

Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.245

(01/2005)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructure des services audiovisuels – Procédures de
communication

**Protocole de commande pour communications
multimédias**

Recommandation UIT-T H.245

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

CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	H.220–H.229
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.349
Architecture des services d'annuaire pour les services audiovisuels et multimédias	H.350–H.359
Architecture de la qualité de service pour les services audiovisuels et multimédias	H.360–H.369
Services complémentaires en multimédia	H.450–H.499
PROCÉDURES DE MOBILITÉ ET DE COLLABORATION	
Aperçu général de la mobilité et de la collaboration, définitions, protocoles et procédures	H.500–H.509
Mobilité pour les systèmes et services multimédias de la série H	H.510–H.519
Applications et services de collaboration multimédia mobile	H.520–H.529
Sécurité pour les systèmes et services multimédias mobiles	H.530–H.539
Sécurité pour les applications et services de collaboration multimédia mobile	H.540–H.549
Procédures d'interfonctionnement de la mobilité	H.550–H.559
Procédures d'interfonctionnement de collaboration multimédia mobile	H.560–H.569
SERVICES À LARGE BANDE ET MULTIMÉDIAS TRI-SERVICES	
Services multimédias à large bande sur VDSL	H.610–H.619

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

Recommandation UIT-T H.245

Protocole de commande pour communications multimédias

Résumé

La présente Recommandation spécifie la syntaxe et la sémantique des messages d'information des terminaux ainsi que les procédures d'utilisation de ces messages pour les négociations dans la bande en début ou en cours de communication. Ces messages portent sur les capacités d'émission et sur les capacités de réception, sur la préférence de mode du côté réception, sur la signalisation de voie logique et sur les commandes et indications. Des procédures de signalisation avec acquittement sont spécifiées pour assurer la fiabilité des communications audiovisuelles et des transmissions de données.

Par rapport à la version 10 (2003-07) de la Rec. H.245, la présente version inclut les éléments suivants:

- prise en charge du document RFC 3389 – Charge utile du protocole RTP concernant le bruit de confort;
- prise en charge des débits binaires dans un flux à multiples charges utiles;
- définition de la capacité L16;
- résolution des conflits entre voies logiques ouvertes H.245;
- définition d'un débit minimal de paquets audio;
- mise à jour du projet d'Annexe G/H.235: Utilisation des protocoles MIKEY+SRTP dans la Rec. H.235;
- expressions compactes de capacités H.245;
- capacité générique de relais sur IP de N circuits à 64K;
- mise à jour de la résolution des conflits entre voies logiques ouvertes H.245;
- procédure de réinitialisation d'appel dans la Rec. UIT-T H.324.

Les produits déclarés comme étant conformes à la version 11 de la Rec. H.245 doivent satisfaire à toutes les prescriptions obligatoires de la présente Recommandation. Les produits de la version 11 peuvent être identifiés par des messages H.245 TerminalCapabilitySet contenant une valeur protocolIdentifier de {itu-t (0) recommendation (0) h (8) 245 version (0) 11}.

Source

La Recommandation UIT-T H.245 a été approuvée le 8 janvier 2005 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

AVANT-PROPOS

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

L'Assemblée mondiale de normalisation des télécommunications (AMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude 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 1 de l'AMNT.

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

NOTE

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

Le respect de cette Recommandation se fait à titre volontaire. Cependant, il se peut que la Recommandation contienne certaines dispositions obligatoires (pour assurer, par exemple, l'interopérabilité et l'applicabilité) et considère que la Recommandation est respectée lorsque toutes ces dispositions sont observées. Le futur d'obligation et les autres moyens d'expression de l'obligation comme le verbe "devoir" ainsi que leurs formes négatives servent à énoncer des prescriptions. L'utilisation de ces formes ne signifie pas qu'il est obligatoire de respecter la Recommandation.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

© UIT 2005

Tous droits réservés. Aucune partie de cette publication ne peut être reproduite, par quelque procédé que ce soit, sans l'accord écrit préalable de l'UIT.

TABLE DES MATIÈRES

		Page
1	Domaine d'application	1
2	Références.....	1
2.1	Références normatives.....	1
2.2	Référence informative	5
3	Définitions	5
4	Abréviations.....	7
5	Généralités	8
5.1	Désignation maître-esclave	9
5.2	Echange de capacités.....	9
5.3	Procédures de signalisation de voie logique.....	10
5.4	Demande de fermeture de voie logique par un terminal récepteur	10
5.5	Modification d'entrées dans le tableau de multiplexage H.223	10
5.6	Demande de mode audiovisuel et de mode données	10
5.7	Détermination du temps de propagation aller et retour	11
5.8	Boucles de maintenance	11
5.9	Commandes et indications.....	11
	Annexe A – Messages: syntaxe	11
	Annexe B – Messages: définitions sémantiques	79
B.1	Messages liés à la désignation maître-esclave.....	79
B.2	Messages liés aux capacités des terminaux	80
B.3	Messages de signalisation de voie logique.....	114
B.4	Messages de signalisation liés au tableau de multiplexage.....	125
B.5	Messages de signalisation liés à une demande concernant le tableau de multiplexage	126
B.6	Messages liés à la demande de mode	127
B.7	Messages liés au temps de propagation aller et retour	132
B.8	Messages liés à la boucle de maintenance.....	132
B.9	Messages liés au mode de communication.....	133
B.10	Messages de demande et de réponse liés aux conférences.....	134
B.11	Messages liés aux multiliaisons.....	137
B.12	Messages liés à la modification du débit d'une voie logique	139
B.13	Commandes	140
B.14	Indications	147
B.15	Messages génériques	155
	Annexe C – Procédures.....	156
C.1	Introduction	156
C.2	Procédures de désignation maître-esclave.....	158
C.3	Procédures d'échange de capacités	169

	Page	
C.4	Procédures de signalisation de voie logique unidirectionnelle.....	177
C.5	Procédures de signalisation de voie logique bidirectionnelle.....	190
C.6	Procédures de fermeture de voie logique	206
C.7	Procédures relatives au tableau de multiplexage H.223	213
C.8	Procédures relatives à la demande d'entrée de multiplexage.....	224
C.9	Procédures de demande de mode	231
C.10	Procédures liées au temps de propagation aller et retour	241
C.11	Procédures relatives à la boucle de maintenance	246
Annexe D – Affectations d'identificateur d'objet.....		256
Annexe E – Définitions relatives à la capacité ISO/CEI 14496-2		259
Annexe F – Définitions relatives à la capacité de gestion du débit de voie logique.....		262
Annexe G – Définitions relatives à la capacité ISO/CEI 14496-1		263
G.1	Identificateur de capacité.....	263
G.2	Paramètres de capacité utilisés pour les négociations de capacité et pour la signalisation de voie logique	264
G.3	Paramètres de capacité utilisés pour la signalisation de voie logique seulement.....	265
Annexe H – Définition des capacités ISO/CEI 14496-3.....		266
Annexe I – Définition des capacités multidébit adaptatif GSM		271
I.1	Définition de la signalisation de mode et du bourrage de bits pour réaliser l'alignement des octets.....	273
Annexe J – Définitions relatives au codec vocal TDMA ACELP		282
Annexe K – Définitions relatives au codec vocal TDMA US1		283
Annexe L – Définitions relatives au codec vocal CDMA EVRC.....		285
Annexe M – Définitions relatives à l'ISO/CEI 13818-7 et à l'UIT-R BS.1196		287
Annexe N – Rfc 3389 – Charge utile du protocole Rtp concernant le bruit de confort.....		288
Annexe O – Identificateur de capacité L16		288
Annexe P – Capacité de flux audio borné.....		289
Annexe Q – Capacité générique de relais sur IP de N circuits à 64 K.....		290
Q.1	Introduction	290
Q.2	Description	291
Appendice I – Aperçu général de la syntaxe ASN.1.....		295
I.1	Introduction à la syntaxe ASN.1	295
I.2	Types de données de base de la syntaxe ASN.1	295
I.3	Types de structures de données	297
I.4	Type d'identificateur d'objet	298
Appendice II – Exemples de procédures H.245.....		299
II.1	Introduction	299
II.2	Entité de signalisation de désignation maître-esclave.....	299

	Page
II.3 Entité de signalisation d'échange de capacités (CESE).....	303
II.4 Entité de signalisation de voie logique (LCSE)	305
II.5 Entité de signalisation de fermeture de voie logique (CLCSE)	307
II.6 Entité de signalisation du tableau de multiplexage (MTSE).....	308
II.7 Entité de signalisation de demande de mode (MRSE, <i>mode request signalling entity</i>).....	310
II.8 Entité de signalisation de temps de propagation aller et retour (RTDSE)	312
II.9 Entité de signalisation de voie logique bidirectionnelle (B-LCSE)	313
Appendice III – Récapitulatif des temporisateurs et des compteurs utilisés dans les procédures.....	315
III.1 Temporisateurs	315
III.2 Compteurs.....	316
Appendice IV – Extensions à la Recommandation H.245.....	317
Appendice V – Procédure remplacementFor	318
Appendice VI – Exemples de paramétrage des capacités H.263.....	320
VI.1 Exemples de paramétrage des couches d'amélioration dans le mode H.245..	320
VI.2 Exemples de paramétrage de la voie vidéo inverse en mode H.245	321
Appendice VII – Procédure et gabarit pour la définition de nouvelles capacités à l'aide des capacités génériques H.245	325
VII.1 Procédure.....	326
VII.2 Gabarit	327
VII.3 Exemple de gabarit – H.261	328
Appendice VIII – Liste des capacités et messages génériques définis dans d'autres Recommandations/Normes que la présente Recommandation.....	329
Appendice IX – Utilisation de la syntaxe ASN.1 dans la présente Recommandation.....	332
IX.1 Etiquetage	332
IX.2 Types	332
IX.3 Contraintes et gammes	332
IX.4 Extensibilité.....	332
Appendice X – Résolution des scénarios conflictuels dans les voies unidirectionnelles et bidirectionnelles.....	332
X.1 Les deux terminaux utilisent des messages bidirectionnels d'ouverture OLC	332
X.2 Le terminal maître propose des messages bidirectionnels d'ouverture OLC et le terminal esclave propose des messages unidirectionnels d'ouverture OLC	333
X.3 Le terminal maître propose des messages unidirectionnels d'ouverture OLC et le terminal esclave propose des messages bidirectionnels d'ouverture OLC	334

	Page
X.4 Le terminal maître propose des messages bidirectionnels d'ouverture OLC avec type nullData et le terminal esclave propose des messages unidirectionnels d'ouverture OLC	335
X.5 Les deux terminaux proposent des messages bidirectionnels d'ouverture OLC avec type nullData	336

Recommandation UIT-T H.245

Protocole de commande pour communications multimédias

1 Domaine d'application

La présente Recommandation spécifie la syntaxe et la sémantique des messages d'information des terminaux ainsi que les procédures d'utilisation de ces messages pour les négociations dans la bande en début ou en cours de communication. Les messages portent sur les capacités d'émission et sur les capacités de réception, sur la préférence de mode du côté réception, sur la signalisation de voie logique et sur les commandes et indications. Des procédures de signalisation avec acquittement sont spécifiées pour assurer la fiabilité 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. Elle s'applique mais n'est pas limitée aux systèmes multimédias qui utilisent les multiplex définis dans les Recommandations UIT-T H.222.0, H.223 et H.225.0. 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 un acheminement correct des données.

NOTE – Il convient cependant de ne pas faire de confusion avec le système de gestion défini dans la Rec. UIT-T 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 Rec. UIT-T H.245 et le flux de données décrit dans la Rec. UIT-T T.120 sont complémentaires.

2 Références

2.1 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée. La référence à un document dans le cadre de la présente Recommandation ne lui donne pas, en tant que document autonome, le statut d'une Recommandation.

- [1] Recommandation UIT-T E.164 (1997), *Plan de numérotage des télécommunications publiques internationales*.
- [2] Recommandation UIT-T G.711 (1988), *Modulation par impulsions et codage (MIC) des fréquences vocales*.
- [3] Recommandation UIT-T G.722 (1988), *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s*.
- [4] Recommandation UIT-T G.723.1 (1996), *Codeurs vocaux: codeur vocal à double débit pour communications multimédias acheminées à 5,3 kbit/s et à 6,3 kbit/s*.
- [5] Recommandation UIT-T G.728 (1992), *Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code*.

- [6] Recommandation UIT-T G.729 (1996), *Codage de la parole à 8 kbit/s par prédiction linéaire avec excitation par séquences codées à structure algébrique conjuguée.*
- [7] Recommandation UIT-T H.221 (1999), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels.*
- [8] Recommandation UIT-T H.222.0 (2000) | ISO/CEI 13818-1:2000, *Technologies de l'information – Codage générique des images animées et du son associé: systèmes.*
- [9] Recommandation UIT-T H.222.1 (1996), *Multiplexage et synchronisation multimédias des communications audiovisuelles en environnement ATM.*
- [10] Recommandation UIT-T H.223 (1996), *Protocole de multiplexage pour communications multimédias à faible débit.*
- [11] Recommandation UIT-T H.224 (2000), *Protocole de commande en temps réel pour les applications simplex utilisant les canaux de données à faible vitesse/à grande vitesse/et à protocole multicouche définis dans la Recommandation H.221.*
- [12] Recommandation UIT-T H.225.0 (2003), *Protocoles de signalisation d'appel et paquets des flux monomédias pour les systèmes de communication multimédias en mode paquet.*
- [13] Recommandation UIT-T H.230 (1999), *Signaux de commande et d'indication synchrones de la trame pour les systèmes audiovisuels.*
- [14] Recommandation UIT-T H.233 (1995), *Système de confidentialité pour les services audiovisuels.*
- [15] Recommandation UIT-T H.234 (1994), *Système de gestion de clés de chiffrement et d'authentification pour les services audiovisuels.*
- [16] Recommandation UIT-T H.235 (2003), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245), plus Amendement 1 (2004).*
- [17] Recommandation UIT-T H.243 (2000), *Procédures pour l'établissement de communications entre trois terminaux audiovisuels ou plus sur des canaux numériques d'un débit allant jusqu'à 1920 kbit/s.*
- [18] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s.*
- [19] Recommandation UIT-T H.262 (2000) | ISO/CEI 13818-2:2000, *Technologies de l'information – Codage générique des images animées et du son associé: données vidéo.*
- [20] Recommandation UIT-T H.263 (1998), *Codage vidéo pour communications à faible débit.*
- [21] Recommandation UIT-T H.310 (1998), *Systèmes et terminaux de communication audiovisuels à large bande.*
- [22] Recommandation UIT-T H.320 (1999), *Systèmes et équipements terminaux visiophoniques à bande étroite.*
- [23] Recommandation UIT-T H.323 (2003), *Systèmes de communication multimédia en mode paquet.*
- [24] Recommandation UIT-T H.324 (2002), *Terminal pour communications multimédias à faible débit.*
- [25] Recommandations UIT-T de la série I.363.x, *Spécification de la couche d'adaptation du mode de transfert asynchrone du RNIS à large bande.*

- [26] 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.*
- [27] Recommandation UIT-T T.30 (2003), *Procédures pour la transmission de documents par télécopie sur le réseau téléphonique général commuté.*
- [28] Recommandation UIT-T T.35 (2000), *Procédure d'attribution des codes définis par l'UIT-T pour les facilités non normalisées.*
- [29] Recommandation UIT-T T.38 (2003), *Procédures de communication de télécopie du Groupe 3 en temps réel sur les réseaux à protocole Internet.*
- [30] Recommandation UIT-T T.51 (1992), *Jeux de caractères latins codés pour services de télématique.*
- [31] Recommandation UIT-T T.84 (1996) | ISO/CEI 10918-3:1997, *Technologies de l'information – Compression et codage numériques des images fixes à modelé continu: extensions.*
- [32] Recommandation UIT-T T.120 (1996), *Protocoles de données pour conférence multimédia.*
- [33] Recommandation UIT-T T.123 (1999), *Piles de protocoles de données propres au réseau pour conférences multimédias.*
- [34] Recommandation UIT-T T.140 (1998), *Protocole de conversation en mode texte pour application multimédia.*
- [35] Recommandation UIT-T T.434 (1999), *Format de transfert de fichiers binaires pour les services télématiques.*
- [36] Recommandation UIT-T V.14 (1993), *Transmission de caractères arithmiques sur des voies supports synchrones.*
- [37] Recommandation UIT-T V.34 (1998), *Modem fonctionnant à des débits allant jusqu'à 33 600 bit/s pour usage sur le réseau téléphonique général commuté et sur les circuits loués point à point à 2 fils de type téléphonique.*
- [38] Recommandation UIT-T V.42 (2002), *Procédures de correction d'erreur pour les équipements de terminaison de circuits de données utilisant la conversion asynchrone/synchrone.*
- [39] Recommandation UIT-T V.140 (1998), *Procédures d'établissement d'une communication entre deux terminaux audiovisuels multiprotocoles sur des canaux numériques à un débit multiple de 64 ou 56 kbit/s.*
- [40] Recommandation UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- [41] Recommandation UIT-T X.681 (2002) | ISO/CEI 8824-2:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un – Spécification des objets informationnels.*
- [42] Recommandation UIT-T X.691 (2002) | ISO/CEI 8825-2:2002: *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact.*
- [43] ISO/CEI 13239:2002, *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Procédures de commande de liaisons de données à haut niveau (HDLC).*

- [44] 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.*
- [45] ISO/CEI 11172-3:1993, *Technologies de l'information – Codage de l'image animée et du son associé pour les supports de stockage numérique jusqu'à environ 1,5 Mbit/s – Partie 3: Audio.*
- [46] ISO/CEI 13818-3:1998, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 3: Son.*
- [47] ISO/CEI 13818-6:1998, *Technologies de l'information – Codage générique des images animées et des informations sonores associées – Partie 6: Extensions pour DSM-CC.*
- [48] ISO/CEI 14496-1:1999, *Technologies de l'information – Codage des objets audiovisuels – Partie 1: Systèmes.*
- [49] ISO/CEI 14496-2:1999, *Technologies de l'information – Codage des objets audiovisuels – Partie 2: Codage visuel.*
- [50] ISO/CEI 14496-3:1999, *Technologies de l'information – Codage des objets audiovisuels – Partie 3: Codage audio.*
- [51] ISO/CEI 14496-3/Amd.1:2000, *Technologies de l'information – codage des objets audiovisuels – Partie 3: Codage audio. Amendement 1: Extensions du codage audio.*
- [52] ISO/CEI TR 9577:1999, *Technologies de l'information – Identification des protocoles dans la couche réseau.*
- [53] ETSI ETS 300 961 (GSM 06.10), *Full rate speech transcoding.*
- [54] ETSI ETS 300 969 (GSM 06.20), *Half rate speech transcoding.*
- [55] ETSI ETS 300 726 (GSM 06.60), *Enhanced Full Rate (EFR) speech transcoding.*
- [56] ETSI ETS 300 963 (GSM 06.12), *Comfort noise aspect for full rate speech traffic channels.*
- [57] ETSI ETS 300 971 (GSM 06.22), *Comfort noise aspects for half rate speech traffic channels.*
- [58] ETSI ETS 300 728 (GSM 06.62), *Comfort noise aspects for Enhanced Full Rate (EFR) speech traffic channels.*
- [59] ETSI ETS 300 964 (GSM 06.31), *Discontinuous Transmission (DTX) for full rate speech traffic channels.*
- [60] ETSI ETS 300 972 (GSM 06.41), *Discontinuous transmission (DTX) for half rate speech traffic channels.*
- [61] ETSI ETS 300 729 (GSM 06.81), *Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels.*
- [62] ETSI ETS 300 962 (GSM 06.11), *Substitution and muting of lost frames for full rate speech traffic channels.*
- [63] ETSI ETS 300 970 (GSM 06.21), *Substitution and muting of lost frames for half rate speech traffic channels.*
- [64] ETSI ETS 300 727 (GSM 06.61), *Substitution and muting of lost frames for Enhanced Full Rate (EFR) speech traffic channels.*
- [65] ETSI ETS 300 965 (GSM 06.32), *Voice Activity Detector (VAD) for full rate speech traffic channels.*

- [66] ETSI ETS 300 973 (GSM 06.42), *Voice Activity Detector (VAD) for half rate speech traffic channels.*
- [67] ETSI ETS 300 730 (GSM 06.82), *Voice activity detection for enhanced full rate speech traffic channels.*
- [68] ETSI ETS 300 724 (GSM 06.53), *ANSI-C code for the GSM Enhanced Full Rate Speech (EFR) speech codec.*
- [69] ETSI EN 301 712 (GSM 06.73), *ANSI-C code for the AMR speech codec.*
- [70] ETSI EN 301 704 (GSM 06.90), *Adaptive Multi-Rate (AMR) speech transcoding.*
- [71] ETSI EN 301 705 (GSM 06.91), *Substitution and muting of lost frames for Adaptive Multi Rate (AMR) speech traffic channels.*
- [72] ETSI EN 301 706 (GSM 06.92), *Comfort noise aspects for Adaptive Multi-Rate (AMR) speech traffic channels.*
- [73] ETSI EN 301 708 (GSM 06.94), *Voice Activity Detection (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels.*
- [74] RCR STD-27H, *Personal Digital Cellular Telecommunication System RCR Standard.*
- [75] TIA/EIA – 136-Rev.A, Part 410, *TDMA Cellular/PCS – Radio Interface, Enhanced Full Rate Voice Codec (ACELP). Formerly IS-641. TIA published standard, 1998.*
- [76] TIA/EIA/IS 641-A (1998), *TDMA Cellular/PCS – Radio Interface, US1 Full Rate Voice Codec.*
- [77] Recommandation UIT-T H.239 (2003), *Gestion des rôles et canaux de média additionnels pour les terminaux de la série H.300.*
- [78] Recommandation UIT-T H.241 (2003), *Procédures vidéo et signaux de commande élargis pour les terminaux de la série H.300.*
- [79] Recommandation UIT-T H.235, Annexe G (2005), *Utilisation du protocole de gestion de clés MIKEY avec le protocole de transport sécurisé en temps réel (SRTP) dans les systèmes H.235.*
- [80] Recommandation UIT-T Y.1413 (2004), *Interfonctionnement des réseaux TDM et MPLS – Interfonctionnement dans le plan utilisateur.*
- [81] IETF RFC 2198 (1997), *RTP Payload for Redundant Audio Data.*
- [82] IETF RFC 2733 (1999), *An RTP Payload Format for Forward Error Correction.*
- [83] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications.*
- [84] IETF RFC 3711 (2004), *The secure Real-time Transport Protocol.*

2.2 Référence informative

- [PWE3] BRYANT (S.), PATE (P.): PWE3 Architecture, <draft-ietf-pwe3-arch-07.txt>, mars 2003.

3 Définitions

La présente Recommandation définit les termes suivants:

3.1 voie logique bidirectionnelle: paire de voies logiques associées entre deux terminaux, une dans chaque sens de transmission.

3.2 capacité: un terminal dispose d'une capacité donnée s'il peut coder et émettre ou recevoir et décoder un signal particulier.

- 3.3 canal:** liaison unidirectionnelle entre deux extrémités.
- 3.4 commande:** message nécessitant une action mais pas de réponse explicite.
- 3.5 flux élémentaire:** terme générique pour un flux binaire codé vidéo, audio ou autre.
- 3.6 entrée:** référence à des éléments d'ensembles ou de tableaux, tels que des ensembles de capacités et des tableaux de multiplexage.
- 3.7 sens direct:** sens de transmission allant du terminal émettant la demande de voie logique bidirectionnelle vers l'autre terminal.
- 3.8 dans la bande:** les messages dans la bande sont ceux qui sont transportés dans le canal ou dans la voie logique auquel ou à laquelle ils se rapportent.
- 3.9 entrant:** une entité de signalisation entrante ne peut pas lancer de procédure mais peut répondre aux messages émanant d'une entité de signalisation distante et aux primitives de son propre utilisateur.
- 3.10 indication:** message contenant des informations mais n'appelant ni action ou ni réponse.
- 3.11 voie logique:** trajet unidirectionnel ou bidirectionnel pour la transmission d'informations.
- 3.12 numéro de voie logique:** numéro identifiant une voie logique unique.
- 3.13 signalisation de voie logique:** ensemble des procédures utilisées pour ouvrir et fermer des voies logiques.
- 3.14 terminal maître:** terminal désigné comme étant le terminal maître par la procédure de désignation maître-esclave définie dans la présente Recommandation, ou par toute autre procédure.
- 3.15 type de média:** forme simple d'information présentée à un utilisateur, ou données représentant cette information: vidéo, audio et texte sont des exemples de type de média.
- 3.16 mode:** ensemble de flux élémentaires qu'un terminal émet, vise à émettre ou voudrait recevoir.
- 3.17 communication multimédia:** émission ou réception de signaux correspondant à au moins deux types de médias simultanément.
- 3.18 hors norme:** non conforme à une norme nationale ou internationale à laquelle il est fait référence dans la présente Recommandation.
- 3.19 sortant:** une entité de signalisation sortante est une entité qui lance une procédure.
- 3.20 multipoint:** interconnexion simultanée d'au moins trois terminaux pour permettre les communications entre plusieurs sites par l'utilisation de ponts de conférence qui orientent de manière centralisée les flux d'information.
- 3.21 demande:** message qui entraîne une action du terminal distant et appelle une réponse immédiate de celui-ci.
- 3.22 réponse:** message apportant la réplique à une demande.
- 3.23 sens inverse:** sens de transmission allant du terminal recevant une demande de voie logique bidirectionnelle vers le terminal ayant émis la demande.
- 3.24 session:** période de communication entre deux terminaux, pouvant être de type conversationnel ou non conversationnel (par exemple consultation d'une base de données).
- 3.25 terminal asservi:** terminal désigné comme étant le terminal asservi par la procédure de désignation maître-esclave définie dans la présente Recommandation, ou par toute autre procédure.
- 3.26 prise en charge d'un mode:** capacité à fonctionner dans un mode donné; cependant une prescription de prise en charge d'un mode donné ne signifie pas que ce mode doit être

effectivement utilisé tout le temps; à moins d'être interdits, d'autres modes peuvent être utilisés par accord mutuel.

3.27 terminal: point d'extrémité quelconque tel qu'un terminal d'utilisateur ou tout autre système de communication comme un pont de conférence ou un serveur d'informations.

3.28 identificateur TSAP: élément d'information utilisé pour multiplexer plusieurs liaisons de transport du même type sur une même entité H.323, toutes les liaisons de transport partageant la même adresse de réseau local (par exemple le numéro de port dans un environnement TCP/UDP/IP). Des identificateurs TSAP peuvent être (pré)attribués par une instance internationale ou être attribués dynamiquement pendant l'établissement d'un appel. Les identificateurs TSAP attribués dynamiquement sont de nature transitoire, c'est-à-dire que leurs valeurs s'appliquent uniquement pendant la durée d'une communication unique.

3.29 voie logique unidirectionnelle: trajet pour la transmission d'un flux élémentaire unique entre deux terminaux.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

AAL	couche d'adaptation ATM (<i>ATM adaptation layer</i>)
AL1,2,3	couches d'adaptation 1, 2 et 3 H.223 (<i>H.223 adaptation layers 1, 2 and 3</i>)
ASN.1	notation de syntaxe abstraite numéro un (<i>abstract syntax notation one</i>)
ATM	mode de transfert asynchrone (<i>asynchronous transfer mode</i>)
B-LCSE	entité de signalisation de voie logique bidirectionnelle (<i>bidirectional logical channel signalling entity</i>)
CESE	entité de signalisation d'échange de capacités (<i>capability exchange signalling entity</i>)
CIF	format intermédiaire commun (d'une image vidéo: voir les Recommandations UIT-T H.261 et H.263) (<i>common intermediate format</i>)
CLCSE	entité de signalisation de fermeture de voie logique (<i>close logical channel signalling entity</i>)
CPCS	sous-couche de convergence de partie commune (de la couche d'adaptation ATM 5) (<i>common part convergence sublayer</i>)
DSM-CC	moyens d'enregistrement numérique/commandes et indications (<i>digital storage media/command and control</i>)
DTMF	multifréquence bitonalité (<i>dual tone multi-frequency</i>)
GOB	groupe de blocs (d'une image vidéo: voir les Recommandations UIT-T 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 (voir les Recommandations UIT-T H.261 et H.263) (<i>hypothetical reference decoder</i>)
IV	vecteur d'initialisation (utilisé pour le chiffrement: voir les Recommandations UIT-T H.233 et H.234) (<i>initialization vector</i>)
LAPM	protocole d'accès de liaison pour modems (<i>link access protocol for modems</i>)
LCSE	entité de signalisation de voie logique (<i>logical channel signalling entity</i>)
MC	entité de commande multipoint H.323 (<i>H.323 multipoint control entity</i>)

MCU	pont de conférence (<i>multipoint control unit</i>)
MLSE	entité de signalisation de boucle de maintenance (<i>maintenance loop signalling entity</i>)
MPI	intervalle d'image minimal (<i>minimum picture interval</i>)
MRSE	entité de signalisation de demande de mode (<i>mode request signalling entity</i>)
MSDSE	entité de signalisation pour la désignation maître-esclave (<i>master-slave determination signalling entity</i>)
MTSE	entité de signalisation de tableau de multiplexage (<i>multiplex table signalling entity</i>)
PCR	référence d'horloge programme (voir la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>program clock reference</i>)
PID	identificateur de paquet (voir la Rec. UIT-T H.222.0 ISO/CEI 13818-1) (<i>packet identifier</i>)
QCIF	format quart de CIF (<i>quarter CIF</i>)
RMESE	entité de signalisation de demande d'entrée de multiplexage (<i>request multiplex entry signalling entity</i>)
RTCP	protocole de commande de transport en temps réel (<i>real-time transport control protocol</i>)
RTDSE	entité de signalisation de temps de propagation aller et retour (<i>round-trip delay signalling entity</i>)
RTGC	réseau téléphonique général commuté
RTP	protocole de transport en temps réel (<i>real-time transport protocol</i>)
SDL	langage de description et de spécification (<i>specification and description language</i>)
SDU	unité de données de service (<i>service data unit</i>)
SE	message d'échange de session (utilisé pour le chiffrement: voir les Recommandations UIT-T 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 (voir 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 déterminer quel terminal sera le terminal maître et quel terminal sera le terminal asservi à des fins de gestion des voies logiques, pour transporter différents signaux de commande et d'indication; pour gérer le débit de différentes voies logiques et de l'ensemble du multiplex, et pour mesurer le temps de propagation aller et retour, d'un terminal vers l'autre et dans l'autre sens. Les procédures sont expliquées de manière plus détaillée ci-dessous.

Après cette introduction générale, on détaille la syntaxe et la sémantique des messages, ainsi que les procédures. La syntaxe a été définie au moyen de la notation ASN.1 [40] 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 sur les procédures définit les protocoles utilisant les messages définis dans les autres paragraphes.

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

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

5.1 Désignation maître-esclave

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

La procédure de désignation maître-esclave permet aux terminaux engagés dans une communication de déterminer quel est le terminal maître et quel est le terminal asservi. Le statut du terminal peut être redéfini à tout moment pendant une communication; toutefois, un terminal ne peut lancer le processus de désignation maître-esclave que si aucune procédure dépendant de son résultat n'est active au niveau local.

5.2 Echange de capacités

Les procédures d'échange de capacités sont prévues pour assurer que les seuls signaux multimédias devant être émis 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 ignorées, ce qui ne doit pas être considéré comme une anomalie. Lorsqu'une capacité dotée d'extensions non comprises par le terminal est reçue, elle doit être acceptée comme si elle ne comportait pas d'extensions.

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

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

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

Ces ensembles de capacités sont conçus pour permettre l'envoi simultané de plusieurs flux contenant un type de média donné. Par exemple, un terminal peut déclarer qu'il a la capacité de recevoir (ou d'envoyer) en même temps deux flux vidéo H.262 indépendants et deux flux audio G.722 indépendants. Des messages de capacités ont été définis pour permettre à un terminal d'indiquer qu'il n'a pas de capacités fixes, mais que ses capacités dépendent des autres modes qui

sont utilisés simultanément. Il est par exemple possible d'indiquer que la vidéo à plus haute résolution peut être décodée quand un algorithme audio plus simple est utilisé; ou que deux séquences vidéo à faible résolution ou une seule séquence unique à haute résolution peuvent être décodées. Il est également possible d'indiquer des compromis entre les capacités d'émission et les capacités de réception.

Des capacités hors norme et des messages de commande peuvent être émis au moyen de la structure NonStandardParameter. Il convient de noter que, bien que la signification des messages hors norme soit définie par des organismes différents, les équipements construits par n'importe quel fabricant peuvent signaler tout message hors norme, pour autant que sa signification soit connue.

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

5.3 Procédures de signalisation de voie logique

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

Une partie de ce protocole concerne l'ouverture de voies bidirectionnelles. Afin d'éviter les incidents qui pourraient survenir au lancement simultané d'événements similaires par deux terminaux, un terminal est défini comme le terminal maître, et l'autre comme le terminal asservi. Un protocole est défini afin de déterminer quel est le terminal maître et quel est le terminal asservi. Cependant, des systèmes utilisant la présente Recommandation peuvent spécifier la procédure indiquée dans la présente Recommandation ou un autre moyen pour déterminer quel sera le terminal maître et quel sera le terminal asservi.

5.4 Demande de fermeture de voie logique par un terminal récepteur

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

5.5 Modification d'entrées dans le tableau de multiplexage H.223

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

5.6 Demande de mode audiovisuel et de mode données

Lorsque le protocole d'échange de capacités se termine, chacun des deux terminaux a connaissance des capacités de l'autre terminal à émettre et recevoir comme cela est spécifié dans les descripteurs

de capacités qui ont été échangés. Un terminal ne doit pas obligatoirement déclarer toutes ses capacités; il ne doit déclarer que celles qu'il vise à utiliser.

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

5.7 Détermination du temps de propagation aller et retour

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

5.8 Boucles de maintenance

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

5.9 Commandes et indications

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

Une commande est définie pour permettre la gestion du débit des voies logiques et du multiplex à partir du terminal distant, notamment dans les cas suivants: interfonctionnement avec des terminaux utilisant des multiplex dans lesquels les débits ne sont disponibles qu'en nombre fini, applications multipoint dans lesquelles les débits provenant de différentes sources doivent être adaptés et commande de débit dans les réseaux encombrés.

Annexe A

Messages: syntaxe

La présente annexe spécifie la syntaxe des messages au moyen de la notation ASN.1 [40]. Pour le codage des messages à l'émission, il faut appliquer les règles de codage compact spécifiées en [42] et utiliser la variante de base alignée. Le premier bit de chaque octet qui est transmis est le bit le plus significatif de l'octet comme cela est spécifié dans la Rec. UIT-T X.691 | ISO/CEI 8825-2.

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

```
-- Exportation de tous les symboles
```

```
-- =====
-- Messages de haut niveau
-- =====
```

```
MultimediaSystemControlMessage ::= CHOICE
```

```

{
    request           RequestMessage,
    response          ResponseMessage,
    command           CommandMessage,
    indication        IndicationMessage,
    ...
}

-- Un RequestMessage donne lieu à une action et demande une réponse immédiate

RequestMessage ::= CHOICE
{
    nonStandard      NonStandardMessage,

    masterSlaveDetermination  MasterSlaveDetermination,

    terminalCapabilitySet      TerminalCapabilitySet,

    openLogicalChannel
    closeLogicalChannel       OpenLogicalChannel,
                              CloseLogicalChannel,

    requestChannelClose       RequestChannelClose,

    multiplexEntrySend        MultiplexEntrySend,

    requestMultiplexEntry     RequestMultiplexEntry,

    requestMode               RequestMode,

    roundTripDelayRequest     RoundTripDelayRequest,

    maintenanceLoopRequest    MaintenanceLoopRequest,

    ...,
    communicationModeRequest  CommunicationModeRequest,
    conferenceRequest          ConferenceRequest,

    multilinkRequest          MultilinkRequest,
    logicalChannelRateRequest LogicalChannelRateRequest,
    genericRequest            GenericMessage
}

-- Un ResponseMessage est la réponse à un RequestMessage

ResponseMessage ::= CHOICE
{
    nonStandard      NonStandardMessage,

    masterSlaveDeterminationAck  MasterSlaveDeterminationAck,
    masterSlaveDeterminationReject MasterSlaveDeterminationReject,

    terminalCapabilitySetAck      TerminalCapabilitySetAck,
    terminalCapabilitySetReject   TerminalCapabilitySetReject,

    openLogicalChannelAck        OpenLogicalChannelAck,
    openLogicalChannelReject     OpenLogicalChannelReject,
    closeLogicalChannelAck       CloseLogicalChannelAck,

    requestChannelCloseAck       RequestChannelCloseAck,
    requestChannelCloseReject    RequestChannelCloseReject,

    multiplexEntrySendAck        MultiplexEntrySendAck,
    multiplexEntrySendReject     MultiplexEntrySendReject,
}

```

```

requestMultiplexEntryAck      RequestMultiplexEntryAck,
requestMultiplexEntryReject   RequestMultiplexEntryReject,

requestModeAck                RequestModeAck,
requestModeReject             RequestModeReject,

roundTripDelayResponse        RoundTripDelayResponse,

maintenanceLoopAck            MaintenanceLoopAck,
maintenanceLoopReject         MaintenanceLoopReject,

... ,
communicationModeResponse     CommunicationModeResponse,

conferenceResponse            ConferenceResponse,

multilinkResponse             MultilinkResponse,

logicalChannelRateAcknowledge LogicalChannelRateAcknowledge,
logicalChannelRateReject      LogicalChannelRateReject,
genericResponse                GenericMessage

}

-- Un CommandMessage exige une action, mais pas de réponse explicite
CommandMessage                ::=CHOICE
{
    nonStandard                NonStandardMessage,

    maintenanceLoopOffCommand  MaintenanceLoopOffCommand,

    sendTerminalCapabilitySet   SendTerminalCapabilitySet,

    encryptionCommand           EncryptionCommand,

    flowControlCommand          FlowControlCommand,

    endSessionCommand           EndSessionCommand,

    miscellaneousCommand        MiscellaneousCommand,

    ... ,
    communicationModeCommand    CommunicationModeCommand,

    conferenceCommand           ConferenceCommand,

    h223MultiplexReconfiguration H223MultiplexReconfiguration,

    newATMVCCCommand           NewATMVCCCommand,

    mobileMultilinkReconfigurationCommand MobileMultilinkReconfigurationCommand,
    genericCommand              GenericMessage

}

-- Un message IndicationMessage est une information qui n'exige ni action
-- ni réponse
IndicationMessage             ::=CHOICE
{
    nonStandard                NonStandardMessage,

```

```

functionNotUnderstood      FunctionNotUnderstood,
masterSlaveDeterminationRelease  MasterSlaveDeterminationRelease,
terminalCapabilitySetRelease  TerminalCapabilitySetRelease,
openLogicalChannelConfirm   OpenLogicalChannelConfirm,
requestChannelCloseRelease   RequestChannelCloseRelease,
multiplexEntrySendRelease    MultiplexEntrySendRelease,
requestMultiplexEntryRelease RequestMultiplexEntryRelease,
requestModeRelease           RequestModeRelease,
miscellaneousIndication      MiscellaneousIndication,
jitterIndication             JitterIndication,
h223SkewIndication           H223SkewIndication,
newATMVCIndication           NewATMVCIndication,
userInput                    UserInputIndication,
...
h2250MaximumSkewIndication   H2250MaximumSkewIndication,
mcLocationIndication         MCLocationIndication,
conferenceIndication          ConferenceIndication,
vendorIdentification         VendorIdentification,
functionNotSupported         FunctionNotSupported,
multilinkIndication          MultilinkIndication,
logicalChannelRateRelease    LogicalChannelRateRelease,
flowControlIndication        FlowControlIndication,
mobileMultilinkReconfigurationIndication  MobileMultilinkReconfigurationIndication,
genericIndication            GenericMessage

```

```

}

```

```

-- SequenceNumber est défini ici car il est utilisé dans un certain nombre
-- de messages

```

```

SequenceNumber ::= INTEGER (0..255)

```

```

-- =====
-- Définitions de message générique
-- =====

```

```

GenericMessage ::= SEQUENCE
{
    messageIdentifier      CapabilityIdentifier,
    subMessageIdentifier    INTEGER(0..127) OPTIONAL,

    messageContent         SEQUENCE OF GenericParameter OPTIONAL,
    ...
}

```

```

-- =====
-- Définitions relatives aux messages hors norme
-- =====

```

```

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

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

NonStandardIdentifier             ::=CHOICE
{
    object                          OBJECT IDENTIFIER,
    h221NonStandard                SEQUENCE
    {
        t35CountryCode             INTEGER (0..255),      -- pays, selon
                                                -- Annexe A/T.35
        t35Extension               INTEGER (0..255),
                                                -- attribué au niveau national sauf si
                                                -- l'indicatif de pays t35CountryCode
                                                -- est 1111 1111 binaire, auquel cas il
                                                -- doit contenir l'indicatif de pays
                                                -- selon Annexe B/T.35
        manufacturerCode           INTEGER (0..65535)    -- attribué au
                                                -- niveau national
    }
}

```

```

-- =====
-- Définitions relatives à la désignation maître-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,
        ...
    },
    ...
}

```

```

}

MasterSlaveDeterminationRelease ::=SEQUENCE
{
    ...
}

-- =====
-- Définitions relatives aux échanges de capacités
-- =====

TerminalCapabilitySet ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,

    protocolIdentifier      OBJECT IDENTIFIER,
                           -- doit être mis à la valeur
                           -- {itu-t (0) recommendation (0) h (8) 245
                           -- version (0) 11}

    multiplexCapability     MultiplexCapability OPTIONAL,

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

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

    ...
}

CapabilityTableEntry ::=SEQUENCE
{
    capabilityTableEntryNumber CapabilityTableEntryNumber,
    capability                Capability OPTIONAL
}

CapabilityDescriptor ::=SEQUENCE
{
    capabilityDescriptorNumber CapabilityDescriptorNumber,
    simultaneousCapabilities  SET SIZE (1..256) OF AlternativeCapabilitySet OPTIONAL
}

AlternativeCapabilitySet ::=SEQUENCE SIZE (1..256) OF CapabilityTableEntryNumber

CapabilityTableEntryNumber ::=INTEGER (1..65535)

CapabilityDescriptorNumber ::=INTEGER (0..255)

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

TerminalCapabilitySetReject ::=SEQUENCE
{
    sequenceNumber          SequenceNumber,
    cause                  CHOICE
    {
        unspecified        NULL,
        undefinedTableEntryUsed NULL,
        descriptorCapacityExceeded NULL,
    }
}

```



```

        tableEntryCapacityExceeded CHOICE
        {
            highestEntryNumberProcessed CapabilityTableEntryNumber,
            noneProcessed NULL
        },
        ...
    },
    ...
}

```

```

TerminalCapabilitySetRelease ::=SEQUENCE
{
    ...
}

```

```

-- =====
-- Définitions relatives aux échanges de capacités: description des capacités
-- de haut niveau
-- =====

```

```

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:
                                                -- millisecondes
        ...
    },
    ...,
    conferenceCapability ConferenceCapability,
    h235SecurityCapability H235SecurityCapability,
    maxPendingReplacementFor INTEGER (0..255),
    receiveUserInputCapability UserInputCapability,
    transmitUserInputCapability UserInputCapability,
    receiveAndTransmitUserInputCapability UserInputCapability,

    genericControlCapability GenericCapability,
    receiveMultiplexedStreamCapability MultiplexedStreamCapability,
    transmitMultiplexedStreamCapability MultiplexedStreamCapability,
    receiveAndTransmitMultiplexedStreamCapability MultiplexedStreamCapability,
    receiveRTPAudioTelephonyEventCapability AudioTelephonyEventCapability,
    receiveRTPAudioToneCapability AudioToneCapability,
    depFecCapability DepFECCapability,
    -- Déconseillé, ne pas utiliser
    multiplePayloadStreamCapability MultiplePayloadStreamCapability,
    fecCapability FECCapability,
    redundancyEncodingCap RedundancyEncodingCapability,
    oneOfCapabilities AlternativeCapabilitySet
}

```

```

H235SecurityCapability ::=SEQUENCE
{
    encryptionAuthenticationAndIntegrity EncryptionAuthenticationAndIntegrity,
    mediaCapability CapabilityTableEntryNumber,
    -- NOTE - La capacité mediaCapability désigne les entrées des
    -- tableaux de capacités du contenant effectivement, de façon directe
    -- ou indirecte, une ou plusieurs des capacités suivantes:
    -- AudioCapability (transmit, receive, ou receiveAndTransmit)+,
    -- VideoCapability, DataApplicationCapability, ou des capacités
    -- similaires indiquées uniquement par un paramètre
    -- NonStandardParameter ou GenericCapability
    ...
}

-- =====
-- Définitions relatives aux échanges de capacités: capacités de multiplexage
-- =====

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

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

VCCapability ::=SEQUENCE
{
    aal1 SEQUENCE
    {
        nullClockRecovery BOOLEAN,
        srtsClockRecovery BOOLEAN,
        adaptiveClockRecovery BOOLEAN,
        nullErrorCorrection BOOLEAN,
        longInterleaver BOOLEAN,
        shortInterleaver BOOLEAN,
        errorCorrectionOnly BOOLEAN,
        structuredDataTransfer BOOLEAN,
        partiallyFilledCells BOOLEAN,
        ...
    } OPTIONAL,
    aal5 SEQUENCE
    {
        forwardMaximumSDUSize INTEGER (0..65535), -- unité: octets
        backwardMaximumSDUSize INTEGER (0..65535), -- unité: octets
        ...
    } OPTIONAL,
    transportStream BOOLEAN,
    programStream BOOLEAN,
}

```

```

availableBitRates          SEQUENCE
{
  type                     CHOICE
  {
    singleBitRate          INTEGER (1..65535), -- unité: 64 kbit/s
    rangeOfBitRates        SEQUENCE
    {
      lowerBitRate         INTEGER (1..65535), -- unité: 64 kbit/s
      higherBitRate        INTEGER (1..65535)  -- unité: 64 kbit/s
    }
  },
  ...
},
...,
aallViaGateway            SEQUENCE
{
  gatewayAddress           SET SIZE(1..256) OF Q2931Address,
  nullClockRecovery        BOOLEAN,
  srtsClockRecovery        BOOLEAN,
  adaptiveClockRecovery    BOOLEAN,
  nullErrorCorrection       BOOLEAN,
  longInterleaver          BOOLEAN,
  shortInterleaver         BOOLEAN,
  errorCorrectionOnly       BOOLEAN,
  structuredDataTransfer    BOOLEAN,
  partiallyFilledCells      BOOLEAN,
  ...
} OPTIONAL
}
H223Capability            ::=SEQUENCE
{
  transportWithI-frames    BOOLEAN, -- Transport de trames I
                                -- selon la Rec. H.245

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

  maximumAL2SDUSize        INTEGER (0..65535), -- unité: octets
  maximumAL3SDUSize        INTEGER (0..65535), -- unité: octets

  maximumDelayJitter       INTEGER (0..1023), -- unité: millisecondes

  h223MultiplexTableCapability CHOICE
  {
    basic                   NULL,
    enhanced                 SEQUENCE
    {
      maximumNestingDepth   INTEGER (1..15),
      maximumElementListSize INTEGER (2..255),
      maximumSubElementListSize INTEGER (2..255),
      ...
    }
  },
  ...,
  maxMUXPDUSizeCapability  BOOLEAN,
  nsrpSupport              BOOLEAN,
  mobileOperationTransmitCapability SEQUENCE

```

```

    {
        modeChangeCapability          BOOLEAN,
        h223AnnexA                    BOOLEAN,
        h223AnnexADoubleFlag          BOOLEAN,
        h223AnnexB                    BOOLEAN,
        h223AnnexBwithHeader          BOOLEAN,
        ...
    } OPTIONAL,
    h223AnnexCCapability              H223AnnexCCapability OPTIONAL,

    bitRate                           INTEGER (1..19200) OPTIONAL, -- unités:
                                                                    -- 100 bit/s

    mobileMultilinkFrameCapability    SEQUENCE
    {
        maximumSampleSize             INTEGER (1..255),      -- unité: octets
        maximumPayloadLength          INTEGER (1..65025),    -- unité: octets
        ...
    } OPTIONAL
}

H223AnnexCCapability                ::= SEQUENCE
{
    videoWithAL1M                    BOOLEAN,
    videoWithAL2M                    BOOLEAN,
    videoWithAL3M                    BOOLEAN,
    audioWithAL1M                    BOOLEAN,
    audioWithAL2M                    BOOLEAN,
    audioWithAL3M                    BOOLEAN,
    dataWithAL1M                     BOOLEAN,
    dataWithAL2M                     BOOLEAN,
    dataWithAL3M                     BOOLEAN,
    alpduInterleaving                BOOLEAN,

    maximumAL1MPDUSize               INTEGER (0..65535),    -- unité: octets
    maximumAL2MSDUSize               INTEGER (0..65535),    -- unité: octets
    maximumAL3MSDUSize               INTEGER (0..65535),    -- unité: octets
    ...,
    rsCodeCapability                  BOOLEAN OPTIONAL
}

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

V75Capability                        ::=SEQUENCE
{
    audioHeader                       BOOLEAN,

```

```

    ...
}

H2250Capability ::=SEQUENCE
{
    maximumAudioDelayJitter    INTEGER(0..1023),    -- unités:
                                -- millisecondes

    receiveMultipointCapability MultipointCapability,
    transmitMultipointCapability MultipointCapability,
    receiveAndTransmitMultipointCapability MultipointCapability,
    mcCapability                SEQUENCE
    {
        centralizedConferenceMC  BOOLEAN,
        decentralizedConferenceMC BOOLEAN,
        ...
    },
    rtcpVideoControlCapability  BOOLEAN,    -- acquittement négatif
                                -- demande de rafraîchissement
                                -- intra complet

    mediaPacketizationCapability MediaPacketizationCapability,
    ...,
    transportCapability         TransportCapability OPTIONAL,
    redundancyEncodingCapability SEQUENCE SIZE(1..256) OF RedundancyEncodingCapability OPTIONAL,
    logicalChannelSwitchingCapability BOOLEAN,
    t120DynamicPortCapability   BOOLEAN
}

MediaPacketizationCapability ::=SEQUENCE
{
    h261aVideoPacketization    BOOLEAN,
    ...,
    rtpPayloadType             SEQUENCE SIZE(1..256) OF RTPPayloadType OPTIONAL
}

RSVPPParameters ::=SEQUENCE
{
    qosMode QOSMode OPTIONAL,
    tokenRate    INTEGER (1..4294967295) OPTIONAL,
                -- débit en octets
                -- par seconde

    bucketSize    INTEGER (1..4294967295) OPTIONAL,
                -- taille en octets

    peakRate    INTEGER (1..4294967295) OPTIONAL,
                -- largeur de bande max. en octets
                -- par seconde

    minPoliced    INTEGER (1..4294967295) OPTIONAL,
                --

    maxPktSize    INTEGER (1..4294967295) OPTIONAL,
                -- taille en octets

    ...
}

QOSMode ::=CHOICE
{
    guaranteedQOS    NULL,
    controlledLoad    NULL,
    ...
}

ATMPParameters ::=SEQUENCE
{
    maxNTUSize    INTEGER(0..65535),    -- unité: octets
    atmUBR        BOOLEAN,              -- débit non spécifié
}

```

```

    atmrtVBR                BOOLEAN,                -- débit variable en
                                -- temps réel
    atmrVBR                 BOOLEAN,                -- débit variable pas
                                -- en temps réel

    atmABR                  BOOLEAN,                -- débit disponible
    atmCBR                  BOOLEAN,                -- débit constant
    ...
}

QOSCapability                ::=SEQUENCE
{
    nonStandardData          NonStandardParameter OPTIONAL,
    rsvpParameters          RSVPParameters OPTIONAL,
    atmParameters           ATMPParameters OPTIONAL,
    ...
}

MediaTransportType          ::=CHOICE
{
    ip-UDP                  NULL,
    ip-TCP                  NULL,
    atm-AAL5-UNIDIR        NULL, -- circuits virtuels utilisés
                                -- unidirectionnel
    atm-AAL5-BIDIR        NULL, -- circuits virtuels utilisés
                                -- bidirectionnel
    ...,
    atm-AAL5-compressed    SEQUENCE
    {
        variable-delta     BOOLEAN,
        ...
    }
}

MediaChannelCapability      ::=SEQUENCE
{
    mediaTransport          MediaTransportType OPTIONAL,
    ...
}

TransportCapability         ::=SEQUENCE
{
    nonStandard             NonStandardParameter OPTIONAL,
    qosCapabilities        SEQUENCE SIZE(1..256) OF QOSCapability OPTIONAL,
    mediaChannelCapabilities SEQUENCE SIZE(1..256) OF MediaChannelCapability OPTIONAL,
    ...
}

RedundancyEncodingCapability ::=SEQUENCE
{
    redundancyEncodingMethod RedundancyEncodingMethod,
    primaryEncoding          CapabilityTableEntryNumber,
    secondaryEncoding        SEQUENCE SIZE(1..256) OF CapabilityTableEntryNumber OPTIONAL,
                                -- il doit s'agir de capacités audio, vidéo ou données,
                                -- non de capacités dérivées; l'ordre de redondance est
                                -- déduit du nombre de codages secondaires
}

RedundancyEncodingMethod   ::=CHOICE
{
    nonStandard             NonStandardParameter,
    rtpAudioRedundancyEncoding NULL,
    ...,
    rtpH263VideoRedundancyEncoding RTPH263VideoRedundancyEncoding
}

```

```

}

RTPH263VideoRedundancyEncoding ::= SEQUENCE
{
    numberOfThreads                INTEGER (1..16),
    framesBetweenSyncPoints        INTEGER (1..256),
    frameToThreadMapping           CHOICE
    {
        roundrobin                 NULL,
        custom                      SEQUENCE SIZE(1..256) OF
        RTPH263VideoRedundancyFrameMapping,
        -- SEQUENCE vide pour la négociation
        -- des capacités;
        -- contenu significatif uniquement
        -- pour OpenLogicalChannel
        ...
    },
    containedThreads               SEQUENCE SIZE(1..256) OF INTEGER (0..15) OPTIONAL,
    -- utilisé uniquement pour
    -- ouvrir des voies logiques
    ...
}

RTPH263VideoRedundancyFrameMapping ::= SEQUENCE
{
    threadNumber                   INTEGER (0..15),
    frameSequence                  SEQUENCE SIZE(1..256) OF INTEGER (0..255),
    ...
}

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

MediaDistributionCapability ::=SEQUENCE
{
    centralizedControl             BOOLEAN,
    distributedControl             BOOLEAN,
    -- pour complément d'étude
    -- dans la Rec. UIT-T H.323
    centralizedAudio              BOOLEAN,
    distributedAudio              BOOLEAN,
    centralizedVideo              BOOLEAN,
    distributedVideo              BOOLEAN,
    centralizedData               SEQUENCE OF DataApplicationCapability OPTIONAL,
    distributedData               SEQUENCE OF DataApplicationCapability OPTIONAL,
    -- pour complément d'étude
    -- dans la Rec. UIT-T H.323
    ...
}

-- =====
-- Définitions relatives aux échanges de capacités: capacités vidéo
-- =====

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

