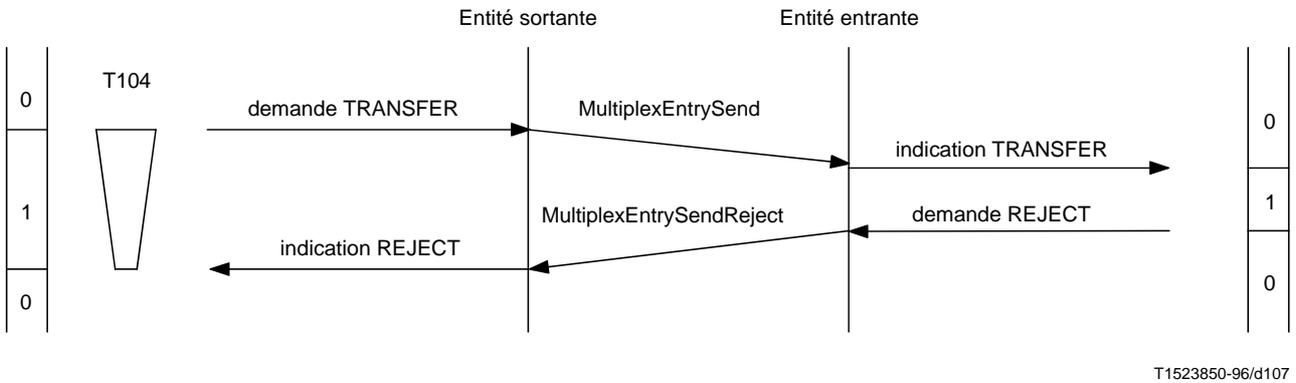


FIGURE II.6-2/H.245

Demande d'envoi de table de multiplexage avec rejet issu de l'utilisateur de l'entité MTSE homologue

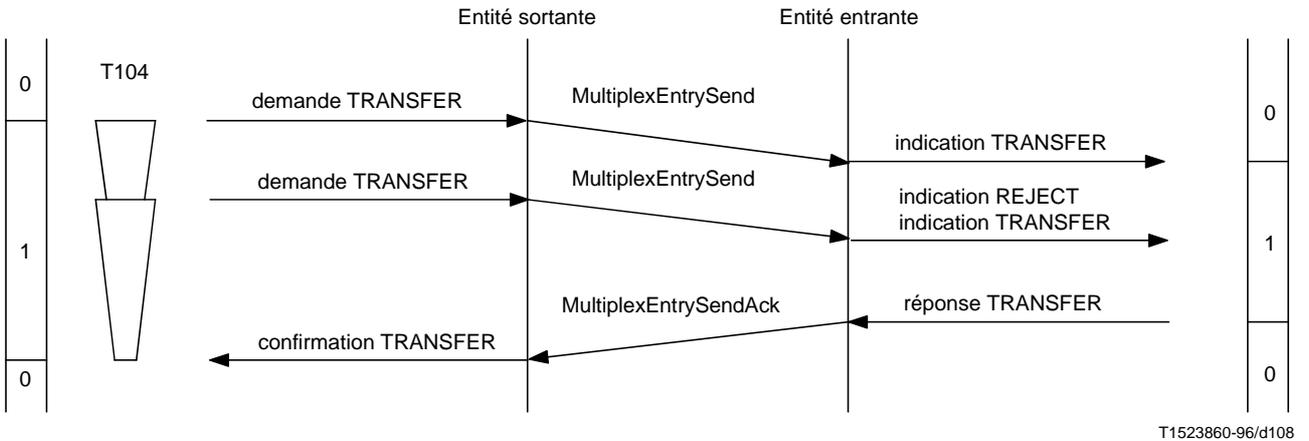
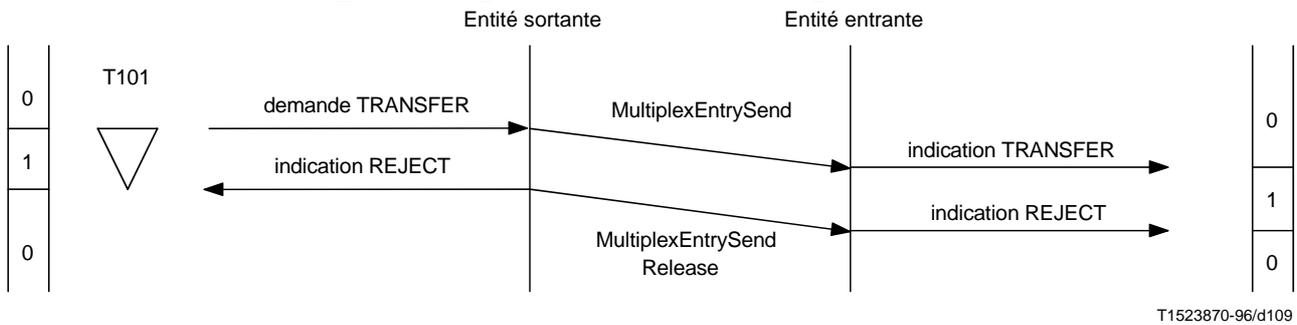