```

```

    is11172VideoCapability      IS11172VideoCapability,
    ...,
    genericVideoCapability      GenericCapability,
    extendedVideoCapability     ExtendedVideoCapability
}

ExtendedVideoCapability ::= SEQUENCE
{
    videoCapability             SEQUENCE OF VideoCapability,
    videoCapabilityExtension    SEQUENCE OF GenericCapability OPTIONAL,
    ...
}

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
    ...,
    videoBadMBsCap             BOOLEAN
}

H262VideoCapability ::= SEQUENCE
{
    profileAndLevel-SPatML      BOOLEAN,
    profileAndLevel-MPatLL      BOOLEAN,
    profileAndLevel-MPatML      BOOLEAN,
    profileAndLevel-MPatH-14    BOOLEAN,
    profileAndLevel-MPatHL      BOOLEAN,
    profileAndLevel-SNRatLL     BOOLEAN,
    profileAndLevel-SNRatML     BOOLEAN,
    profileAndLevel-SpatialatH-14 BOOLEAN,
    profileAndLevel-HPatML      BOOLEAN,
    profileAndLevel-HPatH-14    BOOLEAN,
    profileAndLevel-HPatHL      BOOLEAN,
    videoBitRate                INTEGER (0.. 1073741823) OPTIONAL, -- unité: 400 bit/s
    vbvBufferSize               INTEGER (0.. 262143) OPTIONAL, -- unité: 16 384 bits
    samplesPerLine              INTEGER (0..16383) OPTIONAL, -- unité: échantillons
                                                                    -- par ligne
    linesPerFrame               INTEGER (0..16383) OPTIONAL, -- unité lignes par
                                                                    -- par trame
    framesPerSecond             INTEGER (0..15) OPTIONAL, -- frame_rate_code
    luminanceSampleRate         INTEGER (0..4294967295) OPTIONAL, -- unité: échantillons
                                                                    -- par seconde
    ...,
    videoBadMBsCap             BOOLEAN
}

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
...
slowSqcifMPI        INTEGER (1..3600) OPTIONAL,
                    -- unité: secondes par trame
slowQcifMPI         INTEGER (1..3600) OPTIONAL,
                    -- unité: secondes par trame
slowCifMPI          INTEGER (1..3600) OPTIONAL,
                    -- unité: secondes par trame
slowCif4MPI         INTEGER (1..3600) OPTIONAL,
                    -- unité: secondes par trame
slowCif16MPI        INTEGER (1..3600) OPTIONAL,
                    -- unité: secondes par trame
errorCompensation   BOOLEAN,

enhancementLayerInfo EnhancementLayerInfo OPTIONAL,
h263Options          H263Options OPTIONAL
}

EnhancementLayerInfo ::=SEQUENCE
{
    baseBitRateConstrained    BOOLEAN,
    snrEnhancement            SET SIZE(1..14) OF EnhancementOptions OPTIONAL,
    spatialEnhancement        SET SIZE(1..14) OF EnhancementOptions OPTIONAL,
    bPictureEnhancement       SET SIZE(1..14) OF BEnhancementParameters OPTIONAL,
    ...
}

BEnhancementParameters ::=SEQUENCE
{
    enhancementOptions        EnhancementOptions,
    numberOfBPictures         INTEGER (1..64),
    ...
}

EnhancementOptions ::=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,
    temporalSpatialTradeOffCapability BOOLEAN,
    slowSqcifMPI              INTEGER (1..3600) OPTIONAL, -- unité: secondes par
    -- trame
    slowQcifMPI               INTEGER (1..3600) OPTIONAL, -- unité: secondes par
    -- trame
    slowCifMPI                INTEGER (1..3600) OPTIONAL, -- unité: secondes par
    -- trame
    slowCif4MPI               INTEGER (1..3600) OPTIONAL, -- unité: secondes par
    -- trame
    slowCif16MPI              INTEGER (1..3600) OPTIONAL, -- unité: secondes par
    -- trame
    errorCompensation         BOOLEAN,
    h263Options                H263Options OPTIONAL,
    ...
}

H263Options ::= SEQUENCE
{
    advancedIntraCodingMode    BOOLEAN,

```

```

deblockingFilterMode          BOOLEAN,
improvedPBFFramesMode        BOOLEAN,

unlimitedMotionVectors        BOOLEAN,

fullPictureFreeze            BOOLEAN,
partialPictureFreezeAndRelease  BOOLEAN,
resizingPartPicFreezeAndRelease  BOOLEAN,
fullPictureSnapshot          BOOLEAN,
partialPictureSnapshot        BOOLEAN,
videoSegmentTagging          BOOLEAN,
progressiveRefinement         BOOLEAN,

dynamicPictureResizingByFour  BOOLEAN,
dynamicPictureResizingSixteenthPel  BOOLEAN,
dynamicWarpingHalfPel        BOOLEAN,
dynamicWarpingSixteenthPel    BOOLEAN,

independentSegmentDecoding    BOOLEAN,

slicesInOrder-NonRect        BOOLEAN,
slicesInOrder-Rect           BOOLEAN,
slicesNoOrder-NonRect        BOOLEAN,
slicesNoOrder-Rect           BOOLEAN,

alternateInterVLCMode         BOOLEAN,
modifiedQuantizationMode      BOOLEAN,
reducedResolutionUpdate       BOOLEAN,

transparencyParameters        TransparencyParameters OPTIONAL,
separateVideoBackChannel      BOOLEAN,
refPictureSelection           RefPictureSelection OPTIONAL,
customPictureClockFrequency   SET SIZE (1..16) OF CustomPictureClockFrequency OPTIONAL,
customPictureFormat           SET SIZE (1..16) OF CustomPictureFormat OPTIONAL,
modeCombos                    SET SIZE (1..16) OF H263VideoModeCombos OPTIONAL,
...,
videoBadMBsCap                BOOLEAN,
h263Version3Options           H263Version3Options

```

```

}

```

```

TransparencyParameters ::= SEQUENCE
{
    presentationOrder      INTEGER(1..256),
    offset-x                INTEGER(-262144..262143), -- 1/8 pixel
    offset-y                INTEGER(-262144..262143), -- 1/8 pixel
    scale-x                 INTEGER(1..255),
    scale-y                 INTEGER(1..255),
    ...
}

```

```

RefPictureSelection ::=SEQUENCE
{
    additionalPictureMemory SEQUENCE
    {
        sqcifAdditionalPictureMemory  INTEGER (1..256) OPTIONAL, -- unité: trames
        qcifAdditionalPictureMemory    INTEGER (1..256) OPTIONAL, -- unité: trames
        cifAdditionalPictureMemory     INTEGER (1..256) OPTIONAL, -- unité: trames
        cif4AdditionalPictureMemory    INTEGER (1..256) OPTIONAL, -- unité: trames
        cif16AdditionalPictureMemory   INTEGER (1..256) OPTIONAL, -- unité: trames
        bigCpfAdditionalPictureMemory  INTEGER (1..256) OPTIONAL, -- unité: trames
        ...
    } OPTIONAL,
    videoMux                    BOOLEAN,
    videoBackChannelSend        CHOICE

```

```

{
    none                NULL,
    ackMessageOnly     NULL,
    nackMessageOnly    NULL,
    ackOrNackMessageOnly NULL,
    ackAndNackMessage  NULL,
    ...
},
...,
enhancedReferencePicSelect SEQUENCE
{
    subPictureRemovalParameters SEQUENCE
    {
        mpuHorizMBS          INTEGER (1..128),
        mpuVertMBS           INTEGER (1..72),
        mpuTotalNumber       INTEGER (1..65536),
        ...
    } OPTIONAL,
    ...
}
}

CustomPictureClockFrequency ::=SEQUENCE
{
    clockConversionCode      INTEGER(1000..1001),
    clockDivisor             INTEGER(1..127),
    sqcifMPI                 INTEGER (1..2048) OPTIONAL,
    qcifMPI                  INTEGER (1..2048) OPTIONAL,
    cifMPI                   INTEGER (1..2048) OPTIONAL,
    cif4MPI                  INTEGER (1..2048) OPTIONAL,
    cif16MPI                 INTEGER (1..2048) OPTIONAL,
    ...
}

CustomPictureFormat ::=SEQUENCE
{
    maxCustomPictureWidth   INTEGER(1..2048),      -- unité: 4 pixels
    maxCustomPictureHeight  INTEGER(1..2048),      -- unité: 4 pixels
    minCustomPictureWidth   INTEGER(1..2048),      -- unité: 4 pixels
    minCustomPictureHeight  INTEGER(1..2048),      -- unité: 4 pixels
    mPI                     SEQUENCE
    {
        standardMPI         INTEGER (1..31) OPTIONAL,
        customPCF           SET SIZE (1..16) OF SEQUENCE
        {
            clockConversionCode  INTEGER (1000..1001),
            clockDivisor         INTEGER (1..127),
            customMPI            INTEGER (1..2048),
            ...
        } OPTIONAL,
        ...
    },

    pixelAspectInformation  CHOICE
    {
        anyPixelAspectRatio   BOOLEAN,
        pixelAspectCode       SET SIZE (1..14) OF INTEGER(1..14),
        extendedPAR           SET SIZE (1..256) OF SEQUENCE
        {
            width              INTEGER(1..255),
            height              INTEGER(1..255),
            ...
        },
    },
}

```

```

    } ,
    ...
}

H263VideoModeCombos ::= SEQUENCE
{
    h263VideoUncoupledModes H263ModeComboFlags,
    h263VideoCoupledModes SET SIZE (1..16) OF H263ModeComboFlags,
    ...
}

H263ModeComboFlags ::= SEQUENCE
{
    unrestrictedVector BOOLEAN,
    arithmeticCoding BOOLEAN,
    advancedPrediction BOOLEAN,
    pbFrames BOOLEAN,
    advancedIntraCodingMode BOOLEAN,
    deblockingFilterMode BOOLEAN,
    unlimitedMotionVectors BOOLEAN,
    slicesInOrder-NonRect BOOLEAN,
    slicesInOrder-Rect BOOLEAN,
    slicesNoOrder-NonRect BOOLEAN,
    slicesNoOrder-Rect BOOLEAN,
    improvedPBframesMode BOOLEAN,
    referencePicSelect BOOLEAN,
    dynamicPictureResizingByFour BOOLEAN,
    dynamicPictureResizingSixteenthPel BOOLEAN,
    dynamicWarpingHalfPel BOOLEAN,
    dynamicWarpingSixteenthPel BOOLEAN,
    reducedResolutionUpdate BOOLEAN,
    independentSegmentDecoding BOOLEAN,
    alternateInterVLCMode BOOLEAN,
    modifiedQuantizationMode BOOLEAN,
    ...,
    enhancedReferencePicSelect BOOLEAN,
    h263Version3Options H263Version3Options}

H263Version3Options ::=SEQUENCE
{
    dataPartitionedSlices BOOLEAN,
    fixedPointIDCT0 BOOLEAN,
    interlacedFields BOOLEAN,
    currentPictureHeaderRepetition BOOLEAN,
    previousPictureHeaderRepetition BOOLEAN,
    nextPictureHeaderRepetition BOOLEAN,
    pictureNumber BOOLEAN,
    spareReferencePictures BOOLEAN,
    ...
}

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

```

```

        videoBadMBsCap                BOOLEAN
    }

-- =====
-- Définitions relatives aux échanges de capacités: capacités audio
-- =====

-- Pour un multiplex H.222, les nombres entiers indiquent la taille de la
-- mémoire tampon STD(unité: 256 octets)
-- Pour un multiplex H.223, les nombres entiers indiquent le nombre maximal de
-- trames audio par AL-SDU
-- Pour un multiplex H.225.0, les nombres entiers indiquent le nombre maximal de
-- trames audio par paquet

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

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

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

    g728                                INTEGER (1..256),
    g729                                INTEGER (1..256),
    g729AnnexA                          INTEGER (1..256),
    is11172AudioCapability              IS11172AudioCapability,
    is13818AudioCapability              IS13818AudioCapability,
    ...,
    g729wAnnexB                          INTEGER (1..256),
    g729AnnexAwAnnexB                  INTEGER (1..256),
    g7231AnnexCCapability              G7231AnnexCCapability,
    gsmFullRate                        GSMAudioCapability,
    gsmHalfRate                         GSMAudioCapability,
    gsmEnhancedFullRate                GSMAudioCapability,
    genericAudioCapability              GenericCapability,
    g729Extensions                      G729Extensions,
    vbd                                 VBDCapability,
    audioTelephonyEvent                NoPTAudioTelephonyEventCapability,
    audioTone                           NoPTAudioToneCapability
}

G729Extensions                        ::= SEQUENCE
{
    audioUnit                            INTEGER (1..256) OPTIONAL,
    annexA                              BOOLEAN,
    annexB                              BOOLEAN,
    annexD                              BOOLEAN,
    annexE                              BOOLEAN,
    annexF                              BOOLEAN,
    annexG                              BOOLEAN,
    annexH                              BOOLEAN,
    ...
}

```

```

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

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

    audioSampling32k        BOOLEAN,
    audioSampling44k1      BOOLEAN,
    audioSampling48k        BOOLEAN,

    singleChannel           BOOLEAN,
    twoChannels             BOOLEAN,

    bitRate                 INTEGER (1..448),   -- unit s: 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),  -- unit s: kbit/s
    ...
}

```

```

GSMAudioCapability ::= SEQUENCE
{
    audioUnitSize      INTEGER (1..256),
    comfortNoise        BOOLEAN,
    scrambled            BOOLEAN,
    ...
}

VBDCapability ::= SEQUENCE
{
    type                AudioCapability, -- ne doit pas être "vbd"
    ...
}

-- =====
-- Définitions relatives aux échanges de capacités: capacités de données
-- =====

DataApplicationCapability ::= SEQUENCE
{
    application        CHOICE
    {
        nonStandard    NonStandardParameter,
        t120            DataProtocolCapability,
        dsm-cc          DataProtocolCapability,
        userData        DataProtocolCapability,
        t84SEQUENCE
        {
            t84Protocol  DataProtocolCapability,
            t84Profile    T84Profile
        },
        t434            DataProtocolCapability,
        h224            DataProtocolCapability,
        nlpid           SEQUENCE
        {
            nlpidProtocol DataProtocolCapability,
            nlpidData      OCTET STRING
        },
        dsvdControl     NULL,
        h222DataPartitioning DataProtocolCapability,
        ...,
        t30fax          DataProtocolCapability,
        t140            DataProtocolCapability,
        t38fax          SEQUENCE
        {
            t38FaxProtocol DataProtocolCapability,
            t38FaxProfile    T38FaxProfile
        },
        genericDataCapability GenericCapability
    },
    maxBitRate         INTEGER (0..4294967295), -- unités:
                                                    -- 100 bit/s
    ...
}

DataProtocolCapability ::= CHOICE
{
    nonStandard        NonStandardParameter,
    v14buffered        NULL,
    v42lapm            NULL, -- possibilité de négocier
                        -- le mode V.42 bis
    hdlcFrameTunnelling NULL,
    h310SeparateVCStack NULL,
}

```

```

h310SingleVCStack      NULL,
transparent             NULL,
...
segmentationAndReassembly  NULL,
hdlcFrameTunnelingwSAR    NULL,
v120                   NULL,           -- comme dans la Rec. UIT-T H.230
separateLANStack        NULL,
v76wCompression        CHOICE
{
    transmitCompression    CompressionType,
    receiveCompression     CompressionType,
    transmitAndReceiveCompression  CompressionType,
    ...
},
tcp                     NULL,
udp                     NULL
}

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

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

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

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

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

```



```

T38FaxProfile ::=SEQUENCE
{
    fillBitRemoval          BOOLEAN,
    transcodingJBIG         BOOLEAN,
    transcodingMMR          BOOLEAN,
    ...,
    version                 INTEGER (0..255),
                           -- Version 0, par défaut,
                           -- voir Rec. UIT-T.38 (1998)

    t38FaxRateManagement   T38FaxRateManagement,
                           -- La capacité Data Rate Management par
                           -- défaut est déterminée par le choix de
                           -- la capacité DataProtocolCapability

    t38FaxUdpOptions       T38FaxUdpOptions OPTIONAL,
                           -- Pour le protocole UDP, t38UDPRedundancy
                           -- est l'option par défaut

    t38FaxTcpOptions       T38FaxTcpOptions OPTIONAL
}

T38FaxRateManagement ::= CHOICE
{
    localTCF                NULL,
    transferredTCF          NULL,
    ...
}

T38FaxUdpOptions ::= SEQUENCE
{
    t38FaxMaxBuffer         INTEGER OPTIONAL,
    t38FaxMaxDatagram       INTEGER OPTIONAL,
    t38FaxUdpEC             CHOICE
    {
        t38UDPFEC           NULL,
        t38UDPRedundancy    NULL,
        ...
    }
}

T38FaxTcpOptions ::= SEQUENCE
{
    t38TCPBidirectionalMode BOOLEAN,
    ...
}

-- =====
-- Définitions relatives aux capacités de chiffrement
-- =====

EncryptionAuthenticationAndIntegrity ::=SEQUENCE
{
    encryptionCapability    EncryptionCapability OPTIONAL,
    authenticationCapability AuthenticationCapability OPTIONAL,
    integrityCapability      IntegrityCapability OPTIONAL,
    ...,
    genericH235SecurityCapability GenericCapability OPTIONAL
}

EncryptionCapability ::=SEQUENCE SIZE(1..256) OF MediaEncryptionAlgorithm

MediaEncryptionAlgorithm ::=CHOICE
{
    nonStandard              NonStandardParameter,

```

```

        algorithm                OBJECT IDENTIFIER, -- grand nombre
                                -- d'éléments définis
                                -- dans l'ISO/CEI 9979
    ...
}

AuthenticationCapability        ::=SEQUENCE
{
    nonStandard                  NonStandardParameter OPTIONAL,
    ...,
    antiSpamAlgorithm            OBJECT IDENTIFIER OPTIONAL
}

IntegrityCapability             ::=SEQUENCE
{
    nonStandard                  NonStandardParameter OPTIONAL,
    ...
}

-- =====
-- Définitions relatives aux échanges de capacités: entrées de l'utilisateur
-- =====

UserInputCapability             ::= CHOICE
{
    nonStandard                  SEQUENCE SIZE(1..16) OF NonStandardParameter,
    basicString                  NULL, -- alphanumérique
    iA5String                    NULL, -- alphanumérique
    generalString                NULL, -- alphanumérique
    dtmf                         NULL, -- accepte dtmf avec signal
                                -- et signalUpdate
    hookflash                    NULL, -- accepte le mode de raccrochage
                                -- avec signal
    ...,
    extendedAlphanumeric         NULL,
    encryptedBasicString         NULL, -- chaîne de base chiffrée dans
                                -- encryptedAlphanumeric
    encryptedIA5String          NULL, -- chaîne IA5 chiffrée dans
                                -- encryptedSignalType
    encryptedGeneralString      NULL, -- chaîne générale chiffrée dans
                                -- extendedAlphanumeric.encryptedalphanumeric
    secureDTMF                  NULL -- codage DTMF sécurisé utilisant
                                -- encryptedSignalType
}

-- =====
-- Définitions relatives aux échanges de capacités: conférence
-- =====

ConferenceCapability            ::=SEQUENCE
{
    nonStandardData              SEQUENCE OF NonStandardParameter OPTIONAL,
    chairControlCapability        BOOLEAN,
    ...,
    videoIndicateMixingCapability BOOLEAN,
    multipointVisualizationCapability BOOLEAN OPTIONAL -- identique à H.230 MVC
}

-- =====
-- Définitions relatives aux échanges de capacités: capacité générique
-- =====

GenericCapability               ::=SEQUENCE
{
    capabilityIdentifier          CapabilityIdentifier,

```

```

maxBitRate                INTEGER (0..4294967295) OPTIONAL,
                           -- Unités 100 bit/s
collapsing                SEQUENCE OF GenericParameter OPTIONAL,
nonCollapsing             SEQUENCE OF GenericParameter OPTIONAL,
nonCollapsingRaw         OCTET STRING OPTIONAL,
                           -- Contient typiquement des données
                           -- décrivant la capacité, codées selon
                           -- les règles PER de l'ASN.1
transport                DataProtocolCapability OPTIONAL,
...
}

CapabilityIdentifier      ::=CHOICE
{
    standard              OBJECT IDENTIFIER,
                           -- par exemple
                           -- { itu-t (0) recommendation (0) h (8) 267
                           -- version (0) 2 subIdentifieur (0)}
    h221NonStandard       NonStandardParameter,
    uuid                  OCTET STRING ( SIZE (16) ),
    domainBased           IA5String ( SIZE (1..64) ),
    ...
}

-- NOTE - Les intervalles de valeurs de paramètre ont été choisis de manière à
-- garantir que le préambule GenericParameter, la partie standard de
-- ParameterIdentifier et le codage de ce choix, et le préambule
-- de ParameterValue rentrent dans 2 octets.

GenericParameter        ::=SEQUENCE
{
    parameterIdentifier   ParameterIdentifier,
    parameterValue        ParameterValue,
    supersedes            SEQUENCE OF ParameterIdentifier OPTIONAL,
    ...
}

ParameterIdentifier     ::=CHOICE
{
    standard              INTEGER (0..127), -- assigné au moyen des
                                           -- spécifications de la
                                           -- capacité
    h221NonStandard       NonStandardParameter, -- N.B.
                                           -- NonStandardIdentifieur
                                           -- ne suffit pas dans
                                           -- ce cas
    uuid                  OCTET STRING ( SIZE (16) ), -- dans le cas
                                           -- non-standard
    domainBased           IA5String ( SIZE (1..64) ),
    ...
}

ParameterValue         ::=CHOICE
{
    logical               NULL,
                           -- acceptable
                           -- uniquement si
                           -- toutes
                           -- les entités
                           -- incluent cette
                           -- option
    booleanArray          INTEGER (0..255),
                           -- matrice de 8 types
                           -- logiques
    unsignedMin           INTEGER (0..65535),
                           -- recherche de la
                           -- valeur maximale
                           -- commune
}

```

```

    unsignedMax                INTEGER (0..65535),      -- recherche de la
                                -- valeur maximale
                                -- commune
    unsigned32Min              INTEGER (0..4294967295), -- recherche de la
                                -- valeur maximale
                                -- commune
    unsigned32Max              INTEGER (0..4294967295), -- recherche de la
                                -- valeur maximale
                                -- commune
    octetString                OCTET STRING,          -- chaîne d'octets dans le cas
                                -- dans le cas non-collapsing
                                -- fédérable
    genericParameter           SEQUENCE OF GenericParameter,
    ...
}

```

```

-- =====
-- Définitions relatives aux échanges de capacités: capacité de flux multiplexé
-- =====

```

```

MultiplexedStreamCapability    ::=SEQUENCE
{
    multiplexFormat             MultiplexFormat,
    controlOnMuxStream          BOOLEAN,
    capabilityOnMuxStream       SET SIZE (1..256) OF
AlternativeCapabilitySet OPTIONAL,
    ...
}

```

```

MultiplexFormat                ::= CHOICE
{
    nonStandard                 NonStandardParameter,
    h222Capability              H222Capability,
    h223Capability              H223Capability,
    ...
}

```

```

-- =====
-- Définitions relatives aux échanges de capacités:
-- AudioTelephonyEventCapability et AudioToneCapability
-- =====

```

```

AudioTelephonyEventCapability ::=SEQUENCE
{
    dynamicRTPPayloadType       INTEGER(96..127),
    audioTelephoneEvent         GeneralString, -- Selon <liste de valeurs>
                                -- 3.9/RFC 2833
    ...
}

```

```

AudioToneCapability            ::=SEQUENCE
{
    dynamicRTPPayloadType       INTEGER(96..127),
    ...
}
}

```

```

-- Les définitions suivantes sont comme ci-dessus, mais sans champ de type de
-- charge utile.

```

```

NoPTAudioTelephonyEventCapability ::=SEQUENCE
{
    audioTelephoneEvent         GeneralString, -- Conformément à la liste de
                                -- valeurs <list of values>
                                -- figurant dans le § 3.9/RFC 2833
}

```

```

    ...
}

NoPTAudioToneCapability ::=SEQUENCE
{
    ...
}

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

MultiplePayloadStreamCapability ::=SEQUENCE
{
    capabilities SET SIZE(1..256) OF AlternativeCapabilitySet,
    ...
}

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

DepFECCapability ::=CHOICE -- Déconseillé, à éviter
{
    rfc2733 SEQUENCE
    {
        redundancyEncoding BOOLEAN,
        separateStream SEQUENCE
        {
            separatePort BOOLEAN,
            samePort BOOLEAN,
            ...
        },
        ...
    },
    ...
}

FECCapability ::= SEQUENCE
{
    protectedCapability CapabilityTableEntryNumber,
    fecScheme OBJECT IDENTIFIER OPTIONAL,
    -- désigne le système de codage
    rfc2733Format CHOICE
    {
        rfc2733rfc2198 MaxRedundancy, -- RFC 2198
        rfc2733sameport MaxRedundancy,
        -- Paquet séparé, même port
        rfc2733diffport MaxRedundancy
        -- Paquet séparé, port différent
    } OPTIONAL,
    ...
}

MaxRedundancy ::= INTEGER (1..MAX)

-- =====
-- Définitions relatives à la signalisation des voies logiques
-- =====

```

-- 'Forward' désigne la transmission depuis le terminal émetteur de la demande
 -- initiale de voie logique jusqu'à l'autre terminal, et 'reverse' désigne le
 -- sens inverse de transmission, lorsqu'une voie bidirectionnelle est demandée.

```

OpenLogicalChannel ::= SEQUENCE
{
  forwardLogicalChannelNumber LogicalChannelNumber,

  forwardLogicalChannelParameters SEQUENCE
  {
    portNumber INTEGER (0..65535) OPTIONAL,
    dataType DataType,
    multiplexParameters CHOICE
    {
      h222LogicalChannelParameters H222LogicalChannelParameters,
      h223LogicalChannelParameters H223LogicalChannelParameters,
      v76LogicalChannelParameters V76LogicalChannelParameters,
      ...,
      h2250LogicalChannelParameters H2250LogicalChannelParameters,
      none NULL -- à utiliser avec
                -- une pile séparée
                -- lorsque les paramètres
                -- multiplexParameters
                -- ne sont pas requis
                -- ou appropriés
    },
    ...,
    forwardLogicalChannelDependency LogicalChannelNumber OPTIONAL,
    -- sert également à désigner la
    -- voie logique primaire en cas
    -- de codage vidéo redondant

    replacementFor LogicalChannelNumber OPTIONAL
  },

  -- Permet de spécifier la voie inverse en cas de demande d'ouverture
  -- d'une voie bidirectionnelle

  reverseLogicalChannelParameters SEQUENCE
  {
    dataType DataType,
    multiplexParameters CHOICE
    {
      -- Les paramètres H.222 ne sont jamais présents dans le sens
      -- inverse
      h223LogicalChannelParameters H223LogicalChannelParameters,
      v76LogicalChannelParameters V76LogicalChannelParameters,
      ...,
      h2250LogicalChannelParameters H2250LogicalChannelParameters
    } OPTIONAL,
    -- Non présent pour H.222
    ...,
    reverseLogicalChannelDependency LogicalChannelNumber OPTIONAL,
    -- sert aussi à désigner la voie logique primaire en cas
    -- d'utilisation du codage vidéo redondant
    replacementFor LogicalChannelNumber OPTIONAL
  } OPTIONAL,
  -- Non présent pour une demande
  -- de voie unidirectionnelle
  ...,
  separateStack NetworkAccessParameters OPTIONAL,
  -- Permet au répondeur d'établir la pile
  encryptionSync EncryptionSync OPTIONAL
  -- utilisé uniquement par le
  -- terminal maître

```

```

}

LogicalChannelNumber ::=INTEGER (1..65535)

NetworkAccessParameters ::=SEQUENCE
{
    distribution CHOICE
    {
        unicast NULL,
        multicast NULL, -- Pour complément d'étude
        ... -- dans T.120
    } OPTIONAL,

    networkAddress CHOICE
    {
        q2931Address Q2931Address,
        e164Address IA5String(SIZE(1..128)) (FROM ("0123456789#*,")),
        localAreaAddress TransportAddress,
        ...
    },
    associateConference BOOLEAN,
    externalReference OCTET STRING(SIZE(1..255)) OPTIONAL,
    ...,
    t120SetupProcedure CHOICE
    {
        originateCall NULL,
        waitForCall NULL,
        issueQuery NULL,
        ...
    } OPTIONAL
}

Q2931Address ::=SEQUENCE
{
    address CHOICE
    {
        internationalNumber NumericString(SIZE(1..16)),
        nsapAddress OCTET STRING (SIZE(1..20)),
        ...
    },
    subaddress OCTET STRING (SIZE(1..20)) OPTIONAL,
    ...
}

V75Parameters ::= SEQUENCE
{
    audioHeaderPresent BOOLEAN,
    ...
}

DataType ::=CHOICE
{
    nonStandard NonStandardParameter,
    nullData NULL,
    videoData VideoCapability,
    audioData AudioCapability,
    data DataApplicationCapability,
    encryptionData EncryptionMode,
    ...,
    h235Control NonStandardParameter,
    h235Media H235Media,
}

```

<pre> multiplexedStream redundancyEncoding multiplePayloadStream depFec fec } </pre>	<pre> MultiplexedStreamParameter, RedundancyEncoding, MultiplePayloadStream, DepFECData, -- Déconseillé, à éviter FECDATA </pre>
<pre> H235Media { encryptionAuthenticationAndIntegrity mediaType { nonStandard videoData audioData data ..., redundancyEncoding multiplePayloadStream depFec fec }, ... } </pre>	<pre> ::=SEQUENCE EncryptionAuthenticationAndIntegrity, CHOICE NonStandardParameter, VideoCapability, AudioCapability, DataApplicationCapability, RedundancyEncoding, MultiplePayloadStream, DepFECData, -- Déconseillé, à éviter FECDATA </pre>
<pre> MultiplexedStreamParameter { multiplexFormat controlOnMuxStream ... } </pre>	<pre> ::=SEQUENCE MultiplexFormat, BOOLEAN, </pre>
<pre> H222LogicalChannelParameters { resourceID subChannelID pcr-pid programDescriptors streamDescriptors ... } </pre>	<pre> ::=SEQUENCE INTEGER (0..65535), INTEGER (0..8191), INTEGER (0..8191) OPTIONAL, OCTET STRING OPTIONAL, OCTET STRING OPTIONAL, </pre>
<pre> H223LogicalChannelParameters { adaptationLayerType { nonStandard allFramed allNotFramed al2WithoutSequenceNumbers al2WithSequenceNumbers al3 { controlFieldOctets sendBufferSize }, ..., al1M al2M al3M }, ... } </pre>	<pre> ::=SEQUENCE CHOICE NonStandardParameter, NULL, NULL, NULL, NULL, SEQUENCE INTEGER (0..2), INTEGER (0..16777215) -- unité: octets H223AL1MParameters, H223AL2MParameters, H223AL3MParameters </pre>


```

        segmentableFlag          BOOLEAN,
        ...
    }
H223AL1MParameters              ::=SEQUENCE
{
    transferMode                 CHOICE
    {
        framed                   NULL,
        unframed                 NULL,
        ...
    },
    headerFEC                    CHOICE
    {
        sebch16-7               NULL,
        golay24-12              NULL,
        ...
    },
    crcLength                    CHOICE
    {
        crc4bit                 NULL,
        crc12bit                NULL,
        crc20bit                NULL,
        crc28bit                NULL,
        ...,
        crc8bit                 NULL,
        crc16bit                NULL,
        crc32bit                NULL,
        crcNotUsed              NULL
    },
    rcpcCodeRate                 INTEGER (8..32),
    arqType                      CHOICE
    {
        noArq                   NULL,
        typeIArq                H223AnnexCArqParameters,
        typeIIArq               H223AnnexCArqParameters,
        ...
    },
    alpduInterleaving            BOOLEAN,
    alsduSplitting              BOOLEAN,
    ...,
    rsCodeCorrection             INTEGER (0..127) OPTIONAL
}
H223AL2MParameters              ::=SEQUENCE
{
    headerFEC                    CHOICE
    {
        sebch16-5               NULL,
        golay24-12              NULL,
        ...
    },
    alpduInterleaving            BOOLEAN,
    ...
}
H223AL3MParameters              ::=SEQUENCE
{
    headerFormat                 CHOICE
    {
        sebch16-7               NULL,

```

```

        golay24-12                NULL,
        ...
    },
    crcLength                      CHOICE
    {
        crc4bit                    NULL,
        crc12bit                   NULL,
        crc20bit                   NULL,
        crc28bit                   NULL,
        ...,
        crc8bit                    NULL,
        crc16bit                   NULL,
        crc32bit                   NULL,
        crcNotUsed                 NULL
    },

    rcpcCodeRate                  INTEGER (8..32),

    arqType                       CHOICE
    {
        noArq                      NULL,
        typeIArq                   H223AnnexCArqParameters,
        typeIIArq                  H223AnnexCArqParameters,
        ...
    },

    alpduInterleaving            BOOLEAN,
    ...,
    rsCodeCorrection             INTEGER (0..127) OPTIONAL
}

H223AnnexCArqParameters         ::=SEQUENCE
{
    numberOfRetransmissions      CHOICE
    {
        finite                    INTEGER (0..16),
        infinite                  NULL,
        ...
    },
    sendBufferSize              INTEGER (0..16777215), -- unité: octets
    ...
}

V76LogicalChannelParameters     ::=SEQUENCE
{
    hdlcParameters              V76HDLCPParameters,
    suspendResume               CHOICE
    {
        noSuspendResume          NULL,
        suspendResumewAddress    NULL,
        suspendResumewoAddress   NULL,
        ...
    },
    uIH                         BOOLEAN,
    mode                         CHOICE
    {
        eRM                      SEQUENCE
        {
            windowSize           INTEGER (1..127) ,
            recovery              CHOICE
            {
                rej               NULL,
                sREJ              NULL,
            }
        }
    }
}

```

```

        mSREJ                                NULL,
        ...
    },
    ...
},
uNERM                                        NULL,
...
},
v75Parameters                              V75Parameters,
...
}

V76HDLParameters ::=SEQUENCE
{
    crcLength                                CRCLength,
    n401                                     INTEGER (1..4095),
    loopbackTestProcedure                   BOOLEAN,
    ...
}

CRCLength ::=CHOICE
{
    crc8bit                                 NULL,
    crc16bit                                NULL,
    crc32bit                                NULL,
    ...
}

H2250LogicalChannelParameters ::=SEQUENCE
{
    nonStandard                             SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID                               INTEGER(0..255),
    associatedSessionID                     INTEGER(1..255) OPTIONAL,
    mediaChannel                            TransportAddress OPTIONAL,
    mediaGuaranteedDelivery                 BOOLEAN OPTIONAL,
    mediaControlChannel                     TransportAddress OPTIONAL, -- canal RTCP
                                                -- inverse
    mediaControlGuaranteedDelivery          BOOLEAN OPTIONAL,
    silenceSuppression                       BOOLEAN OPTIONAL,
    destination                             TerminalLabel OPTIONAL,
    dynamicRTPPayloadType                   INTEGER(96..127) OPTIONAL,
    mediaPacketization                       CHOICE
    {
        h261aVideoPacketization             NULL,
        ...
        rtpPayloadType                       RTPPayloadType
    } OPTIONAL,
    ...
    transportCapability                     TransportCapability OPTIONAL,
    redundancyEncoding                       RedundancyEncoding OPTIONAL,
    source                                   TerminalLabel OPTIONAL
}

RTPPayloadType ::= SEQUENCE
{
    payloadDescriptor                       CHOICE
    {
        nonStandardIdentifier               NonStandardParameter,
        rfc-number                           INTEGER (1..32768, ...),
    }
}

```

```

        oid                OBJECT IDENTIFIER,
        ...
    },
    payloadType            INTEGER (0..127) OPTIONAL,
    ...
}

RedundancyEncoding        ::=SEQUENCE
{
    redundancyEncodingMethod RedundancyEncodingMethod,
    secondaryEncoding        DataType OPTIONAL,    -- selon la méthode
    ...

-- La séquence ci-dessous peut être utilisée à la place du champ
-- secondaryEncoding ci-dessus

    rtpRedundancyEncoding    SEQUENCE
    {
        primary                RedundancyEncodingElement OPTIONAL,
                                -- Présent quand redundancyEncoding
                                -- est choisi comme dataType
                                -- dans un OpenLogicalChannel ou
                                -- dans le cadre d'un
                                -- MultiplePayloadStream

        secondary              SEQUENCE OF RedundancyEncodingElement OPTIONAL,
        ...
    } OPTIONAL
}

RedundancyEncodingElement ::=SEQUENCE
{
    dataType                DataType,
    payloadType            INTEGER(0..127) OPTIONAL,
    ...
}

MultiplePayloadStream     ::=SEQUENCE
{
    elements                SEQUENCE OF MultiplePayloadStreamElement,
    ...
}

MultiplePayloadStreamElement ::=SEQUENCE
{
    dataType                DataType,
    payloadType            INTEGER(0..127) OPTIONAL,
    ...
}

DepFECData                ::=CHOICE -- Déconseillé, à éviter
{
    rfc2733                SEQUENCE
    {
        mode                CHOICE
        {
            redundancyEncoding NULL,
            separateStream    CHOICE
            {
                differentPort SEQUENCE
                {
                    protectedSessionID INTEGER(1..255),

```

```

        protectedPayloadType  INTEGER(0..127) OPTIONAL,
        ...
    },
    samePort                  SEQUENCE
    {
        protectedPayloadType  INTEGER(0..127),
        ...
    },
    ...
},
...
},
...
}
}

FECDData ::= CHOICE
{
    rfc2733 SEQUENCE
    {
        protectedPayloadType  INTEGER(0..127),
        fecScheme              OBJECT IDENTIFIER OPTIONAL,
        pktMode                CHOICE
        {
            rfc2198coding      NULL,
            rfc2733sameport    SEQUENCE
            {
                ...
            },
            rfc2733diffport    SEQUENCE
            {
                protectedChannel  LogicalChannelNumber,
                ...
            },
            ...
        },
        ...
    },
    ...
}

TransportAddress ::=CHOICE
{
    unicastAddress      UnicastAddress,
    multicastAddress    MulticastAddress,
    ...
}

UnicastAddress ::=CHOICE
{
    ipAddress SEQUENCE
    {
        network          OCTET STRING (SIZE(4)),
        tsapIdentifier    INTEGER(0..65535),
        ...
    },
    ipXAddress SEQUENCE
    {
        node              OCTET STRING (SIZE(6)),
        netnum            OCTET STRING (SIZE(4)),
        tsapIdentifier    OCTET STRING (SIZE(2)),
        ...
    },
    ip6Address SEQUENCE

```

```

{
    network                OCTET STRING (SIZE(16)),
    tsapIdentifier         INTEGER(0..65535),
    ...
},
netBios                   OCTET STRING (SIZE(16)),
iPSourceRouteAddress     SEQUENCE
{
    routing                CHOICE
    {
        strict             NULL,
        loose              NULL
    },
    network                OCTET STRING (SIZE(4)),
    tsapIdentifier         INTEGER(0..65535),
    route                  SEQUENCE OF OCTET STRING (SIZE(4)),
    ...
},
...,
nsap                      OCTET STRING (SIZE(1..20)),
nonStandardAddress       NonStandardParameter
}

MulticastAddress          ::=CHOICE
{
    iPAddress              SEQUENCE
    {
        network            OCTET STRING (SIZE(4)),
        tsapIdentifier     INTEGER(0..65535),
        ...
    },
    iP6Address             SEQUENCE
    {
        network            OCTET STRING (SIZE(16)),
        tsapIdentifier     INTEGER(0..65535),
        ...
    },
    ...,
    nsap                    OCTET STRING (SIZE(1..20)),
    nonStandardAddress     NonStandardParameter
}

EncryptionSync           ::=SEQUENCE
-- fournit la nouvelle clé et
-- le point de synchronisation

nonStandard              NonStandardParameter OPTIONAL,
synchFlag                INTEGER(0..255) , -- une valeur plus
-- élevée peut être
-- requise pour H.324,
-- doit être le n° de
-- charge utile etc.
-- dynamique
-- pour H.323

h235Key                  OCTET STRING (SIZE(1..65535)),
-- valeur H.235 codée

escrowentry              SEQUENCE SIZE(1..256) OF EscrowData OPTIONAL,
...,
genericParameter        GenericParameter OPTIONAL
}

```

```

EscrowData ::=SEQUENCE
{
    escrowID OBJECT IDENTIFIER,
    escrowValue BIT STRING (SIZE(1..65535)),
    ...
}

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
            -- inverse
            ...,
            h2250LogicalChannelParameters H2250LogicalChannelParameters
        } OPTIONAL, -- non présent pour H.223
        ...,
        replacementFor LogicalChannelNumber OPTIONAL
    } OPTIONAL, -- non présent pour une demande de
    -- voie unidirectionnelle
    ...,
    separateStack NetworkAccessParameters OPTIONAL,
    -- permet au demandeur d'établir
    -- la pile
    forwardMultiplexAckParameters CHOICE
    {
        -- les paramètres H.222 ne sont jamais présents dans l'acquittement
        -- les paramètres H.223 ne sont jamais présents dans l'acquittement
        -- les paramètres V.76 ne sont jamais présents dans l'acquittement
        h2250LogicalChannelAckParameters H2250LogicalChannelAckParameters,
        ...,
    } OPTIONAL,
    encryptionSync EncryptionSync OPTIONAL
    -- utilisé uniquement
    -- par le terminal maître
}

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

```

```

        invalidDependentChannel      NULL,
        replacementForRejected       NULL,
        securityDenied                NULL
    },
    ...
}
OpenLogicalChannelConfirm           ::=SEQUENCE
{
    forwardLogicalChannelNumber     LogicalChannelNumber,
    ...
}

H2250LogicalChannelAckParameters   ::=SEQUENCE
{
    nonStandard                     SEQUENCE OF NonStandardParameter OPTIONAL,
    sessionID                        INTEGER(1..255) OPTIONAL,
    mediaChannel                     TransportAddress OPTIONAL,
    mediaControlChannel              TransportAddress OPTIONAL, -- canal RTCP
    dynamicRTTPayloadType            INTEGER(96..127) OPTIONAL,
                                     -- utilisé uniquement par le terminal
                                     -- maître ou l'entité MC
    ...,
    flowControlToZero                BOOLEAN,
    portNumber                       INTEGER (0..65535) OPTIONAL
}

CloseLogicalChannel                 ::=SEQUENCE
{
    forwardLogicalChannelNumber     LogicalChannelNumber,
    source                           CHOICE
    {
        user                          NULL,
        lcse                          NULL
    },
    ...,
    reason                           CHOICE
    {
        unknown                       NULL,
        reopen                        NULL,
        reservationFailure            NULL,
        ...
    }
}

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

RequestChannelClose                 ::=SEQUENCE
{
    forwardLogicalChannelNumber     LogicalChannelNumber,
    ...,
    qosCapability                    QOSCapability OPTIONAL,
    reason                           CHOICE
    {
        unknown                       NULL,
        normal                        NULL,
        reopen                        NULL,

```



```

        reservationFailure      NULL,
        ...
    }
}

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

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

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

-- =====
-- Définitions relatives au tableau de multiplexage 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), -- répétitions de type
        untilClosingFlag        NULL                -- utilisé pour le
                                                -- dernier élément
    }
}

MultiplexTableEntryNumber      ::=INTEGER (1..15)

```

```

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

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

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

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

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

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

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

RequestMultiplexEntryRejectionDescriptions ::=SEQUENCE
{
    multiplexTableEntryNumber MultiplexTableEntryNumber,
    cause                     CHOICE
    {

```

```

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

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

-- =====
-- Définitions relatives à la demande de mode
-- =====

-- RequestMode est une liste, établie dans l'ordre de préférence, des modes
-- qu'un terminal souhaiterait recevoir.

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 relatives à la demande de mode: description de mode
-- =====

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

```

```

ModeElementType                                     ::= CHOICE
{
    nonStandard                                     NonStandardParameter,
    videoMode                                       VideoMode,
    audioMode                                       AudioMode,
    dataMode                                        DataMode,
    encryptionMode                                 EncryptionMode,
    ...,
    h235Mode                                        H235Mode,
    multiplexedStreamMode                          MultiplexedStreamParameter,
    redundancyEncodingDTMode                       RedundancyEncodingDTMode,
    multiplePayloadStreamMode                      MultiplePayloadStreamMode,
    depFecMode                                      DepFECMode, -- déconseillé, à éviter
    fecMode FECMode
}

ModeElement                                         ::= SEQUENCE
{
    type                                           ModeElementType,

    h223ModeParameters                             H223ModeParameters OPTIONAL,
    ...,
    v76ModeParameters                             V76ModeParameters OPTIONAL,
    h2250ModeParameters                           H2250ModeParameters OPTIONAL,
    genericModeParameters                         GenericCapability OPTIONAL,
    multiplexedStreamModeParameters              MultiplexedStreamModeParameters OPTIONAL,
    logicalChannelNumber                          LogicalChannelNumber OPTIONAL
}

H235Mode                                           ::= SEQUENCE
{
    encryptionAuthenticationAndIntegrity          EncryptionAuthenticationAndIntegrity,

    mediaMode                                     CHOICE
    {
        nonStandard                               NonStandardParameter,
        videoMode                                 VideoMode,
        audioMode                                 AudioMode,
        dataMode                                  DataMode,
        ...
    },
    ...
}

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

RedundancyEncodingDTMode                          ::= SEQUENCE
{
    redundancyEncodingMethod                     RedundancyEncodingMethod,
    primary                                       RedundancyEncodingDTModeElement,
    secondary                                     SEQUENCE OF RedundancyEncodingDTModeElement,
    ...
}

RedundancyEncodingDTModeElement                   ::= SEQUENCE
{
    type                                           CHOICE
    {
        nonStandard                               NonStandardParameter,
        videoMode                                 VideoMode,

```

```

        audioMode      AudioMode,
        dataMode       DataMode,
        encryptionMode EncryptionMode,
        h235Mode       H235Mode,
        ...,
        fecMode        FECMode
    },
    ...
}

MultiplePayloadStreamMode ::=SEQUENCE
{
    elements            SEQUENCE OF MultiplePayloadStreamElementMode,
    ...
}

MultiplePayloadStreamElementMode ::=SEQUENCE
{
    type               ModeElementType,
    ...
}

DepFECMode ::=CHOICE -- déconseillé, à éviter
{
    rfc2733Mode       SEQUENCE
    {
        mode          CHOICE
        {
            redundancyEncoding NULL,
            separateStream    CHOICE
            {
                differentPort SEQUENCE
                {
                    protectedSessionID INTEGER(1..255),
                    protectedPayloadType INTEGER(0..127) OPTIONAL,
                    ...
                },
                samePort      SEQUENCE
                {
                    protectedType ModeElementType,
                    ...
                },
                ...
            },
            ...
        },
        ...
    },
    ...
}

FECMode ::= SEQUENCE
{
    protectedElement ModeElementType,
    fecScheme        OBJECT IDENTIFIER OPTIONAL,
                    -- désigne le système de codage
    rfc2733Format    CHOICE
    {
        rfc2733rfc2198 MaxRedundancy, -- redondance RFC 2198
        rfc2733sameport MaxRedundancy,
                    -- paquet séparé, même port
        rfc2733diffport MaxRedundancy
                    -- paquet séparé, port séparé
    }
}

```

```

    } OPTIONAL,
    ...
}
H223ModeParameters ::=SEQUENCE
{
    adaptationLayerType CHOICE
    {
        nonStandard NonStandardParameter,
        allFramed NULL,
        allNotFramed NULL,
        al2WithoutSequenceNumbers NULL,
        al2WithSequenceNumbers NULL,
        al3 SEQUENCE
        {
            controlFieldOctets INTEGER(0..2),
            sendBufferSize INTEGER(0..16777215) -- unité: octets
        },
        ...,
        al1M H223AL1MParameters,
        al2M H223AL2MParameters,
        al3M H223AL3MParameters
    },
    segmentableFlag BOOLEAN,
    ...
}

```

```

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

```

```

H2250ModeParameters ::=SEQUENCE
{
    redundancyEncodingMode RedundancyEncodingMode OPTIONAL,
    ...
}

```

```

RedundancyEncodingMode ::=SEQUENCE
{
    redundancyEncodingMethod RedundancyEncodingMethod,
    secondaryEncoding CHOICE
    {
        nonStandard NonStandardParameter,
        audioData AudioMode,
        ...
    } OPTIONAL,
    ...
}

```

```

-- =====
-- Définitions relatives à la demande de mode: modes vidéo
-- =====

```

```

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

```

```

        is11172VideoMode          IS11172VideoMode,
        ...,
        genericVideoMode          GenericCapability
    }

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,
        ...
    },
    videoBitRate                   INTEGER(0..1073741823) OPTIONAL,
        -- unité: 400 bit/s
    vbvBufferSize                  INTEGER(0..262143) OPTIONAL,
        -- unité: 16 384 bits
    samplesPerLine                  INTEGER(0..16383) OPTIONAL,
        -- unité: échantillons/lignes
    linesPerFrame                   INTEGER(0..16383) OPTIONAL,
        -- unité: lignes/trame
    framesPerSecond                 INTEGER(0..15) OPTIONAL,
        -- frame_rate_code
    luminanceSampleRate             INTEGER(0..4294967295) OPTIONAL,
        -- unité: échantillons/s
    ...
}

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

```

```

    advancedPrediction          BOOLEAN,
    pbFrames                    BOOLEAN,
    ...,

    errorCompensation          BOOLEAN,
    enhancementLayerInfo       EnhancementLayerInfo OPTIONAL,
    h263Options                 H263Options OPTIONAL
}

IS11172VideoMode              ::=SEQUENCE
{
    constrainedBitstream        BOOLEAN,
    videoBitRate                INTEGER(0..1073741823) OPTIONAL,
                                -- unité: 400 bit/s
    vbvBufferSize              INTEGER(0..262143) OPTIONAL,
                                -- unité: 16 384 bits
    samplesPerLine              INTEGER(0..16383) OPTIONAL,
                                -- unité: échantillons/lignes

    linesPerFrame               INTEGER(0..16383) OPTIONAL,
                                -- unité:lignes par trame
    pictureRate                 INTEGER(0..15) OPTIONAL,
    luminanceSampleRate         INTEGER(0..4294967295) OPTIONAL,
                                -- unité: échantillons/s
    ...
}

```

```

-- =====
-- Définitions relatives à la demande de mode: 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,
    g729AnnexA                  NULL,

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

    is11172AudioMode            IS11172AudioMode,
    is13818AudioMode            IS13818AudioMode,

    ...,

    g729wAnnexB                 INTEGER(1..256),
    g729AnnexAwAnnexB           INTEGER(1..256),
    g7231AnnexCMode              G7231AnnexCMode,
    gsmFullRate                  GSMAudioCapability,
    gsmHalfRate                  GSMAudioCapability,
}

```



```

    gsmEnhancedFullRate      GSMAudioCapability,
    genericAudioMode         GenericCapability,
    g729Extensions           G729Extensions,
    vbd                      VBDMODE
}

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

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

VBDMode                             ::= SEQUENCE
{
    type                            AudioMode,           -- ne doit pas être "vbd"
    ...
}

-- =====
-- Définitions relatives à la demande de mode: modes données
-- =====

DataMode                             ::= SEQUENCE
{
    application                     CHOICE
    {
        nonStandard                NonStandardParameter,
        t120                        DataProtocolCapability,
        dsm-cc                      DataProtocolCapability,
        userData                    DataProtocolCapability,
        t84                          DataProtocolCapability,
        t434                        DataProtocolCapability,
        h224                        DataProtocolCapability,
        nlpid                       SEQUENCE
        {
            nlpidProtocol           DataProtocolCapability,
            nlpidData               OCTET STRING
        },
        dsvdControl                 NULL,
        h222DataPartitioning        DataProtocolCapability,
        ...,
        t30fax                      DataProtocolCapability,
        t140                        DataProtocolCapability,
        t38fax                      SEQUENCE
        {
            t38FaxProtocol          DataProtocolCapability,
            t38FaxProfile           T38FaxProfile
        },
        genericDataMode             GenericCapability
    }
}

```

```

    },
    bitRate                INTEGER (0..4294967295), -- unité: 100 bit/s
    ...
}

-- =====
-- Définitions relatives à la demande de mode: modes de chiffrement
-- =====

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

-- =====
-- Définitions relatives au temps de propagation aller-retour
-- =====

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

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

-- =====
-- Définitions relatives à la boucle de maintenance
-- =====

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

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

MaintenanceLoopReject     ::=SEQUENCE
{
    type                  CHOICE

```

```

    {
        systemLoop                NULL,
        mediaLoop                  LogicalChannelNumber,
        logicalChannelLoop        LogicalChannelNumber,
        ...
    },
    cause                          CHOICE
    {
        canNotPerformLoop        NULL,
        ...
    },
    ...
}

MaintenanceLoopOffCommand        ::=SEQUENCE
{
    ...
}

-- =====
-- Définitions relatives aux modes de communication
-- =====

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

CommunicationModeRequest          ::=SEQUENCE
{
    ...
}

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

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

    terminalLabel                 TerminalLabel OPTIONAL,
    -- s'il est absent, tous les
    -- participants à la
    -- conférence sont visés
    sessionDescription            BMPString (SIZE(1..128)) ,
    -- session de base ISO/CEI 10646-1
    -- (Unicode)
    dataType                      CHOICE
    {
        videoData                 VideoCapability,
        audioData                 AudioCapability,
        data                      DataApplicationCapability,
        ...
    },
    mediaChannel                  TransportAddress OPTIONAL,
    mediaGuaranteedDelivery       BOOLEAN OPTIONAL,
    mediaControlChannel           TransportAddress OPTIONAL,
    -- canal RTCP inverse

```

```

mediaControlGuaranteedDelivery BOOLEAN OPTIONAL,
...
redundancyEncoding RedundancyEncoding OPTIONAL,
sessionDependency INTEGER (1..255) OPTIONAL,
destination TerminalLabel OPTIONAL
}

-- =====
-- Définitions relatives aux demandes pour une conférence
-- =====

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

    makeMeChair NULL, -- identique à H.230 CCA (term->MC)
    cancelMakeMeChair NULL, -- identique à H.230 CIS (term->MC)

    dropTerminal TerminalLabel, -- identique à H.230 CCD(term->MC)

    requestTerminalID TerminalLabel, -- identique à TCP (term->MC)

    enterH243Password NULL, -- identique à H.230 TCS1(MC->term)
    enterH243TerminalID NULL, -- identique à H.230 TCS2/TCI
    -- (MC->term)
    enterH243ConferenceID NULL, -- identique à H.230 TCS3
    -- (MC->term)
    ...
    enterExtensionAddress NULL, -- identique à H.230 TCS4
    -- (GW->term)
    requestChairTokenOwner NULL, -- identique à H.230 TCA (term->MC)
    requestTerminalCertificate SEQUENCE
    {
        terminalLabel TerminalLabel OPTIONAL,
        certSelectionCriteria CertSelectionCriteria OPTIONAL,
        sRandom INTEGER (1..4294967295) OPTIONAL,
        -- épreuve imposée par le demandeur
        ...
    },
    broadcastMyLogicalChannel LogicalChannelNumber, -- similaire à H.230 MCV
    makeTerminalBroadcaster TerminalLabel, -- similaire à H.230 VCB
    sendThisSource TerminalLabel, -- similaire à H.230 VCS
    requestAllTerminalIDs NULL,
    remoteMCRequest RemoteMCRequest
}

CertSelectionCriteria ::=SEQUENCE SIZE (1..16) OF Criteria

Criteria ::=SEQUENCE
{
    field OBJECT IDENTIFIER, -- peut comporter le
    -- type de certificat
    value OCTET STRING (SIZE(1..65535)),
    ...
}

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

McuNumber ::=INTEGER(0..192)

```

TerminalNumber ::=INTEGER(0..192)

-- Définitions relatives aux réponses pour une conférence

```
ConferenceResponse ::=CHOICE
{
  mCTerminalIDResponse SEQUENCE -- réponse à TCP
  { -- (identique à TIP)
    -- envoyée par l'entité MC
    -- uniquement
    terminalLabel TerminalLabel,
    terminalID TerminalID,
    ...
  },
  terminalIDResponse SEQUENCE -- réponse à TCS2 ou TCI
  { -- identique à IIS
    -- (term->MC)
    terminalLabel TerminalLabel,
    terminalID TerminalID,
    ...
  },
  conferenceIDResponse SEQUENCE -- réponse à TCS3
  { -- identique à IIS
    -- (term->MC)
    terminalLabel TerminalLabel,
    conferenceID ConferenceID,
    ...
  },
  passwordResponse SEQUENCE -- réponse à TCS1
  { -- identique à IIS
    -- (term->MC)
    terminalLabel TerminalLabel,
    password Password,
    ...
  },
  terminalListResponse SET SIZE (1..256) OF TerminalLabel,
  videoCommandReject NULL, -- identique à H.230 VCR
  terminalDropReject NULL, -- identique à H.230 CIR
  makeMeChairResponse CHOICE -- identique à H.230 CCR
  {
    grantedChairToken NULL, -- identique à H.230 CIT
    deniedChairToken NULL, -- identique à H.230 CCR
    ...
  },
  ...,
  extensionAddressResponse SEQUENCE -- réponse à TCS4
  {
    extensionAddress TerminalID, -- identique à IIS (term->GW)
    ...
  },
  chairTokenOwnerResponse SEQUENCE -- réponse à TCA
  { -- (identique à TIR)
    -- envoyée par le MC
    -- uniquement
    terminalLabel TerminalLabel,
    terminalID TerminalID,
    ...
  },
}
```

```

terminalCertificateResponse SEQUENCE
{
    terminalLabel TerminalLabel OPTIONAL,
    certificateResponse OCTET STRING (SIZE(1..65535)) OPTIONAL,
    ...
},
broadcastMyLogicalChannelResponse CHOICE
{
    grantedBroadcastMyLogicalChannel NULL, -- similaire à H.230 MVA
    deniedBroadcastMyLogicalChannel NULL, -- similaire à H.230 MVR
    ...
},
makeTerminalBroadcasterResponse CHOICE
{
    grantedMakeTerminalBroadcaster NULL,
    deniedMakeTerminalBroadcaster NULL,
    ...
},
sendThisSourceResponse CHOICE
{
    grantedSendThisSource NULL,
    deniedSendThisSource NULL,
    ...
},
requestAllTerminalIDsResponse RequestAllTerminalIDsResponse,
remoteMCResponse RemoteMCResponse
}

```

```

TerminalID ::=OCTET STRING (SIZE(1..128)) -- selon H.230
ConferenceID ::=OCTET STRING (SIZE(1..32))
Password ::=OCTET STRING (SIZE(1..32))

```

```

RequestAllTerminalIDsResponse ::=SEQUENCE
{
    terminalInformation SEQUENCE OF TerminalInformation,
    ...
}

```

```

TerminalInformation ::=SEQUENCE
{
    terminalLabel TerminalLabel,
    terminalID TerminalID,
    ...
}

```

```

-- =====
-- Définitions relatives à l'entité MC distante
-- =====

```

```

RemoteMCRequest ::=CHOICE
{
    masterActivate NULL,
    slaveActivate NULL,
    deactivate NULL,
    ...
}

```

```

RemoteMCResponse ::=CHOICE
{
    accept NULL,
    reject CHOICE
    {
        unspecified NULL,
    }
}

```

```

        functionNotSupported      NULL,
        ...
    },
    ...
}

-- =====
-- Définitions relatives aux multiliaisons
-- =====

MultilinkRequest ::=CHOICE
{
    nonStandard      NonStandardMessage,

    callInformation  SEQUENCE
    {
        maxNumberOfAdditionalConnections  INTEGER (1..65535),
        ...
    },

    addConnection    SEQUENCE
    {
        sequenceNumber      SequenceNumber, -- Identificateur
                                -- unique de demande
        dialingInformation  DialingInformation,
        ...
    },

    removeConnection SEQUENCE
    {
        connectionIdentifier  ConnectionIdentifier,
        ...
    },

    maximumHeaderInterval SEQUENCE
    {
        requestType          CHOICE
        {
            currentIntervalInformation  NULL,
            requestedInterval          INTEGER (0..65535), -- Intervalle
                                                -- d'en-tête, maximal
                                                -- millisecondes
            ...
        },
        ...
    },
    ...
}

MultilinkResponse ::=CHOICE
{
    nonStandard      NonStandardMessage,

    callInformation  SEQUENCE
    {
        dialingInformation  DialingInformation,
        callAssociationNumber  INTEGER (0..4294967295),
        ...
    },

    addConnection    SEQUENCE
    {
        sequenceNumber      SequenceNumber, -- Même valeur que
                                                -- dans la demande
    }
}

```



```

        responseCode                               CHOICE
        {
            accepted                               NULL,
            rejected                               CHOICE
            {
                connectionsNotAvailable           NULL, -- pour n'importe quelle
                                                    -- raison technique
                userRejected                     NULL,
                ...
            },
            ...
        },
        ...
    },
    ...
}

removeConnection                               SEQUENCE
{
    connectionIdentifier                         ConnectionIdentifier,
    ...
},

maximumHeaderInterval                         SEQUENCE
{
    currentInterval                             INTEGER (0..65535), -- Intervalle
                                                    -- d'en-tête
                                                    -- maximal,
                                                    -- millisecondes
    ...
},
...
}

MultilinkIndication                           ::=CHOICE
{
    nonStandard                                 NonStandardMessage,

    crcDesired                                 SEQUENCE
    {
        ...
    },

    excessiveError                             SEQUENCE
    {
        connectionIdentifier                   ConnectionIdentifier,
        ...
    },
    ...
}

DialingInformation                             ::= CHOICE
{
    nonStandard                                 NonStandardMessage,

    differential                               SET SIZE (1..65535) OF DialingInformationNumber,
                                                    -- liste des numéros pour tous les canaux
                                                    -- supplémentaires;
                                                    -- uniquement les chiffres les moins
                                                    -- significatifs différents des
                                                    -- chiffres du numéro correspondant
                                                    -- au canal initial
}

```

```

        infoNotAvailable          INTEGER (1..65535),
                                   -- nombre maximal de canaux
                                   -- supplémentaires
    ...
}

DialingInformationNumber        ::= SEQUENCE
{
    networkAddress              NumericString (SIZE (0..40)),
    subAddress                  IA5String (SIZE (1..40)) OPTIONAL,
    networkType                 SET SIZE (1..255) OF DialingInformationNetworkType,
    ...
}

DialingInformationNetworkType  ::= CHOICE
{
    nonStandard                 NonStandardMessage,
    n-isdn                     NULL,
    gstn                        NULL,
    ...,
    mobile                      NULL
}

ConnectionIdentifier           ::= SEQUENCE
{
    channelTag                  INTEGER (0..4294967295), -- tiré de H.226
    sequenceNumber              INTEGER (0..4294967295), -- tiré de H.226
    ...
}

-- =====
-- Définitions relatives à la modification de débit d'une voie logique
-- =====

MaximumBitRate                 ::=INTEGER (0.. 4294967295) -- unité: 100 bit/s

LogicalChannelRateRequest      ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    logicalChannelNumber        LogicalChannelNumber,
    maximumBitRate              MaximumBitRate,
    ...
}

LogicalChannelRateAcknowledge  ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    logicalChannelNumber        LogicalChannelNumber,
    maximumBitRate              MaximumBitRate,
    ...
}

LogicalChannelRateReject      ::=SEQUENCE
{
    sequenceNumber              SequenceNumber,
    logicalChannelNumber        LogicalChannelNumber,
    rejectReason                LogicalChannelRateRejectReason,
    currentMaximumBitRate       MaximumBitRate OPTIONAL,
    ...
}

LogicalChannelRateRejectReason ::=CHOICE
{
    undefinedReason             NULL,

```

```

        insufficientResources          NULL,
        ...
    }
LogicalChannelRateRelease             ::=SEQUENCE
{
    ...
}

-- =====
-- Définitions relatives aux 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
                                        CapabilityTableEntryNumber OPTIONAL,

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

        ...
    },
    genericRequest                     NULL,
    ...
}

-- =====
-- Message de commande: chiffrement
-- =====

EncryptionCommand                    ::=CHOICE
{
    encryptionSE                       OCTET STRING, -- selon H.233, mais
                                                -- sans protection
    encryptionIVRequest                NULL,         -- contre les erreurs
                                                -- demande d'un
                                                -- nouvel IV

    encryptionAlgorithmID              SEQUENCE
    {
        h233AlgorithmIdentifier        SequenceNumber,
        associatedAlgorithm              NonStandardParameter
    },
    ...
}

-- =====
-- Message de commande: commande de débit
-- =====

FlowControlCommand                   ::=SEQUENCE
{
    scope CHOICE
    {
        logicalChannelNumber           LogicalChannelNumber,
        resourceID                     INTEGER (0..65535),
    }
}

```

```

        wholeMultiplex          NULL
    },
    restriction                  CHOICE
    {
        maximumBitRate          INTEGER (0..16777215), -- unité: 100 bit/s
        noRestriction           NULL
    },
    ...
}

```

```

-- =====
-- Message de commande: changement ou fin de session
-- =====

```

```

EndSessionCommand              ::=CHOICE
{
    nonStandard                  NonStandardParameter,

    disconnect                   NULL,

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

    ...,

    isdnOptions                  CHOICE
    {
        telephonyMode            NULL,
        v140                      NULL,
        terminalOnHold            NULL,
        ...
    }
}

```

```

-- =====
-- Message de commande: commandes de conférence
-- =====

```

```

ConferenceCommand              ::=CHOICE
{
    broadcastMyLogicalChannel     LogicalChannelNumber, -- similaire à H.230 MCV
    cancelBroadcastMyLogicalChannel LogicalChannelNumber, -- similaire à
                                                -- H.230/annul-MCV

    makeTerminalBroadcaster       TerminalLabel, -- identique
                                                -- à H.230 VCB
    cancelMakeTerminalBroadcaster NULL,          -- identique
                                                -- à H.230/annul-VCB

    sendThisSource                TerminalLabel, -- identique à
                                                -- H.230/VCS
    cancelSendThisSource          NULL,          -- identique
                                                -- à H.230/annul-VCS

    dropConference                 NULL,          -- identique à H.230 CCK
    ...,
    substituteConferenceIDCommand SubstituteConferenceIDCommand
}

```

```

SubstituteConferenceIDCommand      ::=SEQUENCE
{
    conferenceIdentifier            OCTET STRING (SIZE(16)),
    ...
}

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

EncryptionUpdateDirection          ::= CHOICE
{
    masterToSlave                   NULL,
    slaveToMaster                   NULL,
    ...
}

MiscellaneousCommand               ::=SEQUENCE
{
    logicalChannelNumber            LogicalChannelNumber,
    type                            CHOICE
    {
        equaliseDelay               NULL,           -- identique à H.230 ACE
        zeroDelay                   NULL,           -- identique à H.230 ACZ
        multipointModeCommand       NULL,
        cancelMultipointModeCommand NULL,
        videoFreezePicture           NULL,
        videoFastUpdatePicture       NULL,

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

        videoTemporalSpatialTradeOff INTEGER (0..31), -- commande une
                                                    -- valeur de compromis

        videoSendSyncEveryGOB       NULL,
        videoSendSyncEveryGOBCancel NULL,

        ...,
        videoFastUpdateMB           SEQUENCE
        {
            firstGOB                 INTEGER (0..255) OPTIONAL,
            firstMB                   INTEGER (1..8192) OPTIONAL,
            numberOfMBs                INTEGER (1..8192),
            ...
        },
        maxH223MUXPDUsize           INTEGER(1..65535), -- unité: octets
        encryptionUpdate             EncryptionSync,
        encryptionUpdateRequest      EncryptionUpdateRequest,
        switchReceiveMediaOff        NULL,
        switchReceiveMediaOn         NULL,

        progressiveRefinementStart   SEQUENCE
        {
            repeatCount               CHOICE
            {
                doOneProgression      NULL,
                doContinuousProgressions NULL,
                doOneIndependentProgression NULL,
            }
        }
    }
}

```

```

        doContinuousIndependentProgressions          NULL,
        ...,
    },
    ...,
},
progressiveRefinementAbortOne                      NULL,
progressiveRefinementAbortContinuous              NULL,

videoBadMBs                                       SEQUENCE
{
    firstMB                                         INTEGER (1..9216),
    numberOfMBs                                    INTEGER (1..9216),
    temporalReference                              INTEGER (0..1023),
    ...,
},
lostPicture                                       SEQUENCE OF PictureReference,
lostPartialPicture                               SEQUENCE
{
    pictureReference                               PictureReference,
    firstMB                                         INTEGER (1..9216),
    numberOfMBs                                    INTEGER (1..9216),
    ...,
},
recoveryReferencePicture                         SEQUENCE OF PictureReference,
encryptionUpdateCommand                          SEQUENCE -- pour mise à jour de clé
                                                -- acquittée dans H.235V3
{
    encryptionSync                                EncryptionSync,
    multiplePayloadStream                         MultiplePayloadStream OPTIONAL,
    ...,
},
encryptionUpdateAck                              SEQUENCE
{
    synchFlag                                     INTEGER (0..255),
    ...,
}
},
...,
direction                                       EncryptionUpdateDirection OPTIONAL
}

KeyProtectionMethod                             ::=SEQUENCE -- indique comment
                                                -- protéger la nouvelle clé
{
    secureChannel                                 BOOLEAN,
    sharedSecret                                 BOOLEAN,
    certProtectedKey                             BOOLEAN,
    ...,
}

EncryptionUpdateRequest                         ::=SEQUENCE
{
    keyProtectionMethod                          KeyProtectionMethod OPTIONAL,
    ...,
    synchFlag                                    INTEGER (0..255) OPTIONAL
}

PictureReference                               ::=CHOICE
{
    pictureNumber                                INTEGER (0..1023),
    longTermPictureIndex                         INTEGER (0..255),
    ...,
}

```

```

-- =====
-- Message de commande: reconfiguration du multiplex H.223
-- =====

```

```

H223MultiplexReconfiguration ::=CHOICE
{
    h223ModeChange CHOICE
    {
        toLevel0 NULL,
        toLevel1 NULL,
        toLevel2 NULL,
        toLevel2withOptionalHeader NULL,
        ...
    },
    h223AnnexADoubleFlag CHOICE
    {
        start NULL,
        stop NULL,
        ...
    },
    ...
}

```

```

-- =====
-- Message de commande: commande d'une nouvelle voie virtuelle ATM
-- =====

```

```

NewATMVCCCommand ::=SEQUENCE
{
    resourceID INTEGER(0..65535),
    bitRate INTEGER(1..65535), -- unité: 64 kbit/s
    bitRateLockedToPCRClock BOOLEAN,
    bitRateLockedToNetworkClock BOOLEAN,
    aal CHOICE
    {
        aal1 SEQUENCE
        {
            clockRecovery CHOICE
            {
                nullClockRecovery NULL,
                srtsClockRecovery NULL,
                adaptiveClockRecovery NULL,
                ...
            },
            errorCorrection CHOICE
            {
                nullErrorCorrection NULL,
                longInterleaver NULL,
                shortInterleaver NULL,
                errorCorrectionOnly NULL,
                ...
            },
            structuredDataTransfer BOOLEAN,
            partiallyFilledCells BOOLEAN,
            ...
        },
        aal5 SEQUENCE
        {
            forwardMaximumSDUSize INTEGER(0..65535), -- unité: octets

```

```

        backwardMaximumSDUSize    INTEGER (0..65535),  -- unité: octets
        ...
    },
    ...
},
multiplex                        CHOICE
{
    noMultiplex                  NULL,
    transportStream              NULL,
    programStream                NULL,
    ...
},
reverseParameters                SEQUENCE
{
    bitRate                      INTEGER(1..65535),  -- unité: 64 kbit/s
    bitRateLockedToPCRClock      BOOLEAN,
    bitRateLockedToNetworkClock  BOOLEAN,
    multiplex                     CHOICE
    {
        noMultiplex              NULL,
        transportStream          NULL,
        programStream            NULL,
        ...
    },
    ...
},
...
}

```

```

-- =====
-- Message de commande: commande de reconfiguration de multilaisons mobiles
-- =====

```

```

MobileMultilinkReconfigurationCommand ::=SEQUENCE
{
    sampleSize                    INTEGER (1..255),
    samplesPerFrame               INTEGER (1..255),
    status                        CHOICE
    {
        synchronized              NULL,
        reconfiguration           NULL,
        ...
    },
    ...
}

```

```

-- =====
-- Définitions relatives aux messages d'indication
-- =====

```

```

-- =====
-- Message d'indication: fonction non comprise
-- =====

```

-- Permet de retourner une demande, réponse ou commande non reconnue

```

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

```



```

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

```

```

-- Permet de retourner une demande, réponse ou commande non reconnue

```

```

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

```

```

-- =====
-- Message d'indication: conférence
-- =====

```

```

ConferenceIndication ::=CHOICE
{
    sbeNumber INTEGER (0..9), -- identique à H.230
    -- SBE Number

    terminalNumberAssign TerminalLabel, -- identique à H.230 TIA

    terminalJoinedConference TerminalLabel, -- identique à H.230 TIN

    terminalLeftConference TerminalLabel, -- identique à H.230 TID

    seenByAtLeastOneOther NULL, -- identique à H.230 MIV
    cancelSeenByAtLeastOneOther NULL, -- identique à
    -- H.230/annul-MIV

    seenByAll NULL, -- similaire à H.230 MIV
    cancelSeenByAll NULL, -- similaire à H.230 MIV

    terminalYouAreSeeing TerminalLabel, -- identique à H.230 VIN

    requestForFloor NULL, -- identique à H.230 TIF

    ...,
    withdrawChairToken NULL, -- identique à H.230 CCR
    -- MC-> président

    floorRequested TerminalLabel, -- identique à H.230 TIF
    -- MC-> président

    terminalYouAreSeeingInSubPictureNumber TerminalYouAreSeeingInSubPictureNumber,
    videoIndicateCompose VideoIndicateCompose
}

```

```

TerminalYouAreSeeingInSubPictureNumber ::= SEQUENCE
{
    terminalNumber TerminalNumber,
    subPictureNumber INTEGER (0..255),
    ...
}

```

```

VideoIndicateCompose ::= SEQUENCE
{

```

```

        compositionNumber          INTEGER (0..255),
        ...
    }

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

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

        multipointConference      NULL,
        cancelMultipointConference NULL,

        multipointZeroComm        NULL,      -- identique à H.230 MIZ
        cancelMultipointZeroComm  NULL,      -- identique à H.230/annul-MIZ

        multipointSecondaryStatus  NULL,     -- identique à H.230 MIS
        cancelMultipointSecondaryStatus NULL, -- identique à H.230 cancel MIS

        videoIndicateReadyToActivate NULL,   -- identique à H.230 VIR

        videoTemporalSpatialTradeOff  INTEGER (0..31), -- indique le compromis
                                                -- en cours

        ...,
        videoNotDecodedMBS          SEQUENCE
        {
            firstMB                  INTEGER (1..8192),
            numberOfMBS              INTEGER (1..8192),
            temporalReference         INTEGER (0..255),
            ...
        },
        transportCapability          TransportCapability
    },
    ...
}

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

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

```

```

-- =====
-- Message d'indication: décalage temporel de voies logiques H.223
-- =====

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

-- =====
-- Message d'indication: décalage temporel maximal de voies logiques H.225.0
-- =====

H2250MaximumSkewIndication                       ::=SEQUENCE
{
    logicalChannelNumber1                       LogicalChannelNumber,
    logicalChannelNumber2                       LogicalChannelNumber,
    maximumSkew                                INTEGER (0..4095), -- unité: millisecondes
    ...
}

-- =====
-- Message d'indication: indication de l'emplacement de l'entité MC
-- =====

MCLocationIndication                             ::=SEQUENCE
{
    signalAddress                              TransportAddress,
                                                -- adresse de signalisation
                                                -- d'appel H.323 de l'entité MC
    ...
}

-- =====
-- Message d'indication: identification du fabricant
-- =====

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

-- =====
-- Message d'indication: indication de nouvelle voie virtuelle ATM
-- =====

NewATMVCIndication                               ::=SEQUENCE
{
    resourceID                                 INTEGER(0..65535),
    bitRate                                    INTEGER(1..65535),   -- unité: 64 kbit/s
    bitRateLockedToPCRClock                   BOOLEAN,
    bitRateLockedToNetworkClock               BOOLEAN,
    aal                                        CHOICE
    {

```

```

aall                               SEQUENCE
{
  clockRecovery                     CHOICE
  {
    nullClockRecovery               NULL,
    srtsClockRecovery               NULL,
    adaptiveClockRecovery           NULL,
    ...
  },
  errorCorrection                    CHOICE
  {
    nullErrorCorrection              NULL,
    longInterleaver                  NULL,
    shortInterleaver                 NULL,
    errorCorrectionOnly              NULL,
    ...
  },
  structuredDataTransfer             BOOLEAN,
  partiallyFilledCells               BOOLEAN,
  ...
},
aal5                                 SEQUENCE
{
  forwardMaximumSDUSize              INTEGER (0..65535),  -- unité: octets
  backwardMaximumSDUSize             INTEGER (0..65535),  -- unité: octets
  ...
},
...
},
multiplex                           CHOICE
{
  noMultiplex                        NULL,
  transportStream                    NULL,
  programStream                       NULL,
  ...
},
...,
reverseParameters                   SEQUENCE
{
  bitRate                            INTEGER(1..65535),    -- unité: 64 kbit/s
  bitRateLockedToPCRClock            BOOLEAN,
  bitRateLockedToNetworkClock        BOOLEAN,
  multiplex                           CHOICE
  {
    noMultiplex                       NULL,
    transportStream                    NULL,
    programStream                       NULL,
    ...
  },
  ...
}
}

```

```

-- =====
-- Message d'indication: entrées de l'utilisateur
-- =====

```

```

IV8                                 ::= OCTET STRING (SIZE(8))
-- valeur initiale pour
-- blocs chiffrants de 64 bits
IV16                                ::= OCTET STRING (SIZE(16))
-- valeur initiale pour
-- blocs chiffrants de 128 bits

```

```

Params ::= SEQUENCE
{
    iv8 IV8 OPTIONAL, -- vecteur d'initialisation
                    -- de 8 octets
    iv16 IV16 OPTIONAL, -- vecteur d'initialisation
                    -- de 16 octets
    iv OCTET STRING OPTIONAL,
                    -- vecteur d'initialisation
                    -- de longueur arbitraire
    ...
}

UserInputIndication ::= CHOICE
{
    nonStandardNonStandardParameter,
    alphanumericGeneralString,
    ...,
    userInputSupportIndication CHOICE
    {
        nonStandardNonStandardParameter,
        basicString NULL, -- indique une chaîne de base non
                        -- sécurisée
        ia5String IA5String NULL, -- indique une chaîne IA5 non
                        -- sécurisée
        generalString NULL, -- indique une chaîne générale
                        -- non sécurisée
        ...,
        encryptedBasicString NULL, -- indique une chaîne de base
                        -- chiffrée
        encryptedIA5String NULL, -- indique une chaîne IA5
                        -- chiffrée
        encryptedGeneralString NULL -- indique une chaîne générale
                        -- chiffrée
    },
    signal SEQUENCE
    {
        signalType IA5String (SIZE (1) ^ FROM ("0123456789#*ABCD!")),
                    -- contient le caractère fictif "!" si
                    -- encryptedSignalType
                    -- est en cours d'utilisation
        duration INTEGER (1..65535) OPTIONAL,
                    -- millisecondes
        rtp SEQUENCE
        {
            timestamp INTEGER (0..4294967295) OPTIONAL,
            expirationTime INTEGER (0..4294967295) OPTIONAL,
            logicalChannelNumber LogicalChannelNumber,
            ...
        } OPTIONAL,
        ...,
        rtpPayloadIndication NULL OPTIONAL,
        paramS Params OPTIONAL, -- tous paramètres
                    -- "runtime"
    },
    encryptedSignalType OCTET STRING (SIZE(1)) OPTIONAL,
                    -- signalType chiffré
    algorithmOID OBJECT IDENTIFIER OPTIONAL
},
    signalUpdate SEQUENCE
    {
        duration INTEGER (1..65535), -- millisecondes
        rtp SEQUENCE
        {
            logicalChannelNumber LogicalChannelNumber,
            ...
        }
    }
}

```

```

    } OPTIONAL,
    ...
},
extendedAlphanumeric SEQUENCE
{
    alphanumeric GeneralString,
        -- contient une chaîne vide
        -- si encryptedAlphanumeric
        -- est en cours d'utilisation

    rtpPayloadIndication NULL OPTIONAL,
    ...,
    encryptedAlphanumeric SEQUENCE
    {
        algorithmOID OBJECT IDENTIFIER,
        paramS Params OPTIONAL, -- tous paramètres "runtime"
        encrypted OCTET STRING, -- chaîne générale chiffrée
        ...
    } OPTIONAL
},
encryptedAlphanumeric SEQUENCE
{
    algorithmOID OBJECT IDENTIFIER,
    paramS Params OPTIONAL, -- tous paramètres "runtime"
    encrypted OCTET STRING, -- chaîne de base chiffrée
    ...}
}

-- =====
-- Message d'indication: commande de débit
-- =====

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

-- =====
-- Message d'indication: indication de reconfiguration de multilaisons mobiles
-- =====

MobileMultilinkReconfigurationIndication ::=SEQUENCE
{
    sampleSize INTEGER (1..255),
    samplesPerFrame INTEGER (1..255),
    ...
}

END

```

Annexe B

Messages: définitions sémantiques

B.0 La présente annexe donne les définitions sémantiques et les contraintes applicables aux éléments de syntaxe définis dans le paragraphe précédent.

B.0.1 multimediaSystemControlMessage: choix de types de messages. Les messages définis dans la présente Recommandation sont classés comme messages de demande, de réponse, de commande et d'indication.

B.0.2 requestMessage: entraîne une action du terminal distant et requiert une réponse immédiate de celui-ci. Le message nonStandard peut être utilisé pour envoyer des demandes hors norme.

B.0.3 responseMessage: message émis en réponse à un message de demande. Le message nonStandard peut être utilisé pour transmettre des réponses hors norme.

B.0.4 commandMessage: message entraînant une action mais n'impliquant pas de réponse explicite. Le message nonStandard peut être utilisé pour envoyer des commandes hors norme.

B.0.5 indicationMessage: message contenant des informations qui ne nécessitent pas d'action ou de réponse. Le message nonStandard peut être utilisé pour envoyer des indications hors norme.

B.0.6 nonStandardParameter: sert à indiquer un paramètre hors norme; il se compose d'un identificateur et des paramètres effectifs qui sont codés sous la forme d'une chaîne d'octets.

B.0.7 nonStandardIdentifier: sert à identifier un type de paramètre hors normes. Il s'agit soit d'un identificateur d'objet, soit d'un identificateur de type H.221 représenté par une chaîne d'octets comprenant exactement quatre octets, à savoir: L'indicatif de pays est formé de deux octets, dont le premier est conforme à l'Annexe A/T.35. Le second octet est une attribution nationale, à moins que le premier octet ne soit 1111 1111, et dans ce cas le second octet contient l'indicatif de pays conformément à l'Annexe B/T.35. Le code du fabricant du terminal est formé de deux octets attribués au plan national. Les codes de fabricant sont les mêmes que ceux qui sont attribués pour les besoins de la Rec. UIT-T H.320 [22]. Les identificateurs hors normes H.245 peuvent être du type "objet" ou du type "h221NonStandard" à la discrétion du fabricant définissant le message hors normes, étant donné que les identificateurs d'objet et les messages h221NonStandard proviennent de domaines disjoints et ne peuvent pas être confondus. Cependant, étant donné que les messages h221NonStandard sont également utilisés par la Rec. UIT-T H.320, de tels messages proviennent du même domaine que les messages H.320 et devront avoir la même signification.

B.1 Messages liés à la désignation maître-esclave

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

B.1.1 MasterSlaveDetermination

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

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

statusDeterminationNumber est un numéro aléatoire compris entre $0 \dots 2^{24} - 1$.

B.1.2 MasterSlaveDeterminationAcknowledge

Ce message est utilisé pour confirmer si le terminal est le terminal maître ou le terminal asservi, comme cela est indiqué par la décision. Quand la décision indique maître, le terminal recevant ce message est le terminal maître et quand la décision indique esclave, c'est le terminal asservi.

B.1.3 MasterSlaveDeterminationReject

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

B.1.4 MasterSlaveDeterminationRelease

Ce message est envoyé dans le cas d'une expiration de temporisation.

B.2 Messages liés aux capacités des terminaux

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

B.2.1 Aperçu général

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

Ces numéros de capacités sont groupés dans des structures AlternativeCapabilitySet. Chaque AlternativeCapabilitySet indique que le terminal peut fonctionner dans exactement un mode énuméré dans l'ensemble. Par exemple, si un ensemble AlternativeCapabilitySet énumère les 3 valeurs {G.711, G.723.1, G.728}, cela signifie que le terminal peut fonctionner dans l'un quelconque de ces modes audio, mais pas plus d'un.

Ces structures AlternativeCapabilitySet sont groupées en structures simultaneousCapabilities. Chacune de celles-ci indique un ensemble de modes que le terminal est capable d'utiliser simultanément. Par exemple, une structure simultaneousCapability contenant les deux structures AlternativeCapabilitySet {H.261, H.263}, {G.711, G.723.1, G.728} signifie que le terminal peut faire fonctionner simultanément l'un ou l'autre des codecs vidéo avec n'importe lequel des codecs audio. L'ensemble simultaneousCapabilities {{H.261}, {H.261, H.263}, {G.711, G.723.1, G.728}} signifie que le terminal peut faire fonctionner simultanément deux canaux vidéo et un canal audio; un canal vidéo selon H.261, un autre canal vidéo selon H.261 ou H.263, et un canal audio selon G.711, G.723.1 ou G.728.

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

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

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

B.2.2 TerminalCapabilitySet

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

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

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

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

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

B.2.2.1 Tableau de capacités

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

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

B.2.2.2 Descripteurs de capacité

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

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

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

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

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

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

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

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

B.2.2.3 Capacités

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

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

Les choix `receiveAndTransmitVideoCapability`, `receiveAndTransmitAudioCapability`, `receiveAndTransmitDataApplicationCapability`, `receiveAndTransmitUserInputCapability` et `receiveAndTransmitMultiplexedStreamCapability` indiquent la capacité de recevoir et d'émettre symétriquement selon les capacités qui sont respectivement `VideoCapability`, `AudioCapability`, `DataApplicationCapability`, `UserInputCapability` et `MultiplexedStreamCapability`. Ces séquences binaires peuvent servir à indiquer que les capacités de réception et d'émission ne sont pas indépendantes.

Pour apporter des précisions au moyen d'un exemple, un terminal annoncé comme `{{Rx-G.723.1, Rx-G.729}, {Tx-G.723.1, Tx-G.729}}` n'indique aucune limitation symétrique; il peut donc recevoir en mode G.723.1 tout en émettant en mode G.729. En revanche, un terminal annoncé comme `{{RxAndTx-G.723.1, RxAndTx-G729}}` indique une limitation symétrique. Il ne peut donc recevoir en mode G.723.1 tout en émettant en mode G.729.

Le paramètre booléen `h233EncryptionTransmitCapability`, quand il est mis à la valeur "Vrai", indique que le terminal assure le chiffrement selon les Recommandations UIT-T H.233 [14] et H.234 [15].

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

`ConferenceCapability` indique diverses capacités de conférence.

`multipointVisualizationCapability` (similaire à MVC de la H.230) est inclus dans l'ensemble de capacités d'une unité MCU ou d'un terminal pour indiquer qu'il produira ou traitera correctement les signaux `conferenceResponse.broadcastMyLogicalChannel.grantedBroadcastMyLogicalChannel` (similaire au MVA de la Rec. UIT-T H.230) et `conferenceResponse.broadcastMyLogicalChannel.deniedBroadcastMyLogicalChannel` (similaire au MVR de la Rec. UIT-T H.230) en réponse à `conferenceRequest.BroadcastMyLogicalChannel` (similaire au MCV de la Rec. UIT-T H.230).

h235SecurityCapability indique les capacités que le terminal accepte selon la Rec. UIT-T H.235 [16]. Le champ mediaCapability désigne les entrées du tableau de capacités qui ne contiennent pas une capacité de transmission, de réception ou de réception et de transmission audio, vidéo ou d'application de données ou encore une capacité similaire indiquée par un paramètre NonStandardParameter uniquement. La capacité genericH235SecurityCapability signale à la capacité de sécurité contenue dans l'Annexe G/H.235 que l'extrémité prend en charge le protocole MIKEY indiqué [79]. Quand le protocole MIKEY est exécuté au niveau du média, alors la capacité genericParameters contenue dans la capacité genericH235securityCapability contient également les messages MIKEY.

EncryptionAuthenticationAndIntegrity indique quelles capacités de chiffrement, d'authentification et d'intégrité sont prises en charge pour la capacité mediaCapability notifiée. La capacité mediaCapability définit les algorithmes audio, vidéo ou de données acceptés ainsi que les méthodes de répartition assurées (par exemple réception, émission ou réception et émission). Le paramètre maxPendingReplacementFor indique le nombre maximal d'opérations d'ouverture de voie logique autorisées simultanément dans l'état REPLACEMENT PENDING (en attente de remplacement). Cet état intervient lorsqu'une voie logique a été établie à l'aide du paramètre replacementFor sans, toutefois, que la voie logique remplacée soit déjà fermée.

genericControlCapability indique des capacités de commande génériques.

B.2.2.3.1 Exemple (Informatif)

Supposons qu'une extrémité soit capable de prendre en charge divers codecs audio, des variantes de codage d'une voie audio comme le codage DTMF et les données dans la bande vocale (VBD), un ou plusieurs algorithmes de chiffrement, et la redondance selon RFC 2198, mais qu'elle ne cherche pas à prendre en charge toutes les combinaisons possibles de ces capacités. En particulier, elle peut très bien chercher à prendre en charge une voie à multiples charges utiles avec l'un quelconque de ses codecs audio, DTMF, ou VBD pris en charge. Cette extrémité est capable de chiffrer l'une quelconque des charges utiles conformément à un certain algorithme (ou ensemble d'algorithmes), et est capable de transmettre les données VBD avec la redondance selon RFC 2198.

Cette extrémité pourrait ensuite annoncer ses capacités de la manière suivante.

Tout d'abord, les capacités audio de base:

Capacité 1 = g711Ulaw64k

Capacité 2 = g729wAnnexB

Capacité 3 = vbd

Capacité 4 = audioTone

Il existe une capacité d'alternative afin de choisir soit le codage G.711 ou Annexe B/G.729 (mais non les deux):

Capacité 5 = oneOfCapabilities (1, 2)

Ensuite, les données VBD à codage redondant et le transport du codage DTMF 'fiable':

Capacité 6 = RedundancyEncodingCapability (primaire = 3, secondaire = 3)

Capacité 7 = RedundancyEncodingCapability (primaire = 4, secondaire = 4, 4)

Puis le flux à multiples charges utiles, avec redondance pour les données VBD:

Capacité 8 = MultiplePayloadStreamCapability (MPS) (5, 6, 7)

Et, finalement, le flux MPS avec toutes les charges utiles chiffrées:

Capacité 9 = H235SecurityCapability (8) [flux MPS chiffré]

Un ensemble `alternativeCapabilitySet` pourrait donc être composé des capacités (9, 8, 7, 2, 1), à combiner avec des capacités vidéo à option et/ou des capacités de données à option afin de former un descripteur de capacité `CapabilityDescriptor` multimédia.

B.2.2.4 Capacités de multiplexage

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

H222Capability: sert à indiquer les capacités de multiplexage et d'adaptation réseau qui sont propres au multiplex défini dans la Rec. UIT-T H.222.1 [9].

Le paramètre `numberOfVCs` indique combien de voies virtuelles (*VC*, *virtual channel*) ATM simultanées peuvent être acceptées 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 toutes les voies virtuelles qui contiennent des informations audiovisuelles, mais ne comprend pas la voie virtuelle utilisée pour la signalisation selon la Rec. UIT-T Q.2931 [26].

Le paramètre `vcCapability` est un ensemble dont la taille est égale à la valeur `numberOfVCs`, qui indique les capacités présentes pour chaque canal virtuel disponible.

La séquence `aal1` indique, lorsqu'elle est présente, la capacité de prise en charge de la couche d'adaptation 1 de l'ATM et précise quelles options, spécifiées dans les Recommandations UIT-T I.363.x [25], sont acceptées. Les codages sont définis dans le Tableau B.1.

Tableau B.1/H.245 – Codages de la couche d'adaptation 1 de l'ATM

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

La séquence `aal5` indique, lorsqu'elle est présente, la capacité de prise en charge de la couche d'adaptation 5 de l'ATM et précise quelles options spécifiées dans les Recommandations UIT-T I.363.x [25], sont acceptées. Les tailles `forwardMaximumSDUSize` et `backwardMaximumSDUSize` indiquent la taille maximale de l'unité CPCS-SDU dans les sens direct et inverse, mesurée en octets. L'une des couches d'adaptation `aal1` ou `aal5`, voire les deux couches, doivent être présentes.

Les variables booléennes `transportStream` et `programStream`, quand elles sont mises à "Vrai", indiquent la capacité de prendre en charge respectivement les multiplex de flux de transport et de flux de programme [8].

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

La séquence `aal1ViaGateway` indique, lorsqu'elle est présente, la capacité de prise en charge de la couche d'adaptation 1 de l'ATM grâce à des passerelles de conversion AAL1/5. Les codages sont les mêmes que ceux de la séquence `aal1`. La séquence `Q2931Address` indique un ou plusieurs ensembles de numéros et de sous-adresses Q.2931.

H223Capability: sert à indiquer les capacités propres au multiplex selon H.223 [10].

La variable booléenne `transportWithI-frames`, quand elle est mise à "Vrai", indique que le terminal peut envoyer et recevoir des messages sur les canaux de commande en utilisant des trames I du protocole LAPM comme cela est défini dans la Rec. UIT-T V.42 [38].

Les variables booléennes `videoWithAL1`, `videoWithAL2`, `videoWithAL3`, `audioWithAL1`, `audioWithAL2`, `audioWithAL3`, `dataWithAL1`, `dataWithAL2` et `dataWithAL3`, indiquent quand elles sont mises à "Vrai", la capacité de recevoir le type de média spécifié (vidéo, audio ou données) en utilisant la couche d'adaptation indiquée (AL1, AL2 ou AL3).

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

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

h223MultiplexTableCapability: sert à indiquer les capacités du terminal à recevoir et traiter des entrées de tableau de multiplexage.

Le paramètre "basic" indique que le multiplex ne peut recevoir que les descripteurs `MultiplexEntryDescriptor` de base tels qu'ils sont définis dans la Rec. UIT-T H.223 [10].

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

Le paramètre `maximumNestingDepth` indique la profondeur d'imbrication maximale de champs `subElementList` invoqués de façon récurrente. Les paramètres `MultiplexEntryDescriptor` qui n'utilisent pas le champ `subElementList` seront considérés comme ayant une profondeur d'imbrication nulle.

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

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

Le paramètre booléen `maxMUXPDUSizeCapability`, quand il est mis à "Vrai", indique que l'émetteur est capable de limiter la taille de l'unité MUX-PDU selon H.223 qu'il émet. Il n'a pas de signification lorsqu'il fait partie d'une capacité de réception.

Le paramètre booléen `nsrpSupport`, quand il est mis à "Vrai", indique la prise en charge du mode NSRP spécifié dans l'Annexe A/H.324.

MobileOperationTransmitCapability: sert à indiquer la capacité de transmettre les couches de multiplexage décrites à l'Annexe A/H.223 et à l'Annexe B/H.223.

Le paramètre booléen `h223AnnexA`, quand il est mis à "Vrai", indique que le terminal peut transmettre les unités MUX-PDU définies à l'Annexe A/H.223.

Le paramètre booléen `h223AnnexADoubleFlag`, quand il est mis à "Vrai", indique que le terminal peut transmettre les unités MUX-PDU définies à l'Annexe A/H.223 avec le mode double fanion facultatif.

Le paramètre booléen `h223AnnexB`, quand il est mis à "Vrai", indique que le terminal peut transmettre les unités MUX-PDU définies à l'Annexe B/H.223.

Le paramètre booléen `h223AnnexBwithOptionalHeaderField`, quand il est mis à "Vrai", indique que le terminal peut transmettre les unités MUX-PDU définies à l'Annexe B/H.223 avec un champ d'en-tête facultatif.

h223AnnexCCapability: sert à indiquer la capacité de recevoir et de traiter des unités AL-PDU comme décrit à l'Annexe C/H.223, avec la condition suivante.

Les paramètres booléens `videoWithAL1M`, `videoWithAL2M`, `videoWithAL3M`, `audioWithAL1M`, `audioWithAL2M`, `audioWithAL3M`, `dataWithAL1M`, `dataWithAL2M` et `dataWithAL3M`, quand ils sont mis à "Vrai", indiquent la capacité de recevoir le type de média spécifié (vidéo, audio ou données) au moyen de la couche d'adaptation indiquée (AL1M, AL2M ou AL3M).

Le paramètre `alpduInterleaving`, quand il est mis à "Vrai", indique la capacité de recevoir et de traiter des unités AL-PDU faisant l'objet d'un entrelacement.

L'entier maximal `A11MPDUSize` indique le nombre maximal d'octets dans chaque unité PDU que le terminal peut recevoir lorsqu'il utilise la couche d'adaptation AL1M.

Les nombres entiers `maximumA12MSDUSize` et `maximumAL3MSDUSize` indiquent le nombre maximal d'octets de chaque unité SDU que le terminal peut recevoir lorsqu'il utilise la couche d'adaptation AL2M et AL3M, respectivement.

Le paramètre `rsCodeCapability` indique, quand il est mis à "Vrai", la capacité de recevoir les unités AL-PDU pour lesquelles le codage de Reed-Solomon est indiqué.

Le paramètre `bitRate` indique, quand il est présent, le débit d'émission du flux de bits à la sortie d'un multiplexeur H.223.

Le paramètre `mobileMultilinkFrameCapability` indique, quand il est présent, la capacité de recevoir et de traiter des trames multiliasons mobiles de la taille `maximumSampleSize` et de la longueur `maximumPayloadLength` spécifiées. Le paramètre `maximumSampleSize` indique le nombre maximal d'octets dans chaque échantillon que le terminal peut traiter. Le paramètre `maximumPayloadLength` indique la longueur maximale de trames dans les octets que le terminal peut traiter.

V76Capability: sert à indiquer des capacités propres au multiplex V.76.

Le paramètre `suspendResumeCapabilitywAddress` indique la capacité de mettre en œuvre la procédure d'interruption/reprise V.76 avec un champ d'adresse. Le paramètre `suspendResumeCapabilitywoAddress` indique la capacité de mettre en œuvre la procédure d'interruption/reprise V.76 sans champ d'adresse.

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

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

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

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

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

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

Le paramètre `uihCapability` indique la prise en charge des trames UIH selon V.76.

Le paramètre `numOfDLCS` indique le nombre de voies DLC que le multiplex V.76 peut mettre en œuvre.

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

Le paramètre `loopBackTestCapability` indique la prise en charge du bouclage selon la Rec. UIT-T V.76. Le paramètre `n401Capability` indique la valeur maximale du nombre N401 décrit dans la Rec. UIT-T V.76. Le paramètre `maxWindowSizeCapability` indique la taille maximale de la fenêtre que le multiplex V.76 peut accepter.

H2250Capability: sert à indiquer des capacités propres à la couche de paquets H.225.0.

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

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

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

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

Le paramètre `mcCapability` indique la capacité d'un terminal à jouer le rôle d'entité de commande multipoint (MC) dans une conférence à architecture centralisée ou répartie.

Le paramètre `rtcpVideoControlCapability` indique la capacité du terminal à traiter les messages RTCP acquittement négatif (NACK, *negative acknowledgement*) et demande de rafraîchissement intra complet (FIR, *full intra request*).

Le paramètre `MediaPacketizationCapability` indique quel algorithme facultatif de paquets de média est accepté par le point d'extrémité.

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

`rtpPayloadType` indique les systèmes de paquets de charge utile de protocole RTP pris en charge par le point d'extrémité, comme suit.

L'élément "payloadDescriptor" identifie la sémantique associée au type de charge utile: si la composante `rfc-number` est choisie, elle indique le document officiel de l'IETF dans lequel le format de la charge utile est défini. Les documents RFC périmés ne doivent pas être cités en référence ici. Si la composante "oid" est choisie, elle indique un format de charge utile qui est spécifié dans le cadre d'une Recommandation définie par l'UIT ou d'une Norme internationale définie par l'ISO et

enregistrée dans le document correspondant avec cet identificateur d'objet. Cela s'applique également à l'échange de capacités et à l'ouverture de voies logiques. L'élément `payloadDescriptor` doit être rempli comme suit:

- 1) si la Rec. UIT-T H.225.0 spécifie un numéro oid ou rfc à utiliser pour le codec, se conformer à la Rec. UIT-T H.225.0. Autrement;
- 2) si un identificateur oid est décrit pour le codec dans la Rec. UIT-T relative aux codecs, cet identificateur doit être utilisé. Autrement;
- 3) si le codec est défini par une Recommandation UIT-T (sans oid explicite), la composante oid doit être utilisée et doit être celle du numéro de la Recommandation UIT-T comme suit: {itu-t(0) recommandation(0) <lettre> (<lettre>) <numéro>}. Par exemple, Rec. UIT-T G.711 s'exprimerait par l'oid {itu-t(0) recommandation(0) g (7) 711}. Autrement;
- 4) s'il existe un document RFC définissant la paquetsation au niveau du codec, la composante numéro RFC doit être utilisée. Autrement;
- 5) la composante `nonStandardIdentifier` doit être utilisée.

D'autres identifications du type de charge utile (modes facultatifs, versions, débits, etc., s'il y a lieu) doivent figurer dans les structures du type de données du paramètre `OpenLogicalChannel`. Les décodeurs H.245 doivent reconnaître les oid indiqués ci-dessus, outre tout numéro rfc défini pour le codec.

Le champ `payloadType` peut être inclus pour indiquer le type de charge utile associé à ce format. S'il est utilisé lors de l'échange de capacités, ce champ doit être mis à un type de charge utile assigné statiquement, s'il en existe un pour ce format, et uniquement dans ce cas. Sinon, ce champ doit être omis. S'il est utilisé conjointement avec le paramètre `OpenLogicalChannel`, ce champ doit indiquer la valeur du type de charge utile RTP à utiliser (statique ou dynamique), quel que soit le type de charge utile assigné statiquement. A noter que si la valeur du type de charge utile se situe dans une fourchette de 96 à 127, la même valeur doit aussi figurer dans le paramètre `h2250LogicalChannelParameters.dynamicRTPPayloadType`.

`TransportCapability` indique des capacités de transport facultatives telles que qualité de service et type de média.

`redundancyEncodingCapability` indique quels modes de codage avec redondance sont acceptés, le cas échéant. Pour chaque entrée de capacité, `redundancyEncodingMethod` spécifie le type de codage à utiliser, le codage primaire et les codages secondaires pris en charge pour le codage primaire. Le choix des systèmes de codage dépend du mode choisi. `rtpAudioRedundancyEncoding` désigne le codage audio avec redondance; si ce mode est la méthode `redundancyEncodingMethod` choisie, seuls les nombres `CapabilityEntryNumbers` applicables aux codages audio sont valables. `rtpH263VideoRedundancyEncoding` indique que le codage vidéo avec redondance selon la Rec. UIT-T H.263 et l'Annexe N est possible ou qu'une voie logique doit être ouverte au moyen du codage vidéo avec redondance. Des paramètres additionnels sont fournis:

`numberOfThreads` indique le nombre maximal de chemins que l'émetteur/récepteur peut accepter pendant l'échange de capacités; ce paramètre contient le nombre réel de chemins pour un flux spécifique lors de l'ouverture d'une voie logique;

`framesBetweenSyncPoints` définit le nombre maximal de trames vidéo susceptibles d'être transmises (somme des trames contenues dans tous les chemins) entre deux points de synchronisation de tous les chemins pendant l'échange de capacités; ce paramètre définit le nombre réel de trames pour un flux spécifique lors de l'ouverture d'une voie logique;

`frameToThreadMapping` définit les modes acceptés par un émetteur/récepteur pendant l'échange de capacités, ainsi que le mode à utiliser lors de l'ouverture d'une voie logique: la valeur "round-robin" indique que les trames sont affectées à tour de rôle aux chemins, la première trame après un point

de synchronisation étant affecté au chemin 0, la deuxième au chemin 1 et ainsi de suite. Le format personnalisé "custom" permet d'affecter des mappages arbitraires de trames avec les chemins; pendant l'échange de capacités, la prise en charge du format "custom" est indiquée par le choix de cet élément et le codage d'une SEQUENCE arbitraire (éventuellement vide). La prise en charge des formats "custom" implique celle des mappages effectués à tour de rôle.

containedThreads s'applique uniquement aux commandes d'ouverture de voies logiques: ce paramètre indique quels chemins sont transmis à la voie logique à ouvrir. Une voie logique peut contenir de 1 à 15 chemins, mais deux voies logiques ne doivent pas contenir le même chemin.

En cas de codage rtpH263VideoRedundancyEncoding, le paramètre secondaryEncoding ne doit pas être présent; cela s'applique également à H2250ModeParameters et aux structures ASN.1 RedundancyEncoding de la Rec. UIT-T H.245.

Lorsqu'une voie logique est ouverte pour le codage vidéo avec redondance, la voie logique contenant le chemin 0 sera ouvert en premier et toutes les autres voies logiques y feront référence au moyen du paramètre forwardLogicalChannelDependency du message OpenLogicalChannel.

LogicalChannelSwitchingCapability indique la capacité d'un récepteur de sélectionner le flux (par exemple la voie logique) traité à partir des commandes switchReceiveMedia on/off.

t120DynamicPortCapability indique que le point d'extrémité peut envoyer un appel T.120 [32] à une adresse de transport dynamique et non à l'adresse de port normalisé connu défini dans la Rec. UIT-T T.123 [33].

MultipointCapability: sert à indiquer les capacités d'un terminal inhérentes aux conférences multipoint.

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

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

MediaDistributionCapability: sert à indiquer les capacités du terminal concernant l'émission et la réception de médias dans une conférence multipoint. Les paramètres "centralizedControl" et "centralizedAudio" seront mis à "Vrai" pour les terminaux H.323. Si la vidéo est prise en charge, le paramètre "CentralizedVidéo" sera mis à "Vrai". Si les protocoles T.120 sont acceptés, la capacité "Centralized Data T.120 Data Application" sera présente.

Les paramètres audio, commande et vidéo, de type centralisé ou réparti, indiquent la capacité d'un terminal à participer à une conférence faisant intervenir ces types de distribution de médias. Les paramètres "centralizedData" et "distributedData" indiquent la capacité d'un terminal à participer à une conférence avec ces types de distribution de média en ce qui concerne le protocole "Data Application Protocol" particulier. La capacité MediaDistributionCapability est une séquence permettant de définir des capacités simultanées (c'est-à-dire l'audio de type centralisé avec la vidéo de type réparti ou la vidéo de type centralisé avec l'audio de type réparti, ou des capacités de données particulières selon le protocole "Data Application Protocol").

QOSCapabilities indique des capacités de qualité de service telles que les paramètres RSVPParameters et ATMPParameters.

mediaChannelCapabilities indique de quelle manière le média peut être acheminé. IP-UDP indique que le point d'extrémité assure le transport du média sur une couche de réseau IP et une couche de transport UDP. IP-TCP indique que le point d'extrémité assure le transport du média sur une couche de réseau IP et une couche de transport TCP. La valeur atm-AAL5-UNIDIR indique que le point d'extrémité assure le transport du média sur un circuit virtuel unidirectionnel ATM AAL5. La valeur atm-AAL5-BIDIR indique que le point d'extrémité assure le transport du média sur un circuit virtuel bidirectionnel ATM AAL5.

RSVPPParameters indique des informations de paramètres spécifiques concernant le protocole RSVP.

ATMPParameters indique des informations de paramètres spécifiques concernant un circuit virtuel ATM.

QosMode indique si le mode est un mode de qualité de service garantie ou un mode de chargement contrôlé dans lequel aucune limite supérieure n'est appliquée au temps de propagation de bout en bout.

genericMultiplexCapability: sert à indiquer des capacités de multiplexage génériques.

B.2.2.5 Capacités vidéo

VideoCapability indique des capacités vidéo. L'indication de plusieurs capacités à l'intérieur d'une même structure VideoCapability ne correspond pas à l'indication d'une capacité de traitements simultanés. La capacité de traitements simultanés peut être indiquée par des instances de structures VideoCapability dans différents ensembles AlternativeCapabilitySet avec un descripteur CapabilityDescriptor unique.

ExtendedVideoCapability: sert à indiquer des capacités vidéo avec une séquence de structures associées GenericCapability.

videoCapability indique une séquence de capacités vidéo en option; l'une quelconque des capacités de type VideoCapability peut être utilisée avec l'extension videoCapabilityExtension indiquée.

La structure videoCapabilityExtension, si elle est présente, indique une séquence de structures GenericCapability associées à la capacité videoCapability.

La séquence de structures VideoCapability ne doit pas contenir de capacité ExtendedVideoCapability.

Quand elle est utilisée dans un message OpenLogicalChannel, la capacité ExtendedVideoCapability.videoCapability doit contenir exactement une seule capacité VideoCapability.

H.261VideoCapability: sert à indiquer les capacités H.261 [18].

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

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

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

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

Le paramètre stillImageTransmission indique la capacité de transmission d'images fixes telle que spécifiée à l'Annexe D/H.261.

Le paramètre videoBadMBsCap indique, quand il est mis à "Vrai", la capacité d'un codeur à recevoir – ou d'un décodeur à émettre – la commande videoBadMBs. Quand il fait partie d'une capacité d'émission, il indique la capacité du codeur à traiter des commandes videoBadMBs et à prendre des mesures correctives propres à rétablir la qualité vidéo. Quand il fait partie d'une

capacité de réception, il indique la capacité du décodeur à envoyer des indications videoBadMBs appropriées.

H262VideoCapability: sert à indiquer les capacités H.262 [19].

La liste des variables booléennes indique la capacité de traitement de certains profils et de certains niveaux: une valeur mise à "Vrai" indique qu'une telle opération est possible, alors qu'une valeur mise à "Faux" indique qu'une telle opération n'est pas possible. Un codeur doit produire des flux binaires conformes aux spécifications d'un profil et d'un niveau pour lesquels il a indiqué une capacité, mais également à l'intérieur des limites imposées par les champs facultatifs (voir ci-dessous). Un décodeur doit pouvoir accepter tous les flux binaires 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 B.2.

Le paramètre videoBadMBsCap remplit la même fonction dans la capacité H262VideoCapability que dans la capacité H261VideoCapability.

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

Codage ASN.1	Unité du paramètre cité
videoBitRate	400 bit/s
vbvBufferSize	16 384 bits
samplesPerLine	Echantillons par ligne
linesPerFrame	Lignes par trame
framesPerSecond	Index, frame_rate_code, dans le Tableau 6-4/H.262
luminanceSampleRate	Echantillons par seconde

H.263VideoCapability: sert à indiquer les capacités H.263 [20].

Le nombre sqcifMPI indique, s'il est présent, l'intervalle minimal entre images (unité: 1/29,97) pour le codage et le décodage des images de format SQCIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format SQCIF.

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

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

Le nombre cif4MPI indique, s'il est présent, l'intervalle minimal entre images (unité: 1/29,97) pour le codage et le décodage des images de format 4CIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format 4CIF.

Le format cif16MPI indique, s'il est présent, l'intervalle minimal entre images (unité: 1/29,97) pour le codage et le décodage des images de format 16CIF et, s'il n'est pas présent, aucune capacité n'est indiquée pour les images de format 16CIF.

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

Les paramètres booléens unrestrictedVector (Annexe D/H.263), arithmeticCoding (Annexe E/H.263), advancedPrediction (Annexe F/H.263) et pbFrames (Annexe G/H.263), quand ils sont mis à "Vrai", indiquent la capacité de transmettre et/ou de recevoir ces modes facultatifs, définis dans les annexes à la Rec. UIT-T H.263.

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

Le nombre entier `hrd-B` indique, lorsqu'il est présent, le paramètre B du décodeur HRD (Annexe B/H.263) (unité: 128 bits). Quand il n'est pas présent, la valeur par défaut définie dans l'Annexe B/H.263 est applicable. C'est une capacité de réception et cette capacité n'a pas de signification dans des ensembles de capacités d'émission.

Le nombre entier `bppMaxKb` indique, lorsqu'il est présent, le nombre maximal de bits pour une image codée que le récepteur peut recevoir et décoder correctement (unité: 1024 bits). Quand il n'est pas présent, la valeur par défaut définie dans la Rec. UIT-T H.263 est applicable. C'est une capacité de réception et cette capacité n'a pas de signification dans des ensembles de capacités d'émission.

Les capacités suivantes sont destinées à être utilisées dans certaines applications à très faible fréquence image telles que les applications de vidéosurveillance:

le nombre `slowSqcifMPI` indique, s'il est présent, l'intervalle minimal entre images (unité: secondes par trame) pour le codage et/ou le décodage des images de format SQCIF. S'il n'est pas présent et si le nombre `sqcifMPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format SQCIF. Si le nombre `sqcifMPI` est présent, le nombre `slowSqcifMPI` ne devra pas être présent.

Le nombre `slowQcifMPI` indique, s'il est présent, l'intervalle minimal entre images (unité: secondes par trame) pour le codage et/ou le décodage des images de format QCIF. S'il n'est pas présent et si le nombre `qcifMPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format QCIF. Si le nombre `qcifMPI` est présent, le nombre `slowQcifMPI` ne devra pas être présent.

Le nombre `slowCifMPI` indique, s'il est présent, l'intervalle minimal entre images (unité: secondes par trame) pour le codage et/ou le décodage des images de format CIF. S'il n'est pas présent et si le nombre `cifMPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format CIF. Si le nombre `cifMPI` est présent, le nombre `slowCifMPI` ne devra pas être présent.

Le nombre `slowCif4MPI` indique, s'il est présent, l'intervalle minimal entre images (unité: secondes par trame) pour le codage et/ou le décodage des images de format 4CIF. S'il n'est pas présent et si le nombre `cif4MPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format 4CIF. Si le nombre `cif4MPI` est présent, le nombre `slowCif4MPI` ne devra pas être présent.

Le format `slowCif16MPI` indique, s'il est présent, l'intervalle minimal entre images (unité: secondes par trame) pour le codage et/ou le décodage des images de format 16CIF. S'il n'est pas présent et si le nombre `cif16MPI` n'est pas présent, aucune capacité n'est indiquée pour les images de format 16CIF. Si le nombre `cif16MPI` est présent, le nombre `slowCif16MPI` ne devra pas être présent.

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

Le paramètre booléen `errorCompensation`, quand il est mis à "Vrai", indique la capacité de transmettre et de recevoir une information de retour pour la compensation d'erreurs telle que l'illustre l'Appendice I/H.263. Quand il appartient à une capacité d'émission, il indique la capacité du codeur à traiter les indications `videoNotDecodedMB` et à compenser les erreurs. Quand il appartient à une capacité de réception, il indique la capacité du décodeur à détecter les macroblocs (MB) avec erreur, à les traiter en macroblocs non codés et à envoyer des indications `videoNotDecodedMBs` appropriées.