FIGURE II.6-3/H.245

Demande d'envoi de table de multiplexage avec deuxième demande d'envoi de table de multiplexage avant acquittement de la première demande – La première demande n'a pas été suivie d'effet

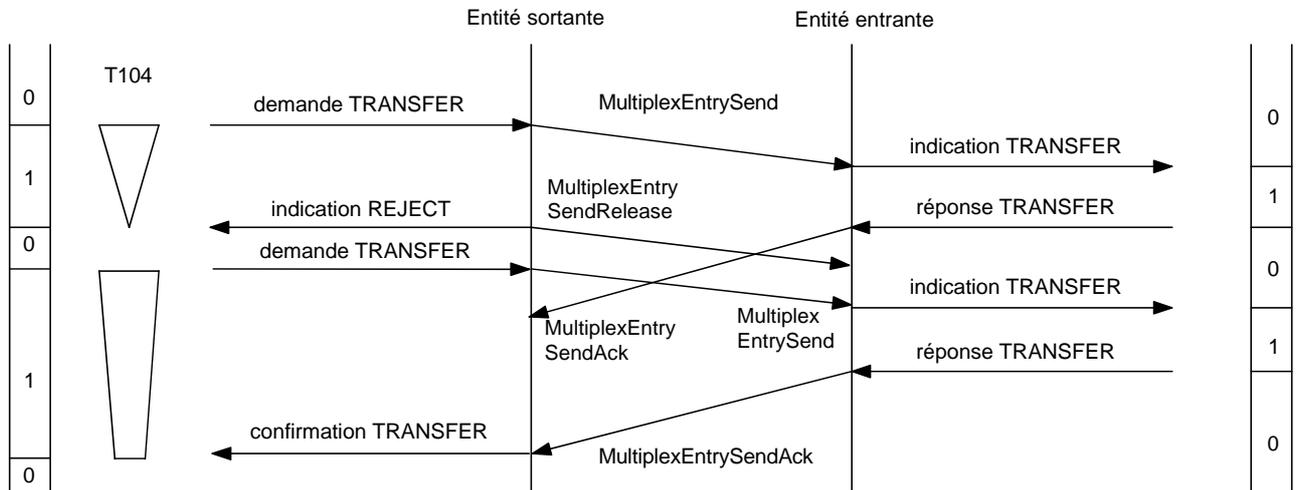
Remplacée par une version plus récente



T1523870-96/d109

FIGURE II.6-4/H.245

Demande d'envoi de table de multiplexage avec expiration du temporisateur T104



T1523880-96/d110

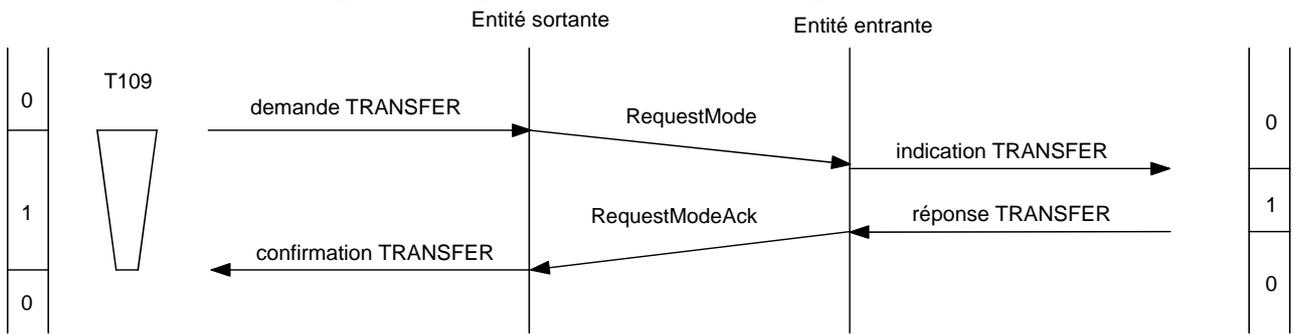
FIGURE II.6-5/H.245

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 à l'entité MTSE sortante – Seule la deuxième demande a été efficace

II.7 Entité de signalisation de demande de mode

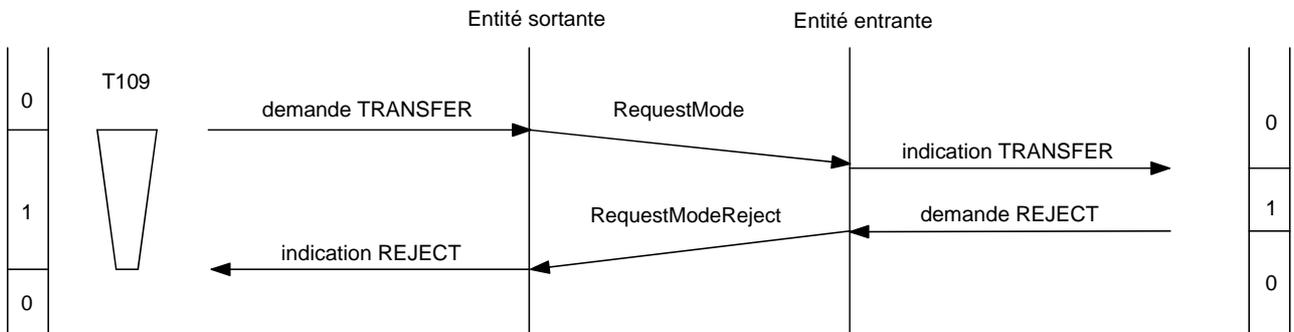
Les Figures II.7-1 à II.7-5 illustrent les échanges entre entités MTSE. Les états IDLE et AWAITING RESPONSE states sont désignés par «0» et «1» respectivement.