S'il est présent, le paramètre `enhancementLayerInfo` indique la capacité du codeur à émettre, ou la capacité du décodeur à recevoir, des flux binaires dans le mode d'échelonnabilité facultatif (Annexe O/H.263). `enhancementLayerInfo` est une séquence qui indique les paramètres de configuration du mode d'échelonnabilité.

S'il est présent, le paramètre `H263Options` indique la capacité de prendre en charge les modes facultatifs spécifiés dans la Rec. UIT-T H.263.

EnhancementLayerInfo: sert à indiquer la capacité d'assurer le mode d'échelonnabilité spécifié dans la Rec. UIT-T H.263.

`baseBitRateConstrained` indique si la couche de base est forcée de ne pas dépasser le débit maximal dans la capacité vidéo, déduction faite de la somme des débits maximaux de chacune des options d'amélioration.

Lorsqu'il est présent, le paramètre `snrEnhancement` indique la présence d'une capacité de couche d'amélioration `snr`. La taille définie indique le nombre de couches `snrEnhancement` que le terminal est à même d'accepter dans une seule voie logique.

Lorsqu'il est présent, le paramètre `spatialEnhancement` indique la présence d'une capacité de couche d'amélioration spatiale. Un flux binaire de couche d'amélioration contient une taille d'image qui est le double de la largeur et/ou le double de la hauteur du format d'image de la couche à laquelle elle fait référence. Pour qu'un terminal puisse assurer une amélioration à échelonnabilité spatiale dans une dimension (largeur ou hauteur), il doit également indiquer la capacité de prendre en charge le format d'image personnalisé associé nécessaire dans la couche d'amélioration. La taille définie indique le nombre de couches `spatialEnhancement` que le terminal est à même d'accepter dans une seule voie logique.

Lorsqu'il est présent, le paramètre `bPictureEnhancement` indique la présence d'une capacité de couche d'amélioration d'images B. La taille définie indique le nombre de couches `bPictureEnhancement` que le terminal est à même d'accepter dans une seule voie logique.

Le paramètre `EnhancementOptions`, dans la séquence `bPictureEnhancement`, indique les options additionnelles qu'un codeur peut émettre ou qu'un décodeur peut recevoir dans les images B.

`numberOfBPictures` indique le nombre maximal d'images B que le terminal peut prendre en charge entre des paires successives d'images de référence d'ancrage utilisées dans la prédiction des images B. Ainsi, si le nombre 2 est spécifié, deux images B peuvent être envoyées entre les paires d'images P ou d'autres images d'ancrage.

EnhancementOptions: sert à indiquer les capacités relatives aux couches d'amélioration à échelonnabilité.

Les paramètres de la séquence `EnhancementOptions` ont la même définition sémantique que les paramètres de même nom de la séquence `H263VideoCapability`.

H263Options: sert à indiquer la capacité d'assurer les modes facultatifs additionnels spécifiés dans la Rec. UIT-T H.263.

Le paramètre `advancedIntraCodingMode`, lorsqu'il est mis à "Vrai", indique la capacité d'émettre ou de recevoir le mode de codage INTRA avancé (de l'Annexe I/H.263).

Le paramètre `deblockingFilterMode`, lorsqu'il est mis à "Vrai", indique la capacité d'émettre ou de recevoir le mode filtre de déblocage (de l'Annexe J/H.263).

Le paramètre `improvedPBFramesMode`, lorsqu'il est mis à "Vrai", indique la capacité d'émettre ou de recevoir le mode trames PB améliorées (de l'Annexe M/H.263).

Le paramètre `unlimitedMotionVectors`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge la gamme des vecteurs cinétiques non restreints lorsque le mode vecteurs cinétiques non restreints (de l'Annexe D/H.263) est également indiqué.

unlimitedMotionVectors aura la valeur "Faux" si unrestrictedVector a la valeur Faux dans la même capacité H263VideoCapability ou le même mode H263VideoMode.

Le paramètre fullPictureFreeze, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir des commandes de gel total d'image comme décrit à l'Annexe L/H.263.

Le paramètre partialPictureFreezeAndRelease, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir des commandes de gel partiel d'image et de fin de gel comme décrit à l'Annexe L/H.263.

Le paramètre resizingPartPicFreezeAndRelease, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir des commandes de redimensionnement de gel partiel d'image et de fin de gel comme décrit à l'Annexe L/H.263.

Le paramètre fullPictureSnapshot, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir des instantanés complets du contenu vidéo comme décrit à l'Annexe L/H.263.

Le paramètre partialPictureSnapshot, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir des instantanés partiels du contenu vidéo comme décrit à l'Annexe L/H.263.

Le paramètre videoSegmentTagging, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir un étiquetage des segments du contenu vidéo comme décrit à l'Annexe L/H.263.

Le paramètre progressiveRefinement, lorsqu'il est mis à "Vrai", indique la capacité du codeur d'envoyer et celle du décodeur de recevoir un étiquetage des raffinements progressifs comme décrit à l'Annexe L/H.263. En outre, lorsque la valeur est "Vrai", le codeur doit répondre aux diverses commandes de raffinement progressif doOneProgression, doContinuousProgressions, doOneIndependentProgression, doContinuousIndependentProgressions, progressiveRefinementAbortOne et progressiveRefinementAbortContinuous. Il doit également insérer les étiquettes de début et de fin de segment de raffinement progressif comme défini dans la Spécification des informations d'amélioration supplémentaire (Annexe L/H.263).

NOTE – L'étiquetage des raffinements progressifs peut être envoyé par un codeur et reçu par un décodeur même en l'absence d'une des diverses commandes susmentionnées.

Le paramètre dynamicPictureResizingByFour, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le sous-mode de redimensionnement des images par un facteur de quatre (avec écrêtage) du mode implicite de rééchantillonnage de l'image de référence (Annexe P/H.263).

Le paramètre dynamicPictureResizingSixteenthPel, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le redimensionnement d'une image de référence à n'importe quelle largeur et hauteur au moyen du mode implicite de rééchantillonnage de l'image de référence (Annexe P/H.263 (avec écrêtage)).

Le paramètre dynamicWarpingHalfPel, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge une opération arbitraire de distorsion d'image dans le mode de rééchantillonnage de l'image de référence (Annexe P/H.263 (avec tout mode de remplissage)) avec une précision au demi-pixel.

Le paramètre dynamicWarpingSixteenthPel, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge une opération arbitraire de distorsion d'image dans le mode de rééchantillonnage de l'image de référence (Annexe P/H.263 (avec tout mode de remplissage)) avec une précision au demi-pixel ou au seizième de pixel.

Si le paramètre `DynamicPictureResizingSixteenthPel` est mis à "Vrai", il en est de même pour le paramètre `DynamicPictureResizingByFour`. Si `DynamicWarpingSixteenthPel` est mis à "Vrai", les paramètres `DynamicWarpingHalfPel`, `DynamicPictureResizingByFour` et `DynamicPictureResizingSixteenthPel` ont la même valeur.

La déclaration de la capacité `dynamicPictureResizingByFour` pour un format d'image donné, appelé ici le format d'image initial, suppose la prise en charge d'un maximum de deux autres formats d'image, appelés ici formats d'image obtenus. Pour un format d'image initial de largeur W et de hauteur H , les formats d'image obtenus admis seront d'une largeur $W/2$ et d'une hauteur $H/2$, ou d'une largeur $W/4$ et d'une hauteur $H/4$, sous réserve qu'ils respectent les conditions suivantes: chaque format d'image obtenu sera admis pour autant que sa largeur ne soit pas inférieure à 128 et que sa hauteur ne soit pas inférieure à 96 (128 et 96 étant respectivement la largeur et la hauteur du format SQCIF). Les formats d'image obtenus seront admis pour les mêmes modes facultatifs – MPI (intervalle minimal entre images) et fréquence d'horloge – que pour le format d'image initial.

Le paramètre `independentSegmentDecoding`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le mode de décodage indépendant des segments (Annexe R/H.263).

Le paramètre `slicesInOrder-NonRect`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le sous-mode du mode structuré par tranches (Annexe K/H.263) dans lequel les tranches sont transmises dans l'ordre et contiennent des macroblocs dans l'ordre de balayage de l'image.

Le paramètre `slicesInOrder-Rect`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le sous-mode du mode structuré par tranches (Annexe K/H.263) dans lequel les tranches sont transmises dans l'ordre, chaque tranche occupant une zone rectangulaire de l'image.

Le paramètre `slicesNoOrder-NonRect`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le sous-mode du mode structuré par tranches (Annexe K/H.263), dans lequel les tranches contiennent des macroblocs dans l'ordre de balayage de l'image et ne doivent pas être impérativement transmises dans l'ordre.

Le paramètre `slicesNoOrder-Rect`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le sous-mode du mode structuré par tranches (Annexe K/H.263), dans lequel les tranches occupent une zone rectangulaire de l'image et ne doivent pas être impérativement transmises dans l'ordre.

Le paramètre `alternateInterVLCMode`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le mode de codage à longueur variable Inter (Annexe S/H.263).

Le paramètre `modifiedQuantizationMode`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le mode de quantification modifié (Annexe T/H.263).

Le paramètre `reducedResolutionUpdate`, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur de prendre en charge le mode de rafraîchissement à résolution réduite défini à l'Annexe Q/H.263.

Le paramètre `videoBadMBsCap` remplit la même fonction dans la capacité `H263VideoCapability` que dans la capacité `H261VideoCapability`.

Le paramètre `dataPartitionedSlices`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge le mode *tranche de données segmentées* défini dans l'Annexe V/H.263. Le paramètre `dataPartitionedSlices` doit être mis à "Faux" si les paramètres `slicesInOrder-NonRect`, `slicesInOrder-Rect`, `slicesNoOrder-NonRect` et `slicesNoOrder-Rect` sont tous mis à "Faux" dans le même message `H263Options`.

Le paramètre `fixedPointIDCT0`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge l'option IDCT 0 de référence définie dans l'Annexe W/H.263.

Le paramètre `interlacedFields`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge le codage de champs entrelacés défini dans l'Annexe W/H.263.

Le paramètre `currentPictureHeaderRepetition`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge la répétition de l'en-tête de l'image affichée, telle que définie dans l'Annexe W/H.263.

Le paramètre `previousPictureHeaderRepetition`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge la répétition de l'en-tête de l'image précédente, telle que définie dans l'Annexe W/H.263.

Le paramètre `nextPictureHeaderRepetition`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur ou d'un décodeur à prendre en charge la répétition de l'en-tête de l'image suivante (avec ou sans indication fiable de référence temporelle), telle que définie dans l'Annexe W/H.263.

Les paramètres `currentPictureHeaderRepetition`, `previousPictureHeaderRepetition` et `nextPictureHeaderRepetition`, lorsqu'ils sont mis à "Vrai" et qu'ils font partie des capacités du récepteur, indiquent qu'un décodeur peut fonctionner de nouveau normalement après altération ou perte d'un en-tête d'image en remplaçant l'en-tête d'image altéré ou perdu par un en-tête d'image transmis conformément aux prescriptions de l'Annexe W/H.263.

Le paramètre `pictureNumber`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur à transmettre des numéros d'image conformément aux prescriptions de l'Annexe W/H.263 ou la capacité d'un codeur à détecter des pertes d'image de référence parmi les numéros d'image transmis.

Le paramètre `spareReferencePictures`, lorsqu'il est mis à "Vrai", indique la capacité d'un codeur à émettre des indications d'image de référence de réserve selon les modalités définies dans l'Annexe W/H.263 ou la capacité d'un décodeur à utiliser une image de référence de réserve s'il ne dispose pas de l'image de référence effective.

TransparencyParameters: sert à indiquer les paramètres définissant une couche vidéo transparente.

Le paramètre `presentationOrder` indique la disposition des couches vidéo transparentes. Pendant l'échange de capacités, ce paramètre prend l'une des valeurs suivantes: 0, 1 ou 2. La valeur 0 indique que le type de transparence arrière-plan de l'image de référence (RPB, *reference picture background*) défini à l'Annexe L/H.263 est pris en charge; la valeur 1 indique qu'une image d'arrière-plan commandée de l'extérieur peut être utilisée; enfin, la valeur 2 indique que le flux binaire peut spécifier l'utilisation d'un type de transparence arrière-plan de l'image de référence ou d'un type de transparence image d'arrière-plan commandée de l'extérieur. Pendant l'ouverture d'une voie logique, la valeur INTEGER (nombre entier) définit l'ordre de présentation. Une couche dotée d'un ordre de présentation plus élevé sera disposée au-dessus d'une couche ayant un ordre de présentation inférieur. On peut visualiser `presentationOrder` comme un axe perpendiculaire à l'écran, les valeurs du paramètre s'accroissant en direction du spectateur.

Les paramètres `offset-x` et `offset-y` indiquent le décalage, exprimé en huitièmes de pixel, de la couche transparente notifiée par rapport à la couche de base. Lorsque ces paramètres sont utilisés dans une capacité, ils indiquent la capacité de décaler l'emplacement de la couche vidéo transparente et leurs valeurs seront limitées à 1, 2, 4 ou 8, l'unité étant le huitième de pixel: une valeur égale à 4, par exemple, indique la capacité de décaler la couche transparente par incrément d'un demi-pixel.

Les paramètres `scale-x` et `scale-y` indiquent le facteur d'échelle à appliquer, dans les coordonnées x et y correspondantes, à la couche transparente notifiée avant l'établissement des couches vidéo, en unités par rapport à la couche de base. Dans un message de capacité, ils indiquent le facteur

d'échelle maximal susceptible d'être appliqué: 1 indique que la remise à l'échelle n'est pas prise en charge, 2 signifie que la taille de la couche peut être doublée ou ne pas être mise à l'échelle, 3 indique que la taille de la couche peut être doublée, triplée ou ne pas être mise à l'échelle, etc.

Le paramètre booléen **separateVideoBackChannel** indique, lorsqu'il est mis à "Vrai", que le terminal peut prendre en charge le mode voie logique séparée: aucune autre capacité vidéo ne doit être indiquée dans la même séquence H263VideoCapability: aucune valeur MPI (intervalle minimal entre images) ne doit être présente et les autres indicateurs de mode et contenus correspondants sont sans objet et doivent être à "Faux" ou absents. Lorsqu'il est envoyé dans un message de demande de mode, le paramètre separateVideoBackChannel avec la valeur "Vrai" doit représenter la seule capacité vidéo de H263VideoMode et indique que le récepteur souhaite recevoir une voie contenant uniquement des données de retour H.263. S'il figure dans le message OpenLogicalChannel, ce paramètre indique que la voie logique est destinée uniquement aux messages vidéo de retour et qu'aucun autre flux binaire vidéo H.263 ne doit être transmis par cette voie logique.

refPictureSelection: sert à indiquer la capacité de prendre en charge le mode de sélection de l'image de référence (Annexe N/H.263) et, éventuellement, la capacité de prendre en charge le mode de sélection de l'image de référence améliorée (Annexe U/H.263).

Si elle est présente, la séquence additionalPictureMemory indique la présence d'une quantité supplémentaire de mémoire, en sus de la quantité pouvant être utilisée par le décodeur normal qui n'accepte pas le mode de sélection de l'image de référence. S'il est absent, ce paramètre indique qu'il n'existe, pour le codeur de l'autre terminal, aucune information concernant la quantité additionnelle de mémoire que le décodeur peut utiliser. Si ce paramètre figure dans H263VideoMode, il indique la présence d'une quantité supplémentaire de mémoire image qui peut être utilisée pour le décodage.

Le paramètre sqcifAdditionalPictureMemory indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images SQCIF ou d'un format inférieur, horizontalement et verticalement, si la prise en charge du format personnalisé pour de telles images est indiquée dans le paramètre customPictureFormat.

Le paramètre qcifAdditionalPictureMemory indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images QCIF ou d'un format inférieur, horizontalement et verticalement, si la prise en charge du format personnalisé pour de telles images est indiquée dans le paramètre customPictureFormat. Le nombre d'espaces mémoire image indiqué dans le paramètre qcifAdditionalPictureMemory ne doit pas être supérieur au nombre d'images indiqué dans le paramètre sqcifAdditionalPictureMemory (s'il est présent).

Le paramètre cifAdditionalPictureMemory indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images CIF ou d'un format inférieur, horizontalement et verticalement, si la prise en charge du format personnalisé pour de telles images est indiquée dans le paramètre customPictureFormat. Le nombre d'espaces mémoire image indiqué dans le paramètre cifAdditionalPictureMemory ne doit pas être supérieur au nombre d'images indiqué dans les paramètres sqcifAdditionalPictureMemory ou qcifAdditionalPictureMemory (s'ils sont présents).

Le paramètre cif4AdditionalPictureMemory indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images 4CIF ou d'un format inférieur, horizontalement et verticalement, si la prise en charge du format personnalisé pour de telles images est indiquée dans le paramètre customPictureFormat. Le nombre d'espaces mémoire image indiqué dans le paramètre cif4AdditionalPictureMemory ne doit pas être supérieur au nombre d'images

indiqué dans les paramètres `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory` ou `cifAdditionalPictureMemory` (s'ils sont présents).

Le paramètre `cif16AdditionalPictureMemory` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images 16CIF ou d'un format inférieur, horizontalement et verticalement, si la prise en charge du format personnalisé pour de telles images est indiquée dans le paramètre `customPictureFormat`. Le nombre d'espaces mémoire image indiqué dans le paramètre `cif16AdditionalPictureMemory` ne doit pas être supérieur au nombre d'images indiqué dans les paramètres `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory`, `cifAdditionalPictureMemory` ou `cif4AdditionalPictureMemory` (s'ils sont présents).

Le paramètre `bigCpfAdditionalPictureMemory` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 qui exige que le décodeur dispose d'une quantité additionnelle de mémoire pour enregistrer le nombre indiqué d'images ayant un format personnalisé de la taille indiquée dans le paramètre `customPictureFormat`, qui sont plus grandes que le format 16CIF, horizontalement et verticalement. Le nombre d'espaces mémoire image indiqué dans le paramètre `bigCpfAdditionalPictureMemory` ne doit pas être supérieur au nombre d'images indiqué dans les paramètres `sqcifAdditionalPictureMemory`, `qcifAdditionalPictureMemory`, `cifAdditionalPictureMemory`, `cif4AdditionalPictureMemory` ou `cif16AdditionalPictureMemory` (s'ils sont présents).

Le paramètre `videoMux` indique, pendant la procédure d'échange de capacités, que le terminal peut prendre en charge le mode `VideoMux` décrit dans l'Annexe N/H.263. Quand ce paramètre est mis à "Vrai", le codeur ou le décodeur peuvent utiliser un flux binaire vidéo contenant un message vidéo de retour. S'il figure dans `H263VideoMode`, ce paramètre indique qu'il est préférable de recevoir les messages vidéo de retour dans le mode `VideoMux`. Lorsqu'ils figurent dans `H263VideoMode`, les paramètres `videoMux` et `separateVideoBackChannel` ne doivent pas être mis à "Vrai".

`videoBackChannelSend` indique quel type de message vidéo de retour est accepté par le terminal. S'il figure dans `H263VideoMode`, il désigne le type de message de retour souhaité.

`none` indique que le codeur ne peut pas envoyer ou que le décodeur ne peut pas recevoir un flux binaire H.263 demandant de retourner des messages, quels qu'ils soient.

`ackMessageOnly` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 demandant que seuls les messages d'acquiescement soient retournés.

`nackMessageOnly` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 demandant que seuls les messages d'acquiescement négatif soient retournés.

`ackOrNackMessageOnly` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 demandant de retourner soit les messages d'acquiescement soit les messages d'acquiescement négatif, pour un flux binaire vidéo particulier.

`ackAndNackMessage` indique que le codeur peut envoyer ou que le décodeur peut recevoir un flux binaire H.263 demandant de retourner les messages d'acquiescement et les messages d'acquiescement négatif.

Le paramètre `enhancedReferencePicSelect`, lorsqu'il est présent, indique la capacité du codeur ou du décodeur à utiliser le mode de sélection de l'image de référence améliorée de l'Annexe U/H.263. S'il peut utiliser ce mode de sélection, le codeur doit aussi pouvoir recevoir les trois messages de commande diversifiés `lostPicture`, `lostPartialPicture` et `recoveryReferencePicture` et prendre les mesures nécessaires pour rétablir la qualité des images décodées à l'extrémité distante.

Le paramètre `subPictureRemovalParameters`, lorsqu'il est présent, indique la capacité de prendre en charge la suppression de la sous-image de l'image de référence selon l'Annexe U/H.263.

Le paramètre `mpuHorizMBs` indique la taille horizontale, en macroblocs, de l'élément d'image minimal pour la suppression de la sous-image de l'image de référence selon l'Annexe U/H.263.

Le paramètre `mpuVertMBs` indique la taille verticale, en macroblocs, de l'élément d'image minimal pour la suppression de la sous-image de l'image de référence selon l'Annexe U/H.263.

Le paramètre `mpuTotalNumber` indique la capacité totale de la mémoire tampon multi-image en cas d'utilisation de la suppression de sous-image selon l'Annexe U/H.263 en unités d'image minimales.

CustomPictureClockFrequency: cette séquence indique la capacité de prendre en charge une fréquence d'horloge image personnalisée lorsqu'elle est présente en tant que capacité, ainsi que les paramètres de fréquence d'horloge image personnalisée lorsqu'elle figure dans les séquences `OpenLogicalChannel` et `RequestMode`.

Lorsque `customPictureClockFrequency` est utilisé dans `OpenLogicalChannel` et qu'il comprend plusieurs éléments, le flux binaire vidéo peut passer d'une fréquence d'horloge image (PCF, *picture clock frequency*) à une autre à l'intérieur de cet ensemble d'éléments et à l'intérieur du même flux binaire vidéo. Même s'il n'existe qu'une seule fréquence PCF dans cet ensemble, si une valeur MPI quelconque est envoyée pour la fréquence PCF normalisée à des niveaux plus élevés dans le même message (notamment dans la même structure `H263VideoCapability`), il est possible, dans le même flux binaire, de basculer entre la fréquence PCF normalisée et la fréquence personnalisée. Si l'on souhaite indiquer que la fréquence PCF ne devrait pas changer dans le flux binaire, des données s'appliquant à une seule fréquence PCF devraient être envoyées (soit uniquement les valeurs MPI pour la fréquence PCF normalisée soit uniquement la fréquence `customPictureClockFrequency`).

Le paramètre `clockConversionCode` indique le code de conversion d'horloge lorsque la fréquence d'horloge image personnalisée est utilisée dans la Rec. UIT-T H.263.

Le paramètre `clockDivisor` indique la représentation binaire naturelle de la valeur du diviseur d'horloge. La fréquence d'horloge image personnalisée est donnée par la formule $1\ 800\ 000 / (\text{diviseur d'horloge} * \text{facteur de conversion d'horloge})$, le résultat étant exprimé en Hz.

S'il est présent, le paramètre `sqcifMPI` indique l'intervalle minimal entre images (unités: $1/(\text{fréquence d'horloge image personnalisée})$) pour le codage et/ou le décodage des images SQCIF. S'il est absent, aucune capacité relative aux images SQCIF n'est indiquée.

S'il est présent, le paramètre `qcifMPI` indique l'intervalle minimal entre images (unités: $1/(\text{fréquence d'horloge image personnalisée})$) pour le codage et/ou le décodage des images QCIF. S'il est absent, aucune capacité relative aux images QCIF n'est indiquée.

S'il est présent, le paramètre `cifMPI` indique l'intervalle minimal entre images (unités: $1/(\text{fréquence d'horloge image personnalisée})$) pour le codage et/ou le décodage des images CIF. S'il est absent, aucune capacité relative aux images CIF n'est indiquée.

S'il est présent, le paramètre `cif4MPI` indique l'intervalle minimal entre images (unités: $1/(\text{fréquence d'horloge image personnalisée})$) pour le codage et/ou le décodage des images 4CIF. S'il est absent, aucune capacité relative aux images 4CIF n'est indiquée.

S'il est présent, le paramètre `cif16MPI` indique l'intervalle minimal entre images (unités: $1/(\text{fréquence d'horloge image personnalisée})$) pour le codage et/ou le décodage des images 16CIF. S'il est absent, aucune capacité relative aux images 16CIF n'est indiquée.

CustomPictureFormat: cette séquence indique la capacité de prendre en charge un format d'image personnalisé lorsqu'elle est présente en tant que capacité, et des paramètres de format d'image personnalisé lorsqu'elle figure dans les séquences `OpenLogicalChannel` et `RequestMode`.

Les paramètres `maxCustomPictureWidth`, `maxCustomPictureHeight`, `minCustomPictureWidth` et `minCustomPictureHeight` indiquent la gamme de formats d'image, en unités de 4 pixels, qu'un codeur ou décodeur peut prendre en charge; ils indiquent également le format d'image demandé en cas d'utilisation avec `RequestMode`.

Le paramètre `standardMPI` indique l'intervalle minimal entre images (unités: 1/29,97) lorsque aucune fréquence d'horloge image personnalisée n'est utilisée.

Le paramètre `customPCF` indique les paramètres de fréquence d'horloge image personnalisée en cas d'utilisation avec le format d'image personnalisé.

Le paramètre `clockConversionCode` indique un code de conversion d'horloge en cas d'utilisation d'une fréquence d'horloge image personnalisée selon la Rec. UIT-T H.263.

Le paramètre `clockDivisor` indique la représentation binaire naturelle de la valeur du diviseur d'horloge. La fréquence d'horloge image personnalisée est donnée par la formule $1\ 800\ 000 / (\text{diviseur d'horloge} * \text{facteur de conversion d'horloge})$, le résultat étant exprimé en Hz.

Le paramètre `customMPI` indique l'intervalle minimal entre images (unités: 1/(fréquence d'horloge image personnalisée)) pour le codage et/ou le décodage d'images dans le format personnalisé.

Le paramètre `pixelAspectInformation` indique la capacité d'un codeur ou décodeur de prendre en charge divers rapports hauteur/largeur en pixels, ainsi que le rapport hauteur/largeur en pixels demandé en cas d'utilisation avec `RequestMode`.

Le paramètre `pixelAspectCode` indique la capacité de prendre en charge le rapport hauteur/largeur en pixels indiqué par le code PAR de la Rec. UIT-T H.263.

Le paramètre `extendedPAR: (width, height)` indique la capacité de prendre en charge le rapport hauteur/largeur en pixels indiqué par le code étendu de rapport hauteur/largeur en pixels (`EPAR, extended pixel aspect ratio`) de la Rec. UIT-T H.263.

H263VidéoModeCombos

Lorsqu'elle est présente, la séquence `h263VideoModeCombos` indique les dépendances entre les modes facultatifs de la Rec. UIT-T H.263. Les combinaisons de modes pour lesquelles des capacités sont indiquées dans `h263VideoModeCombos` ne sont pas implicitement autorisées à être utilisées avec d'autres modes facultatifs notifiés à des niveaux plus élevés du même message `H263Options`, `H263VideoCapability` ou `H263VideoMode`, sauf indication contraire dans le quatrième alinéa du présent paragraphe et dans le troisième alinéa du paragraphe suivant. Autrement dit, si la prise en charge de certaines variables booléennes de mode identiques à celles de la séquence `H263VideoModeCombos` est indiquée à des niveaux plus élevés de la syntaxe dans les séquences `h263Mode` ou `H263Capability`, ces modes ne sont pas censés s'appliquer également, dans des combinaisons découplées, avec les modes déclarés dans la séquence `H263VideoModeCombos`.

Le paramètre `h263VideoUncoupledModes` indique quels modes facultatifs H.263 peuvent être activés ou désactivés indépendamment les uns des autres pour une image selon une syntaxe correcte et quels modes peuvent être activés ou désactivés indépendamment des modes indiqués dans le paramètre `h263VideoCoupledModes` envoyé dans la même séquence `H263VideoModeCombos`.

`h263VideoCoupledModes` indique un ou plusieurs ensembles de modes facultatifs H.263 qui peuvent être activés ou désactivés conjointement pour une image dans un flux binaire H.263, mais pour lesquels la capacité d'activer ou de désactiver indépendamment tout sous-ensemble de ces modes n'est pas implicite. Tout ensemble de modes indiqués comme étant couplés dans un message `h263VideoCoupledModes` peut être utilisé avec la totalité ou un sous-ensemble quelconque des modes indiqués comme étant dissociés dans le message `h263VideoUncoupledModes` à l'intérieur du même message `H263VideoModeCombos`. Il doit y avoir, dans chaque message `H263ModeComboFlags` d'un message `h263VideoCoupledModes`, au moins deux indicateurs booléens mis à "Vrai", et il ne doit pas y avoir d'ensemble d'indicateurs de mode mis à "Vrai" qui indique une combinaison couplée de modes qui n'est pas autorisée, au niveau syntaxique, dans la même image d'un flux binaire H.263.

Certains éléments de service facultatifs de la Rec. UIT-T H.263 ne sont pas inclus dans les fanions de la séquence `H263ModeComboFlags` car on estime peu probable qu'ils nécessiteront un couplage

lors de leur mise en œuvre. Plus précisément, il s'agit des éléments de service spécifiés dans l'Annexe L/H.263 (par exemple fullPictureFreeze, partialPictureFreezeAndRelease et resizingPartPicFreezeAndRelease) ainsi que des options portant sur les formats d'image et sur les fréquences d'horloge d'image. Si la prise en charge de certains de ces éléments est signalée à un niveau supérieur dans le même message H263Options ou H263VideoCapability ou H263VideoMode, ces éléments doivent fonctionner de façon découplée avec les combinaisons de mode signalées dans la séquence H263VideoModeCombos. On trouvera ci-après un exemple relativement complexe de l'utilisation des combinaisons de mode vidéo.

Cet exemple correspond au cas où la séquence H263VideoCapability indique que les options advancedPrediction et unrestrictedVector sont prises en charge. D'autre part, un message H263Options (contenu dans le même message H263VideoCapability), indique que le mode dynamicPictureResizingByFour est pris en charge, et un message H263VideoModeCombos (toujours dans le même message H263VideoCapability) comporte un message h263VideoUncoupledModes qui indique que le mode advancedIntraCodingMode est pris en charge de façon découplée alors qu'un message h263VideoCoupledModes indique que les options modifiedQuantizationMode et slicesInOrder-NonRect sont prises en charge de façon couplée. En d'autres termes, le flux binaire vidéo (ne) peut contenir (que) des images assorties des combinaisons de mode suivantes: None, advancedPrediction, unrestrictedVector, dynamicPictureResizingByFour, advancedPrediction avec unrestrictedVector, advancedPrediction avec dynamicPictureResizingByFour, unrestrictedVector avec dynamicPictureResizingByFour, advancedPrediction avec unrestrictedVector avec dynamicPictureResizingByFour, advancedIntraCodingMode, modifiedQuantizationMode avec slicesInOrder-NonRect, et advancedIntraCodingMode avec modifiedQuantizationMode avec slicesInOrder-NonRect.

H263ModeComboFlags

Les paramètres de la séquence H263ModeComboFlags ont la même signification que les paramètres de même nom des séquences H263VideoCapability et H263Options.

Le paramètre unlimitedMotionVectors doit avoir la valeur "Faux" si le paramètre unrestrictedVector a la valeur "Faux" dans le même message H263VideoUncoupledModes. unlimitedMotionVectors doit avoir la valeur "Faux" si le paramètre unrestrictedVector doit avoir la valeur "Faux" dans le même message H263VideoCoupledModes et dans le message H263VideoUncoupledModes du même message H263VideoModeCombos.

Le paramètre referencePicSelect, lorsqu'il est mis à "Vrai", indique la capacité du codeur ou du décodeur d'utiliser le mode de sélection de l'image de référence de la Rec. UIT-T H.263. Lorsqu'ils sont à "Vrai", les paramètres spécifiques qui définissent comment utiliser le mode de sélection de l'image de référence doivent être envoyés dans le champ refPictureSelection du même message H263Options. referencePicSelect ne doit être à "Vrai" que si refPicturesSelection est présent dans le même H263Options message.

Le paramètre enhancedReferencePicSelect doit avoir la valeur "Faux" si le paramètre referencePicSelect a la valeur "Faux" dans le même message H263VideoUncoupledModes. Le paramètre enhancedReferencePicSelect doit avoir la valeur "Faux" si le paramètre referencePicSelect a la valeur "Faux" dans le même message H263VideoCoupledModes et dans le message H263VideoUncoupledModes du même message H263VideoModeCombos.

Le paramètre dataPartitionedSlices doit avoir la valeur "Faux" si les paramètres slicesInOrder-NonRect, slicesInOrder-Rect, slicesNoOrder-NonRect et slicesNoOrder-Rect ont tous la valeur "Faux" dans le même message H263VideoUncoupledModes. Le paramètre dataPartitionedSlices doit avoir la valeur "Faux" si les paramètres slicesInOrder-NonRect, slicesInOrder-Rect, slicesNoOrder-NonRect et slicesNoOrder-Rect ont tous la valeur "Faux" dans le même message H263VideoCoupledModes et dans le message H263VideoUncoupledModes du même message H263VideoModeCombos.

IS11172 VideoCapability: sert à indiquer les capacités IS11172 [44].

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

Le paramètre `videoBadMBsCap` remplit la même fonction dans la capacité IS11172VideoCapability que dans la capacité H261VideoCapability.

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

Codage ASN.1	Unité du paramètre cité
<code>videoBitRate</code>	400 bit/s
<code>vbvBufferSize</code>	16 384 bits
<code>samplesPerLine</code>	Echantillons par ligne
<code>linesPerFrame</code>	Lignes par trame
<code>pictureRate</code>	Voir section 2.4.3.2 de IS11172-2
<code>luminanceSampleRate</code>	Echantillons par seconde

genericVideoCapability: sert à indiquer des capacités vidéo génériques.

B.2.2.6 Capacités audio

AudioCapability indique les capacités audio. L'indication de plusieurs capacités dans une même structure AudioCapability ne correspond pas à l'indication d'une capacité de traitements simultanés. La capacité de traitements simultanés peut être indiquée par des instances de capacité AudioCapability dans différents ensembles AlternativeCapabilitySet avec un descripteur de capacité CapabilityDescriptor unique.

La capacité d'émettre et/ou de recevoir de l'audio selon des Recommandations UIT-T de la série G est indiquée par un choix de nombres entiers. Quand un multiplex H.222.1 est utilisé, ces nombres se réfèrent à la taille disponible de la mémoire tampon STD (unité: 256 octets). Quand un multiplex H.223 est utilisé, ces nombres se réfèrent au nombre maximal de trames audio par unité AL-SDU. Quand un multiplex H.225.0 est utilisé, ces nombres se réfèrent au nombre maximal de trames audio par paquet: un point d'extrémité doit pouvoir recevoir n'importe quel nombre de trames par paquet jusqu'à la limite maximale indiquée dans la capacité AudioCapability; en outre, le point d'extrémité ne doit pas émettre un nombre de trames par paquet supérieur à la limite maximale indiquée dans sa capacité AudioCapability d'émission. La signification exacte des codages est indiquée dans le Tableau B.4.

Tableau B.4/H.245 – Codages audio selon des Recommandations UIT-T de la série G

Codage ASN.1	Signification sémantique du codage
<code>g711Alaw64k</code>	Audio G.711 à 64 kbit/s, loi A
<code>g711Alaw56k</code>	Audio G.711 à 56 kbit/s, loi A tronquée jusqu'à 7 bits
<code>g711Ulaw64k</code>	Audio G.711 à 64 kbit/s, loi μ
<code>g711Ulaw56k</code>	Audio G.711 à 56 kbit/s, loi μ tronquée à 7 bits
<code>g722-64k</code>	Audio G.722, 7 kHz à 64 kbit/s

Tableau B.4/H.245 – Codages audio selon des Recommandations UIT-T de la série G

Codage ASN.1	Signification sémantique du codage
g722-56k	Audio G.722, 7 kHz à 56 kbit/s
g722-48k	Audio G.722, 7 kHz à 48 kbit/s
g7231	G.723.1 soit à 5,3 soit à 6,3 kbit/s
g728	Audio G.728 à 16 kbit/s
g729	Audio G.729 à 8 kbit/s
g729AnnexA	Audio selon l'Annexe A/G.729 à 8 kbit/s
g729wAnnexB	Audio G.729 à 8 kbit/s avec suppression des silences selon l'Annexe B
g729AnnexAwAnnexB	Audio selon l'Annexe A/G.729 à 8 kbit/s avec suppression des silences selon l'Annexe B
g7231AnnexCCapability	G.723.1 avec l'Annexe C/G.723.1
gsmFullRate	Transcodage de la parole à plein débit (GSM 06.10)
gsmHalfRate	Transcodage de la parole à mi-débit (GSM 06.20)
gsmEnhancedFullRate	Transcodage de la parole à plein débit amélioré (EFR, <i>enhanced full rate</i>) (GSM 06.60)
g729Extensions	G.729 Extensions

G7231: sert à indiquer la capacité de traitement par le codec audio G.723.1. Le paramètre `maxA1-sduAudioFrames` indique le nombre maximal de trames audio par unité AL-SDU. Le paramètre booléen `silenceSupression`, quand il est mis à Vrai, indique la capacité d'utiliser la suppression des silences définie dans l'Annexe A/G.723.1.

G.7231AnnexCCapability: sert à indiquer la capacité de traitement par le codec audio Annexe C/G.723.1. Le paramètre `maxA1-sduAudioFrames` indique le nombre maximal de trames audio par unité AL-SDU. Le paramètre booléen `silenceSupression`, quand il est mis à Vrai, indique la capacité d'utiliser la compression des silences définie dans l'Annexe A/G.723.1. Le paramètre `g723AnnexCAudioMode` ne devra pas être présent quand le paramètre `G7231AnnexCCapability` est inclus dans un message `TerminalCapabilitySet`, mais devra être présent quand le paramètre `G7231AnnexCCapability` est inclus dans le message `OpenLogicalChannel`. Les champs `highRateMode0`, `highRateMode1`, `lowRateMode0`, `lowRateMode1`, `sidMode0`, et `sidMode1`, indiquent le nombre d'octets par trame pour chacun des modes audio et de protection d'erreurs selon la Rec. UIT-T G.723.1 et l'Annexe C/G.723.1 qui seront utilisés sur la voie logique.

IS11172AudioCapability: sert à indiquer la capacité de traitement du signal audio codé selon l'ISO/CEI 11172-3 [45].

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

IS13818AudioCapability: sert à indiquer la capacité de traitement du signal audio codé selon l'ISO/CEI 13818-3 [46].

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

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

Tableau B.5/H.245 – Codages pour les canaux multiples selon l'ISO/CEI 13818-3

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

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

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

Le nombre entier bitRate indique la capacité maximale en termes de débit audio, en kbit/s.

GSMAudioCapability: sert à indiquer les capacités relatives aux codecs audio de transcodage de la parole à plein débit, à mi-débit et à plein débit amélioré GSM. Le paramètre audioUnitSize indique le nombre maximal d'octets à envoyer dans chaque paquet; le paramètre comfortNoise, lorsqu'il est mis à "Vrai", indique la capacité de prendre en charge le traitement du bruit de confort pour la voie de conversation à plein débit, à mi-débit et à plein débit amélioré (GSM 06.12, GSM 06.22, GSM 06.62) et le paramètre scrambled, lorsqu'il est mis à "Vrai" indique la capacité de prendre en charge le brouillage des bits pour les voies de conversation à plein débit, à mi-débit et à plein débit amélioré (GSM 06.10, GSM 06.20, GSM 06.60).

genericAudioCapability: sert à indiquer des capacités audio génériques.

g729Extensions: sert à indiquer la capacité des modes facultatifs supplémentaires de la Rec. UIT-T G.729. Ce code ne sera pas utilisé pour indiquer les capacités correspondant aux Annexes A et B pour lesquelles existent les codes **g729AnnexA**, **g729AnnexB** et **g729AnnexAwAnnexB**.

audioUnit:

- pour un multiplex H.222, indique la taille du tampon STD en unités de 256 octets;
- pour un multiplex H.223, le nombre maximal de trames audio par unité AL-SDU;
- pour un multiplex H.225.0, le nombre maximal de trames audio par unité par paquet.

audioUnit sera présent pour l'échange de capacité. Il peut être présent pour une demande de mode.

annexA, lorsque ce paramètre est "Vrai" indique la capacité d'émettre et de recevoir des signaux audio Annexe A/G.729 à 8 kbit/s, au lieu de signaux de type G.729 (corps de la Recommandation).

annexB, lorsque ce paramètre est "Vrai" indique la capacité de suppression des silences selon l'Annexe B/G.729.

annexD, lorsque ce paramètre est "Vrai" indique la capacité d'émettre et de recevoir des signaux audio Annexe D/G.729 à 6,4 kbit/s.

annexE, lorsque ce paramètre est "Vrai" indique la capacité d'émettre et de recevoir des signaux audio Annexe E/G.729 à 11,8 kbit/s.

annexF, lorsque ce paramètre est "Vrai" indique la capacité de suppression des silences selon l'Annexe F/G.729.

annexG, lorsque ce paramètre est "Vrai" indique la capacité de suppression des silences selon l'Annexe G/G.729.

annexH, lorsque ce paramètre est "Vrai" indique la capacité de basculement entre le fonctionnement à 6,4 kbit/s (Annexe D/G.729) et 11,8 kbit/s (Annexe E/G.729).

audioTelephonyEvent peut être inclus afin d'indiquer la prise en charge d'événements téléphoniques dans la bande vocale conformément à RFC 2833. Les événements pris en charge doivent être décrits dans la capacité audioTelephoneEvent comme indiqué dans la <liste de valeurs> figurant à la section 3.9 du RFC 2833. Les événements 0-15 (correspondant à des chiffres du code DTMF 0-9, *, #, A, B, C, D) sont les seuls événements obligatoires.

audioTone peut être inclus afin d'indiquer la prise en charge de tonalités dans la bande vocale conformément à RFC 2833.

B.2.2.7 Capacités d'application de données

DataApplicationCapability indique les capacités de mise en œuvre de transmission de données. L'indication de plusieurs capacités dans une même structure DataApplicationCapability ne correspond pas à l'indication d'une capacité de traitements simultanés. La capacité de traitements simultanés peut être indiquée par des instances de structure DataApplicationCapability dans différents ensembles AlternativeCapabilitySet avec un descripteur CapabilityDescriptor unique.

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

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

DataApplicationCapability: est une énumération d'applications de données et de débits. Chaque application de données indiquée doit avoir pour support une ou plusieurs capacités de protocoles DataProtocolCapability.

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

Le paramètre `t120` indique la capacité de prise en charge du protocole T.120 [32].

Le paramètre `dsm-cc` indique la capacité de prise en charge du protocole DSM-CC [47].

Le paramètre `userData` indique la capacité de prise en charge des données d'un utilisateur non spécifié émanant de ports de données externes.

Le paramètre `t84` indique la capacité de prise en charge du transfert d'images (JPEG, JBIG, télécopie Gr.3/4) de type T.84 [31].

Le paramètre `t434` indique la capacité de prise en charge du transfert de fichiers binaires télématiques T.434 [35].

Le paramètre `h224` indique la capacité de prise en charge du protocole de commande de dispositif simplex en temps réel H.224 [11].

Le paramètre `nlpid` indique la capacité de prise en charge du protocole de la couche Réseau spécifié par `nlpidData` défini dans l'ISO/CEI TR 9577 [52]. Ces protocoles incluent notamment le protocole Internet (IP, *Internet protocol*) et le protocole point à point (PPP) de l'IETF.

NOTE – L'usage de NLPID est largement décrit dans le document RFC 1490, "*Multiprotocol Interconnect over Frame Relay*".

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

Le paramètre `h222DataPartitioning` indique la capacité à modifier et restreindre la subdivision des données conformément à la Rec. UIT-T H.262, comme cela est spécifié dans la Rec. UIT-T H.222.1, dans laquelle les données d'amélioration sont transmises par un canal de données figurant dans la liste des capacités `DataProtocolCapability`.

`t30fax`: ce codage indique la capacité d'utiliser le mode analogique (G3V) de l'Annexe C/T.30, comme spécifié dans la Rec. UIT-T T.39 pour les modes DSVF/MSVF.

`t140`: ce codage indique la capacité d'utiliser le protocole de conversation par texte T.140, comme spécifié dans la Rec. UIT-T T.140.

`t38fax`: ce codage indique un protocole de données conforme à la Rec. UIT-T T.38 [29].

Les champs **`version`**, **`t38FaxRateManagement`**, **`t38FaxUdpOptions`** et **`t38FaxTcpOptions`** sont définis dans la Rec. UIT-T T.38.

Le booléen `fillBitRemoval` indique, lorsqu'il est à "Vrai", que la passerelle/le terminal a la capacité de supprimer et d'insérer des bits de remplissage.

Le booléen `transcodingJBIG` indique, lorsqu'il est à "Vrai", que la passerelle a la capacité d'opérer un transcodage en temps réel entre la compression de ligne et JBIG pour le transfert sur le réseau IP.

Le booléen `transcodingMMG` indique, lorsqu'il est à "Vrai", que la passerelle a la capacité d'opérer un transcodage en temps réel entre la compression de ligne et MMG pour le transfert sur le réseau IP.

`genericDataCapability` indique des capacités de données génériques. Lorsque `maxBitRate` est inclus dans `genericDataCapability`, sa valeur sera identique à celle de `maxBitRate` figurant dans `DataApplicationCapability`.

`DataProtocolCapability`: contient une énumération de protocoles de données.

`v14buffered` indique la capacité de prise en charge d'une application de données spécifiée, en utilisant la conversion V.14 [36] avec une mémoire tampon.

v42lapm indique la capacité de prise en charge d'une application de données spécifiée, en utilisant le protocole LAPM défini dans la Rec. UIT-T V.42 [38].

hdlcFrameTunnelling indique la capacité de prise en charge d'une application de données spécifiée, en utilisant la tunnellation de trames HDLC. Voir la section 4.5.2 de l'ISO/CEI 13239 [43].

h310SeparateVCStack indique la capacité de prise en charge d'une application de données spécifiée, en utilisant la pile de protocoles définie dans la Rec. UIT-T H.310 pour le transport des messages H.245 dans un canal virtuel ATM différent de celui utilisé pour les communications audiovisuelles.

h310SingleVCStack indique la capacité de prise en charge d'une application de données spécifiée, en utilisant la pile de protocoles définie dans la Rec. UIT-T H.310 pour le transport des messages H.245 dans le même canal virtuel ATM que celui utilisé pour les communications audiovisuelles.

transparent indique la capacité de prise en charge d'une application de données spécifiée, en utilisant un transfert de données transparent.

v120: l'utilisation de v120 fait l'objet d'un complément d'étude dans la Rec. UIT-T H.323.

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

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

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

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

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

La paramètre v76wCompression indique la capacité de prise en charge d'une compression de données sur un canal de données V.76.

Le paramètre tcp indique la capacité de prise en charge du protocole TCP/IP pour cette application.

Le paramètre udp indique la capacité de prise en charge du protocole UDP pour cette application.

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

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

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

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

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

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

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

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

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

g3FacsMH200x100 indique qu'une image de télécopie Groupe 3 MH (Huffman modifié) codée sur deux niveaux en séquentiel à résolution normale (200×100 ppi) peut être utilisée.

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

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

g4FacsMMR200x200 indique qu'une image de télécopie Gr. 4 MMR (modification de Reed 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, peut être utilisée pour une taille d'image allant jusqu'à 720×576 .

digPhotoMedSeq indique qu'une image couleur codée selon JPEG en séquentiel peut être utilisée pour une taille d'image allant jusqu'à 1140×1152 .

digPhotoMedProg indique qu'une image couleur codée selon JPEG en progressif peut être utilisée pour une taille d'image allant jusqu'à 1440×1152 .

digPhotoHighSeq indique qu'une image couleur codée selon JPEG en séquentiel peut être utilisée pour une taille d'image allant jusqu'à 2880×2304 .

digPhotoHighProg indique qu'une image couleur codée selon JPEG en progressif peut être utilisée pour une taille d'image allant jusqu'à 2880×2304 .

B.2.2.8 Capacités de chiffrement, d'authentification et d'intégrité

La séquence EncryptionCapability, si elle est présente, indique les capacités de chiffrement d'un terminal pour chaque type de média. Le domaine d'application du chiffrement indique si celui-ci doit être appliqué au flux binaire entier, à une partie du flux binaire d'une manière normalisée ou à

une partie du flux binaire d'une manière non normalisée. L'algorithme choisit l'algorithme de chiffrement.

La séquence `AuthenticationCapability`, si elle est présente, indique que les éléments d'authentification de la Rec. UIT-T H.235 [16] sont pris en charge par le terminal. `antiSpamAlgorithm` indique la méthode et l'algorithme utilisés pour assurer les contre-mesures pour faire face aux attaques de submergement et de refus de service.

La séquence `IntegrityCapability`, si elle est présente, indique que les éléments d'intégrité de la Rec. UIT-T H.235 [16] sont pris en charge par le terminal.

B.2.2.9 Capacités de conférence

La séquence `ConferenceCapability` indique des capacités de conférence telles que la capacité de prendre en charge la conduite de la conférence comme décrit dans la Rec. UIT-T H.243.

Le paramètre `videoIndicateMixingCapability` sera défini comme le VIM de la Rec. UIT-T H.230.

B.2.2.10 Capacités d'entrée utilisateur

`UserInputCapabilities` indique quels paramètres du message `UserInputIndication` sont pris en charge par le terminal. Le paramètre `basicString` indique que le terminal accepte l'option `basicString` de `userInputSupportIndication`; `iA5String` indique que le terminal accepte l'option `iA5String` de `userInputSupportIndication` et `generalString` indique que le terminal accepte l'option `generalString` de `userInputSupportIndication`. `Dtmf` indique que le terminal accepte le mode dtmf avec les éléments `signal` et `signalUpdate` du message `userInputIndication`. `Hookflash` indique que le terminal accepte le signal de raccrochage avec les éléments `signal` et `signalUpdate` du message `userInputIndication`.

Pour le codage DTMF sécurisé, le paramètre `UserInputCapabilities` indique quels paramètres sont chiffrés dans le message `UserInputIndication`.

`encryptedBasicString` indique que le terminal prend en charge l'option `encryptedalphanumeric` de `UserInputIndication`.

`encryptedIA5String` indique que le terminal prend en charge l'option `encryptedSignalType` de `UserInputIndication`.

`encryptedGeneralString` indique que le terminal prend en charge l'option `encryptedalphanumeric` de `extendedAlphanumeric` de `UserInputIndication`.

`secureDTMF` indique que le terminal prend en charge le type `encryptedSignalType` à l'intérieur du signal de codage DTMF sécurisé.

B.2.2.11 Capacités génériques

Le type **GenericCapability** permet de spécifier de nouvelles capacités de manière telle qu'il n'est pas nécessaire de publier une nouvelle version de la syntaxe H.245. Ce moyen générique de spécifier des capacités permet aux dispositifs raccordés à un réseau (entités MC par exemple) de déterminer le mode de fonctionnement commun le plus élevé sans connaître en détails la capacité utilisée. Il permet de définir des descriptions de capacités fondées sur des normes de l'UIT-T ou d'autres normes (y compris les descriptions de capacités propriétaires). Les descriptions de capacités fondées sur des normes de l'UIT-T doivent être annexées à la présente Recommandation. Les descriptions de capacités fondées sur d'autres normes peuvent être publiées sous n'importe quelle forme appropriée.

Le champ **capabilityIdentifiant** indique le type de capacité défini. Les descriptions de capacités fondées sur des normes de l'UIT-T utiliseront l'identificateur d'objet **standard**, tandis que les descriptions de capacités fondées sur d'autres normes et les descriptions de capacités propriétaires utiliseront l'élément **standard**, **h221NonStandard**, **uuid** ou **domainBased**, selon le cas.

Le champ **subIdentifieur** indique un type ou un ensemble de paramètres associé au champ **capabilityIdentifieur**.

Le champ **maxBitRate** indique le débit maximal auquel la capacité peut fonctionner au moment de l'échange des capacités ainsi que le débit effectif à utiliser au moment de la signalisation d'ouverture de voie logique. Il sera présent chaque fois qu'une valeur peut être indiquée et lorsque la spécification de la description de capacité considérée l'exige. Il est défini séparément de sorte que les intermédiaires sur le trajet de signalisation puissent savoir quelle largeur de bande est utilisée sans connaître en détails chaque capacité.

Les paramètres de la capacité peuvent être décrits sous la forme d'une combinaison quelconque des champs **collapsing**, **nonCollapsing** et **nonCollapsingRaw**, avec le champ **transport**, tel que spécifié dans la description de capacité.

Le champ **collapsing** indique des capacités qui sont décrites de manière telle qu'une entité MC puisse combiner les capacités d'un certain nombre de points d'extrémités et construire un ensemble de capacités commun en utilisant un ensemble simple de règles sans connaître en détails le codec individuel.

Le champ **nonCollapsing** indique des capacités décrites dans la même syntaxe que pour le champ **collapsing** mais qui ne peuvent pas être traitées par une entité MC. Dans ce cas, la sémantique de **ParameterValue** est modifiée pour indiquer uniquement des valeurs et non des règles de centralisation. Par exemple, **unsignedMin** et **unsignedMax** ont la même sémantique et indiquent simplement un paramètre entier de 16 bits.

Le champ **nonCollapsingRaw** indique des capacités au moyen d'une chaîne d'octets. Typiquement, ce champ peut être constitué d'une structure de données codée selon les règles PER de l'ASN.1. A noter qu'une entité MC doit obligatoirement connaître les capacités décrites de cette manière pour pouvoir les utiliser.

Le champ **transport** indique des paramètres de transport propres à la capacité décrite.

Au moment de la spécification des descriptions de capacités, il est recommandé de définir autant de paramètres que possible dans la catégorie **collapsing**, car seuls les paramètres définis de cette manière seront traités à coup sûr et pas simplement retransmis par les éléments de réseau.

Le paramètre **GenericCapabilities**, qui inclut aussi bien des séquences de type **collapsing** que des séquences de type **nonCollapsing**, ne devrait pas inclure de structures de paramètre **GenericParameter** de différents types (**collapsing**, **nonCollapsing**) qui utilisent le même identificateur **parameterIdentifieur**.

NOTE 1 – Une telle réutilisation du même identificateur **parameterIdentifieur** pourrait provoquer une collision de valeurs d'identificateur **parameterIdentifieur** si le paramètre était transféré automatiquement à un système, par exemple à un système H.320, qui ne possède pas le moyen de distinguer entre paramètres de type **collapsing** et paramètres de type **nonCollapsing**.

Le champ normal **parameterIdentifieur** d'un paramètre **GenericParameter** ne devrait pas être assigné à la valeur 0.

NOTE 2 – Une telle assignation à la valeur 0 interférerait avec le transfert automatique à la signalisation H.320, par exemple comme cela est effectué dans l'Annexe A/H.239 et dans la Rec. UIT-T H.241.

GenericParameter indique un ou plusieurs paramètres de capacité.

Le champ **parameterIdentifieur** permet d'indiquer la valeur des paramètres de type standard (c'est-à-dire définis dans la description de capacités) et des paramètres de type propriétaire. Les paramètres définis dans la description de capacité utilisent la forme **standard**, qui identifie chaque paramètre par un entier. Les paramètres qui sont des extensions propriétaires utilisent la forme **h221NonStandard**, **uuid** ou **domainBased**.

Le champ **parameterValue** indique la valeur du paramètre. La présence d'un paramètre **logical** indique que le point d'extrémité prend en charge l'option que le paramètre représente. Le champ **booleanArray** contient jusqu'à huit variables booléennes indépendantes. Les champs **unsignedMin** et **unsignedMax** indiquent un paramètre au moyen d'un entier non signé de 16 bits. Les champs **unsigned32Min** et **unsigned32Max** indiquent un paramètre au moyen d'un entier non signé de 32 bits. Le champ **octetString** indique un paramètre sous la forme d'une chaîne d'octets. Le champ **genericParameter** indique une séquence de paramètres qui ont été regroupés à ce niveau de la hiérarchie des capacités.

Pour qu'une entité MC puisse combiner des descriptions de capacité de plusieurs extrémités pour former une description de capacité commune, dans le cas d'une capacité dont l'entité MC n'a pas de connaissance intégrée, elle doit d'abord ignorer tous les paramètres qui ne sont pas pris en charge par toutes les extrémités susceptibles – ainsi qu'en a décidé l'entité MC – d'utiliser une capacité particulière. Alors, pour chacun des paramètres des extrémités candidates ayant le même **parameterIdentifier**, l'entité MC doit:

- exécuter un "et" logique dans le cas du champ **booleanArray**;
- choisir la valeur minimale dans le cas du champ **unsignedMin** ou **unsigned32Min**;
- choisir la valeur maximale dans le cas du champ **unsignedMax** ou **unsigned32Max**.

Le champ **supersedes** permet à une description de capacité de contenir un groupe de paramètres parmi lesquels un seul doit être sélectionné lorsqu'une description de capacité commune est déterminée. Il pourrait servir dans le cas d'un codec vidéo qui prend en charge des résolutions SQCIF, QCIF et CIF avec différents intervalles minimaux entre images. La valeur de **parameterIdentifier** se rapporte à un paramètre au même niveau d'imbrication. Plusieurs champs **supersedes** sont inclus avec un paramètre de manière à pouvoir exprimer un arbre de dépendances du paramètre, comme dans la description de capacité H.262. Chacun des paramètres identifiés dans le champ **supersedes** doit être ignoré de la description de capacité commune. Les paramètres que les paramètres éliminés remplacent seront ignorés à leur tour et ce processus sera répété jusqu'à ce que tous les paramètres remplacés soient ignorés.

Cette opération aboutit à la description de capacité commune.

NOTE 3 – Une entité MC qui a une connaissance intégrée d'une description de capacité particulière peut utiliser son propre ensemble de règles pour produire une description de capacité commune.

B.2.2.12 Capacités des flux multiplexés

Le paramètre **multiplexedStreamCapability** indique la capacité de prendre en charge un flux multiplexé sur une voie logique unique.

Le paramètre **multiplexFormat** indique le protocole de multiplexage pris en charge.

Le paramètre **controlOnMuxStream**, lorsqu'il est mis à la valeur "Vrai", indique que la signalisation sur voie logique du flux multiplexé est assurée sur la voie de commande transportée sur le flux multiplexé. Lorsque ce paramètre est mis à la valeur "Faux", la signalisation sur voie logique du flux multiplexé est assurée sur la voie de commande H.245. Lorsque le paramètre **controlOnMuxStream** est mis à la valeur "Faux" et que le paramètre **multiplexFormat** correspond au paramètre **H223Capability**, une seule voie logique, au plus, doit être ouverte pour le flux multiplexé H.223. Le paramètre **controlOnMuxStream** doit être mis à la valeur "Faux" lorsque le paramètre **MultiplexFormat** correspond au paramètre **h222Capability**.

Le paramètre **capabilityOnMuxStream**, lorsqu'il est présent, indique l'ensemble de capacités du flux multiplexé. Ces capacités sont indiquées à l'aide d'un ensemble **AlternativeCapabilitySet**. Cet ensemble **AlternativeCapabilitySet** ne doit pas inclure la capacité **multiplexedStreamTransmission**. Lorsqu'il est absent, l'ensemble de capacités du flux multiplexé doit être échangé sur la voie de commande qui est transportée sur le flux multiplexé, une fois la voie logique du flux multiplexé ouverte.

B.2.2.13 Charge utile RTP pour la capacité d'événement de téléphonie audio et la capacité de tonalité audio

Le paramètre `receiveRTPAudioTelephonyEventCapability` peut être inclus pour indiquer la prise en charge d'événements téléphoniques audio dans la bande selon le Document RFC 2833. Le paramètre `dynamicRTPPayloadType` indique quel type de charge utile RTP dynamique doit être utilisé pour transporter ces événements. Les événements pris en charge doivent être décrits dans le paramètre `audioTelephoneEvent` décrit dans la <liste de valeurs> figurant dans la section 3.9 du Document RFC 2833. Les événements 0 à 15 (correspondant aux éléments numériques DTMF 0 à 9 *, #, A, B, C et D) sont les seuls événements obligatoires.

Le paramètre `receiveRTPAudioToneCapability` peut être inclus pour indiquer la prise en charge de tonalités audio dans la bande selon le Document RFC 2833. Le paramètre `dynamicRTPPayloadType` indique quel type de charge utile RTP dynamique doit être utilisé pour transporter ces tonalités.

B.2.2.14 Flux à multiples charges utiles

Un flux à multiples charges utiles (MPS, *multiple payload stream*) contient des paquets représentant un unique flux média logique, c'est-à-dire que les paquets représentent tous des codages de ce même flux pendant des intervalles de temps spécifiés. Afin de permettre l'identification et la corrélation des divers codages utilisés, tous les paquets contenus dans un même flux MPS DOIVENT transporter des identificateurs de type de charge utile situés au même endroit dans le paquet et DEVRAIENT utiliser des marqueurs temporels du même format et extraits d'une unique source de synchronisation (par exemple, les charges utiles du protocole RTP devraient utiliser la même source de synchronisation (SSRC)). Le plus souvent, ces paquets représenteront des intervalles de temps séquentiels sans recouvrement et choisiront simplement des codages distincts pour des intervalles distincts, mais il y a des cas où des variantes de codage représentent des intervalles à recouvrement, tels que, quand un événement se produit au milieu d'un intervalle de codage, celui-ci doit être codé distinctement dans la variante de codage. Cela peut se produire, par exemple, quand une tonalité DTMF est détectée au milieu d'un intervalle de codage vocal et devrait être envoyée au moyen d'un événement téléphonique selon RFC 2833. Dans ce cas, le marqueur temporel contenu dans le paquet d'événement téléphonique correspondra à un instant situé au milieu de l'intervalle de codage vocal. Les paquets de durée égale à zéro peuvent être utilisés lorsque l'événement de flux représenté ne possède aucune durée mesurable. Il est également possible d'utiliser le protocole RFC 2198 afin d'envoyer un paquet plusieurs fois, en l'imbriquant dans un paquet ayant d'autres types de charge utile et d'autres intervalles de temps.

En ouvrant une voie logique contenant un flux MPS, chaque flux situé à l'intérieur du MPS possède son propre débit, qui est indépendant des autres valeurs de débit de d'autres flux. Etant donné que le média contenu dans un flux MPS est effectivement entrelacé (c'est-à-dire qu'un seul flux contenu dans une voie de flux MPS sera en transmission à tout moment), le débit total pour la voie de flux MPS est le maximum des valeurs de débit de tous les flux MPS.

Le débit d'une voie de flux MPS peut être commandé au moyen de diverses commandes H.245, comme avec des voies sans flux MPS. Si le débit de la voie est réglé de façon à être inférieur au débit d'un flux particulier, celui-ci ne peut alors plus être utilisé afin de transmettre des média. Par exemple, si une voie de flux MPS est ouverte avec des flux G.729 et G.711 et si l'ordre de commande de débit est utilisé afin de régler le débit de la voie à 32 kbit/s, alors l'extrémité ne peut transmettre qu'au moyen du protocole G.729.

Le débit du flux particulier d'une voie MPS peut également être commandé au moyen de diverses commandes H.245. Dans ce cas, le débit n'aura d'influence que sur ce flux particulier. De nouveau, le débit de la voie sera le maximum des valeurs de débit de tous les flux MPS, sauf si le débit de la voie entière a été diminué.

Ainsi, quand on fait appel à un flux à multiples charges utiles, il y a deux valeurs de débit à considérer. La première est le débit implicite de la voie, qui est le maximum des valeurs de débit de tous les flux MPS. La seconde est le débit maximal de la voie entière, tel que signalé au moyen de diverses commandes H.245 (p.ex., commande de débit). Quand une commande de débit ou d'autres commandes H.245 analogues servent à supprimer des restrictions de débit dans la voie, alors le débit de celle-ci est, une fois de plus, considéré comme étant le maximum des valeurs de débit de tous les flux MPS.

NOTE – Etant donné que tous les paquets doivent absolument représenter des codages d'un unique flux d'origine (de destination), il n'est pas approprié d'inclure des types de média distincts, comme l'audio et la vidéo, bien que les paquets de type Données représentant des données extraites du flux média (comme les chiffres du code DTMF détectés dans un flux audio) puissent être une variante de représentation ou de codage et soient appropriés.

B.2.2.15 Correction d'erreur directe

Une extrémité peut annoncer l'aptitude à exécuter une correction d'erreur directe. Quand elle annonce une capacité RFC 2733, l'extrémité a la possibilité de signaler que les données de correction FEC peuvent être envoyées par un flux distinct ou par le même flux (au moyen d'un codage redondant), conformément à RFC 2198. Cette capacité permet à l'extrémité d'indiquer (au moyen d'un numéro d'entrée dans une table de capacités) quels codecs peuvent être utilisés dans un flux de correction FEC.

Si l'extrémité qui envoie le message **OpenLogicalChannel** cherche à utiliser le protocole RFC 2198 (et que cette capacité soit prise en charge par le destinataire) afin de transporter les données de correction FEC, cette extrémité doit utiliser le codage de redondance du type de données, **DataType redundancyEncoding**, y compris le codage VBD, par exemple comme codage de type primaire, **primary** et la correction FEC du type de données, **DataType fec**, en tant que codage de type secondaire, **secondary**. Le type de charge utile pour les paquets RFC 2198 doit être spécifié dans le champ **dynamicPayloadType** du message **OpenLogicalChannel**. Le type de charge utile pour le codage primaire, **primary**, et la correction Des données de correction FEC peut être signalée dans le champ **payloadType** des champs **primary** et **secondary** du message **RedundancyEncodingElement**.

Si une extrémité cherche à transmettre des données de correction FEC dans un flux distinct, elle a deux possibilités: transmettre vers le même port que les données protégées de correction FEC ou vers un port différent. Dans le cas de la transmission vers un port différent, l'extrémité doit utiliser explicitement un message **OpenLogicalChannel** distinct pour le flux de correction FEC. Le type de données, **dataType**, choisi doit être **fec** et ne doit pas être contenu dans un champ **redundancyEncoding**. Elle doit choisir l'option **mode.separateStream.differentPort** et inclure l'identificateur de session du flux protégé et, facultativement, le type de charge utile du média protégé si la voie considérée transporte des types de flux à multiples charges utiles, p. ex. un flux MPS. Dans le cas de la transmission dans un flux distinct, mais vers le même port en tant que média protégé, les données de correction FEC doivent être signalées dans le cadre d'un flux MPS. Dans ce cas, un des éléments du flux MPS sera l'audio protégé et l'autre élément sera la correction **fec**. Dans ce cas, l'extrémité choisira le champ **mode.separateStream.samePort** et annoncera le type de charge utile du flux protégé.

B.2.3 TerminalCapabilitySetAcknowledge

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

Le numéro `sequenceNumber` doit être le même que le numéro `sequenceNumber` du message TerminalCapabilitySet pour lequel ce message constitue une confirmation.

B.2.4 TerminalCapabilitySetReject

Ce message est utilisé pour refuser un message TerminalCapabilitySet provenant de l'entité CESE homologue.

Le numéro sequenceNumber doit être le même que le numéro sequenceNumber du message TerminalCapabilitySet pour lequel ce message constitue un acquittement négatif.

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

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

Codage ASN.1	Raison
unspecified	Aucune raison de refus n'est spécifiée.
undefinedTableEntryUsed	Un descripteur de capacités fait référence à une entrée du tableau capabilityTable qui n'est pas définie.
descriptorCapacityExceeded	Le terminal n'a pas pu stocker toutes les informations du message TerminalCapabilitySet.
tableEntryCapacityExceeded	Le terminal n'a pas pu stocker davantage d'entrées que celles qui étaient indiquées dans highestEntryNumberProcessed ou sinon n'a pu enregistrer aucune d'elles.

B.2.5 TerminalCapabilitySetRelease

Ce message est envoyé en cas d'expiration de temporisation.

B.3 Messages de signalisation de voie logique

Cet ensemble de messages est destiné à la signalisation de voie logique. Le même ensemble de messages est utilisé pour la signalisation d'une voie logique unidirectionnelle et pour la signalisation d'une voie logique bidirectionnelle; certains paramètres ne sont cependant présents que dans le cas de la signalisation d'une voie logique bidirectionnelle.

Le sens "direct" désigne le sens de transmission allant du terminal émettant la demande initiale de voie logique vers l'autre terminal et le sens "inverse" désigne le sens de transmission opposé, dans le cas d'une demande de voie logique bidirectionnelle.

B.3.1 OpenLogicalChannel

Ce message est utilisé pour tenter d'ouvrir une voie logique unidirectionnelle entre une entité LCSE sortante et une entité LCSE entrante homologue et pour ouvrir une voie logique bidirectionnelle entre une entité B-LCSE sortante et une entité B-LCSE entrante homologue.

forwardLogicalChannelNumber: sert à indiquer le numéro de la voie logique directe qui doit être ouverte.

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

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

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

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

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

Si le paramètre est de type `nullData`, la voie logique ne sera pas utilisée pour la transmission de flux de données élémentaires, mais uniquement pour les informations de la couche d'adaptation. Au cas où la vidéo doit être transmise dans un sens uniquement mais où un protocole de retransmission doit être utilisé, tel que AL3 défini dans la Rec. UIT-T H.223, une voie de retour est nécessaire pour transmettre les demandes de retransmission. Il peut également être utilisé pour décrire une voie logique contenant uniquement des références d'horloge programme (PCR, *program clock references*) dans le cas des flux de transport H.222.1 [9].

Un type de données, `dataType`, contenu dans le message `h235Media`, sert à spécifier le chiffrement de la voie logique; le type de données réel est indiqué à l'intérieur du message `H235Media`, en même temps que la spécification du chiffrement.

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

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

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

Le paramètre `distribution` devra être présent quand l'adresse `networkAddress` sera de type `localAreaNetwork` et devra indiquer si l'adresse `networkAddress` est une adresse de transport pour une transmission à un seul destinataire ou à plusieurs destinataires.

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

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

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

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

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

Si le paramètre est `multiplexedStream`, la voie logique sera utilisée pour le transport de signaux audio/vidéo ou de données sous la forme d'un flux multiplexé comme spécifié. Les champs du paramètre `MultiplexedStreamParameter` ont la même signification que les champs de même nom du paramètre `MultiplexedStreamCapability`.

`forwardLogicalChannelDependency` indique le numéro de voie logique dont dépend la voie directe qui doit être ouverte.

`reverseLogicalChannelDependency` indique le numéro de voie logique dont dépend la voie inverse qui doit être ouverte.

Le paramètre `replacementFor` indique que la voie logique à ouvrir *remplacera* la voie logique spécifiée déjà ouverte. Ce paramètre doit être utilisé uniquement pour désigner des voies logiques déjà établies (état ESTABLISHED). Les voies logiques ouvertes à l'aide de ce paramètre ne doivent pas acheminer de données tant que le trafic n'a pas cessé sur la voie logique établie désignée. Dans ce cas, il ne sera jamais demandé aux décodeurs de médias de décoder des données provenant simultanément des deux voies logiques. Lorsque le trafic sur la voie logique nouvellement établie a commencé, l'ancienne voie logique doit immédiatement être fermée. Les récepteurs peuvent procéder à l'acquittement des voies logiques ouvertes à l'aide du mécanisme `replacementFor`, étant entendu que les ancienne et nouvelle voies logiques ne doivent pas être utilisées simultanément, la capacité de décodage du récepteur n'étant donc pas dépassée.

Le champ `encryptionSync` doit être utilisé par le terminal maître lors de l'acquittement de l'ouverture d'une voie par un terminal asservi. Il indique la valeur de la clé de chiffrement et le point de synchronisation auquel la clé devrait être utilisée. Pour les systèmes selon la Rec. UIT-T H.323, l'indicateur `syncFlag` doit être mis à la valeur du numéro de charge utile dynamique RTP qui correspond à la clé.

La structure **H222LogicalChannelParameters** est utilisée pour indiquer les paramètres spécifiques à l'utilisation de la Rec. UIT-T H.222.1 [9]. Elle doit être présente dans la structure `forwardLogicalChannelParameters` et ne doit pas être présente dans la structure `reverseLogicalChannelParameters`.

Le numéro `resourceID` indique dans quelle voie virtuelle ATM doit être transportée la voie logique. Les moyens par lesquels ce paramètre est associé à une voie virtuelle ATM ne sont pas spécifiés dans la présente Recommandation. Lorsque le format de flux multiplexé de la Rec. UIT-T H.222.0 est utilisé dans la Rec. UIT-T H.323, ce paramètre contient le numéro de voie logique du flux multiplexé dans lequel cette voie logique doit être multiplexée.

Le numéro `subChannelID` indique quel sous-canal H.222.1 est utilisé pour la voie logique. Il doit être égal à l'identificateur de paquet PID dans un flux de transport et à l'identificateur de flux `stream_id` dans un flux de programme.

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

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 UIT-T H.222.0 et H.222.1, qui décrivent le programme dont les informations à transporter dans la voie logique font partie.

Le champ `streamDescriptors` est une chaîne d'octets facultative qui, si elle est présente, contient un ou plusieurs descripteurs, comme cela est spécifié dans les Recommandations UIT-T H.222.0 et H.222.1, qui décrivent les informations à transporter dans la voie logique.

La structure **H223LogicalChannelParameters** est utilisée pour indiquer des paramètres spécifiques à l'utilisation de la Rec. UIT-T H.223 [10]. Elle doit être présente dans les structures `forwardLogicalChannelParameters` et `reverseLogicalChannelParameters`.

Le paramètre `adaptationLayerType` indique le type de couche d'adaptation et les options à utiliser sur la voie logique. Les codages sont les suivants: `nonStandard`, `al1Framed` (mode trame AL1), `al1NotFramed` (mode sans trame AL1), `al2WithoutSequenceNumbers` (AL2 sans numéro de séquence présent), `al2WithSequenceNumbers` (AL2 avec numéros de séquence présents) et `al3` (AL3, indiquant le nombre d'octets du champ de commande qui seront présents et la taille de la mémoire tampon d'émission, B_S , qui sera utilisée, la taille étant mesurée en octets), `al1M` (couche AL1M définie à l'Annexe C/H.223 avec les paramètres spécifiés), `al2M` (couche AL2M définie à l'Annexe C/H.223 avec les paramètres spécifiés) ou `al3M` (couche AL3M définie à l'Annexe C/H.223 avec les paramètres spécifiés).

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.

H223AL1MParameters: sert à indiquer les paramètres propres à l'utilisation de la couche d'adaptation AL1M.

`transferMode` indique si le mode trame ou sans trame est utilisé.

`headerFEC` indique si le mode FEC est SEBCH(16,7) ou Golay(24,12).

La longueur du champ CRC pour la charge utile est définie par `crcLength` comme étant de 4, 8, 12, 16, 20, 28 ou 32 bits ou par `crcNotUsed`.

`rcpcCodeRate` indique le débit des codes RCPC avec les valeurs de 8/8, 8/9, ..., 8/32.

`arqType` indique le mode de fonctionnement ARQ: `noARQ` indique qu'il n'y a pas de retransmission, `typeIARq` indique le mode ARQ de type I et `typeIIARq` indique le mode ARQ de type II.

Le paramètre `alpduInterleaving`, lorsqu'il est mis à "Vrai", indique le recours à l'entrelacement des unités AL-PDU.

Le paramètre `alsduSplitting`, lorsqu'il est mis à "Vrai", indique le recours au fractionnement des unités AL-SDU.

`rsCodeCorrection` indique la capacité de correction par code RS comme étant 0, 1, ..., 127 octets. Un nombre fixe de symboles (octets) de parité de code RS correspondant à `rsCodeCorrection` est ajouté à chaque champ CRC et unité AL-SDU de longueur variable. Lorsque le codage RS est utilisé, `typeIIARq` et `alpduInterleaving` ne sont pas pris en charge.

H223AL2MParameters: sert à indiquer les paramètres propres à l'utilisation de la couche d'adaptation AL2M.

`headerFEC` indique si le mode FEC est SEBCH(16,5) ou Golay(24,12).

Le paramètre `alpduInterleaving`, lorsqu'il est mis à "Vrai", indique le recours à l'entrelacement des unités de données de protocole AL-PDU.

H223AL3MParameters: sert à indiquer les paramètres propres à l'utilisation de la couche d'adaptation AL3M.

Cette séquence a les mêmes paramètres que la séquence `AL1MParameters`, sauf que les paramètres `transferMode` et `alsduSplitting` n'y figurent pas.

H223AnnexCARqParameters

numberOfRetransmissions indique le nombre maximal de retransmissions autorisé: finite désigne une limite finie du nombre de retransmissions choisie entre 0 et 16, et infinite indique que le nombre de retransmissions est illimité. Lorsque la valeur du nombre numberOfRetransmissions est égale à 0, cela indique que le champ de commande est utilisé pour le mode de fractionnement mais qu'il n'y a pas de retransmission.

sendBufferSize indique la taille, en octets, de la mémoire tampon d'émission qui sera utilisée.

V76LogicalChannelParameters: est utilisé pour indiquer les paramètres propres à l'utilisation de la Rec. UIT-T V.76.

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

Le paramètre suspendResume est utilisé pour indiquer que la voie peut utiliser les procédures d'interruption-reprise pour interrompre d'autres voies logiques. Trois options de voie peuvent être choisies: pas d'interruption-reprise sur la voie, interruption-reprise utilisant une adresse ou interruption-reprise sans adresse comme cela est défini dans la Rec. UIT-T V.76. Le paramètre suspendResumewAddress indique que la voie avec interruption-reprise doit utiliser le champ d'adresse tel qu'il est défini dans la Rec. UIT-T V.76. Le paramètre suspendResumewoAddress indique que la voie avec interruption-reprise ne doit pas utiliser le champ d'adresse.

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

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

Pour la description des paramètres n401, windowSize et loopbackTestProcedure voir 12.2.1/V.42 et ses paragraphes. Pour les besoins de la Rec. UIT-T V.70, le paramètre n401 devra être codé en octets.

Le paramètre crcLength est un paramètre facultatif indiquant la longueur du contrôle de redondance cyclique (CRC) utilisé dans le mode de reprise sur erreur. En l'absence de ce paramètre, la longueur d'un contrôle de redondance cyclique (CRC) par défaut devra être utilisée. Le paramètre crc8bit indique l'utilisation d'un contrôle de redondance cyclique (CRC) sur 8 bits, le paramètre crc16bit indique l'utilisation d'un contrôle de redondance cyclique (CRC) sur 16 bits et le paramètre crc32bit indique l'utilisation d'un contrôle de redondance cyclique (CRC) sur 32 bits, comme cela est défini dans la Rec. UIT-T V.76.

Le paramètre recovery est un paramètre facultatif indiquant les procédures de reprise sur erreur définies dans la Rec. UIT-T V.76. En l'absence de ce paramètre, la procédure de reprise sur erreur par défaut devra être utilisée. Le paramètre sREJ indique l'utilisation de la procédure de refus sélectif de trame et le paramètre mSREJ indique l'utilisation de la procédure de refus sélectif multiple comme cela est défini dans la Rec. UIT-T V.76.

Le paramètre uIH indique l'utilisation de trames UIH selon la Rec. UIT-T V.76.

Le paramètre rej indique l'utilisation de la procédure de refus selon la Rec. UIT-T V.76.

Le paramètre V75Parameters est utilisé pour indiquer un paramètre propre à l'utilisation de la Rec. UIT-T V.75. Le paramètre audioHeaderPresent indique la présence de l'en-tête audio selon V.75.

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

L'identificateur `sessionID` est un identificateur unique de session RTP ou T.120 dans une conférence. Il est utilisé par le terminal émetteur pour faire référence à la session à laquelle la voie logique se rapporte. Seul le terminal maître peut créer l'identificateur de session. Il y a par convention trois sessions primaires. La première de ces sessions avec un identificateur de session égal à 1 est la session audio, la seconde session primaire avec un identificateur de session égal à 2 est la session vidéo et la troisième session avec un identificateur de session égal à 3 est la session de données. Un terminal asservi peut ouvrir une session additionnelle en fournissant un identificateur de session égal à 0 dans le message `openLogicalChannel`. Le terminal maître créera un identificateur de session unique et le placera dans le message `openLogicalChannelAck`.

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

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

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

Le paramètre `mediaGuaranteedDelivery` indique si le transport sous-jacent de médias doit être ou non choisi pour garantir la transmission des données.

Le paramètre `mediaControlChannel` indique le canal de commande de médias sur lequel l'émetteur du message `OpenLogicalChannel` écoutera les messages de commande de médias pour cette session. Ce champ est présent uniquement quand un canal de commande de médias est nécessaire.

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

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

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

Le paramètre `dynamicRTPPayloadType` indique la valeur de charge utile dynamique. Lorsqu'il est utilisé, le champ `RTPPayloadType.payloadType` et la valeur de ce champ doivent correspondre.

Le paramètre `mediaPacketization` indique quel algorithme facultatif de paquets de médias est utilisé.

Le paramètre `redundancyEncoding` indique que la méthode de codage avec redondance spécifiée dans ce paramètre doit être utilisée pour la voie logique à ouvrir. Le codage primaire est défini par le paramètre `dataType` de `forwardLogicalChannelParameters` ou de `reverseLogicalChannelParameters`, respectivement. Le type de codage avec redondance à appliquer pour cette voie logique est identifiée par le paramètre `redundancyEncodingMethod`, le codage secondaire est spécifié dans le paramètre `secondaryEncoding`. Le type `DataType` (audio, vidéo, etc.) choisi pour les codages primaire et secondaire doit concorder et être conforme à la méthode `redundancyEncodingMethod` choisie. Le paramètre `source` permet d'identifier le numéro de terminal de l'émetteur du message `OpenLogicalChannel`.

L'ouverture d'une voie protégée par redondance, comme spécifié dans le protocole RFC 2198, est obtenue au moyen du codage **`dataType.redundancyEncoding`**. Ce champ permet la signalisation d'un type primaire de données et d'un certain nombre de types de données secondaires, **`secondary`**. Ce champ permet également d'utiliser le protocole RFC 2198 avec un "flux à multiples charges utiles" et avec une correction d'erreur directe.

En ouvrant une voie logique, le type de charge utile RTP pour le paquet RFC 2198 est spécifié par le champ **`dynamicPayloadType`** contenu dans le message **`OpenLogicalChannel`** ou par le champ **`payloadType`** contenu dans la structure **`multiplePayloadStreamElement`**. Les types de charge utile pour les types primaire et secondaire de charge utile sont spécifiés dans la structure **`RedundancyEncodingElement`**, en même temps que le type de données, **`DataType`**, des données primaires ou secondaires.

Quand le codage de redondance RFC 2198 est utilisé, le champ **`redundancyEncodingMethod`** doit être mis à **`rtpRedundancyEncoding`**. De même, quand on utilise le protocole RFC 2198 et que l'on construit la SEQUENCE **`RedundancyEncoding`**, seule la SEQUENCE **`rtpRedundancyEncoding`** doit être utilisée. Les champs **`RedundancyEncoding.secondaryEncoding`** et **`RedundancyEncoding.rtpRedundancyEncoding`** ne doivent pas être utilisés en même temps.

Quand le chiffrement est spécifié pour une voie transportant de multiples charges utiles, le codage de redondance au moyen du protocole RFC 2198 sert à préserver les types de charge utile effectivement transmis. Le type de charge utile `Encapsulating` est mis à la valeur spécifiée dans le champ `syncFlag` de l'élément `encryptionSync`.

h235 Key: permet d'inclure et de spécifier la méthode de protection des clés de session propres aux médias lorsqu'elles sont transmises entre deux extrémités. Le codage de ce champ est une valeur ASN.1 imbriquée comme décrit dans la Rec. UIT-T H.235.

La séquence `EscrowData` spécifie le type et le contenu de tout mécanisme de retenue pour garantie de clé utilisé. Des types et contenus spécifiques peuvent être nécessaires lors de la mise en œuvre lorsque le chiffrement des médias est activé.

`T120SetupProcedure` indique comment la conférence T.120 doit être établie. Pour `originateCall` et `waitForCall`, l'appelant devrait déterminer le nom numérique de la conférence T.120 à partir de l'identificateur de conférence (CID) H.323 (comme décrit dans la Rec. UIT-T H.323) et envoyer l'unité PDU appropriée (si le point d'extrémité est le terminal maître, il doit envoyer une demande de type invitation et si c'est un terminal asservi, il doit envoyer une demande de type entrée). Pour l'option `issueQuery`, l'appelant doit d'abord émettre une demande d'interrogation puis établir la conférence T.120 en fonction du contenu de la réponse à l'interrogation (comme décrit dans la Rec. UIT-T T.124).

B.3.2 `OpenLogicalChannelAcknowledge`

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

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

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

Le paramètre `reverseLogicalChannelParameters` est présent uniquement en cas de réponse à une demande de voie logique bidirectionnelle.

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

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

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

Le paramètre `FlowControlToZero` indique si l'émetteur est autorisé à lancer la transmission sur la voie logique. S'il est mis à "Vrai", il indique que l'émetteur ne devrait pas transmettre de données sur la voie logique tant qu'il n'a pas reçu un message `FlowControl` ultérieur s'appliquant à la voie logique et l'autorisant à le faire. Si ce paramètre est mis à "Faux" ou s'il est absent, l'émetteur est autorisé à lancer la transmission dès que la voie est établie.

Le paramètre `replacementFor` indique que la voie logique à ouvrir *remplacera* la voie logique spécifiée déjà ouverte. Ce paramètre doit être utilisé uniquement pour désigner des voies logiques déjà établies (état ESTABLISHED). Les voies logiques ouvertes à l'aide de ce paramètre ne doivent pas acheminer de données tant que le trafic n'a pas cessé sur la voie logique établie désignée. Dans ce cas, il ne sera jamais demandé aux décodeurs de médias de décoder des données provenant simultanément des deux voies logiques. Lorsque le trafic sur la voie logique nouvellement établie a commencé, l'ancienne voie logique doit immédiatement être fermée. Les récepteurs peuvent procéder à l'acquittement des voies logiques ouvertes à l'aide du mécanisme `replacementFor`, étant entendu que les anciennes et nouvelles voies logiques ne doivent pas être utilisées simultanément, la capacité de décodage du récepteur n'étant donc pas dépassée.

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

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

Le champ `encryptionSync` doit être utilisé par le terminal maître afin d'indiquer la valeur de la clé de chiffrement et le point de synchronisation auquel la clé devrait être utilisée. Pour les systèmes selon la Rec. UIT-T H.323, l'indicateur `syncFlag` doit être mis à la valeur de charge utile RTP dynamique qui correspond à la clé.

La structure `H2250LogicalChannelAckParameters` est utilisée pour indiquer les paramètres propres à l'utilisation de la Rec. UIT-T H.225.0.

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

Le paramètre `mediaChannel` indique une adresse de transport `transportAddress` devant être utilisée pour la voie logique. Il doit être présent dans le message `OpenLogicalChannelAck` quand le transport s'effectue vers un seul destinataire. Si le paramètre `transportAddress` correspond à une transmission vers plusieurs destinataires, le terminal maître est responsable de la création de

l'adresse de transport pour plusieurs destinataires et devra inclure l'adresse dans le message OpenLogicalChannel. Un terminal asservi qui souhaite ouvrir une nouvelle voie de transmission vers plusieurs destinataires devra mettre des zéros dans le champ transportAddress à l'usage de plusieurs destinataires. Le terminal maître créera et fournira l'adresse transportAddress de transmission vers plusieurs destinataires dans le message OpenLogicalChannelAck pour le terminal asservi. Il convient de noter que l'entité de commande multipoint (MC) utilisera l'instruction communicationModeCommand pour spécifier les détails concernant toutes les sessions RTP dans la conférence.

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

Le paramètre mediaControlChannel indique le canal de commande de médias sur lequel l'émetteur du message openLogicalChannelAck écoutera les messages de commande de médias pour cette session. Ce champ est présent uniquement quand un canal de commande de médias est nécessaire.

Le paramètre dynamicRTPPayloadType indique la valeur de charge utile dynamique qui est utilisée dans la Rec. UIT-T H.323 pour l'autre algorithme de paquets vidéo H.261 selon la Rec. UIT-T H.225.0. Ce champ est présent uniquement quand une charge utile RTP dynamique est utilisée.

Le champ portNumber est utilisé dans l'Annexe C/H.323 lorsque le point d'extrémité de réception estime que l'information B-HLI donnée par le champ portNumber du message OpenLogicalChannel est inadaptée et indique une autre valeur, celle qu'il faut utiliser.

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

B.3.3 OpenLogicalChannelReject

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

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

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

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

Tableau B.7/H.245 – Motifs de refus d'un message OpenLogicalChannel

Codage ASN.1	Motif
unspecified	Aucun motif n'est spécifié pour le refus.
unsuitableReverseParameters	Ce paramètre sera uniquement utilisé pour refuser une demande de voie logique bidirectionnelle quand le seul motif du refus est que la structure demandée reverseLogicalChannelParameters est inappropriée. Un tel refus doit immédiatement être suivi de la procédure d'initialisation pour ouvrir une voie logique bidirectionnelle similaire mais pouvant être acceptée.
dataTypeNotSupported	Le terminal n'a pas été capable d'utiliser le paramètre dataType indiqué dans OpenLogicalChannel.
dataTypeNotAvailable	Le terminal n'a pas été capable d'utiliser simultanément le paramètre dataType indiqué dans OpenLogicalChannel avec le paramètre dataType correspondant aux voies logiques qui sont déjà ouvertes.
unknownDataType	Le terminal n'a pas compris le paramètre dataType indiqué dans OpenLogicalChannel.
dataTypeALCombinationNotSupported	Le terminal n'a pas été capable d'utiliser simultanément le paramètre dataType indiqué dans OpenLogicalChannel avec le type de couche d'adaptation indiqué dans la structure H223logicalChannelParameters.
multicastChannelNotAllowed	La voie de multidiffusion n'a pas pu être ouverte.
insufficientBandwidth	La voie n'a pas pu être ouverte parce que la permission d'utiliser la largeur de bande demandée pour la voie logique a été refusée.
separateStackEstablishmentFailed	Une demande de lancement du module de données d'une communication sur une pile distincte a échoué.
invalidSessionID	Tentative par le terminal asservi de fixer l'identificateur SessionID lors de l'ouverture d'une voie logique vers le terminal maître.
masterSlaveConflict	Tentative par le terminal asservi d'ouvrir une voie logique pour laquelle le terminal maître a déterminé qu'un conflit pouvait survenir (voir C.4.1.3 et C.5.1.3).
waitForCommunicationMode	Tentative d'ouvrir la voie logique avant que l'entité de commande multipoint MC ait transmis le message CommunicationModeCommand.
invalidDependentChannel	Tentative d'ouvrir la voie logique avec une voie dépendante spécifiée qui n'est pas présente.
replacementForRejected	La voie logique ne peut être ouverte à l'aide du paramètre replacementFor. L'émetteur peut renouveler la tentative d'abord en fermant la voie logique à remplacer, puis en ouvrant la voie de remplacement.

B.3.4 OpenLogicalChannelConfirm

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

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

B.3.5 CloseLogicalChannel

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

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

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

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

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

Codage ASN.1	Origine
user	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, qui peut résulter d'une erreur de protocole.

reason indique pourquoi la voie est fermée. reservationFailure indique qu'une réservation QS n'a pas pu être effectuée pour la voie et qu'elle est donc en cours de fermeture. reopen indique que le point d'extrémité devrait fermer la voie, puis ouvrir à nouveau une voie au moyen des procédures OpenLogicalChannel. Cela peut se produire, par exemple, si un appel multipoint est réduit à un appel point à point, des extrémités ayant quitté la conférence.

B.3.6 CloseLogicalChannelAcknowledge

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

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

B.3.7 RequestChannelClose

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 voie logique directe dont la fermeture est demandée.

qosCapability indique les paramètres QS qui étaient utilisés sur la voie.

reason indique le motif de la demande de fermeture. reservationFailure indique qu'une réservation QS n'a pas pu être effectuée pour la voie et qu'elle est donc en cours de fermeture. reopen indique que le point d'extrémité devrait fermer la voie, puis ouvrir à nouveau une voie au moyen des procédures OpenLogicalChannel. Cela peut se produire, par exemple, si un appel multipoint est réduit à un appel point à point, des extrémités ayant quitté la conférence.

B.3.8 RequestChannelCloseAcknowledge

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

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

B.3.9 RequestChannelCloseReject

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

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

Le champ `cause` indique le motif du refus de la demande de fermeture de la voie logique. La seule valeur valable de ce champ est "unspecified".

B.3.10 RequestChannelCloseRelease

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

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

B.4 Messages de signalisation liés au tableau de multiplexage

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

B.4.1 MultiplexEntrySend

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

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

La structure `MultiplexEntryDescriptors` est un ensemble comprenant entre 1 et 15 descripteurs `MultiplexEntryDescriptor`.

MultiplexEntryDescriptor: décrit une entrée unique du tableau de multiplexage. Il inclut le numéro `MultiplexTableEntryNumber` et une liste d'éléments `MultiplexElement`. Une liste incomplète d'éléments indique que l'entrée est désactivée.

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

Dans chaque descripteur `MultiplexEntryDescriptor`, le champ `repeatCount` du dernier élément `MultiplexElement` de la liste `elementList` devra être mis à la valeur "untilClosingFlag" et le champ `repeatCount` de tous les autres éléments `MultiplexElement` de la liste `elementList` devra être mis à la valeur "finite". Ceci garantit que toutes les entrées du tableau de multiplexage définissent un schéma de séquence de multiplexage de longueur indéfinie, qui se répète jusqu'au fanion de fermeture de l'unité MUX-PDU. Un descripteur `MultiplexEntryDescriptor` avec un champ `elementList` incomplet doit indiquer une entrée désactivée.

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

B.4.2 MultiplexEntrySendAcknowledge

Ce message est utilisé pour confirmer la réception d'un ou plusieurs descripteurs `multiplexEntryDescriptor` d'un message `MultiplexEntrySend` émis par l'entité MTSE homologue.

Le numéro `sequenceNumber` devra être le même que le numéro `sequenceNumber` du message `MultiplexEntrySend` pour lequel le message est une confirmation.

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

B.4.3 `MultiplexEntrySendReject`

Ce message est utilisé pour refuser un ou plusieurs descripteurs `multiplexEntryDescriptor` d'un message `MultiplexEntrySend` envoyé par l'entité MTSE homologue.

Le numéro `sequenceNumber` doit être le même que le numéro `sequenceNumber` du message `MultiplexEntrySend` pour lequel ce message est un refus.

La structure `MultiplexEntryRejectionDescriptions` spécifie quelles entrées du tableau sont refusées et quel est le motif du refus. Les motifs du refus sont indiqués dans le Tableau B.9.

Tableau B.9/H.245 – Motifs du refus d'un message `MultiplexEntrySend`

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

B.4.4 `MultiplexEntrySendRelease`

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

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

B.5 Messages de signalisation liés à une demande concernant le tableau de multiplexage

Cet ensemble de messages est utilisé pour demander la retransmission d'un ou plusieurs descripteurs `MultiplexEntryDescriptor` de l'émetteur au récepteur.

B.5.1 `RequestMultiplexEntry`

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

`entryNumbers` est la liste des numéros `MultiplexTableEntryNumber` des descripteurs `MultiplexEntryDescriptor` pour lesquels la retransmission est demandée.

B.5.2 `RequestMultiplexEntryAcknowledge`

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

`entryNumbers` est la liste des numéros `MultiplexTableEntryNumber` des descripteurs `MultiplexEntryDescriptor` qui seront transmis.

B.5.3 `RequestMultiplexEntryReject`

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

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

La structure RequestMultiplexEntryRejectionDescriptions spécifie quelles sont les entrées du tableau qui sont refusées et quelle est le motif de ce refus. Les motifs de refus sont indiqués dans le Tableau B.10.

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

Codage ASN.1	Motif
unspecified	Aucun motif de refus n'est spécifié.

B.5.4 RequestMultiplexEntryRelease

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

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

B.6 Messages liés à la demande de mode

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

B.6.1 RequestMode

Ce message est utilisé pour demander des modes de transmission particuliers au terminal émetteur. Il s'agit d'une liste, par ordre de préférence (le mode préféré étant le premier), des modes que le terminal souhaite recevoir. Chaque mode est décrit en utilisant un paramètre ModeDescription.

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

ModeDescription: est un ensemble constitué d'un ou plusieurs éléments ModeElement.

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

Le paramètre type est utilisé pour indiquer le type de flux élémentaire qui est demandé. C'est un choix entre les modes VideoMode, AudioMode, DataMode, EncryptionMode et H235Mode. H235Mode indique une demande de médias chiffrés.

Le mode multiplexedStreamMode indique le mode de transmission du flux multiplexé demandé. Les champs de MultiplexedStream ont la même signification que les champs de même nom de la capacité MultiplexedStreamCapability.

La structure **h223ModeParameters** est utilisée pour déclarer les paramètres propres à l'utilisation de la Rec. UIT-T H.223 [10].

La structure adaptationLayerType indique quelle couche d'adaptation et quelles options sont demandées pour le type demandé. Les codages sont les suivants: nonStandard, al1Framed (mode tramé AL1), al1NotFramed (mode non tramé AL1), al2WithoutSequenceNumbers (AL2 sans numéro de séquence présent), al2WithSequenceNumbers (AL2 avec numéros de séquence présent), al3 (AL3, avec indication du nombre d'octets du champ de commande qui seront présents, ainsi que de la taille en octets de la mémoire tampon d'émission, B_s, qui sera utilisée), al1M (couche A11M définie à l'Annexe C/H.223 avec les paramètres spécifiés), al2M (couche A12M définie à l'Annexe C/H.223 avec les paramètres spécifiés) ou al3M (couche A13M définie à l'Annexe C/H.223 avec les paramètres spécifiés).

Le paramètre `segmentableFlag`, quand il est égal à "Vrai", indique que le multiplexage avec segmentation est demandé, et quand il est égal à "Faux", il indique que le multiplexage sans segmentation est demandé.

La séquence **h2250ModeParameters** contient des informations spécifiques à utiliser avec les Recommandations UIT-T H.225.0 et H.323.

Le paramètre `redundancyEncodingMode` (s'il est présent) spécifie la méthode *redundancyEncodingMethod* à utiliser et le codage *secondaryEncoding* à appliquer comme codage par redondance. Le codage primaire est spécifié par l'élément *type* contenu dans *ModeElement*.

La structure **genericModeParameters** indique des paramètres de mode génériques.

Les paramètres `multiplexedStreamModeParameters` indiquent la voie logique du flux multiplexé à laquelle cette demande de mode s'applique: la voie logique est identifiée par le champ `logicalChannelNumber`.

logicalChannelNumber: si elle est présente, cette structure indique la voie logique pour laquelle le mode spécifié est demandé. La structure `logicalChannelNumber` ne devrait être utilisée que pour spécifier une voie logique ouverte.

B.6.1.1 Mode vidéo

`VideoMode` est un choix de modes vidéo.

H261VideoMode: sert à indiquer la résolution d'image demandée (soit le format QCIF ou CIF), le débit (unité: 100 bit/s) et la transmission d'images fixes.

H262VideoMode: sert à indiquer le profil et le niveau demandés; les champs facultatifs indiquent, s'ils sont présents, les valeurs demandées pour les paramètres indiqués. Les champs facultatifs sont des nombres entiers dont les unités sont définies dans le Tableau B.2.

H263VideoMode: sert à indiquer la résolution d'image demandée (format SQCIF, QCIF, CIF, 4CIF et 16CIF ou un format d'image personnalisé) et le débit (unité: 100 bit/s). Lors d'une communication avec une extrémité prenant en charge la version 8 de la Rec. UIT-T H.245 ou une version antérieure, il n'est pas possible de demander seulement un format d'image personnalisé. Donc, lors de la réception d'une champ `RequestMode` en provenance d'une extrémité prenant en charge la version 8 de la Rec. UIT-T H.245 ou une version antérieure, si ce champ `RequestMode` contient un format d'image personnalisé, ce format devrait être considéré comme étant la résolution demandée plutôt que celle qui est indiquée dans le champ "resolution" du mode `H263VideoMode`.

Les paramètres booléens `unrestrictedVector`, `arithmeticCoding`, `advancedPrediction` et `pbFrames`, quand ils sont mis à la valeur "Vrai", indiquent qu'il est demandé d'utiliser ces modes facultatifs qui sont définis dans les annexes à la Rec. UIT-T H.263.

Le paramètre booléen `errorCompensation`, quand il est mis à Vrai, indique que le codeur est en mesure de traiter les indications `videoNotDecodedMB` et de compenser les erreurs comme cela est illustré à l'Appendice I/H.263. Il n'est pas nécessaire que le codeur réponde aux indications `videoNotDecoded`. Un pont de conférence (MCU, *multipoint control unit*) ne peut pas nécessairement répondre à toutes les indications.

EnhancementOptions: sert à indiquer les paramètres de la couche d'amélioration à échelonnabilité demandés.

H263Options: sert à indiquer les modes facultatifs de la Rec. UIT-T H.263 demandés.

IS11172VideoMode: sert à indiquer une demande de `constrainedBitstream`; les champs facultatifs indiquent, s'ils sont présents, les valeurs demandées pour les paramètres indiqués. Les champs facultatifs sont des nombres entiers dont les unités sont définies dans le Tableau B.3.

La structure **genericVideoMode** indique des paramètres de mode vidéo génériques.

B.6.1.2 Mode audio

AudioMode est un choix de modes audio.

La signification exacte des codages de l'audio selon les Recommandations UIT-T de la série G est indiquée dans le Tableau B.4. Il y a quatre options pour l'audio selon G.723.1, permettant de demander l'un ou l'autre débit (le faible débit de 5,3 kbit/s ou le débit élevé de 6,3 kbit/s) avec ou sans utilisation de la suppression des silences.

G7231AnnexCMode: est utilisé pour demander le codage audio selon l'Annexe C/G.723.1. La structure maxA1-sduAudioFrames indique le nombre maximal de trames audio par unité AL-SDU demandé. La variable booléenne silenceSuppression, quand elle est mise à Vrai, demande l'utilisation de la compression des silences définie dans l'Annexe A/G.723.1. Les champs g723AnnexCAudioMode, highRateMode0, highRateMode1, lowRateMode0, lowRateMode1, sidMode0 et sidMode1 indiquent le nombre d'octets par trame demandé pour chacun des modes audio et de protection contre les erreurs selon la Rec. UIT-T G.723.1 et l'Annexe C/G.723.1.

IS11172AudioMode: est utilisé pour demander le codage audio selon l'ISO/CEI 11172-3 [45].

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

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

Le paramètre multichannelType indique quel mode multivoie est demandé: singleChannel, twoChannelStereo et twoChannelDual permettent respectivement de demander un fonctionnement à canal unique, à canal stéréo et à deux canaux.

Le paramètre bitRate indique le débit audio demandé, en kbit/s.

IS13818AudioMode: est utilisé pour demander le codage audio selon l'ISO/CEI 13818-3 [46].

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

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

La structure multichannelType indique quel mode multivoie est demandé comme cela est spécifié dans le Tableau B.11.

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

Codage ASN.1	Signification sémantique du codage
singleChannel	Mode monocanal utilisant la configuration 1/0. Mode canal unique (comme dans l'ISO/CEI 11172-3).
twoChannelStereo	Mode à deux canaux utilisant la configuration 2/0. Mode canal stéréo ou à deux canaux (comme dans l'ISO/CEI 11172-3).
twoChannelDual	Deux canaux, utilisant la configuration 2/0. Mode à deux canaux (comme dans l'ISO/CEI 11172-3).
threeChannels2-1	Mode à trois canaux utilisant la configuration 2/1. Canaux gauche, droit et canal d'ambiance unique.

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

Codage ASN.1	Signification sémantique du codage
threeChannels3-0	Mode à trois canaux utilisant la configuration 3/0. Canaux gauche, centre et droit, sans canal d'ambiance.
fourChannels2-0-2-0	Mode à quatre canaux, utilisant la configuration 2/0 + 2/0. Canaux gauche et droit du premier programme et canaux gauche et droit du second programme.
fourChannels2-2	Mode à quatre canaux utilisant la configuration 2/2. Canaux gauche et droit et canaux d'ambiance gauche et droit.
fourChannels3-1	Mode à quatre canaux utilisant la configuration 3/1. Canaux gauche, centre et droit, et canal d'ambiance unique.
fiveChannels3-0-2-0	Mode à cinq canaux utilisant la configuration 3/0 + 2/0. Canaux gauche, centre et droit du premier programme et canaux gauche et droit du second programme.
fiveChannels3-2	Mode à cinq canaux utilisant la configuration 3/2. Canaux gauche, centre et droit, canaux d'ambiance gauche et droit.

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

Le paramètre booléen `multilingual`, quand il est mis à la valeur "Vrai", permet de demander jusqu'à sept canaux multilingues.

Le paramètre `bitRate` indique le débit audio demandé, en kbit/s.

La structure `genericAudioMode` indique des paramètres de mode audio génériques.

B.6.1.3 Mode données

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

Le paramètre `bitRate` indique le débit demandé (unité: 100 bit/s).

Le paramètre `t120` permet de demander l'utilisation du protocole T.120 [32].

Le paramètre `dsm-cc` permet de demander l'utilisation du protocole DSM-CC [47].

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

Le paramètre `t84` permet de demander l'utilisation de la Rec. UIT-T T.84 [31] pour le transfert de ces images (JPEG, JBIG, télécopie Gr.3/4).

Le paramètre `t434` permet de demander l'utilisation de la Rec. UIT-T T.434 [35] pour le transfert de fichiers binaires télématiques.

Le paramètre `h224` permet de demander l'utilisation du protocole H.224 [11] de commande en temps réel pour les applications simplex.

Le paramètre `nlpid` permet de demander l'utilisation de l'application de transmission de données de couche Réseau spécifiée.

Le paramètre `dsvdControl` permet de demander l'utilisation du terminal DSVD afin de prendre en charge un canal de commande hors bande.

La structure `h222DataPartitioning` permet de demander l'application de l'usage modifié et restreint des subdivisions de données de H.262, comme cela est spécifié dans la Rec. UIT-T H.222.1, dans laquelle les données d'amélioration sont transmises par un canal de données figurant dans la liste des capacités `DataProtocolCapability`.

Le paramètre `t30fax` permet de demander l'utilisation du mode analogique (G3V) défini dans l'Annexe C/T.30, comme spécifié dans la Rec. UIT-T T.39 pour les modes DSVF/MSVF.

Le paramètre `t140` permet de demander l'utilisation du protocole de conversation par texte T.140, comme spécifié dans la Rec. UIT-T T.140.

Le paramètre `t38fax` permet de demander l'utilisation de la Rec. UIT-T T.38 [29].

La structure **`genericDataMode`** indique des paramètres de mode données génériques. Lorsque `maxBitRate` est inclus dans `genericDataMode`, sa valeur sera identique à la valeur de `maxBitRate` figurant dans `DataMode`.

B.6.1.4 Mode de chiffrement

`EncryptionMode` est un choix de modes de chiffrement.

Le paramètre `h233Encryption` permet de demander l'utilisation du chiffrement selon les Recommandations UIT-T H.233 [14] et UIT-T H.234 [15].

B.6.2 RequestModeAcknowledge

Ce message est envoyé pour confirmer que le terminal émetteur a l'intention d'utiliser, à l'émission, un des modes demandés par le terminal récepteur.

Le numéro `sequenceNumber` doit être identique au numéro `sequenceNumber` du message `RequestMode` pour lequel ce message est une confirmation.

Le champ `response` indique l'action retenue par le terminal distant. Les valeurs possibles de ce champ sont indiquées dans le Tableau B.12.

Tableau B.12/H.245 – Réponses de confirmation du message RequestMode

Codage ASN.1	Signification
<code>willTransmitMostPreferredMode</code>	Le terminal émetteur passera au mode préféré du récepteur.
<code>willTransmitLessPreferredMode</code>	Le terminal émetteur passera à un des modes préférés du récepteur, mais pas le premier.

B.6.3 RequestModeReject

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

Le numéro `sequenceNumber` devra être identique au numéro `sequenceNumber` du message `RequestMode` pour lequel ce message est une réponse.

Le champ `cause` indique le motif du refus du mode demandé. Les valeurs de ce champ sont indiquées dans le Tableau B.13.

Tableau B.13/H.245 – Réponses de refus du message RequestMode

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

B.6.4 RequestModeRelease

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

B.7 Messages liés au temps de propagation aller et retour

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

B.7.1 RoundTripDelayRequest

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

Le numéro sequenceNumber est utilisé pour étiqueter les instances de RoundTripDelayRequest de sorte que la réponse correspondante puisse être identifiée.

B.7.2 RoundTripDelayResponse

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

Le numéro sequenceNumber devra être identique au numéro sequenceNumber du message RoundTripDelayRequest pour lequel ce message est une réponse.

B.8 Messages liés à la boucle de maintenance

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

B.8.1 MaintenanceLoopRequest

Ce message est envoyé pour demander un type particulier de boucle. Les paramètres de type mediaLoop et logicalChannelLoop permettent de demander la boucle d'une seule voie logique indiquée par le numéro LogicalChannelNumber, alors que le paramètre de type systemLoop se rapporte à toutes les voies logiques. La définition exacte de ces types est propre au système et sort du domaine de la présente Recommandation.

B.8.2 MaintenanceLoopAcknowledge

Ce message est utilisé pour confirmer que le terminal exécutera la boucle demandée.

B.8.3 MaintenanceLoopReject

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

Un terminal peut utiliser le motif canNotPerformLoop pour indiquer qu'il n'a pas la capacité d'exécuter la boucle demandée.

B.8.4 MaintenanceLoopCommandOff

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.

B.9 Messages liés au mode de communication

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

B.9.1 CommunicationModeCommand

Le message **CommunicationModeCommand** est envoyé par une entité de commande multipoint H.323 pour spécifier le mode de communication pour chaque type de médias: monodiffusion ou multidiffusion. Ce message peut entraîner le passage de la conférence centralisée à la conférence décentralisée, et vice versa. Un tel passage peut entraîner la fermeture de toutes les voies logiques existantes et l'ouverture de nouvelles voies.

Le message **CommunicationModeCommand** spécifie toutes les sessions de la conférence. Pour chaque session, il spécifie l'identificateur de session RTP, un identificateur associé de session RTP s'il y a lieu, une étiquette de terminal s'il y a lieu, une description de session, le type de données des sessions (par exemple G.711) et une adresse de monodiffusion ou de multidiffusion pour les voies de médias et de commande de médias, selon qu'il est approprié, pour la configuration et le type de conférence. Si un codage par redondance est utilisé, le paramètre `communicationModeTableEntry` spécifie également la méthode `redundancyEncodingMethod` comme format de codage secondaire.

Le message **CommunicationModeCommand** communique les modes d'émission que les extrémités doivent utiliser dans une conférence. Il ne transmet pas les modes de réception, car ceux-ci sont spécifiés par les messages **OpenLogicalChannel** envoyés par l'entité de commande multipoint aux extrémités.

Le message **CommunicationModeCommand** est censé définir les modes de conférence. Il est donc envoyé après l'indication **multipointConference** qui notifie à un point d'extrémité qu'il doit se conformer aux commandes de l'entité de commande multipoint. Les extrémités devraient attendre un message **CommunicationModeCommand** avant d'ouvrir des voies logiques lorsqu'ils ont reçu une indication **multipointConference**.

Les extrémités recevant un message **CommunicationModeCommand** utilisent le champ **terminalLabel** de chaque entrée de tableau pour déterminer si l'entrée peut être utilisée à des fins de traitement. Les entrées qui ne contiennent pas de champ **terminalLabels** s'appliquent à toutes les extrémités participant à la conférence. Les entrées qui ont un champ **terminalLabel** sont des commandes destinées à des extrémités spécifiques qui correspondent au champ **terminalLabel**. Par exemple, lorsque des flux audio provenant de toutes les extrémités sont associés à une adresse multidiffusion (une session), l'entrée du tableau correspondant au mode audio, à l'adresse de média et à l'adresse de commande de média ne contiendra pas de champ **terminalLabel**. Lorsque l'entrée du tableau ordonne à un point d'extrémité d'envoyer ses signaux vidéo à une adresse multidiffusion, l'entité de commande multipoint inclura l'étiquette **terminalLabel** de ce point d'extrémité.

Le paramètre `SessionDependency` est défini par l'entité de commande multipoint pour indiquer quand une session dépend d'une autre session pour obtenir un décodage significatif des données.

Le champ **destination**, dans la séquence `CommunicationModeTableEntry` indique le point d'extrémité avec lequel le point d'extrémité émetteur doit ouvrir une voie logique. Si le champ **destination** existe dans la séquence `CommunicationModeTableEntry`, le point d'extrémité doit utiliser la destination comme champ **destination** dans les paramètres `H2250LogicalChannelParameters` du message `OpenLogicalChannel`.

Le message `CommunicationModeCommand` peut servir à ordonner des extrémités participant à une conférence (ou à un appel point à point) de changer de mode (en indiquant un nouveau mode avec la voie `mediaChannel` déjà utilisée) ou d'utiliser, à l'émission, une nouvelle adresse (en indiquant le mode alors utilisé, mais avec une nouvelle voie `mediaChannel`). De même, un point d'extrémité qui reçoit un message `CommunicationModeCommand` indiquant le mode alors utilisé mais pas de voie `mediaChannel` devrait fermer la voie appropriée et tenter une nouvelle ouverture au moyen des

procédures OpenLogicalChannel, lorsque le message OpenLogicalChannelAck contient l'adresse à laquelle le point d'extrémité enverra le média.

B.9.2 CommunicationModeRequest

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

B.9.3 CommunicationModeResponse

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

B.10 Messages de demande et de réponse liés aux conférences

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

B.10.1 TerminalListRequest

Ce message équivaut à la commande TCU de H.230 telle que décrite dans la Rec. UIT-T H.243.