Remplacée par une version plus récente



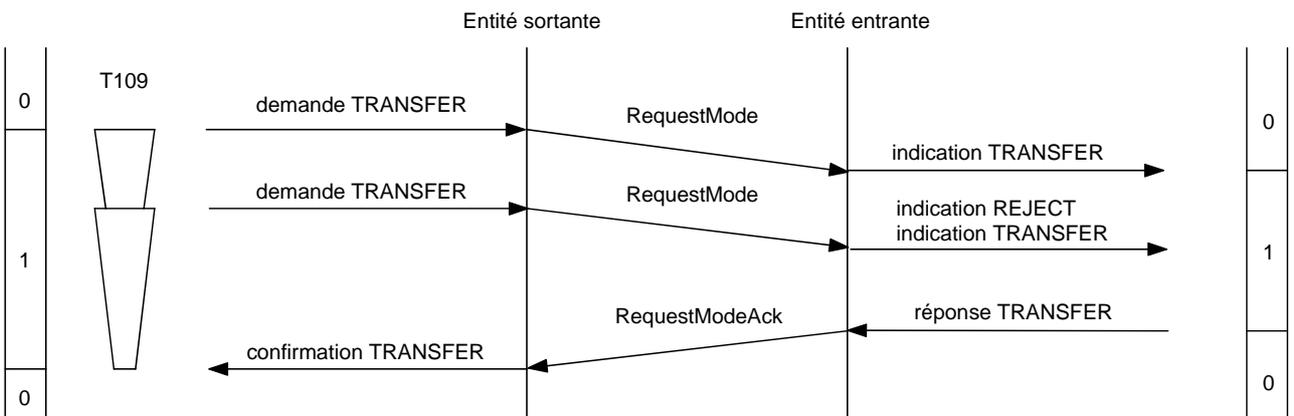
T1523890-96/d111

FIGURE II.7-1/H.245
Demande de mode suivie d'effet



T1523900-96/d112

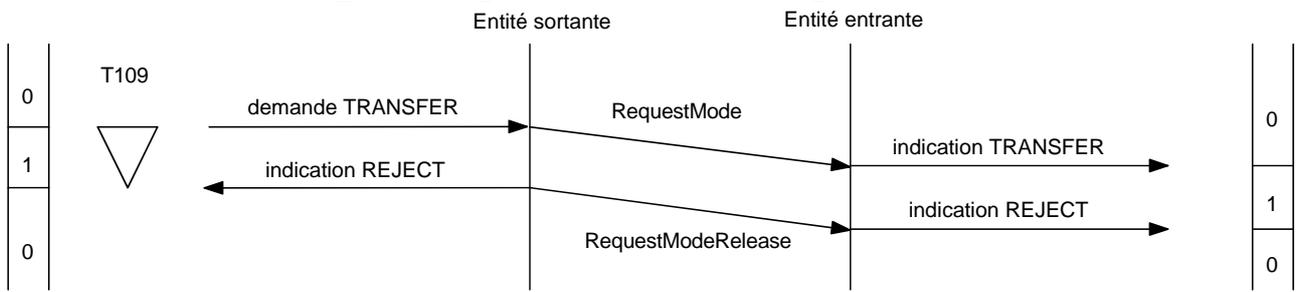
FIGURE II.7-2/H.245
Demande de mode avec rejet issu de l'utilisateur de l'entité MTSE homologue



T1523910-96/d113

FIGURE II.7-3/H.245
Demande de mode avec deuxième demande de mode avant acquittement de la première demande –
La première demande n'a pas été suivie d'effet

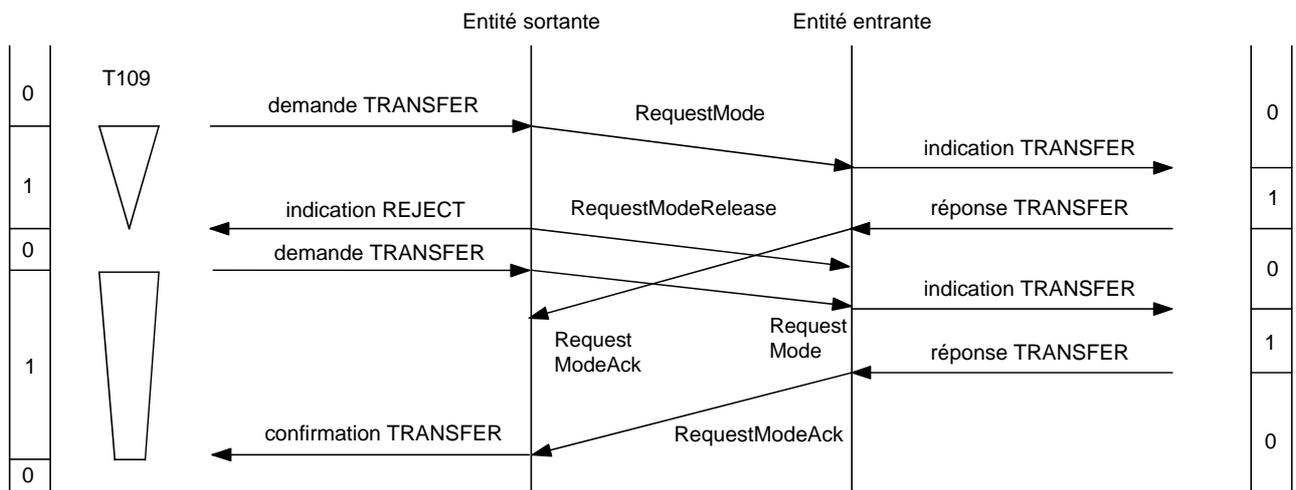
Remplacée par une version plus récente



T1523920-96/d114

FIGURE II.7-4/H.245

Demande de mode non suivie d'effet avec expiration du temporisateur T109



T1523930-96/d115

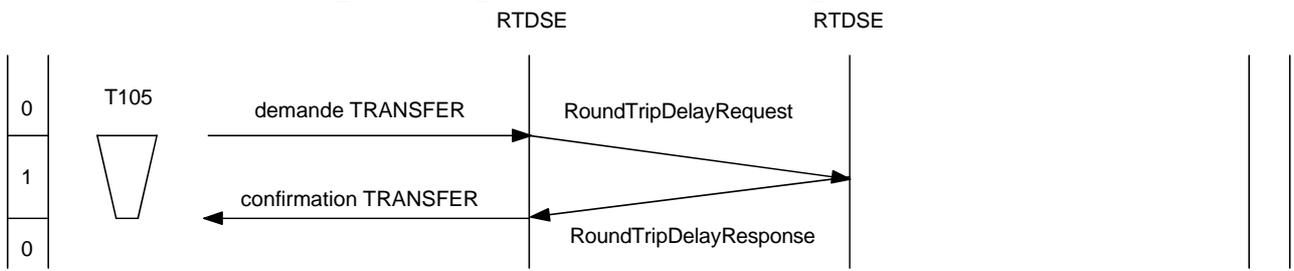
FIGURE II.7-5/H.245

**Demande de mode avec expiration du temporisateur T109, suivie d'une deuxième demande de mode –
Le premier message RequestModeAck est ignoré à l'entité MRSE sortante –
Seule la deuxième demande a été suivie d'effet**

II.8 Entité de signalisation de temps de propagation aller et retour

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».