B.10.2 TerminalListResponse

Ce message équivaut à une séquence de numéros terminalNumber telle que décrite dans la Rec. UIT-T H.230.

B.10.3 MakeMeChair

Ce message équivaut au CCA tel que décrit dans la Rec. UIT-T H.230.

B.10.4 CancelMakeMeChair

Ce message équivaut au CIS tel que décrit dans la Rec. UIT-T H.230.

B.10.5 MakeMeChairResponse

Ce message équivaut à n'importe quel CIT selon la Rec. UIT-T H.230 si le jeton du rôle de président est accepté ou à n'importe quel CCR selon la Rec. UIT-T H.230 si le jeton du rôle de président est refusé.

B.10.6 DropTerminal

Ce message équivaut au CCD tel que décrit dans la Rec. UIT-T H.230.

B.10.7 TerminalDropReject

Ce message équivaut au CIR tel que décrit dans la Rec. UIT-T H.230.

B.10.8 RequestTerminalID

Ce message équivaut au TCP tel que décrit dans la Rec. UIT-T H.230.

B.10.9 MCTerminalIDResponse

Ce message équivaut au TIP tel que décrit dans la Rec. UIT-T H.230.

B.10.10 EnterH.243PasswordRequest

Ce message équivaut au TCS1 tel que décrit dans la Rec. UIT-T H.230.

B.10.11 PasswordResponse

Ce message équivaut à IIS tel que décrit dans la Rec. UIT-T H.230.

B.10.12 EnterH.243TerminalIDRequest

Ce message équivaut à TCS2/TCI tel que décrit dans la Rec. UIT-T H.230.

B.10.13 TerminalIDResponse

Ce message équivaut à IIS tel que décrit dans la Rec. UIT-T H.230.

B.10.14 EnterH.243ConferenceIDRequest

Ce message équivaut à TCS3 tel que décrit dans la Rec. UIT-T H.230.

B.10.15 ConferenceIDResponse

Ce message équivaut à IIS tel que décrit dans la Rec. UIT-T H.230.

B.10.16 VideoCommandReject

Ce message équivaut à VCR tel que décrit dans la Rec. UIT-T H.230.

B.10.17 EnterExtensionAddressRequest

Ce message équivaut à TCS4 tel que décrit dans la Rec. UIT-T H.230.

B.10.18 EnterExtensionAddressResponse

Ce message équivaut à IIS tel que décrit dans la Rec. UIT-T H.230.

B.10.19 RequestChairControlTokenOwner

Ce message équivaut à TCA tel que décrit dans la Rec. UIT-T H.230 pour le jeton du rôle de président.

B.10.20 ChairTokenOwnerResponse

Ce message équivaut à TIR tel que décrit dans la Rec. UIT-T H.230 pour le jeton du rôle de président.

B.10.21 RequestTerminalCertificate

Ce message est envoyé par tout point d'extrémité participant à une conférence à son entité de commande multipoint. Il permet à un point d'extrémité d'obtenir le certificat numérique pour l'utilisateur d'un terminal déterminé. Le terminal demandeur peut, à titre facultatif, inclure son propre terminalCertificate et une chaîne d'épreuve est chiffrée au moyen de la clé privée.

Le paramètre CertSelectionCriteria définit un ensemble de certificats acceptables pour le demandeur. Le répondeur (l'entité de commande multipoint) devrait tenter de satisfaire à ces critères. CertSelectionCriteria peut être présent avec terminalLabel. Dans ce cas, l'entité de commande multipoint peut appliquer les critères pour choisir un certificat approprié parmi ceux qui sont présentés par le terminal spécifié ou demander à ce terminal un certificat qui correspond aux critères et qu'il renvoie ensuite au terminal demandeur initial.

La réponse peut contenir le certificat numérique et, facultativement, une signature associée au certificat dans les conditions suivantes:

- si l'émetteur de la réponse terminalCertificateResponse n'a pas un certificat approprié, ce message peut être retourné sans certificat (et, par conséquent, sans structure certificateResponse);
- si un point d'extrémité demande le certificat d'un autre point d'extrémité dans une conférence multipoint (défini par la structure terminalLabel), l'entité de commande multipoint qui répond doit renvoyer un certificat associé au point d'extrémité demandé (figurant dans la structure certificateResponse);

- la structure `certificateResponse` devrait être présente. Si l'entité de commande multipoint présente le certificat au demandeur pour le compte d'un autre point d'extrémité, il y aura un lien cryptographique entre les signatures et le certificat de l'entité. Ce résultat sera obtenu de l'une des deux manières suivantes;
 - la clé privée utilisée pour protéger les éléments de clé de session distribués au cours de l'échange le plus récent doit être utilisée;
 - en l'absence d'échange, ou si la clé ne convient pas pour la signature, le certificat utilisé pendant la procédure la plus récente visant à authentifier le point d'extrémité sera la source de la clé privée.

B.10.22 TerminalCertificateResponse

Ce message retourne le certificat numérique et une chaîne `responseString` qui est chiffrée au moyen de la clé privée pour un terminal déterminé.

B.10.23 BroadcastMyLogicalChannel

Ce message est similaire au MCV de la Rec. UIT-T H.230 mis en œuvre selon la procédure décrite au § 6.3.2.2/H.243 mais ne se rapporte qu'à une seule voie logique et a un message de réponse `broadcastMyLogicalChannelResponse` qui confirme la réception de la demande. A noter qu'en cas d'application de la procédure MCV décrite au § 6.3.2.1/H.243 (c'est-à-dire lorsqu'une des deux extrémités d'une liaison terminal-MCU ou inter-MCU n'a pas de capacité de visualisation multipoint), on utilise à la place la forme `conferenceCommand` de la structure `BroadcastMyLogicalChannel`.

B.10.24 BroadcastMyLogicalChannelResponse

Ce message permet d'accepter ou de refuser la demande `BroadcastMyLogicalChannel`.

B.10.25 MakeTerminalBroadcaster

Ce message est similaire au VCB de la Rec. UIT-T H.230 et a un message de réponse `makeTerminalBroadcasterResponse` qui confirme la réception de la demande.

B.10.26 MakeTerminalBroadcasterResponse

Ce message permet d'accepter ou de refuser la demande `MakeTerminalBroadcaster`.

B.10.27 SendThisSource

Ce message est similaire au VCS de la Rec. UIT-T H.230 et a un message de réponse `sendThisSourceResponse` qui confirme la réception de la demande.

B.10.28 SendThisSourceResponse

Ce message permet d'accepter ou de refuser la demande `SendThisSource`.

B.10.29 RequestAllTerminalsIDs

Ce message est envoyé par un point d'extrémité à l'entité de commande multipoint d'une conférence pour obtenir toutes les étiquettes et tous les identificateurs de terminal des participants à la conférence.

B.10.30 RequestAllTerminalIDsResponse

Ce message, qui est une réponse à la demande `RequestAllTerminalIDs`, contient une liste de toutes les extrémités intervenant dans une conférence avec les étiquettes et identificateurs de terminal considérés.

B.10.31 RemoteMCRequest

Ce message est envoyé par une entité de commande multipoint active à une autre entité de commande multipoint pour l'activer ou la désactiver. Un message RemoteMCRequest proposant un choix entre les options masterActivate et slaveActivate peut être envoyé par une entité de commande multipoint active à une entité de commande multipoint inactive pour lui donner le rôle de maître ou d'esclave dans une connexion en cascade. Un message RemoteMCRequest avec l'option deActivate peut être envoyé par une entité de commande multipoint maîtresse à une entité de commande multipoint asservie déjà active pour la désactiver.

B.10.32 RemoteMCResponse

Ce message est envoyé pour indiquer l'acceptation ou le rejet d'une demande RemoteMCRequest. L'acceptation d'une telle demande est déterminée par les critères suivants:

choix de l'option activateSlave

Le récepteur est inactif et l'émetteur de la demande a lancé l'appel avec la valeur INVITE du paramètre conferenceGoal dans le message d'établissement (Setup) H.225 ou le récepteur de la demande a lancé l'appel avec la valeur JOIN du paramètre conferenceGoal dans le message d'établissement (Setup) H.225.

Choix de l'option activateMaster

Le récepteur est inactif et l'émetteur de la demande a lancé l'appel avec la valeur CREATE du paramètre conferenceGoal dans le message d'établissement (Setup) H.225.

Choix de l'option deActivate

Le récepteur est une entité de commande multipoint active.

Si les conditions énoncées ci-dessus ne sont pas remplies, la demande devrait être rejetée avec le motif invalidConfiguration.

Les extrémités ne prenant pas en charge la mise en cascade font appel au motif functionNotSupported pour justifier un rejet.

B.11 Messages liés aux multiliaisons

Les messages multilinkRequest, multilinkResponse et multilinkIndication sont utilisés pour prendre en charge l'utilisation de groupes de canaux conformément à la Rec. UIT-T H.226, comme spécifié dans l'Annexe F/H.324. Ces messages permettent d'assurer l'ajout et la suppression de connexions physiques, l'échange automatique d'adresses de réseau (numéros de téléphone) et la commande du fonctionnement selon la Rec. UIT-T H.226.

B.11.1 callInformation – Demande et réponse

Le message MultilinkRequest.callInformation est utilisé par l'entité initiale, comme défini dans l'Annexe F/H.324, pour demander l'information nécessaire pour établir et associer des connexions physiques supplémentaires. Le nombre maximal de connexions supplémentaires que l'émetteur est capable d'établir est envoyé dans le paramètre maxNumberOfAdditionalConnections.

Le message MultilinkResponse.callInformation inclut le paramètre DialingInformation, dont le contenu est décrit plus loin, ainsi qu'un paramètre callAssociationNumber. Celui-ci doit contenir un nombre aléatoire de 32 bits appartenant à une distribution uniforme. Pour les échanges ultérieurs de messages de type callInformation dans la même session, il faut utiliser la même valeur de paramètre callAssociationNumber.

B.11.2 addConnection – Demande et réponse

Le message MultilinkRequest.addConnection peut être utilisé par l'entité qui répond, comme défini dans l'Annexe F/H.324, pour demander à l'entité initiale d'ajouter des connexions physiques. La

structure `DialingInformation` doit indiquer les connexions qu'il faut ajouter. Le paramètre `sequenceNumber` doit être incrémenté de 1, modulo 256, à chaque nouveau message `MultilinkRequest.addConnection` envoyé.

Dès réception de ce message, l'entité initiale doit répondre par un message `MultilinkResponse.addConnection` indiquant soit qu'elle a l'intention d'ajouter la connexion comme demandé, soit qu'elle n'en a pas l'intention, et précisant le code de motif approprié. Le paramètre `sequenceNumber` doit avoir la même valeur que celle qu'il a dans le message `MultilinkRequest.addConnection` correspondant.

B.11.3 removeConnection – Demande et réponse

Le message `MultilinkRequest.removeConnection` peut être utilisé par l'entité initiale ou par l'entité qui répond, comme défini dans l'Annexe F/H.324, pour demander que l'extrémité distante supprime un canal de l'ensemble des canaux H.226. Il est utilisé dans le cadre de la procédure de l'Annexe F/H.324 pour supprimer des connexions physiques. Le paramètre `connectionIdentifier` doit identifier le canal à supprimer, en utilisant la numérotation des canaux reçue via H.226 en provenance du terminal recevant le message `MultilinkRequest.removeConnection`.

Le message `MultilinkResponse.removeConnection` doit être envoyé en réponse, après que le canal ait été supprimé de l'ensemble des canaux H.226, et ce message doit indiquer que le canal en question n'est plus utilisé (ou ne l'a jamais été). Le paramètre `connectionIdentifier` doit avoir la même valeur que celle qu'il a dans le message `MultilinkRequest.removeConnection` correspondant.

B.11.4 maximumHeaderInterval – Demande et réponse

Le message `MultilinkRequest.maximumHeaderInterval` peut être utilisé pour demander le véritable intervalle d'en-tête maximal H.226 utilisé par l'émetteur distant sans en demander la modification (paramètre `currentIntervalInformation`) ou pour utiliser une certaine valeur à la place de la valeur courante (paramètre `requestedInterval`, valeur en millisecondes).

Le message `MultilinkResponse.maximumHeaderInterval` doit être envoyé en réponse. Si la demande correspondante était une demande d'information sur la valeur courante, le terminal doit indiquer, dans la réponse, la valeur que son émetteur utilise comme intervalle d'en-tête maximal. Si la demande correspondante spécifiait une certaine valeur à utiliser, le terminal doit tenter de satisfaire cette demande en modifiant l'intervalle d'en-tête maximal utilisé par son émetteur. Que l'intervalle d'en-tête maximal soit modifié ou non, la réponse doit indiquer la nouvelle valeur utilisée (qui peut être différente de la valeur demandée).

B.11.5 Indications relatives aux multiliaisons

Le message `MultilinkIndication.crcDesired` peut être envoyé par un terminal pour indiquer qu'il souhaite que le terminal distant envoie le code CRC associé aux données H.226 facultatif dans tous les ensembles de données ultérieurs. Le terminal récepteur peut facultativement satisfaire à la demande: aucun acquittement explicite ni aucune réponse explicite n'est exigé.

Le message `MultilinkIndication.excessiveError` peut être envoyé pour indiquer au terminal distant qu'un nombre d'erreurs excessif est reçu sur une connexion donnée. Le moyen utilisé par le terminal pour déterminer le taux d'erreurs ou le critère utilisé pour déterminer ce qui est excessif est défini localement au niveau de ce terminal. La connexion est indiquée au moyen du paramètre `connectionIdentifier`. Dès réception de ce message, le terminal en question peut choisir de prendre une mesure correctrice. La mesure particulière à prendre n'est pas spécifiée.

B.11.6 DialingInformation

Le type `DialingInformation` est utilisé pour fournir des informations de numérotation explicites (numéros de téléphone) afin de permettre l'établissement automatique de connexions physiques. Le paramètre "differential" permet de donner une liste de paramètres `DialingInformationNumber`, un

pour chaque connexion supplémentaire potentielle. La longueur de cette liste correspond au nombre maximal de connexions supplémentaires disponibles. Si ces informations ne sont pas disponibles, on choisit le paramètre `infoNotAvailable`, qui indique uniquement le nombre de connexions supplémentaires qui sont disponibles.

B.11.7 DialingInformationNumber

Le type `DialingInformationNumber` comprend jusqu'à trois sous-paramètres qui fournissent, pour une connexion physique donnée, les informations de numérotation qui sont différentes des informations correspondantes pour une connexion initiale déjà établie.

Le paramètre `networkAddress` doit inclure la partie la moins significative (partie droite) du numéro de téléphone associé à la connexion considérée, jusqu'au chiffre le plus significatif qui est différent du numéro associé à une connexion établie initialement (et y compris ce chiffre), et ne doit inclure aucun chiffre plus significatif. Si le numéro associé à la connexion considérée est identique au numéro associé à la connexion initiale, le paramètre `networkAddress` doit être composé d'une chaîne de longueur nulle (étant donné que les numéros de téléphone ne comportent pas de chiffre différent).

NOTE – On utilise la méthode consistant à indiquer les chiffres différents plutôt que la chaîne de chiffres E.164 complète car les premiers chiffres du numéro à composer peuvent varier en fonction de la situation géographique des deux terminaux; par exemple s'ils sont situés dans la même ville ou non.

Si on utilise une sous-adresse pour la numérotation et si la sous-adresse associée à la connexion considérée est différente de celle qui est associée à la connexion initiale, l'entité qui répond doit inclure la sous-adresse, en intégralité, dans le paramètre facultatif `subAddress`.

Les types de réseau pris en charge pour la connexion (RTGC, RNIS ou les deux) doivent être indiqués dans le paramètre `networkType`.

B.11.8 DialingInformationNetworkType

Le type `DialingInformationNetworkType` indique un type de réseau à commutation de circuits, à savoir `n-isdn` (RNIS-BE), `gstn` (RTGC), ou `mobile` (Mobile).

B.11.9 ConnectionIdentifier

Le type `ConnectionIdentifier` est utilisé pour identifier de manière univoque une connexion physique individuelle dans un ensemble de canaux H.226, une combinaison de paramètres `channelTag` et `sequenceNumber` d'un en-tête H.226 étant utilisée. Si aucune étiquette de canal (`ChannelTag`) n'a été spécifiée dans l'en-tête, il faut utiliser la valeur zéro pour le paramètre `channelTag`.

B.12 Messages liés à la modification du débit d'une voie logique

Les messages `LogicalChannelRateRequest`, `LogicalChannelRateAcknowledge`, `LogicalChannelRateReject` et `LogicalChannelRateRelease` servent à modifier le débit d'une voie logique. La procédure d'utilisation de ces messages consiste pour un terminal à demander un débit cible pour une voie logique donnée et pour le terminal distant à accepter ou rejeter la demande.

Ces messages fournissent un niveau d'interaction amélioré, en autorisant la demande de débit cible par opposition à la contrainte de débit maximal imposée par la commande `FlowControlCommand` et en indiquant en retour si la demande est acceptée ou rejetée.

B.12.1 LogicalChannelRateRequest

Ce message est utilisé par un terminal pour demander une modification du débit de la voie logique sur laquelle il reçoit des données.

`sequenceNumber` sert à étiqueter les instances de `LogicalChannelRateRequest` de manière à pouvoir identifier la réponse correspondante.

logicalChannelNumber indique la voie logique à laquelle la demande de modification de débit se rapporte.

maximumBitRate indique le débit maximal demandé (unité: 100 bit/s) pour la voie logique.

B.12.2 LogicalChannelRateAcknowledge

Ce message est envoyé pour accepter une demande de modification du débit d'une voie logique.

sequenceNumber sera identique au sequenceNumber du message LogicalChannelRateRequest auquel ce message sert de réponse.

logicalChannelNumber indique la voie logique à laquelle la demande de modification de débit se rapporte.

maximumBitRate indique le débit maximal (unité: 100 bit/s) accepté par le terminal pour la voie logique.

B.12.3 LogicalChannelRateReject

Ce message est envoyé pour rejeter une demande de modification du débit d'une voie logique.

sequenceNumber sera identique au sequenceNumber du message LogicalChannelRateRequest auquel ce message sert de réponse.

logicalChannelNumber indique la voie logique à laquelle la demande de modification de débit se rapporte.

rejectReason indique le motif du rejet de la demande. Les motifs actuellement définis sont les suivants: motif non défini et ressources insuffisantes.

currentMaximumBitRate indique le débit maximal (unité: 100 bit/s) que le terminal va utiliser sur la voie logique, pour l'émission de données.

B.12.4 LogicalChannelRateRelease

Ce message est envoyé en cas d'expiration de temporisation.

B.13 Commandes

Un message de commande appelle une action mais pas de réponse explicite.

B.13.1 SendTerminalCapabilitySet

specificRequest donne l'ordre au terminal distant d'indiquer ses capacités d'émission et de réception en envoyant au moins un ensemble TerminalCapabilitySet contenant les informations demandées, comme cela est spécifié ci-dessous. Cette commande peut être envoyée à n'importe quel moment pour obtenir les capacités du terminal distant, par exemple, par suite d'une interruption ou de tout autre motif de défaillance; cependant, de telles commandes ne doivent pas être envoyées répétitivement sans raison majeure.

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

Le paramètre booléen multiplexCapability, quand il est mis à "Vrai", permet de demander la transmission du paramètre MultiplexCapability.

La structure capabilityTableEntryNumbers est un ensemble de numéros CapabilityTableEntryNumbers indiquant les entrées CapabilityTableEntrys dont le terminal demande la transmission.

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

`genericRequest` exige que le terminal distant envoie l'ensemble de ses capacités.

B.13.2 Encryption

Cette commande est utilisée pour échanger des capacités de chiffrement et pour donner l'instruction de transmettre un vecteur d'initialisation (IV, *initialization vector*), voir les Recommandations UIT-T H.233 [14] et UIT-T H.234 [15].

`encryptionSE` est un message d'échange de sessions (SE, *session exchange*) selon H.233, sauf que les bits de protection contre les erreurs décrits dans la Rec. UIT-T H.233 ne doivent pas être appliqués.

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

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

B.13.3 FlowControl

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

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

`maximumBitRate` (unité: 100 bit/s) est moyenné, 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 qui sont pris en compte dans le calcul du débit ne sont pas indiqués dans la présente Recommandation, mais doivent être spécifiés dans des Recommandations utilisant la présente Recommandation.

Chaque transmission de cette commande concerne une voie logique spécifique ou l'ensemble du multiplex. Plusieurs commandes de ce type peuvent être valables en même temps, jusqu'à atteindre le nombre de voies logiques plus une, pour la limite concernant le multiplex tout entier.

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

B.13.4 EndSession

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

`disconnect` indique que la connexion sera abandonnée.

gstnOptions: choix entre plusieurs possibilités à la fin de la session H.245, quand un modem de la série V est utilisé sur le RTGC.

Les options possibles sont indiquées dans le Tableau B.14.

Tableau B.14/H.245 – Options après la commande EndSessionCommand quand un modem de la série V est utilisé sur le RTGC

Codage ASN.1	Option
telephonyMode	Le terminal doit lancer les procédures de libération définies dans la Recommandation de la série V relative au modem, sauf qu'il ne doit pas libérer physiquement la connexion RTGC.
v8bis	Le terminal doit déclencher les procédures de libération définies dans la Recommandation de la série V relative au modem et commencer une session V.8 bis.
v34DSVD	Le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour prendre en charge le mode V.70.
v34DuplexFAX	Le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour prendre en charge le mode FAX T.30 [27].
v34H324	Le terminal doit maintenir la connexion de modem V.34, mais l'utiliser pour prendre en charge le mode de la Rec. UIT-T H.324 [24].

isdnOptions: choix entre plusieurs possibilités à la fin de la session H.245 quand un terminal de communication numérique est utilisé sur un réseau numérique.

Les options possibles sont présentées dans le Tableau B.15.

Tableau B.15/H.245 – Options après la commande EndSessionCommand quand un terminal de communication numérique est utilisé sur un réseau numérique

Codage ASN.1	Option
telephonyMode	Le terminal doit lancer les procédures de libération définies dans la Recommandation spécifiant les communications sur le canal numérique auquel il est connecté, sauf qu'il ne doit pas libérer physiquement la connexion numérique.
v140	Le terminal doit lancer les procédures de libération définies dans la Recommandation spécifiant les communications sur le canal numérique auquel il est connecté et lancer une session V.140 [39].
terminalOnHold	Le terminal doit lancer les procédures de "terminal maintenu en garde", qui sont définies dans la Recommandation spécifiant les communications sur le canal numérique auquel il est connecté.

B.13.5 MiscellaneousCommand

Cette commande est utilisée pour toute une série de commandes, dont certaines sont décrites dans les Recommandations UIT-T H.221 [7] et H.230 [13].

logicalChannelNumber indique le numéro de la voie logique à laquelle la commande s'applique. Ce numéro doit indiquer une voie logique ouverte pour les données vidéo quand le type est l'un des types suivants: videoFreezePicture, videoFastUpdatePicture, videoFastUpdateGOB, videoTemporalSpatialTradeOff, videoSendSyncEveryGOB, videoFastUpdateMB, et videoSendSyncEveryGOBCancel, lostPicture, lostPartialPicture et recoveryReferencePicture. Quand le type est l'un des types suivants: equaliseDelay, zeroDelay, multipointModeCommand ou cancelMultipointModeCommand impliquant plusieurs voies logiques, le numéro logicalChannelNumber devra être arbitraire, mais devra être un numéro LogicalChannelNumber valable (c'est-à-dire dans la gamme 1-65535) et le récepteur devra ignorer la valeur.

equaliseDelay et zeroDelay devront avoir la même signification que les commandes ACE et ACZ définies dans la Rec. UIT-T H.230 [13].

multipointModeCommand donne l'ordre à un terminal récepteur de répondre à toutes les demandes requestMode présentées par le pont de conférence. Un exemple de changement de mode est un changement de mode audio de G.711 à G.728.

cancelMultipointModeCommand annule une commande multipointModeCommand qui a été préalablement émise.

videoFreezePicture donne l'ordre au décodeur vidéo de rafraîchir l'image vidéo courante puis d'afficher l'image gelée jusqu'à la réception du signal approprié de commande de libération d'image gelée.

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

videoFastUpdateGOB donne l'ordre au codeur vidéo distant d'effectuer un rafraîchissement accéléré d'au moins un groupe de blocs (GOB). firstGOB indique le numéro du premier groupe de blocs (GOB) devant être rafraîchi et numberOfGOBs indique le nombre de groupes de blocs (GOB) devant être rafraîchis. VideoFastUpdateGOB ne sera utilisé qu'avec des algorithmes de compression de signaux vidéo définissant des groupes de blocs (GOB), par exemple les algorithmes H.261 et H.263. La numérotation des groupes de blocs (GOB) se fait comme indiqué dans la Rec. UIT-T H.263, même en cas d'application de la Rec. UIT-T H.261. Le premier groupe de blocs (GOB) de l'image est le groupe de blocs (GOB) numéro 0, le deuxième groupe de blocs (GOB) est le groupe de blocs (GOB) numéro 1, et ainsi de suite. L'analyse des groupes de blocs (GOB) aux fins de l'interprétation du paramètre numberOfGOBs doit être effectuée conformément à la norme de codage vidéo appropriée, de manière que le deuxième groupe de blocs (GOB) d'une image de format CIF H.261 se situe à droite du premier groupe blocs (GOB), alors qu'il se situe au-dessous de celui-ci sur des images de format QCIF H.261 et de format H.263.

videoTemporalSpatialTradeOff donne l'ordre au codeur vidéo distant de modifier son compromis entre la résolution temporelle et la résolution spatiale. Une valeur de 0 régit une haute résolution spatiale et une valeur de 31 régit une 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 résolution spatiale ou de fréquence de trame.

videoSendSyncEveryGOB donne l'ordre au codeur vidéo distant d'utiliser la synchronisation pour chaque groupe de blocs (GOB) comme cela est défini dans la Rec. UIT-T H.263 [20], jusqu'à ce que la commande videoSendSyncEveryGOBCancel soit reçue, moment à partir duquel le codeur vidéo distant peut décider de la fréquence de synchronisation des groupes de blocs (GOB). Ces commandes ne doivent être utilisées qu'avec un codage vidéo conforme à la Rec. UIT-T H.263.

videoFastUpdateMB donne l'ordre au codeur vidéo distant d'effectuer un rafraîchissement accéléré d'un ou de plusieurs macroblocs. Le paramètre firstGOB indique le numéro du premier groupe de blocs (GOB) devant être rafraîchi, le paramètre firstMB indique le numéro du premier macrobloc devant être rafraîchi. Le paramètre numberOfMBs indique le nombre de macroblocs devant être rafraîchi. VideoFastUpdateMB ne sera utilisé qu'avec les algorithmes de compression d'image qui définissent des macroblocs, par exemple les algorithmes H.261 et H.263. Les terminaux peuvent répondre à cette commande par un rafraîchissement des groupes de blocs (GOB) comprenant les macroblocs demandés. La numérotation des groupes de blocs (GOB) se fait comme indiquée dans la Rec. UIT-T H.263, même en cas d'application de la Rec. UIT-T H.261. Le premier groupe de blocs (GOB) de l'image est le groupe de blocs (GOB) numéro 0, le deuxième groupe de blocs (GOB) est le groupe de blocs (GOB) numéro 1, et ainsi de suite. L'un ou l'autre ou l'un et l'autre des paramètres firstGOB et firstMB doivent être présents. Lorsque le paramètre firstGOB est présent et que le paramètre firstMB est absent, le premier macrobloc devant être rafraîchi est le premier macrobloc du groupe de blocs (GOB) indiqué. Lorsque les paramètres firstGOB et firstMB sont tous deux présents, le paramètre first MB est indiqué par rapport au début du paramètre firstGOB indiqué, de manière que le premier macrobloc du groupe de blocs (GOB) indiqué soit considéré

comme étant le macrobloc numéro 1. Lorsque le paramètre firstGOB est absent et que le paramètre firstMB est présent, le paramètre firstMB est indiqué par rapport à l'angle supérieur gauche de l'image, le macrobloc correspondant étant considéré comme le macrobloc numéro 1. L'ordre d'analyse des macroblocs dans le reste du groupe de blocs (GOB) et au-delà de ce point est défini comme étant l'ordre d'analyse de la norme de codage vidéo appliquée. L'ordre d'analyse à partir du troisième groupe de blocs (GOB) d'une image de format CIF H.261 commence donc à partir du macrobloc numéro 1, en sautant vers le bas les trois rangées du groupe de blocs (GOB) pour atteindre le macrobloc numéro 33 dans la onzième colonne de la sixième rangée, pour remonter verticalement et commencer à analyser le groupe de blocs (GOB) suivant à partir de la douzième colonne de la quatrième rangée.

maxH223MUXPDUsize donne l'ordre à l'émetteur de réduire la taille des unités MUX-PDU H223 qu'il envoie à un nombre maximal d'octets spécifié.

encryptionUpdate et EncryptionUpdateRequest permettent d'émettre et de distribuer de nouvelles clés à utiliser pour le chiffrement des voies de médias indiquées.

SwitchReceiveMediaOn et switchReceiveMediaOff peuvent être utilisées par une entité de commande multipoint (MC) pour ordonner à un point d'extrémité de passer d'une voie monodiffusion à une voie multidiffusion lorsque l'entité MC+MP mélange les signaux audio. Dans ce cas, lorsque le flux de l'entité MC comprend les signaux audio du terminal, l'entité MC+MP peut basculer le point d'extrémité vers un flux à un seul destinataire qui contiendrait un mélange spécial pour le terminal, les signaux audio étant supprimés.

switchReceiveMediaOff permet à une entité de commande multipoint d'indiquer à un point d'extrémité qu'une voie logique déterminée ne doit pas être utilisée pour recevoir des médias.

switchReceiveMediaOn permet à une entité de commande multipoint d'indiquer à un point d'extrémité qu'une voie logique déterminée doit être utilisée pour recevoir des médias.

doOneProgression ordonne au codeur vidéo de commencer à produire une séquence de raffinement progressif. Dans ce mode, le codeur produit des signaux vidéo comprenant une image suivie de zéro, une ou plusieurs trames de raffinement de la qualité de la même image. Le codeur demeure dans ce mode jusqu'à ce qu'il détermine qu'un niveau de fidélité acceptable a été atteint ou jusqu'à réception de la commande progressiveRefinementAbortOne. En outre, il doit insérer l'étiquette de début de segment de raffinement progressif et l'étiquette de fin de segment de raffinement progressif pour marquer le début et la fin du raffinement progressif comme le définit la spécification des informations d'amélioration supplémentaire de l'Annexe L/H.263.

doContinuousProgressions ordonne au codeur vidéo de commencer à produire des séquences de raffinement progressif. Dans ce mode, le codeur produit des signaux vidéo comprenant une image suivie de zéro, une ou plusieurs trames de raffinement de la qualité de la même image. Lorsque le codeur détermine qu'un niveau de fidélité acceptable a été atteint ou lorsque la commande progressiveRefinementAbortOne est reçue, il met fin au raffinement de l'image en cours et lance le raffinement progressif d'une autre image. La séquence de raffinement progressif se poursuit jusqu'à réception de la commande progressiveRefinementAbortContinuous. En outre, le codeur doit insérer des étiquettes de début de segment de raffinement progressif et de fin de segment de raffinement progressif pour marquer le début et la fin de chaque raffinement progressif comme le définit la spécification des informations d'amélioration supplémentaire de l'Annexe L/H.263.

doOneIndependentProgression ordonne au codeur vidéo de lancer une séquence de raffinement progressif indépendant. Dans ce mode, le codeur produit des signaux vidéo comprenant une image Intra suivie de zéro, une ou plusieurs trames de raffinement de la qualité de la même image. Le codeur demeure dans ce mode jusqu'à ce qu'il détermine qu'un niveau de fidélité acceptable a été atteint ou jusqu'à réception de la commande progressiveRefinementAbortOne. En outre, il doit insérer l'étiquette de début de segment de raffinement progressif et l'étiquette de fin de segment de

raffinement progressif pour marquer le début et la fin du raffinement progressif comme le définit la spécification des informations d'amélioration supplémentaire de l'Annexe L/H.263.

`doContinuousIndependentProgressions` ordonne au codeur vidéo de commencer à produire des séquences de raffinement progressif indépendant. Dans ce mode, le codeur produit des signaux vidéo comprenant une image Intra suivie de zéro, une ou plusieurs trames de raffinement de la qualité de la même image. Lorsque le codeur détermine qu'un niveau de fidélité acceptable a été atteint ou lorsque la commande `progressiveRefinementAbortOne` est reçue, il met fin au raffinement progressif en cours et lance un autre raffinement progressif indépendant pour une autre image. La séquence de raffinement progressif indépendant se poursuit jusqu'à réception de la commande `progressiveRefinementAbortContinuous`. En outre, le codeur doit insérer des étiquettes de début de segment de raffinement progressif et de fin de segment de raffinement progressif pour marquer le début et la fin de chaque raffinement progressif indépendant comme le définit la spécification des informations d'amélioration supplémentaire de l'Annexe L/H.263.

`progressiveRefinementAbortOne` ordonne au codeur vidéo de mettre fin à la fonction `doOneProgression`, à la fonction `doOneIndependentProgression` ou au raffinement progressif en cours dans la séquence de raffinement progressif des fonctions `doContinuousProgressions` ou `doContinuousIndependentProgressions`.

`progressiveRefinementAbortContinuous` ordonne au codeur vidéo de mettre fin à la fonction `doContinuousProgressions` ou `doContinuousIndependentProgressions`.

La commande `videoBadMBs` ordonne au codeur vidéo distant de prendre des mesures correctives lorsqu'un ensemble de macroblocs (MB) n'a pas été reçu de façon satisfaisante. Le codeur doit utiliser les informations ainsi transmises pour prendre des mesures visant à rétablir la qualité vidéo. Contrairement à la commande `videoNotDecodedMBs`, la commande `videoBadMBs` n'indique pas de manière précise comment le décodeur a traité l'ensemble spécifié de macroblocs (MB). Le codeur devrait réagir à cette commande en veillant à ce que l'ensemble spécifié de macroblocs ne soit pas utilisé aux fins de la prévision des images vidéo après réception de la commande par le codeur. Bien que n'étant pas définie, l'action concrète que doit prendre le codeur peut consister en toute mesure corrective appropriée, comme l'envoi d'une trame INTRA. Cette commande ne doit pas être transmise par un décodeur vidéo si le codeur distant correspondant n'a pas indiqué la capacité `videoBadMBsCap`. Cette commande ne doit être utilisée qu'avec des algorithmes de codage vidéo définissant des macroblocs (MB), de type H.261, H.262, IS11172 et H.263, par exemple. La numérotation des macroblocs (MB) se fait en suivant l'ordre d'analyse par balayage de l'image, le macrobloc (MB) supérieur gauche de l'image étant le macrobloc numéro 1, les numéros des macroblocs (MB) allant en augmentant d'abord de gauche à droite puis de haut en bas.

La commande `lostPicture` ordonne au codeur vidéo distant de prendre des mesures correctives en raison de la perte ou de la dégradation des images indiquées. Celles-ci sont indiquées par le numéro `pictureNumber` (numéro d'image provisoire) ou par l'indice `longTermPictureIndex` (indice d'image durable). Un codeur doté de la capacité spécifiée dans l'Annexe U/H.263 (sélection d'image de référence améliorée, avec ou sans suppression de sous-image) ou dans W.6.3.12/H.263 (numéro d'image) doit pouvoir comprendre ce message et prendre des mesures correctives.

La commande `lostPartialPicture` ordonne au codeur vidéo distant de prendre des mesures correctives lorsqu'un ensemble de macroblocs (MB) n'a pas été reçu de façon satisfaisante. Cette commande est identique à la commande `videoBadMBs` à ceci près que l'image est indiquée par le numéro `pictureNumber` (numéro d'image provisoire) ou par l'indice `longTermPictureIndex` (indice d'image durable). Un codeur doté de la capacité spécifiée dans l'Annexe U/H.263 (sélection d'image de référence améliorée, avec ou sans suppression de sous-image) ou dans W.6.3.12/H.263 (numéro d'image) doit pouvoir comprendre ce message et prendre des mesures correctives.

La commande `recoveryReferencePicture` ordonne au codeur vidéo distant de n'utiliser les images indiquées qu'à des fins de prévision. Celles-ci sont indiquées par le numéro `pictureNumber` (numéro

d'image provisoire) ou par l'indice longTermPictureIndex (indice d'image durable). Un codeur doté de la capacité spécifiée dans l'Annexe U/H.263 (sélection d'image de référence améliorée, avec ou sans suppression de sous-image) ou dans W.6.3.12/H.263 (numéro d'image) doit pouvoir comprendre ce message et prendre des mesures correctives. Ce message peut être envoyé depuis un décodeur dans le cas de figure où les images indiquées ont été convenablement reçues et décodées et où les autres images (non spécifiées) ont été dégradées par la transmission.

La commande encryptionUpdateCommand doit être utilisée dans la Rec. UIT-T H.235 pour la procédure améliorée de mise à jour des clés afin de distribuer de nouvelles données de clés de session (voir B.2.6.2/H.235). La commande multiplePayloadStream n'est utilisée que lorsqu'un flux à multiples charges utiles doit être renouvelé en terme de clés, auquel cas le type dynamique de charge utile contenu dans EncryptionSync doit être ignoré.

La commande encryptionUpdateAck doit être utilisée dans la Rec. UIT-T H.235 pour la procédure améliorée de mise à jour des clés afin d'autoriser la réception de l'acquittement – par le système esclave – de nouvelles données de clés de session sur une voie logique détenue par le système maître (voir B.2.6.2/H.235).

La commande direction doit indiquer le sens (masterToSlave ou slaveToMaster) de la voie logique sur laquelle les données de clé sont en cours de distribution (voir B.2.6.2/H.235).

B.13.6 ConferenceCommand

La structure BroadcastMyLogicalChannel sera identique à la commande MCV H.230 utilisée conformément à la procédure décrite au § 6.3.2.1/H.243 mais ne se rapportera qu'à une seule voie logique. A noter qu'en cas d'utilisation de la procédure MCV préférée décrite au § 6.3.2.2/H.243 (c'est-à-dire lorsque les deux extrémités d'une liaison terminal-MCU ou inter-MCU ont une capacité de visualisation multipoints), on utilise en lieu et place la forme conferenceRequest de la structure BroadcastMyLogicalChannel.

La structure CancelBroadcastMyLogicalChannel sera identique à Cancel-MCV selon H.230 mais ne se rapportera qu'à une seule voie logique.

La structure MakeTerminalBroadcaster sera définie comme VCB selon H.230.

La structure CancelMakeTerminalBroadcaster sera définie comme Cancel-VCB.

La structure SendThisSource sera définie comme VCS selon H.230.

La structure CancelSendThisSource sera définie comme Cancel-VCS selon H.230.

La structure DropConference sera définie comme CCK selon H.230.

SubstituteConferenceIDCommand permet à une entité de commande multipoint active de modifier l'identificateur de conférence et de déplacer le destinataire de cette commande vers une autre conférence. Ce destinataire utilisera l'identificateur nouvellement attribué dans tous les futurs messages de signalisation d'appel.

B.13.7 H.223MultiplexReconfiguration

h223ModeChange ordonne à l'émetteur de faire passer le niveau du mode de multiplexage comme décrit à l'Annexe C/H.324 au niveau 0, au niveau 1, au niveau 2 ou au niveau 2 avec l'en-tête facultatif décrit à l'Annexe B/H.223.

h223AnnexADoubleFlag ordonne à l'émetteur de lancer ou de cesser l'utilisation du mode double fanion indiqué à l'Annexe A/H.223.

B.13.8 NewATMVCCCommand

Cette commande est utilisée pour ordonner au terminal distant d'ouvrir une voie virtuelle ATM avec les paramètres donnés.

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

Le paramètre `bitRate` indique le débit (unité: 64 kbit/s) de la voie virtuelle, mesuré au niveau du point AAL-SAP.

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

Le paramètre `bitRateLockedToNetworkClock` indique que le débit de la voie virtuelle est verrouillé sur l'horloge du réseau local. Ceci ne permet pas de garantir que ce verrouillage existera au niveau du récepteur, car il est possible qu'aucune horloge de réseau commune ne soit disponible.

Le paramètre `aal` indique quelle couche d'adaptation ATM sera utilisée et précise ses paramètres.

La séquence `aal1` indique quelles options, spécifiées dans la Rec. UIT-T I.363.x [25], sont prises en charge pour la couche d'adaptation ATM 1. Les codages sont définis dans le Tableau B.1.

La séquence `aal5` indique quelles options, spécifiées dans la Rec. UIT-T I.363.x [25], sont prises en charge pour la couche d'adaptation ATM 5. Les paramètres `forwardMaximumSDUSize` et `backwardMaximumSDUSize` indiquent la taille maximale d'unité CPCS-SDU dans les sens direct et inverse, en octets.

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

B.13.9 MobileMultilinkReconfigurationCommand

Cette commande est utilisée pour ordonner à l'émetteur de modifier la configuration de trame multiliasion comme indiqué dans l'Annexe H/H.324.

Le paramètre `sampleSize` indique la taille d'un échantillon en octets. Un échantillon correspond au nombre d'octets qui seront répartis entre les voies physiques disponibles.

Le paramètre `samplesPerFrame` indique la longueur de la charge utile multiliasion en échantillons.

Le paramètre `status` indique l'état du récepteur au moment où celui-ci envoie ce message de commande. En cas de synchronisation, cela indique que le récepteur a établi la synchronisation de trames et qu'il ordonne à l'émetteur de commencer à émettre la trame d'en-tête comprimé. En cas de reconfiguration, cela ordonne à l'émetteur de modifier la taille de l'échantillon et/ou la longueur de la trame et à commencer à émettre la trame d'en-tête complet.

B.14 Indications

Une indication contient des informations n'appelant ni action ni réponse.

B.14.1 FunctionNotUnderstood

Cette indication est utilisée pour renvoyer à l'émetteur des demandes, des réponses et des commandes non comprises.

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

NOTE – L'indication `FunctionNotUnderstood` était appelée `FunctionNotSupported` dans la version 1 de la présente Recommandation. Le nom a été modifié pour permettre l'ajout d'une indication `FunctionNotSupported` plus performante tout en conservant la compatibilité vers l'arrière avec la syntaxe de la version 1.

B.14.2 MiscellaneousIndication

Cette indication est utilisée pour diverses indications dont certaines sont présentes dans les Recommandations UIT-T H.221 [7] et UIT-T H.230 [13].

logicalChannelNumber indique le numéro de voie logique auquel s'applique l'indication. Ce numéro doit indiquer un numéro de voie logique ouverte pour les signaux vidéo quand le type vaut videoIndicateReadyToActivate ou videoTemporalSpatialTradeOff. Quand le type vaut multipointConference, cancelMultipointConference, multipointZeroComm, cancelMultipointZeroComm, multipointSecondaryStatus ou cancelMultipointSecondaryStatus impliquant l'utilisation de plusieurs voies logiques, le numéro logicalChannelNumber sera choisi arbitrairement et sera mis à une valeur LogicalChannelNumber valable (c'est-à-dire comprise entre 1 et 65535) et le récepteur ignorera cette valeur.

logicalChannelInactive sert à indiquer que le contenu de la voie logique ne représente pas un signal normal. Elle est analogue à AIM ou à VIS défini dans la Rec. UIT-T H.230.

logicalChannelActive est complémentaire de logicalChannelInactive. Elle est analogue à AIA et à VIA définis dans la Rec. UIT-T H.230. MultipointZeroComm, cancelMultipointZeroComm, multipointSecondaryStatus et cancelMultipointSecondaryStatus auront respectivement la même signification que MIZ, cancelMIZ, MIS et cancelMIS, définis dans la Rec. UIT-T H.230.

multipointConference indique que le terminal est associé à une conférence multipoint H.243 et qu'il est prévu que le terminal obéisse à une symétrisation du débit. La symétrisation du débit binaire doit cependant être mise en application par des commandes FlowControlCommand. Il convient de noter que multipointConference a exactement la même signification que MCC dans la Rec. UIT-T H.230. Il convient de noter également que multipointConference, comme MCC, n'exige pas de symétrie de mode.

videoIndicateReadyToActivate aura la même signification que VIR défini dans la Rec. UIT-T H.230, c'est-à-dire qu'il est transmis par un terminal dont l'utilisateur a décidé de ne pas envoyer la vidéo avant d'avoir également reçu la vidéo à partir de l'autre terminal.

videoTemporalSpatialTradeOff indique au décodeur vidéo distant quel est le compromis actuel entre la résolution temporelle et la résolution spatiale. Une valeur de 0 indique une résolution spatiale élevée et une valeur de 31 indique une fréquence image élevée. Les valeurs comprises entre 0 et 31 indiquent de façon monotone une fréquence d'image élevée. Les valeurs réelles ne correspondent pas à des valeurs précises de résolution spatiale ou de fréquence d'image. Un terminal qui a déclaré une capacité videoTemporalSpatialTradeOffCapability transmettra cette indication à chaque fois qu'il modifie son compromis et quand une voie logique vidéo est initialement ouverte.

videoNotDecodedMBs indique au codeur vidéo distant qu'un ensemble de macroblocs a été reçu de façon erronée et que n'importe quel macrobloc dans l'ensemble spécifié a été traité comme non codé. Le codeur peut utiliser ces informations pour compenser les erreurs de transmission, comme cela est illustré dans l'Appendice I/H.263. Le paramètre firstMB indique le numéro du premier macrobloc traité comme étant non codé, et le paramètre numberOfMBs indique le nombre de macroblocs traités comme étant non codés. La numérotation des macroblocs est effectuée de manière le macrobloc situé dans l'angle supérieur gauche de l'image soit considéré comme étant le macrobloc numéro 1, le numéro de chaque macrobloc augmentant de gauche à droite, puis de haut en bas dans l'ordre d'analyse par balayage (de telle sorte que dans l'hypothèse d'une image comportant un total de N macroblocs, le macrobloc inférieur droit est considéré comme étant le macrobloc numéro N). La référence temporelle de l'image contenant des macroblocs non décodés est indiquée dans le paramètre temporalReference. Cette indication ne sera utilisée qu'avec l'algorithme de compression d'image H.263.

B.14.3 JitterIndication

Cette indication sert à préciser la quantité de gigue d'une voie logique telle qu'elle a été évaluée par le terminal récepteur. Elle peut être utile pour choisir le débit et la gestion de la mémoire tampon des voies 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 limiter les fluctuations de la mémoire tampon du décodeur pour éviter les surremplissages ou sous-remplissages, compte tenu de la gigue existante. Si le codeur accepte cette option, cela permettra le bon fonctionnement des modèles existants de mémoires tampons de décodeurs vidéo, quelle que soit l'amplitude de la gigue reçue, de même que le bon fonctionnement avec un retard minimal.

Quand le paramètre scope est du type logicalChannelNumber, les informations s'appliquent à une voie logique donnée, quand il est du type resourceID, les informations s'appliquent au canal virtuel ATM donné et quand il est du type wholeMultiplex, les informations s'appliquent à l'ensemble du multiplex.

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

L'élément estimatedReceivedJitterMantissa indique la mantisse de l'évaluation de gigue telle qu'elle est indiquée dans le Tableau B.16.

Tableau B.16/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 indiqué dans la Tableau B.17.

Tableau B.17/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 sait simplement que l'estimation de gigue est supérieure à 7,5 secondes.

Le compteur `skippedFrameCount` indique le nombre de trames qui a été omis par le décodeur depuis que le dernier message `JitterIndication` a été reçu. Etant donné que la valeur maximale pouvant être codée est égale à 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` dans la Rec. UIT-T H.262 [19].

B.14.4 H.223SkewIndication

Cette indication est utilisée 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 paramètre `skew` indique, en millisecondes, le retard devant être appliqué aux données appartenant à la voie logique `logicalChannelNumber2`, observée à la sortie du multiplex, pour réaliser la synchronisation avec la voie logique `logicalChannelNumber1`, observée à la sortie du multiplex. Le paramètre `skew` inclut les différences de durée d'échantillonnage, de retard introduit par le codeur, et de retard introduit par la mémoire tampon de l'émetteur, et est mesuré par rapport à l'instant de transmission du premier bit de données représentant un échantillon donné. Le retard réel nécessaire à la synchronisation dépend de la mise en œuvre du décodeur et doit être considéré localement lors de la conception du récepteur.

B.14.5 NewATMVCIndication

Cette indication est utilisée pour indiquer les paramètres de la voie virtuelle ATM que le terminal a l'intention d'ouvrir.

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

Le paramètre `bitRate` indique le débit (unité: 64 kbit/s) de la voie virtuelle, mesuré au niveau du point AAL-SAP.

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

La structure `bitRateLockedToNetworkClock` indique que le débit de la voie virtuelle est verrouillé sur l'horloge de réseau local. Cela ne garantit pas que ce verrouillage existera au niveau du récepteur car il est possible qu'aucune horloge de réseau commune ne soit disponible.

Le paramètre `aal` indique quelle couche d'adaptation ATM sera utilisée et précise ses paramètres.

La séquence `aal1` de séquence indique quelles options de la couche d'adaptation 1 de l'ATM, comme cela est spécifié dans la Rec. UIT-T I.363 [25], sont prises en charge. Les codages sont définis dans le Tableau B.1.

La séquence `aal5` indique quelles options de la couche d'adaptation 5 de l'ATM, comme cela est spécifié dans la Rec. UIT-T I.363 [25], sont prises en charge. Les paramètres `forwardMaximumSDUSize` et `backwardMaximumSDUSize` indiquent la taille maximale d'unité CPCS-SDU dans les sens direct et inverse, en octets.

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

B.14.6 `UserInputIndication`

Cette indication est utilisée pour les messages d'entrée de l'utilisateur.

`alphanumeric` est une chaîne de caractères codés selon la Rec. UIT-T T.51 [30]. Elle pourrait être utilisée pour les entrées au clavier (équivalent: tonalités DTMF).

`userInputSupportIndication`: sert à indiquer au terminal distant quels types de chaînes `GENERALSTRING` sont acceptées par le terminal.

NOTE 1 – Il est prévu que la plupart des mises en œuvre de décodeurs PER ne pourront pas décoder d'autres chaînes que celles de l'alphabet IA5. Cette indication devrait être utilisée pour "prévenir" le terminal distant de ne pas essayer d'élaborer des algorithmes de codage en longueur variable sophistiqués.

Si l'indication `DTMF` est envoyée au moyen du protocole RTP et dans l'indication `UserInputIndication` sous forme alphanumérique, elle doit être codée dans la séquence `extendedAlphanumeric` et le fanion `rtppayloadIndication` doit être inclus.

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

Le paramètre booléen `basicString`, quand il est égal à la valeur "Vrai", indique que les caractères 0-9,* et # sont pris en charge.

Le paramètre booléen `ia5String`, quand il est mis à la valeur "Vrai", indique que le jeu complet des caractères `IA5String` est pris en charge.

Le paramètre booléen `generalString`, quand il est mis à la valeur "Vrai", indique que l'ensemble complet des caractères `GeneralString` est pris en charge.

L'opérateur booléen `encryptedBasicString`, quand il a la valeur "Vrai", indique une chaîne de base chiffrée.

L'opérateur booléen `encryptedIA5String`, quand il a la valeur "Vrai", indique une chaîne IA5 chiffrée.

L'opérateur booléen `encryptedGeneralString`, quand il a la valeur "Vrai", indique une chaîne générale chiffrée.

Le paragraphe 8.7/H.235 décrit les procédures pour le code DTMF chiffré H.245 ainsi que la façon de déployer les champs: `encryptedAlphanumeric` contenu dans `UserInputIndication` (= chaîne de base chiffrée), `encryptedSignalType` contenu dans `signal` (= chaîne IA5 chiffrée) et `encryptedAlphanumeric` contenu dans `extendedAlphanumeric` (= chaîne générale chiffrée).

Les indications **`signal`** et **`signalUpdate`** peuvent être utilisées lorsque l'on souhaite une commande précise de l'alignement du mode DTMF ou du signal de raccrochage avec les signaux audio dans la voie logique associée et lorsqu'il est nécessaire de commander la durée du mode DTMF ou d'en obtenir une indication.

L'indication **`signal`** désigne l'élément de signalisation qui doit être produit en cas d'envoi à une passerelle RTPC, qui a été détecté dans le flux audio en cas d'envoi émanant d'une passerelle RTPC ou qui doit être notifié entre d'autres combinaisons de extrémités. Lorsque l'indication **`signal`** est reçue par une passerelle vers le RTPC, celle-ci envoie l'élément de signalisation spécifié dans la voie du RTPC; en cas de réception par une passerelle vers un autre terminal de la série H, **`signal`** sera transformé en message dans le protocole du terminal connecté. Les passerelles produisent des messages **`signal`** (et **`signalUpdate`**) pour indiquer la détection des éléments de signalisation dans les

données audio reçues d'un point d'extrémité RTPC ou convertissent les messages correspondants d'autres protocoles.

Le paramètre **signalType** est mis à "!" (point d'exclamation) pour indiquer un signal de raccrochage ou à l'une des valeurs "0123456789*#ABCD" pour indiquer une tonalité DTMF.

NOTE 2 – Le signal de raccrochage est un état de raccrochage temporaire (d'une demi-seconde en général), souvent utilisé pour commander les fonctions de l'équipement de commutation. Il peut être impossible pour une passerelle de produire ou de détecter un signal de raccrochage en raison des caractéristiques de la voie du RTPC ou de la configuration locale (pour éviter une activation non souhaitée des fonctions de l'équipement associé). Par conséquent, la capacité de transmettre ou de recevoir des indications de signal de raccrochage est déclarée séparément dans la structure **UserInputCapability**.

Le paramètre **duration** indique la durée totale de la tonalité si elle est connue ou une première estimation de cette durée si la tonalité est en cours au moment de la transmission de l'indication **signal**. Si **duration** est omis, le récepteur utilisera une valeur par défaut appropriée en fonction des caractéristiques de la configuration locale et du réseau. Le paramètre **duration** sera ignoré si un signal de raccrochage ("!") est indiqué.

Le paramètre **signalUpdate** révisé l'estimation de la durée totale ou déclare la durée réelle mesurée de la tonalité détectée ou à générer. Il devrait être transmis de manière à arriver bien avant l'expiration de la durée estimée précédemment notifiée dans une indication **signal** ou **signalUpdate**; autrement, la durée révisée sera ignorée étant donné que le récepteur aura déjà mis fin à la tonalité. Il convient de noter qu'il n'est pas nécessaire d'envoyer une indication **signalUpdate** si la durée totale a été notifiée dans l'indication **signal**.

Le paramètre **rtp** contient les paramètres nécessaires pour aligner la tonalité ou le signal de raccrochage avec un flux RTP/UDP (H.323). Il ne doit être inclus dans la séquence **signalUpdate** que si plusieurs messages de signalisation ont été envoyés pour spécifier différents numéros LogicalChannelNumber et s'il est nécessaire d'indiquer quel signal doit être mis à jour.

Le paramètre **timestamp** spécifie, en termes d'horodatage selon le protocole RTP du codeur primaire sur la voie audio associée, le moment auquel la tonalité ou le signal de raccrochage devraient être émis (transmis ou introduits dans le flux de données audio). La tonalité ou le signal de raccrochage ne doivent pas être émis avant la lecture des données audio ayant le même horodatage; ils devraient être émis le plus rapidement possible après cet instant, mais pas plus tard que l'instant spécifié par **expirationTime**. L'émetteur d'une indication ne doit pas affecter au paramètre **timestamp** un instant qui se trouve "dans le futur"; **timestamp** est généralement mis à une valeur égale à celle qui correspond aux données audio dont l'envoi est en cours ou qui ont été le plus récemment envoyées sur la voie audio associée. Si le paramètre **timestamp** n'est pas spécifié, le signal doit être transmis ou introduit dès qu'il est reçu.

Le paramètre **expirationTime** spécifie, en termes d'horodatage selon le protocole RTP du codeur primaire sur la voie audio associée, le moment auquel la tonalité ou le signal de raccrochage seront considérés comme "caducs" et ignorés par le récepteur. Les extrémités qui reçoivent l'indication **signal** et ne sont pas en mesure de réagir dans le délai spécifié par l'horodateur **expirationTime** sur la voie associée doivent ignorer le message. Si un instant **expirationTime** n'est pas spécifié par l'émetteur, le message peut quand même être ignoré en raison de la configuration locale du destinataire.

Le paramètre **logicalChannelNumber** doit spécifier le numéro LogicalChannelNumber de la voie audio associée, contexte dans lequel les paramètres **timestamp** et **expirationTime** ont une signification.

A partir de l'indication reçue, une entité de commande multipoint doit convertir les horodateurs et le numéro de voie logique de manière appropriée pour chaque voie de sortie lorsqu'elle transmet l'indication à chaque point d'extrémité récepteur (les horodateurs peuvent changer si les données audio sont transcodées ou mélangées dans une entité MP). Une entité de commande multipoint

(MC) qui reçoit une indication après le délai **expirationTime** peut ignorer le message immédiatement sans le transmettre; dans le cas contraire, l'entité MC transmettra immédiatement toutes les demandes sans attendre le moment spécifié par **timestamp**.