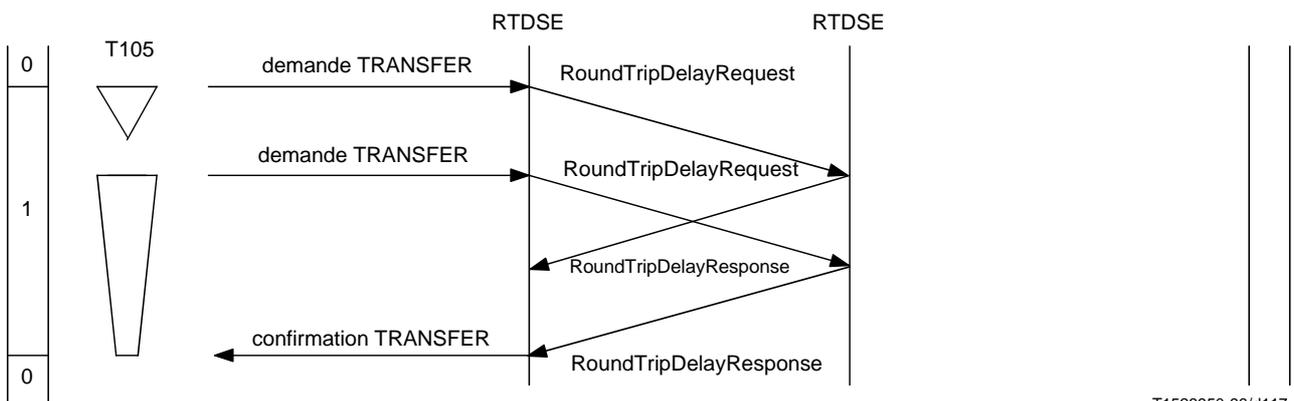
Remplacée par une version plus récente



T1523940-96/d116

FIGURE II.8-1/H.245

Procédure de détermination du temps de propagation aller et retour



T1523950-96/d117

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**

Remplacée par une version plus récente

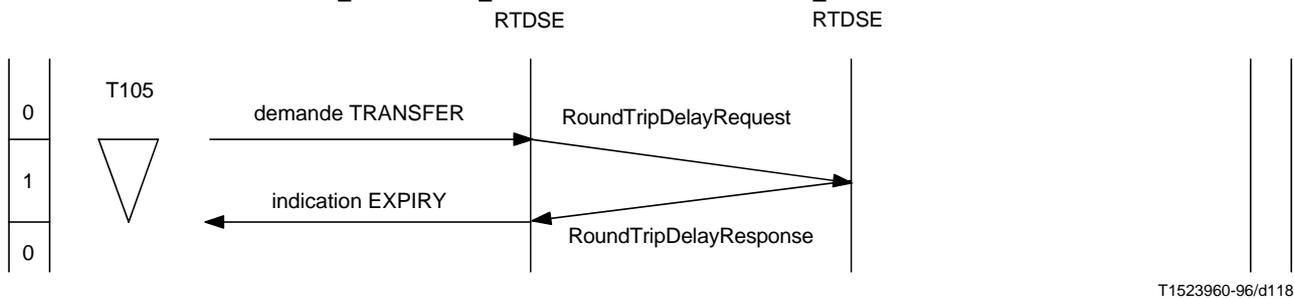


FIGURE II.8-3/H.245

Procédure de détermination du temps de propagation aller et retour avec expiration du temporisateur T105

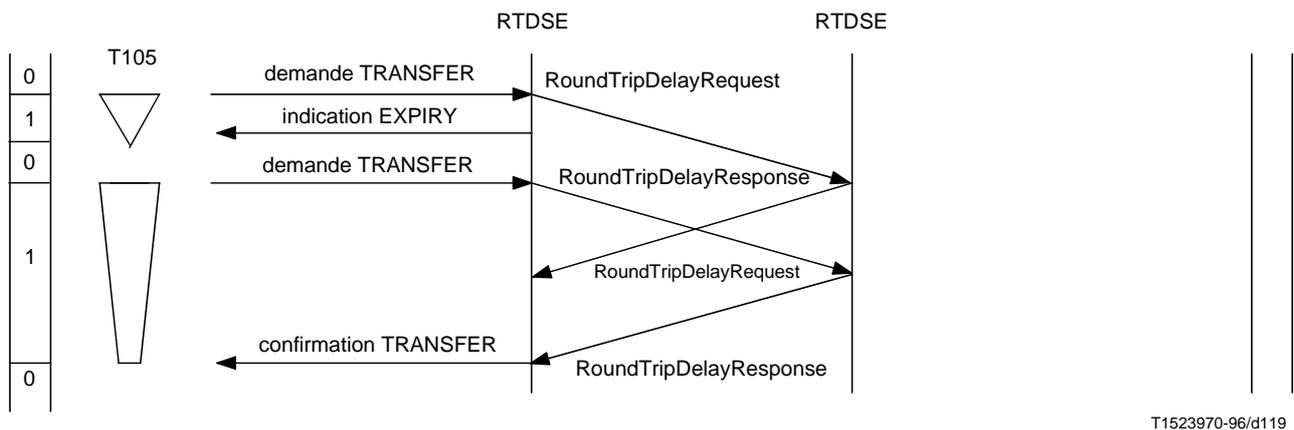


FIGURE II.8-4/H.245

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 RoundTripDelay issu de la première procédure arrive pendant la deuxième procédure et est ignoré

Remplacée par une version plus récente

Appendice III

Résumé des temporisateurs et compteurs de procédure

Le présent appendice énumère les temporisateurs et les compteurs spécifiés dans le paragraphe 8.

III.1 Temporisateurs

Le Tableau III.1-1 montre les temporisateurs spécifiés dans la présente Recommandation.

TABLEAU III.1-1/ H.245

Temporisateurs de procédure

Temporisateur	Procédure	Définition
T106	Détermination maître/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'acquittement ne peut être reçu
T101	Echange 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 unidirectionnelle ou de canal logique bidirectionnel	Ce temporisateur est utilisé dans l'état AWAITING ESTABLISHMENT et AWAITING RELEASE. Il spécifie la durée maximale pendant laquelle aucun message OpenLogicalChannelAck ou OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu
T108	Voie logique SEND	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
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 pendant 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 pendant 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.2-1 montre les compteurs spécifiés dans la présente Recommandation.

TABLEAU III.2-1/ H.245

Compteurs de procédure

Temporisateur	Procédure	Définition
N100	Détermination maître/esclave	Ce compteur spécifie le nombre maximal d'envois de messages MasterSlaveDetermination au cours de l'état OUTGOING AWAITING RESPONSE

Remplacée par une version plus récente

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 Maintenance: 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 Z Langages de programmation