Les extrémités doivent utiliser l'indication **alphanumeric** pour acheminer l'entrée d'utilisateur DTMF si les autres extrémités n'ont pas indiqué la capacité de recevoir des indications DTMF à l'aide de la structure **UserInputCapability**.

Un point d'extrémité doté de la capacité de recevoir des indications DTMF au moyen de l'indication **signal** doit également être en mesure d'accepter des indications **alphanumeric** pour qu'il y ait une compatibilité avec des terminaux plus anciens. Une indication **alphanumeric** peut être traitée comme une séquence d'une ou de plusieurs indications **signal**, les éléments **duration**, **timestamp** et **expirationTime** étant omis et les caractères non valides de **signalType** étant ignorés.

Si l'indication DTMF est envoyée au moyen du protocole RTP, conformément au § 10.5/H.323, et dans l'indication **UserInputIndication** sous forme de signal, le fanion **rtpPayloadIndication** doit être inclus.

Dans le cadre d'une utilisation normale, une passerelle qui détecte une indication DTMF dans le flux de données audio provenant d'une voie RTPC enverra l'indication **signal** dès qu'une tonalité est détectée, avec une estimation relativement élevée de l'élément **duration** et commencera à mesurer la durée de la tonalité. A la fin de la tonalité, l'indication **signalUpdate** est envoyée pour notifier la durée totale mesurée. Si la tonalité n'a pas cessé mais si la durée mesurée est proche de l'estimation précédente (de sorte que celle-ci peut être inférieure à la durée mesurée avant la réception d'une indication **signalUpdate**), une indication **signalUpdate** est envoyée pour accroître la valeur estimée. Le réalisateur est libre de déterminer la fréquence d'envoi de l'indication **signalUpdate**, l'estimation initiale de la durée communiquée dans l'indication **signal** ainsi que les augmentations des estimations ultérieures, mais des précautions doivent être prises pour ne pas surcharger le réseau avec un grand nombre de messages **signalUpdate** et éviter une expiration prématurée des estimations précédentes.

Dans des conditions normales d'utilisation d'un point d'extrémité qui n'est pas une passerelle, l'élément **signal** indiquera la durée totale de la tonalité qui doit être émise par la passerelle. Toutefois, dans certaines applications, il peut être souhaitable d'assurer à l'utilisateur une commande interactive en temps réel de la durée de la tonalité. Dans ce cas, les indications **signal** et **signalUpdate** seraient utilisées comme indiqué dans le paragraphe précédent pour les passerelles, l'indication **signal** étant envoyée dès que des données sont introduites par l'utilisateur (par exemple par l'activation d'une touche ou par une commande à l'écran) au moyen d'une durée estimée de la tonalité et l'indication **signalUpdate** étant utilisée pour envoyer des mises à jour tant que les entrées de l'utilisateur sont en cours et pour indiquer la durée totale lorsque ces entrées sont désactivées.

B.14.7 ConferenceIndication

Le paramètre **sbeNumber** sera défini comme le numéro SBE selon la Rec. UIT-T H.230.

Le paramètre **terminalNumberAssign** sera défini comme l'indication TIA selon la Rec. UIT-T H.230.

Le paramètre **terminalJoinedConference** sera défini comme l'indication TIN selon la Rec. UIT-T H.230.

Le paramètre **terminalLeftConference** sera défini comme l'indication TID selon la Rec. UIT-T H.230.

Le paramètre **seenByAtLeastOneOther** sera défini comme l'indication MIV selon la Rec. UIT T H.230.

Le paramètre **cancelSeenByAtLeastOneOther** sera défini comme l'indication cancel-MIV selon la Rec. UIT-T H.230.

Le paramètre `seenByAll` sera défini comme l'indication MIV selon la Rec. UIT-T H.230.

Le paramètre `cancelSeenByAll` sera défini comme l'indication MIV selon la Rec. UIT-T H.230.

Le paramètre `terminalYouAreSeeing` sera défini comme l'indication VIN selon la Rec. UIT-T H.230.

Le paramètre `requestForFloor` sera défini comme l'indication TIF selon la Rec. UIT-T H.230 et envoyé par un terminal à l'entité de commande multipoint.

Le paramètre `WithdrawChairToken` sera défini comme l'indication CCR selon la Rec. UIT-T H.230 et envoyé par l'entité de commande multipoint au détenteur du jeton de rôle de président.

Le paramètre `FloorRequested` sera défini comme l'indication TIF selon la Rec. UIT-T H.230 lorsqu'il sera envoyé par l'entité de commande multipoint au détenteur du jeton de rôle de président. Cette demande comprend l'élément `TerminalLabel` du terminal demandeur.

`terminalYouAreSeeingInSubPictureNumber` sera défini comme VIN2 selon la Rec. UIT-T H.230. `subPictureNumber` est défini comme N selon les Figures 2-4/H.243.

`videoIndicateCompose` sera défini comme VIC selon la Rec. UIT-T H.230. `compositionNumber` est défini comme M selon le Tableau 4/H.243.

B.14.8 H2250MaximumSkewIndication

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

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

B.14.9 MCLocationIndication

Cette indication est envoyée par l'entité de commande multipoint pour indiquer aux autres terminaux l'adresse de signalisation qui doit être utilisée pour atteindre l'entité de commande multipoint.

B.14.10 VendorIdentificationIndication

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

B.14.11 FunctionNotSupported

Cette indication est utilisée pour renvoyer à l'émetteur les demandes, les réponses et les commandes qui n'ont pas été comprises.

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

Si un terminal reçoit une demande, une réponse ou une commande qu'il ne comprend pas, soit parce qu'il s'agit d'un message non normalisé ou parce que ce message a été défini dans une révision ultérieure de la présente Recommandation, il devra répondre en envoyant le message `FunctionNotSupported`.

Si un terminal reçoit une demande, une réponse ou une commande ayant un codage incorrect, il mettra la valeur `syntaxError` dans le paramètre `cause`. Si le codage est correct, mais si les valeurs codées sont sémantiquement incorrectes, le terminal mettra la valeur `semanticError` dans le paramètre `cause`. Si le message est une extension non reconnue à l'un des messages

MultimediaSystemControlMessage, RequestMessage, ResponseMessage ou CommandMessage, le terminal devra mettre la valeur unknownFunction dans le paramètre cause.

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

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

B.14.12 FlowControlIndication

Cette indication est utilisée pour signaler au terminal distant que le terminal a ajusté son débit maximal sortant, soit en réponse à une commande FlowControlCommand entrante soit parce que le terminal souhaite ajuster son débit sortant. Elle permet à un terminal de signaler toute modification du débit maximal sortant, dans les limites fixées au moment de l'ouverture de voie logique et par les capacités du terminal.

Tout terminal recevant une commande FlowControlCommand doit répondre par une indication FlowControlIndication pour indiquer le nouveau débit maximal qu'il a fixé.

Les champs de l'indication FlowControlIndication ont la même signification que les champs du même nom contenus dans la commande FlowControlCommand.

B.14.13 MobileMultilinkReconfigurationIndication

Cette indication est utilisée pour signaler au récepteur que l'émetteur modifiera la valeur de la taille de l'échantillon et/ou la valeur des échantillons par trame de l'en-tête de la trame d'information, comme indiqué dans l'Annexe H/H.324. Cette indication doit être émise pendant le mode en-tête complet mais elle ne doit pas être émise pendant le mode en-tête comprimé.

Le paramètre sampleSize indique la taille d'un échantillon en octets. Un échantillon correspond au nombre d'octets qui seront répartis entre les voies physiques disponibles.

Le paramètre samplesPerFrame indique la longueur de la charge utile multiliason en échantillons.

B.15 Messages génériques

Le type **GenericMessage** permet de spécifier de nouveaux éléments RequestMessage, CommandMessage, ResponseMessage et IndicationMessage de telle manière qu'une nouvelle version de syntaxe H.245 n'ait pas besoin d'être publiée. Cette méthode permet de définir des messages aussi bien normalisés que non normalisés.

NOTE 1 – Les structures de type GenericMessage définies dans la présente Recommandation devraient être énumérées dans les annexes de la présente Recommandation. Les structures de type GenericMessage définies dans d'autres Recommandations de l'UIT-T devraient faire l'objet de références dans un appendice à la présente Recommandation. Les structures de type GenericMessage définies en dehors de l'UIT-T pourront être publiées sous toute forme appropriée.

Le champ **messageIdentifiant** indique le type de message unique. Les identificateurs de message fondés sur l'UIT-T doivent utiliser la structure normalisée OBJECT IDENTIFIER, alors que d'autres identificateurs de message, normalisés et non normalisés, doivent utiliser, selon le cas, une des structures suivantes: "standard", "h221NonStandard", "uuid" ou "domainBased".

Le champ facultatif **subMessageIdentifiant** indique un sous-message associé à l'identificateur de message, messageIdentifiant.

Le champ **messageContents** indique les paramètres du message.

Afin d'éviter tous problèmes d'ambiguïté et d'interopérabilité, un identificateur de paramètre normalisé, **standard.ParameterIdentifier**, ayant la valeur 0, ne devrait pas être défini pour utilisation dans le champ messageContents.

NOTE 2 – Certaines Recommandations définissent des procédures automatiques afin de convertir les paramètres génériques, GenericParameters, du système de signalisation H.245 au système de signalisation par codec de signal BAS, utilisé dans la Rec. UIT-T H.320. Ces procédures utilisent la valeur 0 au lieu d'un identificateur parameterIdentifier normalisé, en tant que signal spécial balisant la fin d'une liste d'éléments de type GenericParameter.

Le champ genericRequest est un message générique GenericMessage servant à envoyer un message générique, RequestMessage.

Le champ genericResponse est un message générique GenericMessage servant à envoyer un message générique, ResponseMessage.

Le champ genericCommand est un message générique GenericMessage servant à envoyer un message générique, CommandMessage.

Le champ genericIndication est un message générique GenericMessage servant à envoyer un message générique, IndicationMessage.

Annexe C

Procédures

C.1 Introduction

La présente annexe définit les procédures génériques de commande de système multimédia utilisant les messages définis dans la présente Recommandation. Les Recommandations utilisant la présente Recommandation indiqueront quelles sont les procédures applicables et définiront d'éventuelles spécifications particulières.

La présente annexe décrit les procédures permettant d'exécuter les fonctions suivantes:

- désignation maître-esclave;
- échange de capacités de terminal;
- signalisation de voie logique unidirectionnelle;
- signalisation de voie logique bidirectionnelle;
- demande de fermeture de voie logique par un terminal récepteur;
- modification d'entrée de tableau de multiplexage H.223;
- demande d'entrée de multiplexage;
- demande de mode d'émission de récepteur à émetteur;
- détermination du temps de propagation aller et retour;
- boucle de maintenance.

C.1.1 Méthode de spécification

Les procédures sont généralement spécifiées dans le présent paragraphe au moyen de diagrammes SDL qui constituent une spécification graphique des procédures et incluent la spécification des mesures à prendre en cas d'anomalie.

C.1.2 Communication entre l'entité de protocole et l'utilisateur de protocole

L'interaction avec l'utilisateur d'une fonction particulière est spécifiée en termes de primitives transférées à l'interface entre l'entité de protocole et l'utilisateur du protocole. Les primitives ont pour objectif de définir des procédures de protocole et non pas de spécifier ou de rendre obligatoire une mise en œuvre particulière. Il peut y avoir un certain nombre de paramètres associés à chaque primitive.

Pour rendre la spécification plus claire, on définit des états de protocole. 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 par un diagramme de changement d'état. La séquence autorisée apporte des contraintes aux actions de l'utilisateur et définit les réponses possibles de l'entité de protocole.

Un paramètre de primitive décrit comme néant équivaut à l'absence de ce paramètre.

C.1.3 Communication entre entités homologues

Les informations de protocole sont transférées vers l'entité de protocole homologue par l'intermédiaire des messages appropriés définis dans l'Annexe A. Certaines entités de protocole décrites ont des variables d'état qui leur sont associées. Un certain nombre d'entités de protocole décrites comportent également des temporisateurs qui leur sont associés.

Un temporisateur est décrit par la notation T_n , n étant un numéro. Dans les diagrammes SDL, l'armement d'un temporisateur signifie que le temporisateur est chargé à une valeur spécifiée et la temporisation démarre. La suspension de temporisation signifie que le temporisateur est arrêté et que sa valeur au moment de la suspension est retenue. La fin de temporisation signifie qu'un temporisateur a fonctionné pendant la durée spécifiée et qu'il a atteint la valeur zéro.

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

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

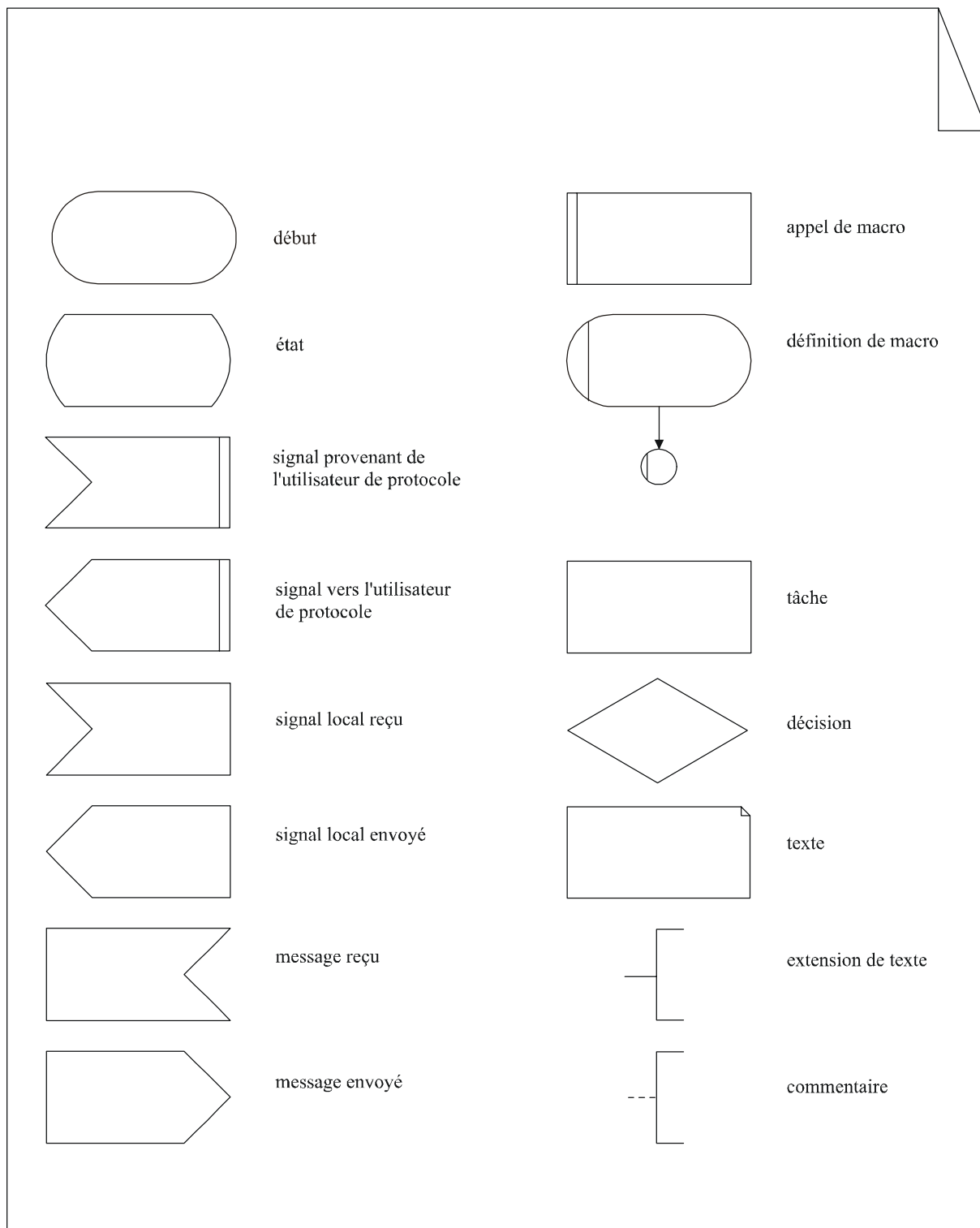
Certaines entités de protocole définissent une primitive d'erreur pour indiquer les états d'erreur de protocole à une entité de gestion.

C.1.4 Diagrammes SDL

Les diagrammes SDL indiquent les actions à prendre dans les interactions autorisées avec l'utilisateur du protocole, de même qu'à la réception de messages provenant de l'entité de protocole homologue. Les primitives qui ne sont pas autorisées pour un état donné, tel que spécifié par les diagrammes de changement d'états ne sont pas indiquées dans les diagrammes SDL. Les réponses à la réception de messages inappropriés sont cependant décrites dans les diagrammes SDL.

C.1.5 Symboles des diagrammes SDL

Les symboles des diagrammes SDL sont décrits dans la Figure C.1.



H.245_FC.1

Figure C.1/H.245 – Symboles des diagrammes SDL

C.2 Procédures de désignation maître-esclave

C.2.1 Introduction

Des conflits peuvent survenir quand au moins deux terminaux mis en communication déclenchent des événements similaires simultanément, pour lesquels des ressources sont disponibles pour une seule manifestation de l'événement, comme l'ouverture de voies logiques. Pour résoudre de tels conflits, un des terminaux peut jouer le rôle de terminal maître et l'autre ou les autres terminaux

peuvent chacun jouer le rôle de terminal asservi. Les procédures décrites ci-dessous permettent aux terminaux en communication de désigner le terminal maître et le ou les terminaux asservis.

Le protocole décrit ici est désigné comme l'entité de signalisation pour la désignation maître-esclave (MSDSE, *master-slave determination signalling entity*). Il y a une instance d'entité MSDSE dans chaque terminal mis en communication.

L'un des terminaux peut déclencher la procédure de désignation maître-esclave en présentant 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. Alors que la primitive d'indication DETERMINE indique le résultat, elle n'indique pas que le résultat est connu du terminal distant. La primitive de confirmation DETERMINE indique le résultat et confirme que ce résultat est également connu du terminal distant. Un terminal ne peut lancer la procédure de désignation maître-esclave que si aucune procédure dépendant de son résultat n'est activée localement.

Un terminal devra réagir aux procédures reposant sur la connaissance du résultat et déclenchées par le terminal distant à tout moment après que le résultat de la désignation soit connu du terminal local. Ceci peut avoir lieu avant que le terminal local ait reçu confirmation que le terminal distant a également connaissance du résultat. Un terminal ne devra pas déclencher des procédures qui dépendent de la connaissance du résultat avant d'avoir reçu confirmation que le terminal distant a également connaissance du résultat de l'instance en cours de la procédure de désignation.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, c'est la spécification formelle qui sera applicable.

C.2.1.1 Aperçu général du protocole – Déclenchement par l'utilisateur local

Une procédure de désignation maître-esclave est déclenchée quand la primitive de demande DETERMINE est présentée par l'utilisateur de l'entité MSDSE. Un message MasterSlaveDetermination est envoyé à l'entité MSDSE homologue et le temporisateur T106 démarre. Si un message MasterSlaveDeterminationAck est reçu en réponse au message MasterSlaveDetermination, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive de confirmation DETERMINE que la procédure de désignation maître-esclave a réussi et un message MasterSlaveDeterminationAck est envoyé à l'entité MSDSE homologue. Si, cependant, un message MasterSlaveDeterminationReject est reçu en réponse au message MasterSlaveDetermination, alors un nouveau numéro de désignation maître-esclave est généré, la temporisation T106 redémarre et un autre message MasterSlaveDetermination est envoyé. Si après l'envoi d'un message MasterSlaveDetermination un nombre de fois égal à N100 fois, un message MasterSlaveDeterminationAck n'a toujours pas été reçu, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que la procédure de désignation maître-esclave n'a pas donné de résultat.

Si le temporisateur T106 vient à expiration, alors l'utilisateur de l'entité MSDSE est informé par la primitive d'indication REJECT et un message MasterSlaveDeterminationRelease est envoyé à l'entité MSDSE homologue.

C.2.1.2 Aperçu général du protocole – Déclenchement par l'utilisateur distant

Quand un message MasterSlaveDetermination est reçu par l'entité MSDSE, une procédure de désignation maître-esclave est déclenchée. Si cette procédure renvoie un résultat déterminé, alors l'utilisateur est informé du résultat de la désignation maître-esclave par la primitive d'indication DETERMINE; un message MasterSlaveDeterminationAck est envoyé à l'entité MSDSE homologue et le temporisateur T106 démarre. Si un message MasterSlaveDeterminationAck est reçu en réponse au message MasterSlaveDeterminationAck, alors le temporisateur T106 est arrêté et l'utilisateur est informé par la primitive de confirmation DETERMINE que la procédure de désignation maître-esclave a réussi.

Si le temporisateur T106 vient à expiration, l'utilisateur MSDSE est informé par la primitive d'indication REJECT.

Si, cependant, la procédure de désignation maître-esclave renvoie un résultat indéterminé, alors le message MasterSlaveDeterminationReject est envoyé à l'entité MSDSE homologue.

C.2.1.3 Aperçu général du protocole – Déclenchement simultané

Quand un message MasterSlaveDetermination est reçu par l'entité MSDSE qui a elle-même déclenché une procédure de désignation maître-esclave et attend un message d'acquiescement MasterSlaveDeterminationAck ou de refus MasterSlaveDeterminationReject, alors une procédure de désignation maître-esclave est déclenchée. Si cette procédure renvoie un résultat déterminé, l'entité MSDSE répond comme si la procédure avait été déclenchée par l'utilisateur distant et les procédures décrites ci-dessus sont applicables.

Si, cependant, la procédure de désignation maître-esclave renvoie un résultat indéterminé, alors un nouveau numéro de désignation maître-esclave est généré et l'entité MSDSE répond comme si la procédure avait de nouveau été déclenchée par l'utilisateur de l'entité MSDSE locale comme cela est décrit ci-dessus.

C.2.1.4 Procédure de désignation maître-esclave

La procédure suivante est utilisée pour déterminer quel terminal est le terminal maître à partir des valeurs des paramètres terminalType et statusDeterminationNumber. En premier lieu, les valeurs terminalType sont comparées et le terminal ayant le numéro de type de terminal le plus grand est désigné comme étant le terminal maître. Si les numéros de type de terminal sont les mêmes, les numéros statusDeterminationNumber sont comparés en utilisant l'arithmétique modulo pour désigner le terminal maître.

Si les deux terminaux ont des valeurs de champ terminalType égales et si la différence entre les valeurs de champ statusDeterminationNumber modulo 2^{24} est 0 ou 2^{23} , un résultat indéterminé est obtenu.

C.2.2 Communication entre l'entité MSDSE et l'utilisateur de l'entité MSDSE

C.2.2.1 Primitives entre l'entité MSDSE et l'utilisateur de l'entité MSDSE

Les communications entre l'entité MSDSE et l'utilisateur de l'entité MSDSE sont effectuées en utilisant les primitives indiquées dans le Tableau C.1.

Tableau C.1/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 aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

C.2.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) la primitive DETERMINE est utilisée pour lancer la procédure de désignation maître-esclave et pour renvoyer le résultat obtenu à partir de cette même procédure.

La primitive de demande DETERMINE est utilisée pour lancer la procédure de désignation maître-esclave.

La primitive d'indication DETERMINE est utilisée pour indiquer le résultat de la procédure de désignation maître-esclave. Etant donné que le résultat de la procédure peut ne pas être connu du terminal distant, le terminal ne devra pas lancer des procédures qui dépendent de la connaissance du résultat, bien qu'il doive réagir à toutes les procédures dépendant de la connaissance du résultat.

La primitive de confirmation DETERMINE est utilisée pour indiquer le résultat de la procédure de désignation maître-esclave et pour indiquer que ce résultat est connu des deux terminaux. Le terminal peut déclencher des procédures qui dépendent de la connaissance du résultat et doit réagir à de telles procédures;

- b) la primitive REJECT indique que la procédure de désignation maître-esclave a échoué;
- c) la primitive ERROR signale les erreurs de l'entité MSDSE à une entité de gestion.

C.2.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.1 est la suivante:

- a) le paramètre TYPE indique le statut du terminal. Il a la valeur de MASTER (maître) ou SLAVE (esclave);
- b) la valeur ERRCODE indique le type d'erreur de l'entité MSDSE. Le Tableau C.5 indique les valeurs que le paramètre ERRCODE peut prendre.

C.2.2.4 Etats d'entité MSDSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité MSDSE et l'utilisateur de l'entité MSDSE.

Etat 0: IDLE (repos)

Aucune procédure de désignation maître-esclave n'a été déclenchée.

Etat 1: OUTGOING AWAITING RESPONSE (entité sortante en attente de réponse)

L'utilisateur de l'entité MSDSE locale a demandé une procédure de désignation maître-esclave. Une réponse est attendue de l'entité MSDSE distante.

Etat 2: INCOMING AWAITING RESPONSE (entité entrante en attente de réponse)

L'entité MSDSE distante a lancé la procédure de désignation maître-esclave dans l'entité MSDSE locale. Un acquittement a été envoyé à l'entité MSDSE distante et une réponse est attendue en provenance de l'entité MSDSE distante.

C.2.2.5 Diagramme de changement d'état

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 C.2.

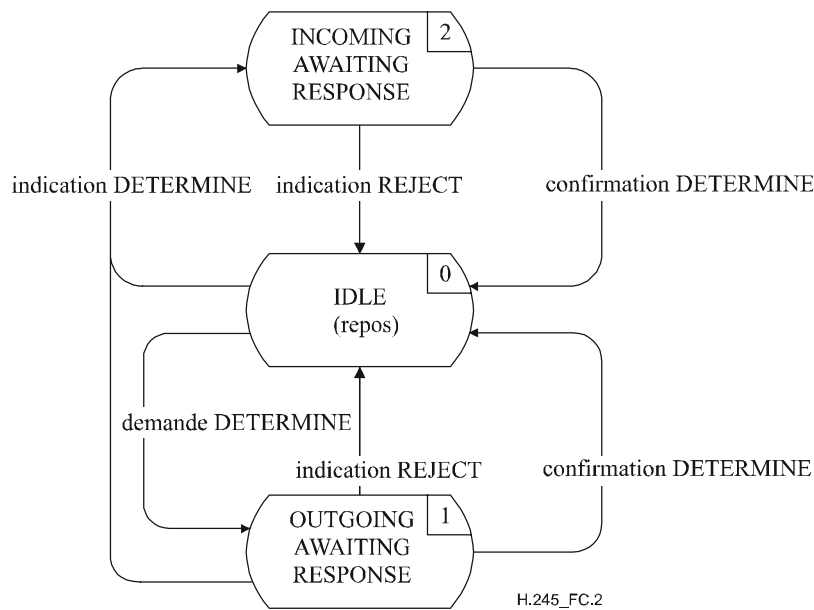


Figure C.2/H.245 – Diagramme de changement d'état pour la séquence de primitives dans l'entité MSDSE

C.2.3 Communication entre les entités MSDSE homologues

C.2.3.1 Messages d'entité MSDSE

Le Tableau C.2 indique les messages et les champs des entités MSDSE, définis dans l'Annexe A, qui se rapportent au protocole spécifique aux entités MSDSE.

Tableau C.2/H.245 – Noms et champs des messages des entités MSDSE

Fonction	Message	Champ
détermination	MasterSlaveDetermination	terminalType statusDeterminationNumber
	MasterSlaveDeterminationAck	decision
	MasterSlaveDeterminationReject	cause
reprise sur erreur	MasterSlaveDeterminationRelease	–

C.2.3.2 Variables d'état d'une entité MSDSE

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

sv_TT

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

sv_SDNUM

Cette variable d'état contient le numéro de désignation maître-esclave pour ce terminal.

sv_STATUS

Cette variable d'état est utilisée pour enregistrer le résultat de la dernière procédure de désignation maître-esclave. Elle a les valeurs de "master", "slave" et "indeterminate".

sv_NCOUNT

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

C.2.3.3 Temporisateurs d'entité MSDSE

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

T106

Ce temporisateur est utilisé pendant l'état OUTGOING AWAITING RESPONSE et pendant l'état INCOMING AWAITING RESPONSE. Il spécifie le temps maximal autorisé pendant lequel aucun message d'acquittement ne peut être reçu.

C.2.3.4 Compteurs d'entité MSDSE

Le paramètre suivant est spécifié pour l'entité MSDSE:

N100

Ce paramètre spécifie la valeur maximale de sv_NCOUNT.

C.2.4 Procédures d'entité MSDSE

C.2.4.1 Introduction

La Figure C.3 récapitule les primitives d'entité MSDSE et leurs paramètres, ainsi que les messages.

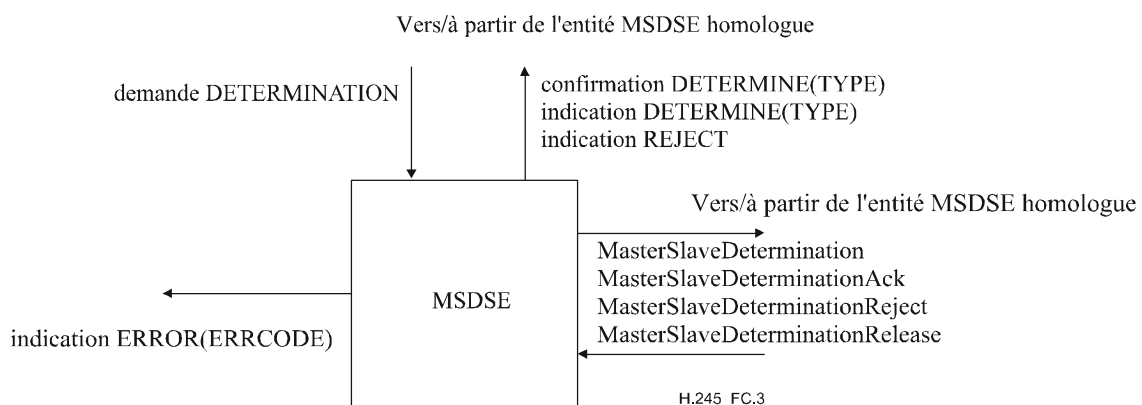


Figure C.3/H.245 – Primitives et messages dans l'entité MSDSE

C.2.4.2 Valeurs par défaut des paramètres des primitives

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent les valeurs indiquées dans le Tableau C.3.

Tableau C.3/H.245 – Valeurs par défaut des paramètres des primitives

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

C.2.4.3 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement indiqué dans les messages SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.4.

Tableau C.4/H.245 – Valeurs par défaut des champs des messages

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

C.2.4.4 Valeurs du paramètre ERRCODE

Le Tableau C.5 indique les valeurs que le paramètre ERRCODE de la primitive d'indication ERROR peut prendre pour l'entité MSDSE.

Tableau C.5/H.245 – Valeurs du paramètre ERRCODE dans l'entité MSDSE

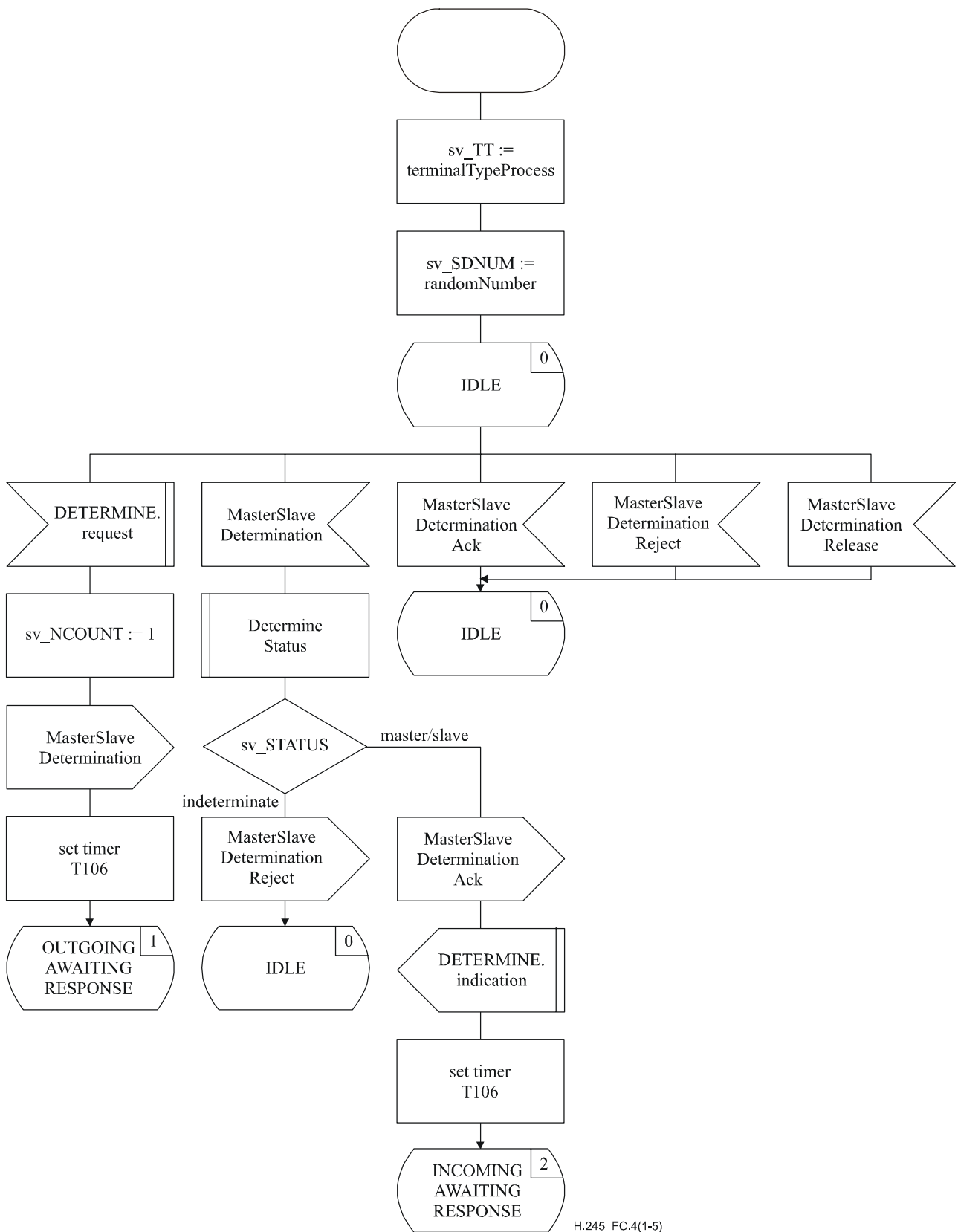
Type d'erreur	Code d'erreur	Condition d'erreur	Etat
Pas de réponse de l'entité MSDSE distante	A	Fin de temporisation locale T106	OUTGOING AWAITING RESPONSE INCOMING AWAITING RESPONSE
Le terminal distant ne voit pas la réponse de l'entité MSDSE locale	B	Fin de temporisation distante T106	OUTGOING AWAITING RESPONSE INCOMING AWAITING RESPONSE
Message inapproprié	C	MasterSlaveDetermination	INCOMING AWAITING RESPONSE
	D	MasterSlaveDeterminationReject	INCOMING AWAITING RESPONSE
Valeur de champ incohérente	E	MasterSlaveDeterminationAck. decision != sv_STATUS	INCOMING AWAITING RESPONSE
Nombre maximal de tentatives	F	sv_NCOUNT == N100	OUTGOING AWAITING RESPONSE

C.2.4.5 Diagrammes SDL

Les procédures de l'entité MSDSE sont décrites sous forme de diagrammes SDL dans la Figure C.4.

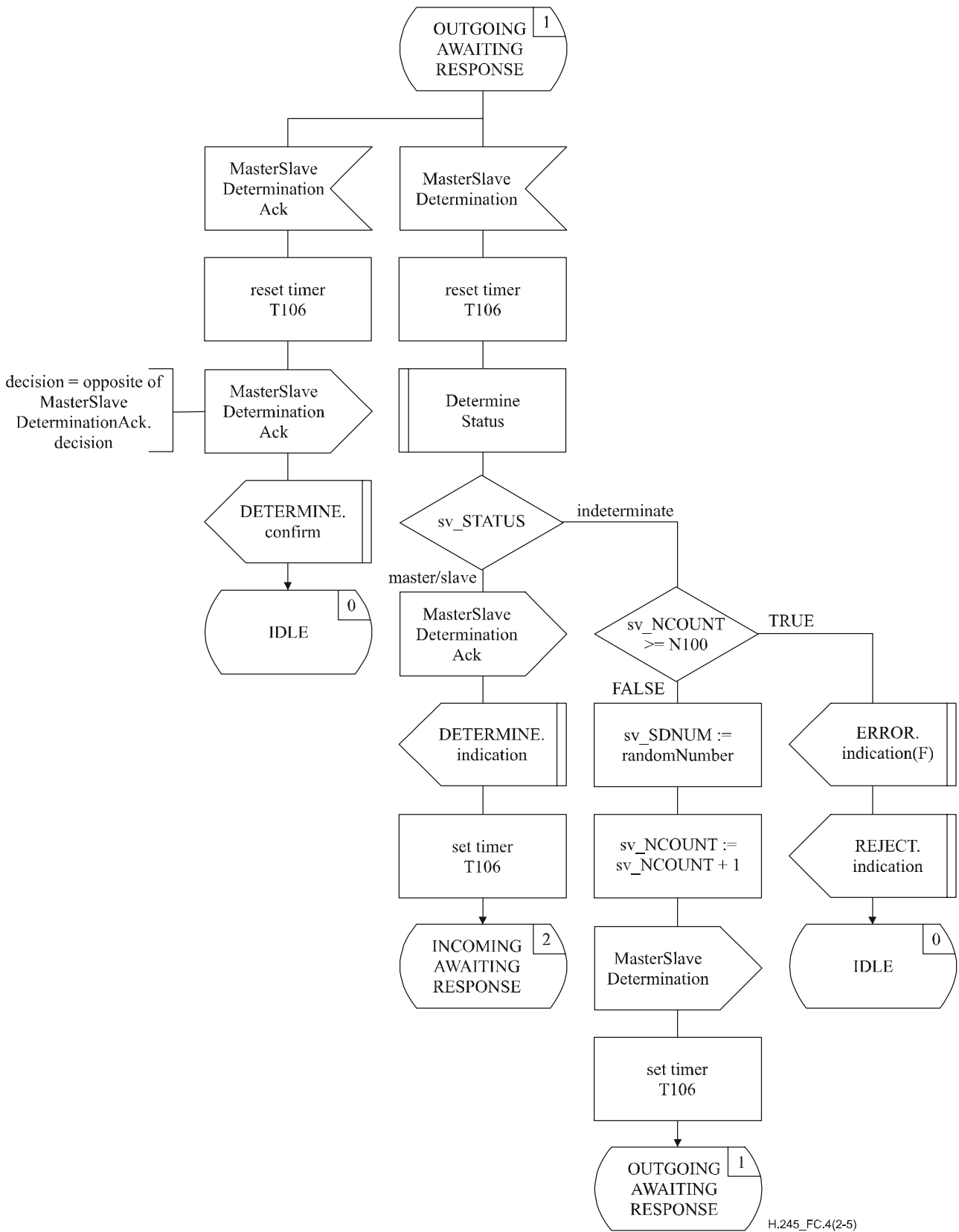
La structure terminalTypeProcess est une procédure qui renvoie un nombre identifiant différents types de terminaux tels que les terminaux, les ponts de conférence et les passerelles.

La structure randomNumber est une procédure qui renvoie un nombre aléatoire dans la gamme $0 \dots 2^{24} - 1$.



H.245_FC.4(1-5)

Figure C.4/H.245 – Diagramme SDL de l'entité MSDSE (feuille 1 de 5)



H.245_FC.4(2-5)

Figure C.4/H.245 – Diagramme SDL de l'entité MSDSE (feuille 2 de 5)

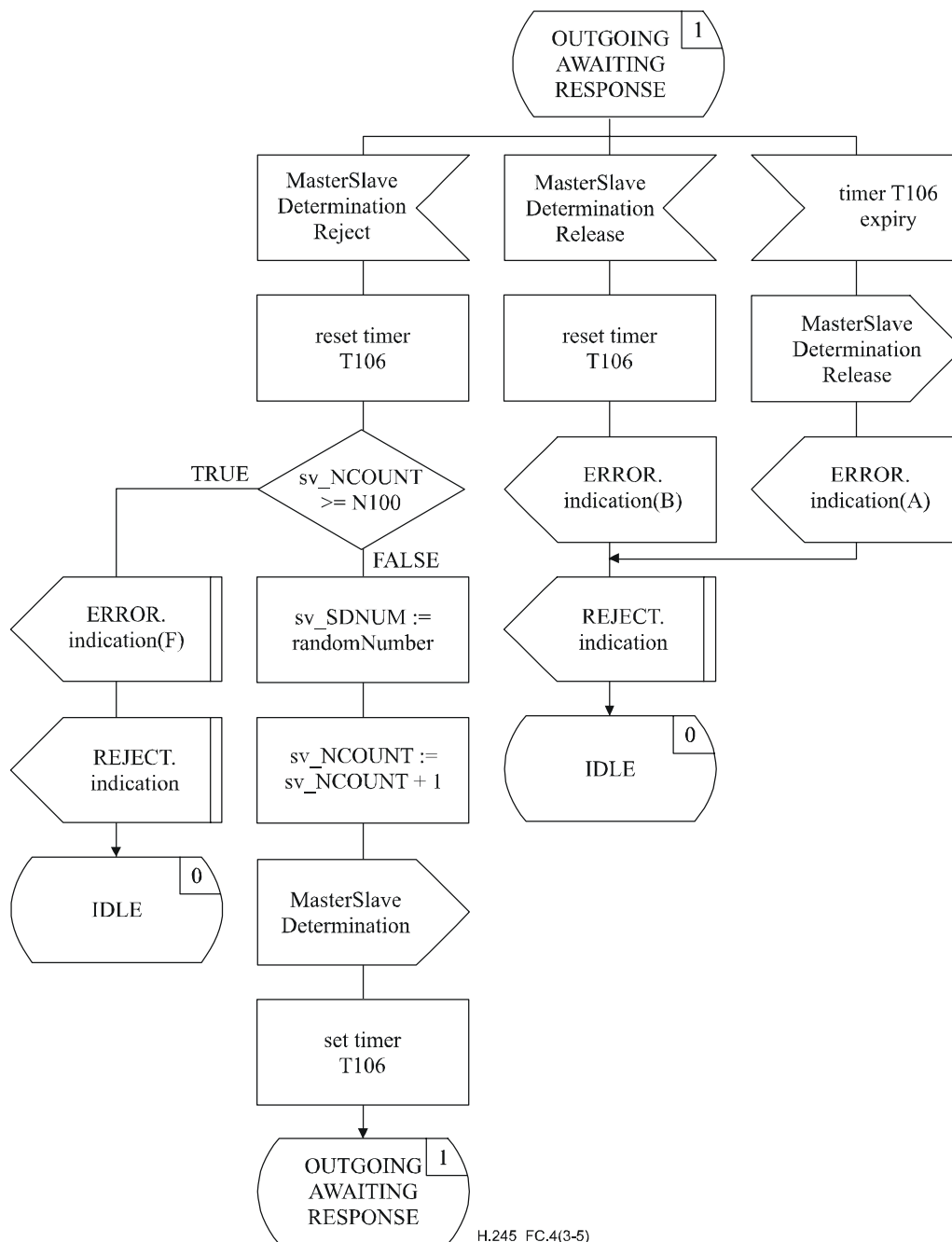


Figure C.4/H.245 – Diagramme SDL de l'entité MSDSE (feuille 3 de 5)

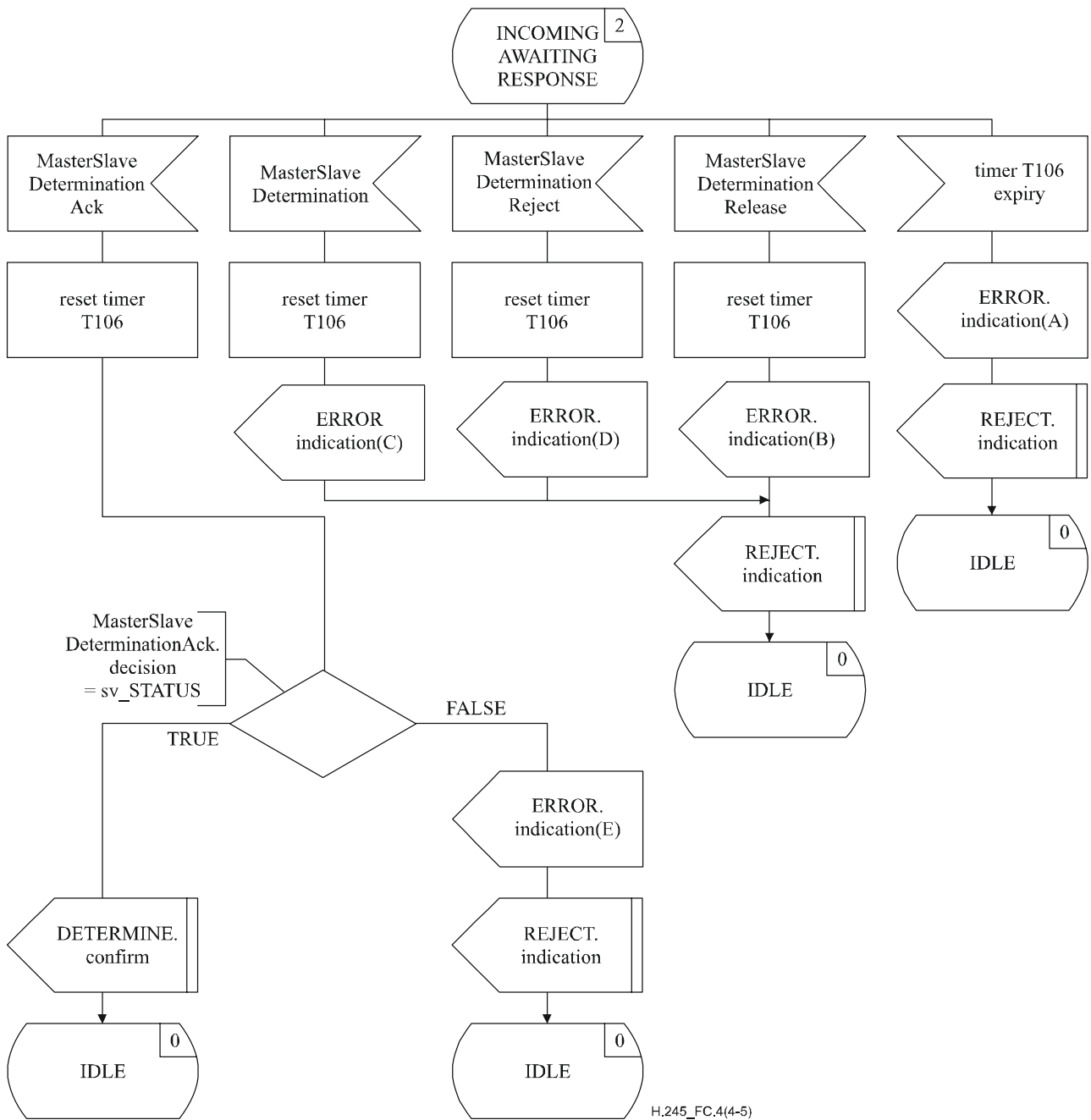
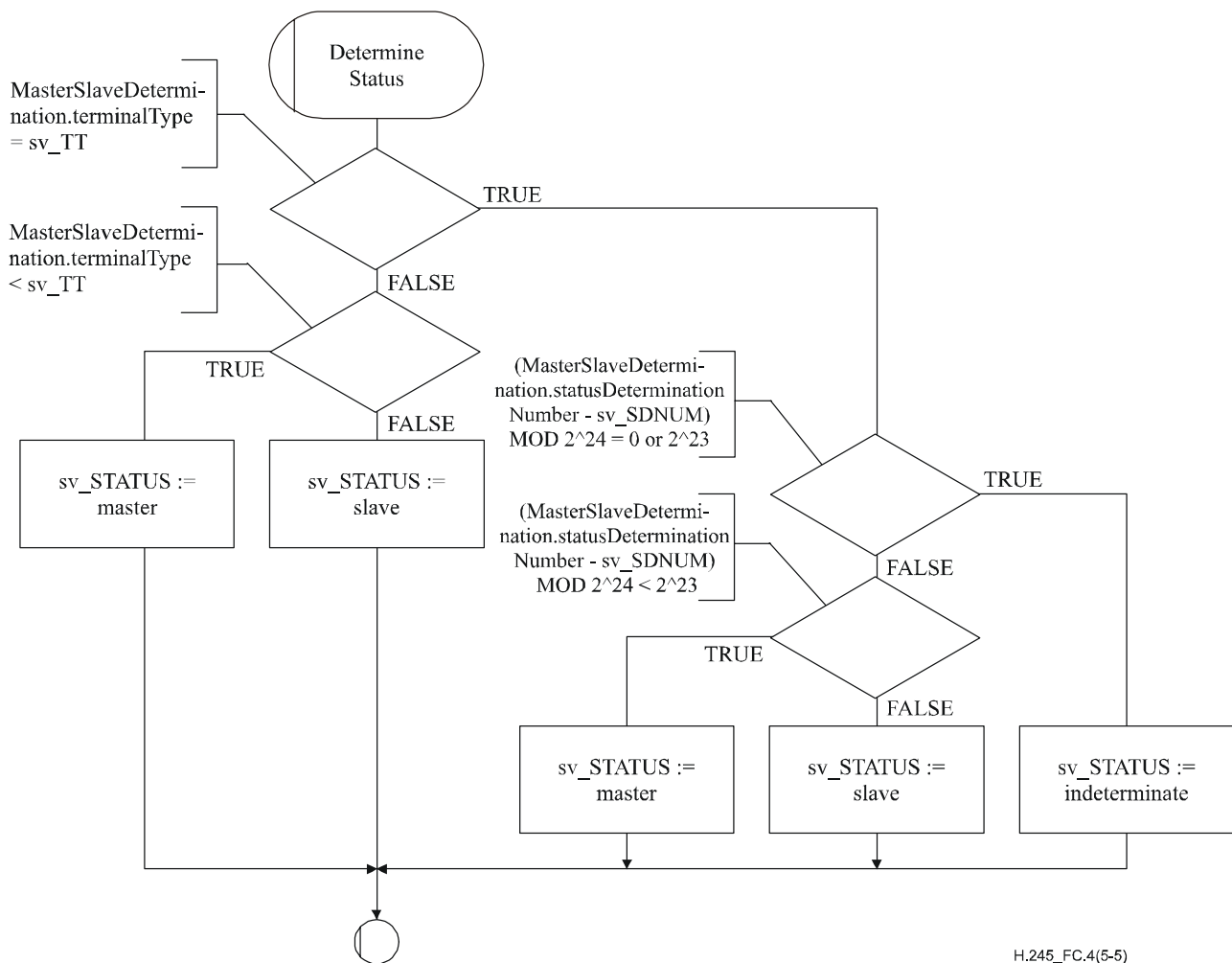


Figure C.4/H.245 – Diagramme SDL de l'entité MSDSE (feuille 4 de 5)



H.245_FC.4(5-5)

Figure C.4/H.245 – Diagramme SDL de l'entité MSDSE (feuille 5 de 5)

C.3 Procédures d'échange de capacités

C.3.1 Introduction

Ces procédures sont utilisées par les terminaux pour faire connaître leurs capacités et sont appelées "entité de signalisation d'échange de capacités" (CESE, *capability exchange signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité CESE et l'utilisateur de l'entité CESE. Les informations de protocole sont transférées vers l'entité CESE homologue par l'intermédiaire des messages appropriés définis dans l'Annexe A. Il y a une entité CESE sortante et une entité CESE entrante. A chacune des extrémités entrante et sortante, il y a une instance d'entité CESE pour chaque appel.

Tous les terminaux destinés à être utilisés dans des applications point à point ou ceux connectés à un pont de conférence devront pouvoir identifier un message TerminalCapabilitySet et sa structure ainsi que les valeurs de capacité qui sont obligatoires pour ces applications; toutes les valeurs de capacité non reconnues seront ignorées et il n'en découlera aucune défaillance.

L'échange de capacités peut être effectué à n'importe quel moment. Il peut signaler les capacités changées comme les capacités inchangées. Les capacités inchangées ne devraient pas être envoyées de façon répétitive sans raison majeure.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

C.3.1.1 Aperçu général du protocole – Entité CESE sortante

Un échange de capacités est déclenché quand la primitive de demande TRANSFER est présentée par l'utilisateur dans l'entité CESE sortante. Un message TerminalCapabilitySet est envoyé à l'entité CESE entrante homologue et le temporisateur T101 démarre. Si un message TerminalCapabilitySetAck est reçu en réponse au message TerminalCapabilitySet, alors le temporisateur T101 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER que l'échange de capacités a abouti. Si, cependant, un message TerminalCapabilitySetReject est reçu en réponse au message TerminalCapabilitySet, alors le temporisateur T101 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité CESE homologue a refusé l'échange de capacités.

Si la temporisation T101 vient à expiration, alors l'utilisateur de l'entité CESE sortante est informé par la primitive d'indication REJECT et un message TerminalCapabilitySetRelease est envoyé.

C.3.1.2 Aperçu général du protocole – Entité CESE entrante

Quand un message TerminalCapabilitySet est reçu par l'entité CESE entrante, l'utilisateur est informé de la demande d'échange de capacités par la primitive d'indication TRANSFER. L'utilisateur de l'entité CESE entrante signale l'acceptation de la demande d'échange de capacités en présentant la primitive de réponse TRANSFER et un message TerminalCapabilitySetAck est envoyé à l'entité CESE sortante homologue. L'utilisateur de l'entité CESE entrante signale le refus de la demande d'échange de capacités en présentant la primitive de demande REJECT et un message TerminalCapabilitySetReject est envoyé à l'entité CESE sortante homologue.

C.3.2 Communication entre l'entité CESE et l'utilisateur de l'entité CESE

C.3.2.1 Primitives entre l'entité CESE et l'utilisateur de l'entité CESE

Les communications entre l'entité CESE et l'utilisateur de l'entité CESE s'effectuent en utilisant les primitives indiquées dans le Tableau C.6.

Tableau C.6/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	PROTOID MUXCAP CAPTABLE CAPDESCRIPTORS	– (Note 1)	–
REJECT	CAUSE	SOURCE CAUSE	Non défini (Note 2)	Non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

C.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.

C.3.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués au Tableau C.6 est la suivante:

- a) le paramètre PROTOID est le paramètre d'identification du protocole. Il est mappé au champ protocolIdentifier du message TerminalCapabilitySet et transmis de façon transparente à l'utilisateur de l'entité CESE homologue. Ce paramètre est obligatoire;
- b) le paramètre MUXCAP est le paramètre des capacités du multiplex. Il est mappé au champ multiplexCapability du message TerminalCapabilitySet et transmis de façon transparente à l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- c) le paramètre CAPTABLE est le paramètre du tableau de capacités. Il peut y avoir une ou plusieurs entrées de tableau de capacités décrites à l'aide de ce paramètre. Ce paramètre est mappé au champ capabilityTable du message TerminalCapabilitySet et transmis de façon transparente à l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- d) 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 mappé au champ CapabilityDescriptors du message TerminalCapabilitySet et transmis de façon transparente à l'utilisateur de l'entité CESE homologue. Ce paramètre est facultatif;
- e) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur "USER" ou "PROTOCOL". Ce dernier cas peut provenir d'une fin de temporisation;
- f) le paramètre CAUSE indique le motif du refus du paramètre CAPTABLE ou CAPDESCRIPTORS. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

C.3.2.4 Etats d'entité CESE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité CESE et l'utilisateur de l'entité CESE.

Les états correspondant à une entité CESE sortante sont:

Etat 0: IDLE (repos)

L'entité CESE est au repos.

Etat 1: AWAITING RESPONSE (attente de la réponse)

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

Les états correspondant à une entité CESE entrante sont:

Etat 0: IDLE

L'entité CESE est au repos.

Etat 1: AWAITING RESPONSE (attente de la réponse)

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

C.3.2.5 Diagramme de changement d'état

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 C.5 et C.6.

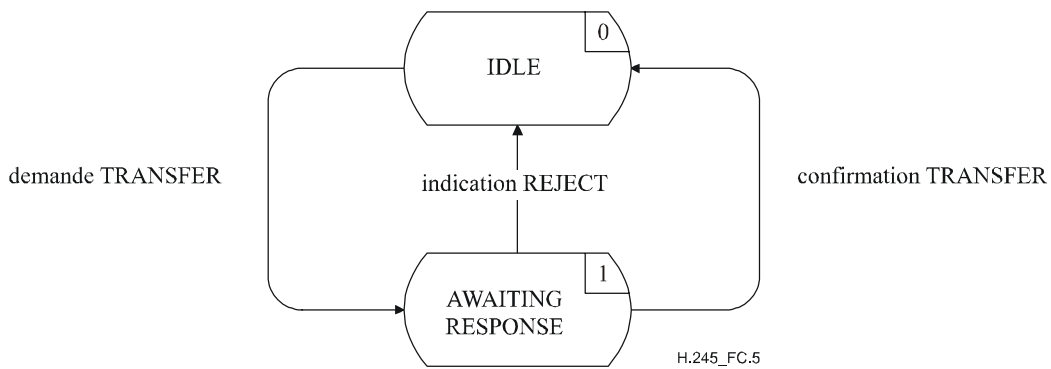


Figure C.5/H.245 – Diagramme de changement d'état pour la séquence de primitives d'une entité CESE sortante

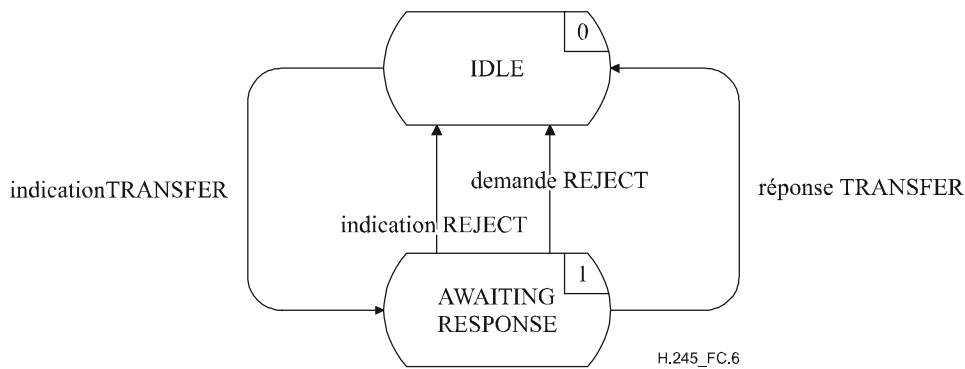


Figure C.6/H.245 – Diagramme de changement d'état pour la séquence de primitives d'une entité CESE entrante

C.3.3 Communication entre les entités CESE homologues

C.3.3.1 Messages

Le Tableau C.7 montre les messages et les champs des entités CESE, définis dans l'Annexe A, qui se rapportent au protocole des entités CESE.

Tableau C.7/H.245 – Noms et champs de messages de l'entité CESE

Fonction	Message	Sens	Champ
transfert	TerminalCapabilitySet	S → E (Note)	sequenceNumber protocolIdentifier multiplexCapability capabilityTable capabilityDescriptors
	TerminalCapabilitySetAck	S ← E	sequenceNumber
rejet	TerminalCapabilitySetReject	S ← E	sequenceNumber cause
réinitialisation	TerminalCapabilitySetRelease	S → E	–
NOTE – Sens: S – sortant, E – entrant.			

C.3.3.2 Variables d'état de l'entité CESE

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

out_SQ

Cette variable d'état est utilisée pour indiquer le message TerminalCapabilitySet le plus récent. Elle est incrémentée de un et mappée au champ sequenceNumber du message TerminalCapabilitySet avant la transmission du message TerminalCapabilitySet. L'opération arithmétique exécutée sur la variable out_SQ est modulo 256.

La variable d'état suivante est définie 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 à l'entité CESE homologue.

C.3.3.3 Temporisateurs CESE

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

T101

Ce temporisateur est utilisé dans l'état AWAITING RESPONSE (attente de la réponse). Il spécifie le temps maximal écoulé avant qu'un message TerminalCapabilitySetAck ou TerminalCapabilitySetReject puisse être reçu.

C.3.4 Procédures de l'entité CESE

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

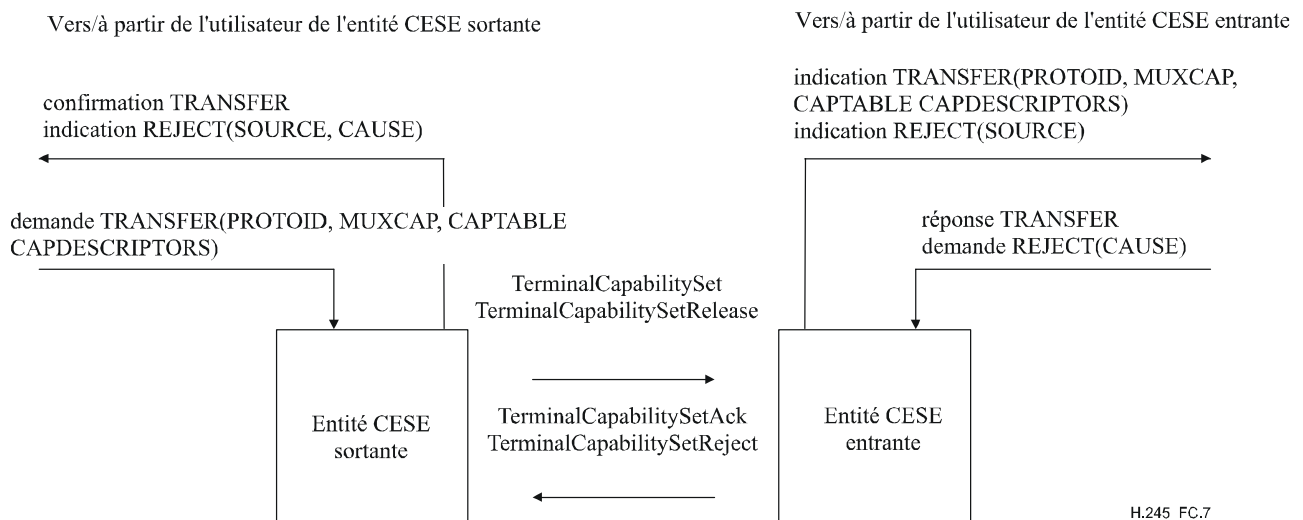


Figure C.7/H.245 – Primitives et messages dans l'entité de signalisation d'échange de capacités (CESE)

C.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 C.8, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau C.8/H.245 – Valeurs par défaut des paramètres des primitives

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

C.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 les valeurs indiquées dans le Tableau C.9.

Tableau C.9/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 vaut "null", c'est-à-dire est absent.		

C.3.4.3 Diagrammes SDL

Les procédures de l'entité CESE sortante et de l'entité CESE entrante sont décrites sous forme de diagrammes SDL dans les Figures C.8 et C.9 respectivement.

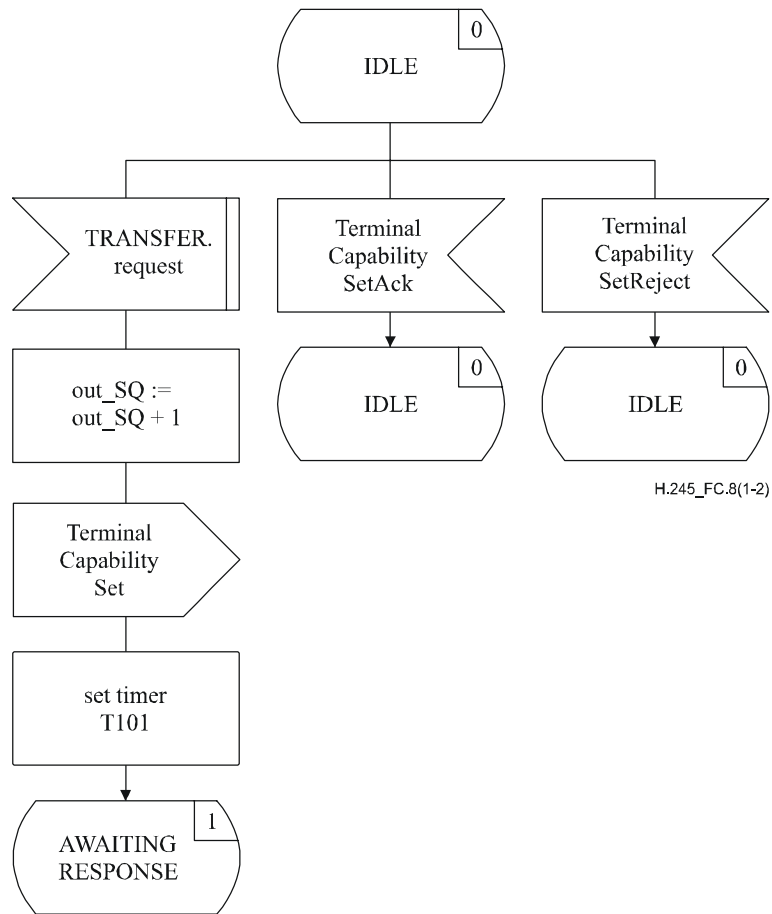
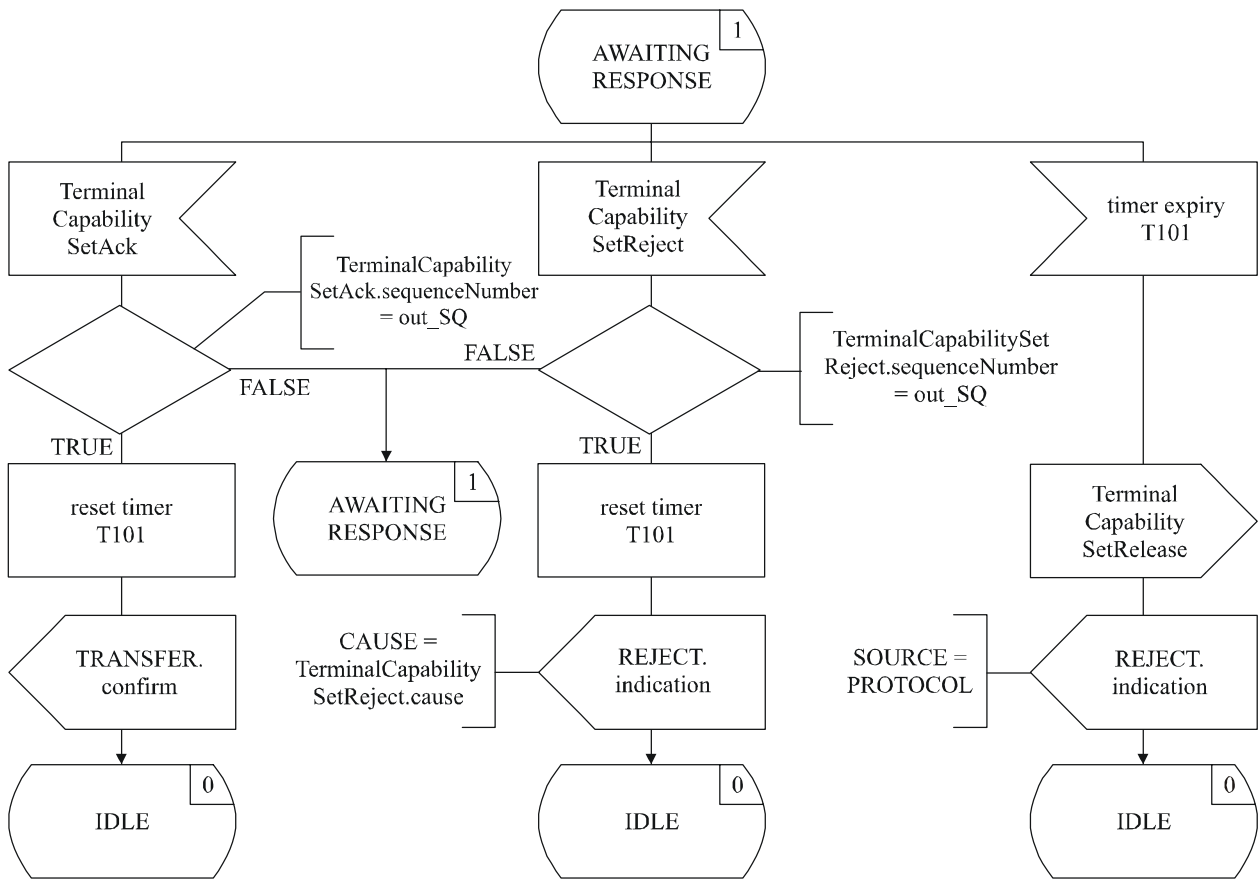
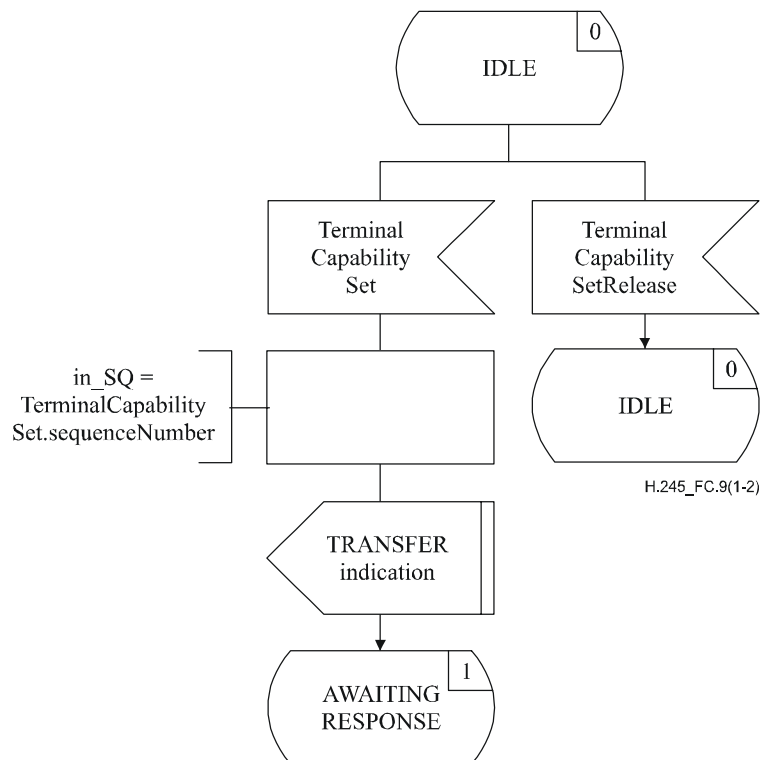


Figure C.8/H.245 – Diagramme SDL de l'entité CESE sortante (feuille 1 de 2)



H.245_FC.8(2-2)

Figure C.8/H.245 – Diagramme SDL de l'entité CESE sortante (feuille 2 de 2)



H.245_FC.9(1-2)

Figure C.9/H.245 – Diagramme SDL de l'entité CESE entrante (feuille 1 de 2)

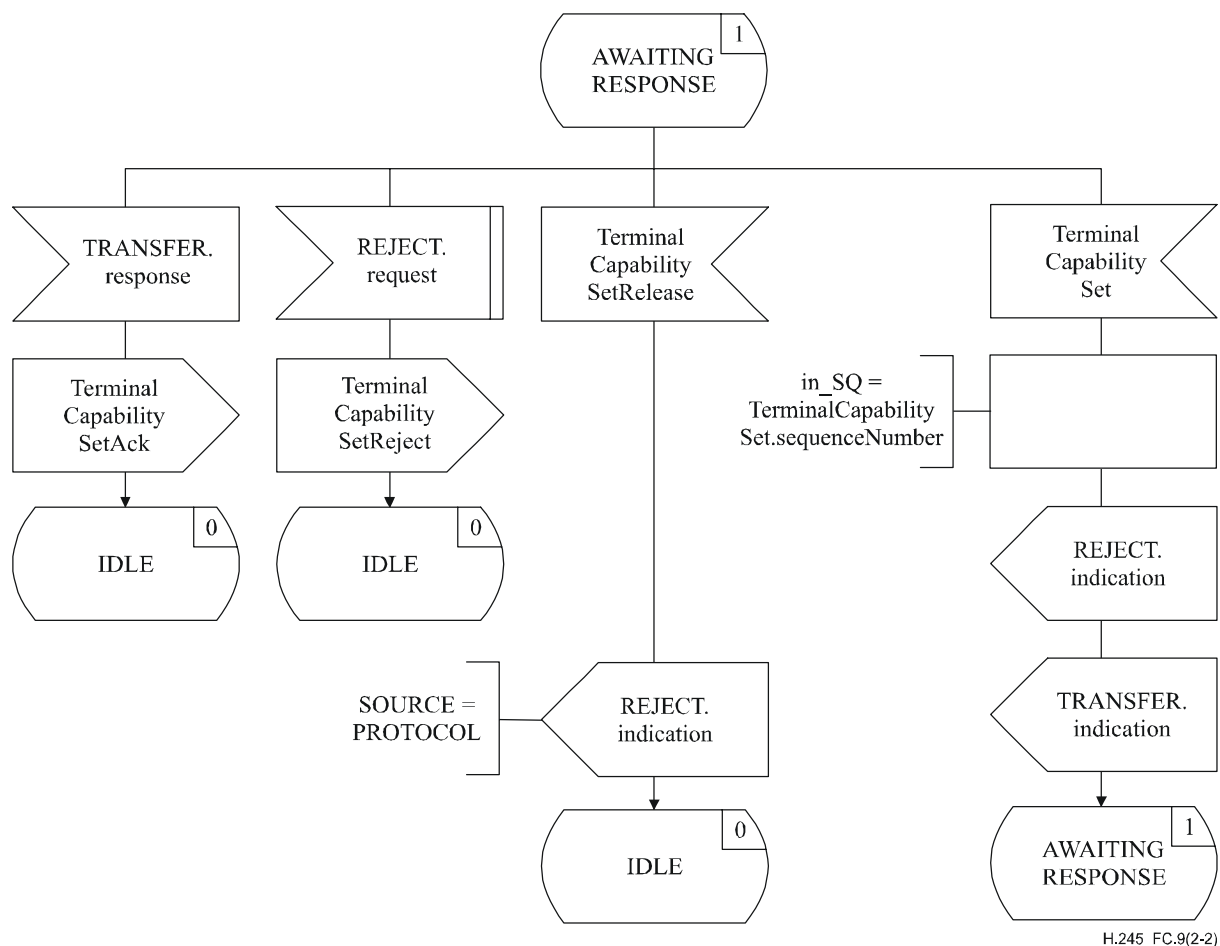


Figure C.9/H.245 – Description SDL de l'entité CESE entrante (feuille 2 de 2)

C.4 Procédures de signalisation de voie logique unidirectionnelle

C.4.1 Introduction

Le protocole spécifié ici permet d'ouvrir et de fermer de façon fiable des voies logiques unidirectionnelles en utilisant des procédures avec acquittement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de voie logique (LCSE, *logical channel signalling entity*). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité LCSE et l'utilisateur de l'entité LCSE ainsi qu'en termes d'états d'entité LCSE. Les informations de protocole sont transmises à l'entité LCSE homologue au moyen des messages appropriés définis dans l'Annexe A.

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 relation entre une entité LCSE entrante et une entité LCSE sortante d'un côté, autre que celle obtenue au moyen des primitives provenant de l'utilisateur d'entité LCSE et allant vers ce même utilisateur. Les conditions d'erreur de l'entité LCSE sont signalées.

Les données ne seront envoyées que sur une voie logique située dans l'état ESTABLISHED. Si les données sont reçues sur une voie logique qui n'est pas dans l'état ESTABLISHED, les données seront ignorées et il faudra considérer qu'aucune défaillance ne s'est produite.

La commutation de mode devrait s'effectuer par la fermeture et l'ouverture des voies logiques existantes, ou par l'ouverture de nouvelles voies logiques.

NOTE – Certaines Recommandations qui utilisent la présente Recommandation peuvent définir certaines voies logiques par défaut. Celles-ci devront être considérées comme étant dans l'état ESTABLISHED dès le début de la communication et ne devront pas être ouvertes en utilisant ces procédures. Cependant, ces voies peuvent être fermées par ces procédures et elles peuvent ultérieurement être réouvertes pour répondre au même objectif ou à un objectif différent.

Un terminal n'ayant plus la capacité de traiter les signaux sur une voie logique devrait prendre des mesures appropriées: celles-ci devraient comprendre la fermeture de la voie logique et la transmission au terminal distant des informations appropriées relatives aux capacités (modifiées).

Le texte suivant est un aperçu général du fonctionnement du protocole dans les entités LCSE. En cas de divergence entre celui-ci et la spécification formelle, la spécification formelle sera applicable.

C.4.1.1 Aperçu général du protocole

L'ouverture d'une voie logique est déclenchée quand la primitive de demande ESTABLISH est présentée par l'utilisateur de l'entité LCSE sortante. Un message OpenLogicalChannel, contenant les paramètres de voie logique dans le sens direct mais n'incluant ces mêmes paramètres dans le sens inverse, est envoyé à l'entité LCSE entrante homologue, et le temporisateur T103 démarre. Si un message OpenLogicalChannelAck est reçu en réponse au message OpenLogicalChannel, alors le temporisateur T103 s'arrête et l'utilisateur est informé par la primitive de confirmation ESTABLISH que l'ouverture de la voie logique a abouti. La voie logique peut alors être utilisée pour transmettre les informations de l'utilisateur. Si, cependant, un message OpenLogicalChannelReject est reçu en réponse au message OpenLogicalChannel, alors le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité LCSE homologue a refusé l'établissement de la voie logique.

Si la temporisation T103 vient à expiration pendant cette période, alors l'utilisateur est informé par la primitive d'indication RELEASE et un message CloseLogicalChannel est envoyé à l'entité LCSE entrante homologue.

Une voie logique qui a été établie avec succès peut être fermée quand la primitive de demande RELEASE est présentée par l'utilisateur de l'entité LCSE sortante. Un message CloseLogicalChannel est envoyé à l'entité LCSE entrante homologue et le temporisateur T103 démarre. Quand un message CloseLogicalChannelAck est reçu, le temporisateur T103 est arrêté et l'utilisateur est informé que la voie logique a bien été fermée par la primitive de confirmation RELEASE.

Si la temporisation T103 vient à expiration pendant cette période, alors l'utilisateur est informé par la primitive d'indication RELEASE.

Avant que l'un des messages OpenLogicalChannelAck et OpenLogicalChannelReject ait été reçu en réponse à un message OpenLogicalChannel préalablement envoyé, l'utilisateur de l'entité LCSE sortante peut fermer la voie logique en utilisant la primitive de demande RELEASE.

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

C.4.1.2 Aperçu général du protocole – Entité LCSE entrante

Quand un message OpenLogicalChannel est reçu par l'entité LCSE entrante, l'utilisateur est informé de la demande d'ouverture d'une nouvelle voie logique par la primitive d'indication ESTABLISH. L'utilisateur de l'entité LCSE entrante signale l'acceptation de la demande d'ouverture de voie logique en présentant la primitive de réponse ESTABLISH, et un message OpenLogicalChannelAck est envoyé à l'entité LCSE sortante homologue. La voie logique peut alors être utilisée pour recevoir les informations de l'utilisateur. L'utilisateur de l'entité LCSE entrante signale le refus de la demande d'établissement de voie logique en présentant la primitive de

demande RELEASE et un message OpenLogicalChannelReject est envoyé à l'entité LCSE sortante homologue.

Une voie logique qui a bien été établie peut être fermée quand le message CloseLogicalChannel est reçu par l'entité LCSE entrante. L'utilisateur de l'entité LCSE entrante est informé par la primitive d'indication RELEASE, et le message CloseLogicalChannelAck est envoyé à l'entité LCSE sortante homologue.

C.4.1.3 Résolution des conflits

Des conflits peuvent survenir quand des demandes d'ouverture de voies logiques sont déclenchées en même temps. Il peut être possible de déterminer qu'il y a un conflit à partir de la connaissance des capacités échangées.

Les terminaux doivent être en mesure de détecter qu'un conflit est survenu ou pourrait survenir, et ils doivent réagir comme suit:

avant que des voies logiques puissent être ouvertes, l'un des terminaux doit être désigné comme le terminal maître et l'autre comme le terminal asservi. Le protocole défini au § C.2 est un mécanisme permettant d'effectuer cette désignation. Le terminal maître doit refuser immédiatement toute demande émanant du terminal asservi qu'il identifie comme une demande conflictuelle. Le terminal asservi peut identifier de tels conflits, mais devra répondre à la demande émanant du terminal maître, en sachant que sa demande précédente sera refusée.

NOTE – De tels conflits pourraient être provoqués par des ressources restreintes du terminal, par exemple quand les capacités d'émission et de réception sont interdépendantes, comme c'est le cas pour un terminal pouvant utiliser un grand nombre d'algorithmes audio, mais pouvant uniquement décoder avec le même algorithme que celui utilisé pour le codage.

Pour éviter autant que possible que les extrémités tentent d'ouvrir des voies logiques conflictuelles lorsque le point d'extrémité asservi est soumis à des limitations de capacités symétriques, il est recommandé de procéder comme suit. Lorsque le terminal maître et le terminal asservi ont indiqué leurs choix de capacités de réception pour un type de média particulier, le terminal asservi doit tenter d'ouvrir une voie logique pour la capacité préférée du terminal maître dont il assure la mise en œuvre conformément à l'ordre de priorité indiqué par le terminal maître pour ses capacités. Le terminal maître, quant à lui, doit tenter d'ouvrir une voie logique pour sa capacité préférée dont le terminal asservi assure la mise en œuvre, conformément à l'ordre de priorité qu'il a indiqué pour ses capacités.

Par exemple, si le terminal maître a déclaré être à même d'assurer la mise en œuvre des capacités G.723.1, G.729 et G.711, et que le terminal asservi ait indiqué être à même d'assurer la mise en œuvre des capacités G.711 et G.729, la capacité préférable étant citée en premier dans les deux cas, le terminal maître et le terminal asservi doivent tenter d'ouvrir des voies logiques pour la capacité G.729.

Après que la demande d'ouverture d'une voie logique a été refusée par le terminal maître pour cause de conflit masterSlaveConflict ou équivalente, il incombe au terminal asservi d'ouvrir une voie non conflictuelle.

Lorsque le terminal asservi détecte un conflit et que le terminal maître ne refuse pas une voie logique ouverte conflictuelle, le terminal asservi doit fermer la voie conflictuelle. Dans le cas de voies logiques conflictuelles dues à des limitations de capacités symétriques, le terminal asservi doit ouvrir une voie logique appropriée en appliquant la procédure de remplacement et, en temps utile, fermer la voie logique conflictuelle.

C.4.1.4 Résolution de conflit entre voies unidirectionnelles et bidirectionnelles

Un autre type de conflit qui pourrait se produire se présente quand les deux extrémités tentent d'ouvrir une voie du même type, mais qu'une d'entre elles essaye de l'ouvrir en tant que voie unidirectionnelle et l'autre en tant que voie bidirectionnelle.

En telle occurrence, le système maître doit rejeter la voie avec une valeur de cause égale à masterSlaveConflict, et le système asservi devrait déterminer s'il a besoin d'ouvrir une voie non conflictuelle ou ne rien faire de plus.

Quand le système asservi détecte un conflit et que le système maître ne rejette pas une voie logique ouverte en conflit, ce système asservi devrait fermer la voie conflictuelle.

L'Appendice X contient des scénarios contribuant à préciser comment résoudre de tels conflits.

C.4.2 Communication entre l'entité LCSE et l'utilisateur de l'entité LCSE

C.4.2.1 Primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE

Les communications entre l'entité LCSE et l'utilisateur de l'entité LCSE sont effectuées en utilisant les primitives indiquées dans le Tableau C.10.

Tableau C.10/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
ESTABLISH	FORWARD_PARAM	FORWARD_PARAM	– (Note 1)	–
RELEASE	CAUSE	SOURCE CAUSE	Non défini (Note 2)	–
ERROR	non défini	ERRCODE	Non défini	Non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'existe pas.				

C.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 indique les erreurs de l'entité LCSE à une entité de gestion.

C.4.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.10 est la suivante:

- le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie logique. Ce paramètre est mappé au champ des paramètres forwardLogicalChannelParameters du message OpenLogicalChannel et est transmis de façon transparente 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 la voie logique. Le paramètre SOURCE a la valeur de "USER" ou de "LCSE", indiquant soit l'utilisateur de l'entité LCSE, soit l'entité LCSE elle-même. Le deuxième cas peut être le résultat d'une erreur de protocole;

- c) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité LCSE homologue a refusé une demande d'établissement de voie logique. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "LCSE";
- d) le paramètre ERRCODE indique le type d'erreur d'entité LCSE. Le Tableau C.14 indique les valeurs autorisées du paramètre ERRCODE.

C.4.2.4 Etats d'entité LCSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE, de même que l'échange de messages entre des entités LCSE homologues. Les états sont spécifiés de façon distincte pour chacune des entités LCSE sortante et entrante. Les états correspondant à l'entité LCSE sortante sont:

Etat 0: RELEASED

La voie logique est libre. Elle ne devra pas être utilisée pour envoyer des données sortantes.

Etat 1: AWAITING ESTABLISHMENT

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

Etat 2: ESTABLISHED

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

Etat 3: AWAITING RELEASE

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

Les états correspondant à l'entité LCSE entrante sont les suivants:

Etat 0: RELEASED

La voie logique est libre. Elle ne devra pas être utilisée pour recevoir des données entrantes.

Etat 1: AWAITING ESTABLISHMENT

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

Etat 2: ESTABLISHED

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

C.4.2.5 Diagramme de changement d'état

La séquence autorisée de primitives entre l'entité LCSE et l'utilisateur de l'entité LCSE est définie ici. La séquence autorisée de primitives fait référence à des états de l'entité LCSE du point de vue de l'utilisateur de l'entité LCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités LCSE sortante et LCSE entrante, comme cela est indiqué à la Figure C.10 et à la Figure C.11 respectivement.

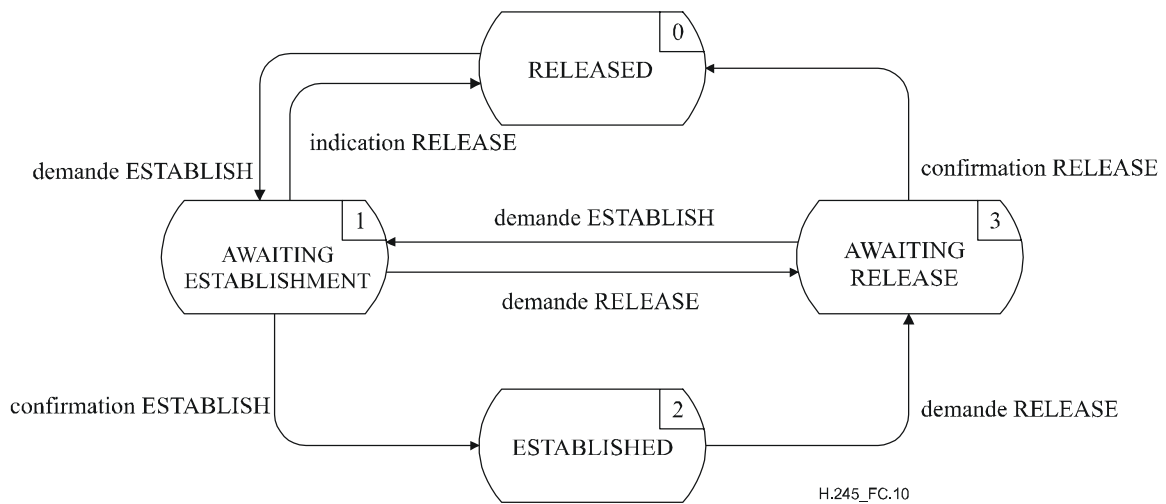


Figure C.10/H.245 – Diagramme de changement d'état pour la séquence de primitives d'une entité LCSE sortante

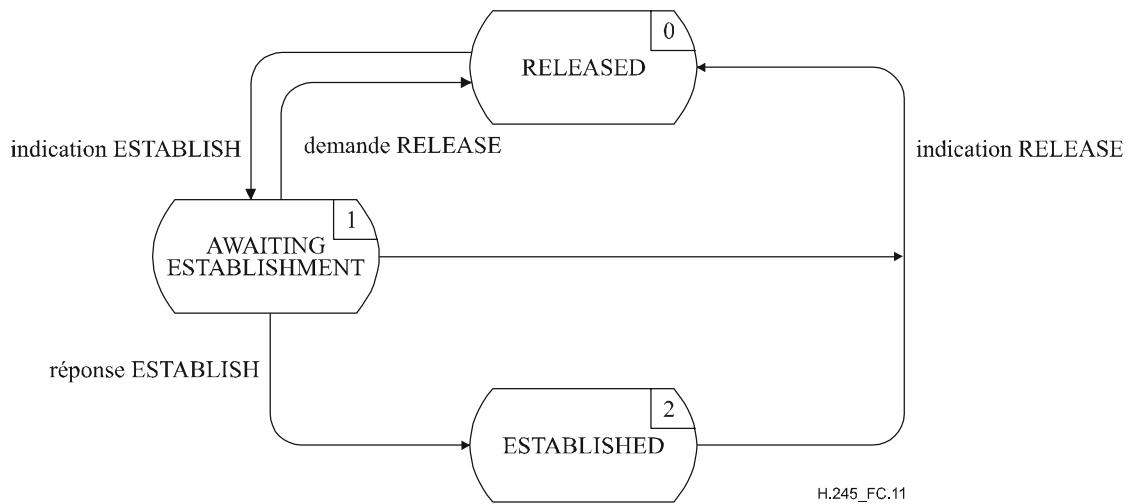


Figure C.11/H.245 – Diagramme de changement d'état pour la séquence de primitives d'une entité LCSE entrante

C.4.3 Communications entre les entités LCSE homologues

C.4.3.1 Messages d'entité LCSE

Le Tableau C.11 indique les messages et les champs des entités LCSE, définis dans l'Annexe A, qui se rapportent au protocole d'entité LCSE.

Tableau C.11/H.245 – Noms et champs de messages des entités LCSE

Fonction	Message	Sens	Champ
établissement	OpenLogicalChannel	S → E (Note)	forwardLogicalChannelNumber forwardLogicalChannelParameters
	OpenLogicalChannelAck	S ← E	forwardLogicalChannelNumber
	OpenLogicalChannelReject	S ← E	forwardLogicalChannelNumber cause
libération	CloseLogicalChannel	S → E	forwardLogicalChannelNumber source
	CloseLogicalChannelAck	S → E	forwardLogicalChannelNumber
NOTE – Sens: S – sortant, E – entrant.			

C.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 faire la distinction entre les entités LCSE sortantes. Elle est initialisée lors de l'initialisation de l'entité LCSE sortante. La valeur de la variable out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés à partir d'une entité LCSE sortante. En ce qui concerne les messages reçus par une entité LCSE sortante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de la variable out_LCN.

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

in_LCN

Cette variable d'état permet de faire la distinction entre les entités LCSE entrantes. Elle est initialisée lors de l'initialisation de l'entité LCSE entrante. La valeur de la variable in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés par une entité LCSE entrante. En ce qui concerne les messages reçus dans une entité LCSE entrante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de la variable in_LCN.

C.4.3.3 Temporiseurs d'entité LCSE

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

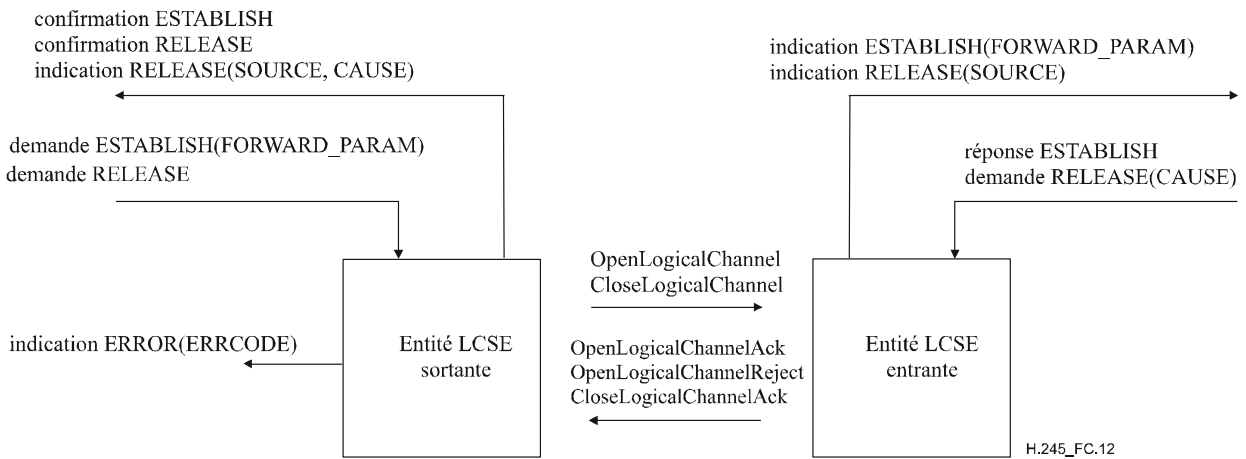
T103

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 et CloseLogicalChannelAck ne peut être reçu.

C.4.4 Procédures d'entité LCSE

C.4.4.1 Introduction

La Figure C.12 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités LCSE entrante et sortante.



H.245_FC.12

Figure C.12/H.245 – Primitives et messages de l'entité de signalisation de voie logique (LCSE)

C.4.4.2 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 les valeurs indiquées dans le Tableau C.12.

Tableau C.12/H.245 – Valeurs par défaut des paramètres des primitives

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

NOTE – Un paramètre de primitive doit être codé comme valant "null" si un champ de message indiqué n'est pas présent dans le message.

C.4.4.3 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.13.

Tableau C.13/H.245 – Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut (Note 1)
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 valant "null", c'est-à-dire est absent.		
NOTE 2 – Les paramètres reverseLogicalChannelParameters ne sont pas codés dans les procédures de signalisation de voie logique unidirectionnelle.		

C.4.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition d'erreur particulière. Le Tableau C.14 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 C.14/H.245 – Valeurs du paramètre ERRCODE de l'entité LCSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
Message inapproprié	A	OpenLogicalChannelAck	RELEASED
	B	OpenLogicalChannelRequest	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
Aucune réponse provenant de l'entité LCSE homologue	D	Fin de temporisation T103	AWAITING ESTABLISHMENT AWAITING RELEASE

C.4.4.5 Diagrammes SDL

Les procédures de l'entité LCSE sortante et de l'entité LCSE entrante sont décrites sous forme de diagrammes SDL dans les Figures C.13 et C.14 respectivement.

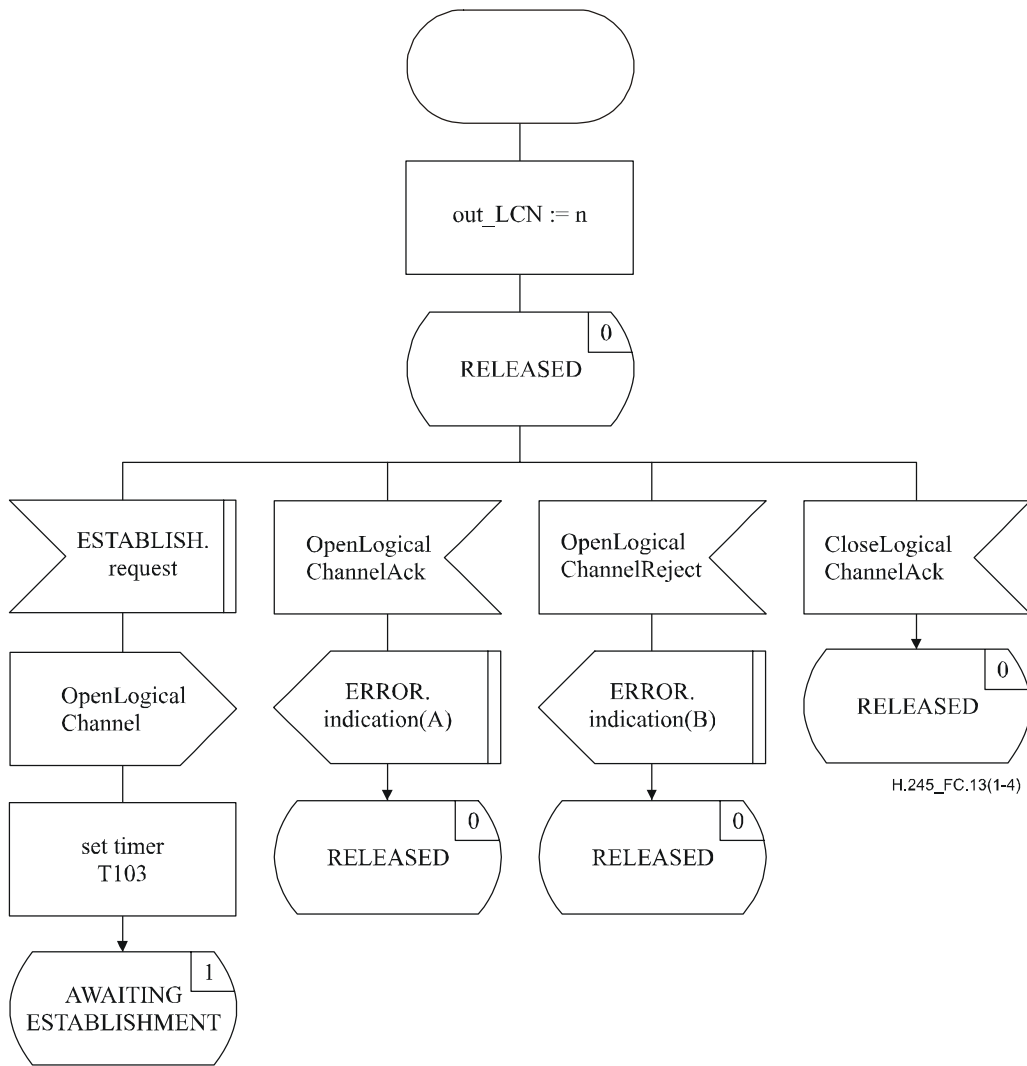
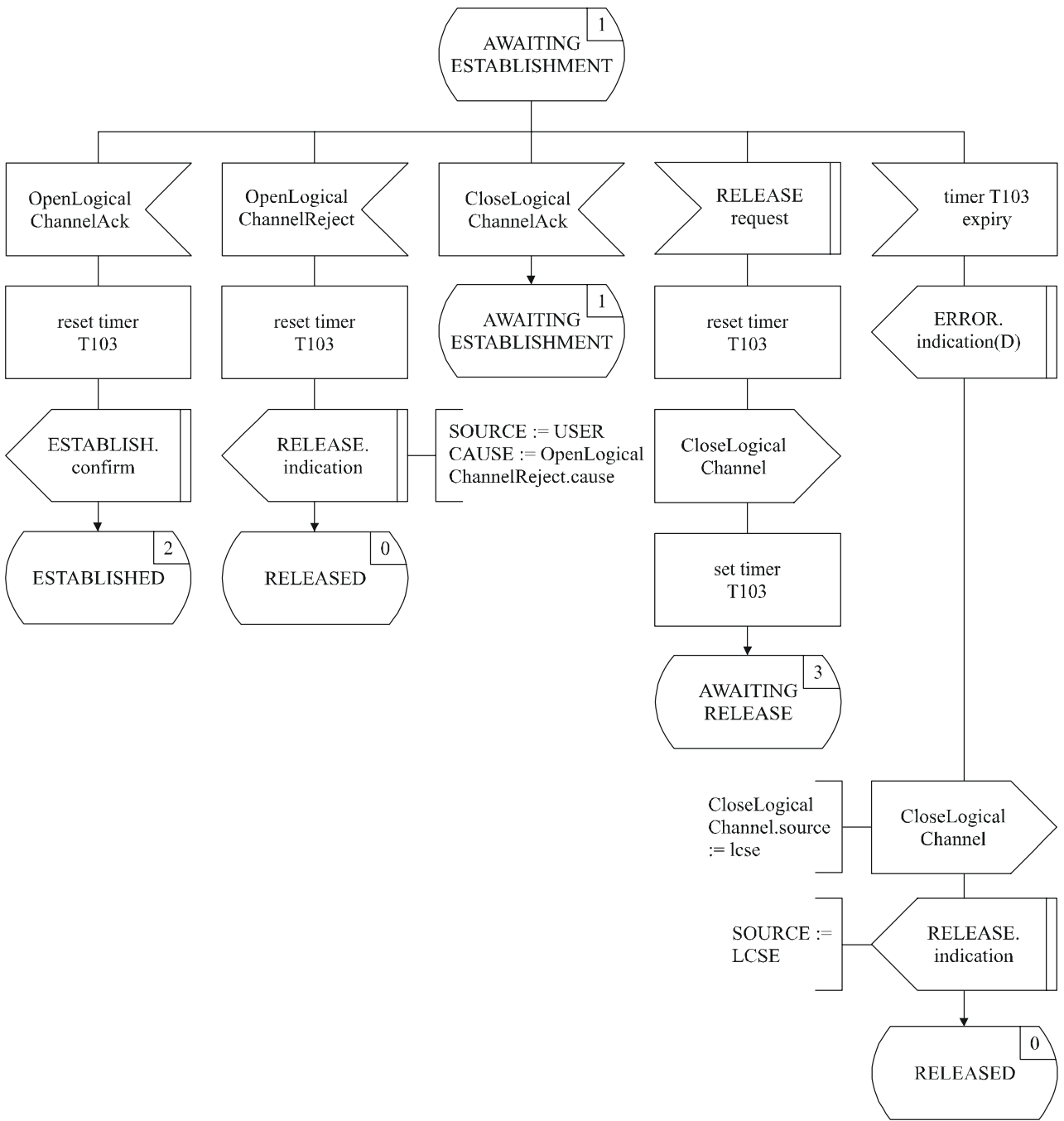


Figure C.13/H.245 – Diagramme SDL de l'entité LCSE sortante (feuille 1 de 4)



H.245_C.13(2-4)

Figure C.13/H.245 – Diagramme SDL de l'entité LCSE sortante (feuille 2 de 4)

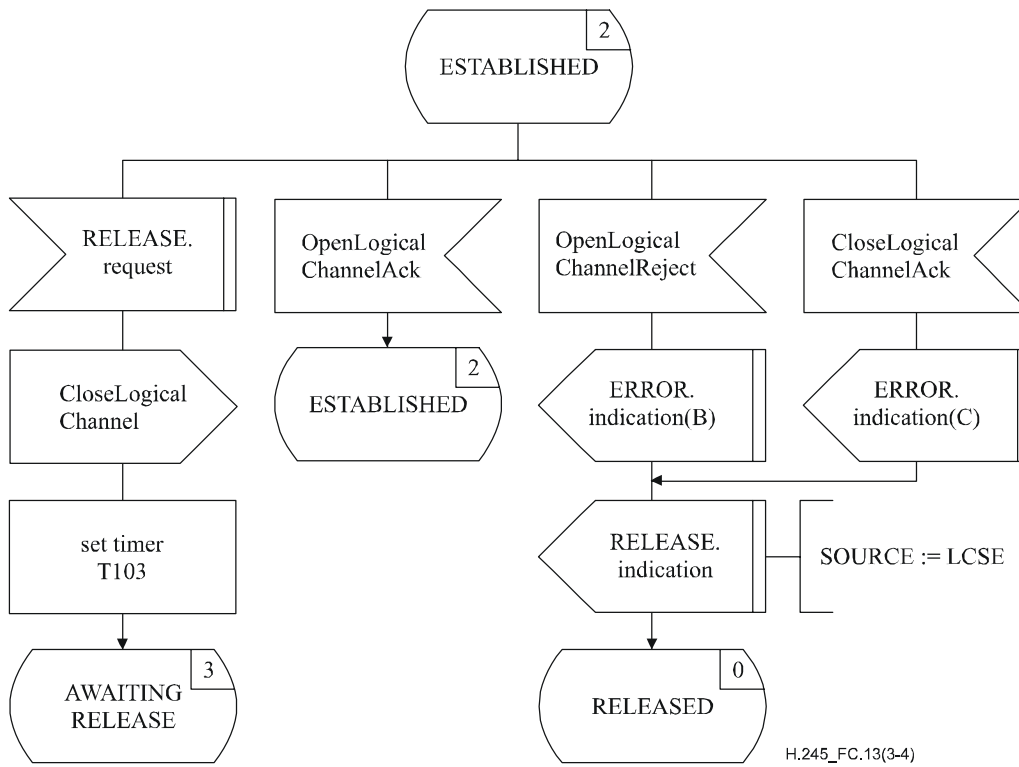


Figure C.13/H.245 – Diagramme SDL de l'entité LCSE sortante (feuille 3 de 4)

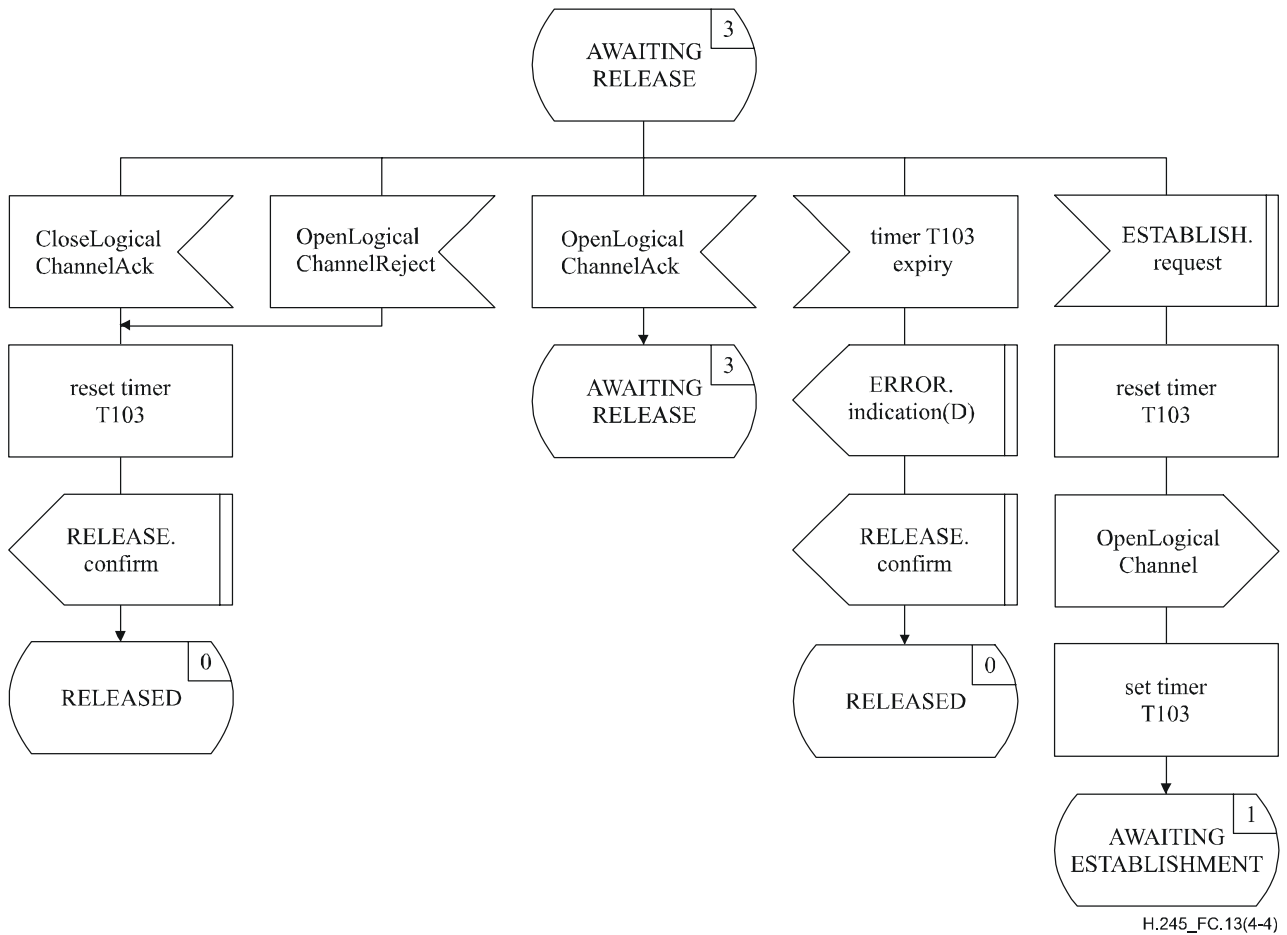


Figure C.13/H.245 – Diagramme SDL de l'entité LCSE sortante (feuille 4 de 4)

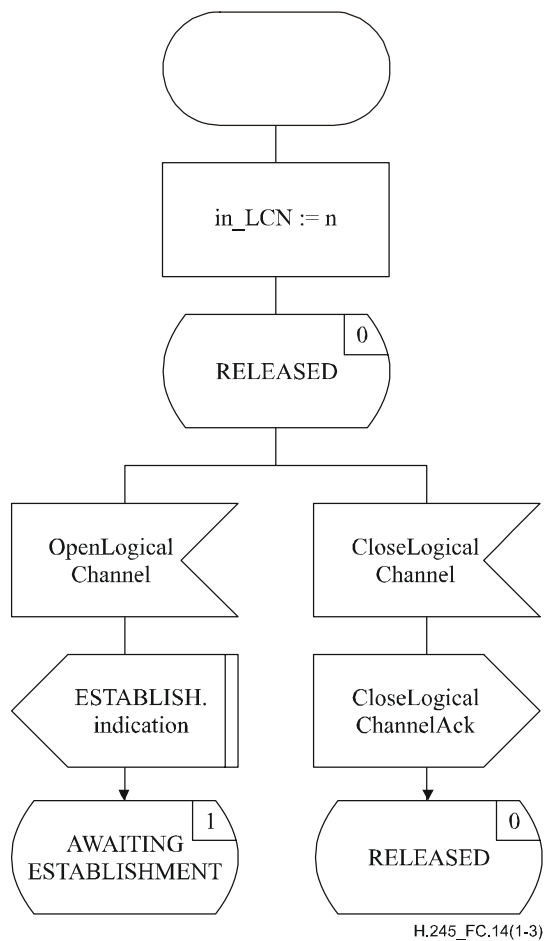


Figure C.14/H.245 – Diagramme SDL de l'entité LCSE entrante (feuille 1 de 3)

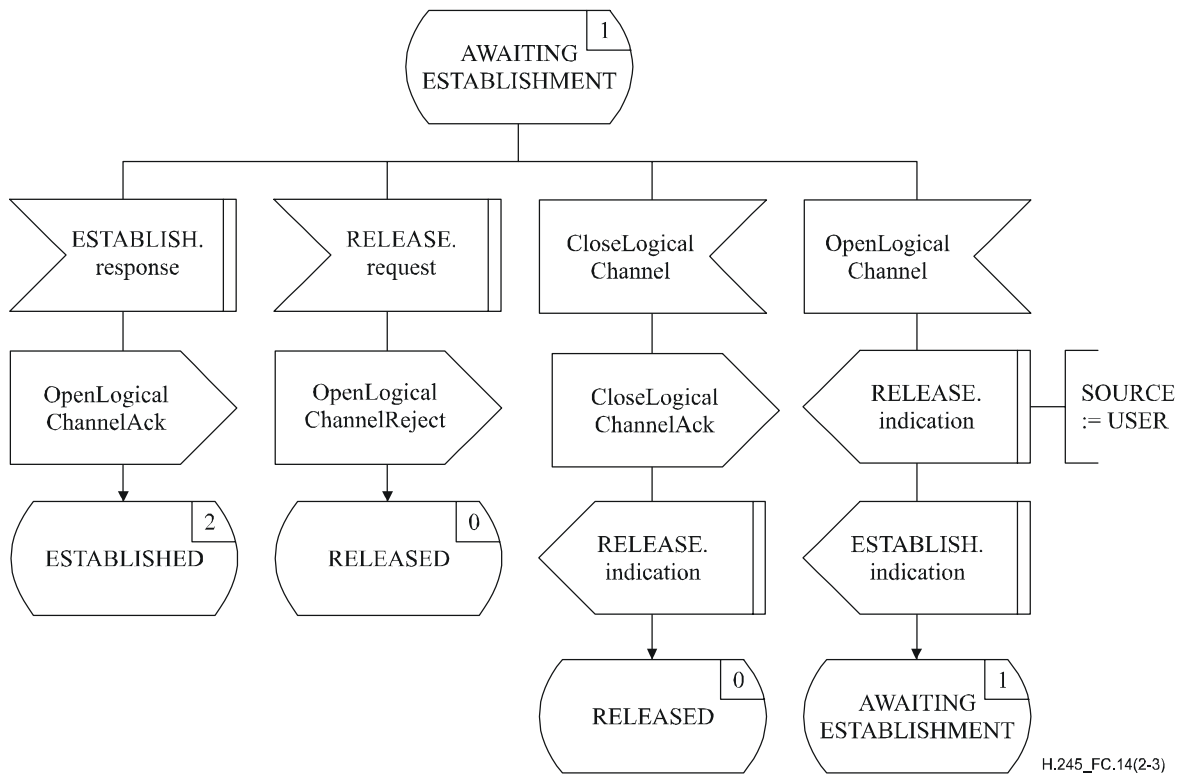


Figure C.14/H.245 – Diagramme SDL de l'entité LCSE entrante (feuille 2 de 3)

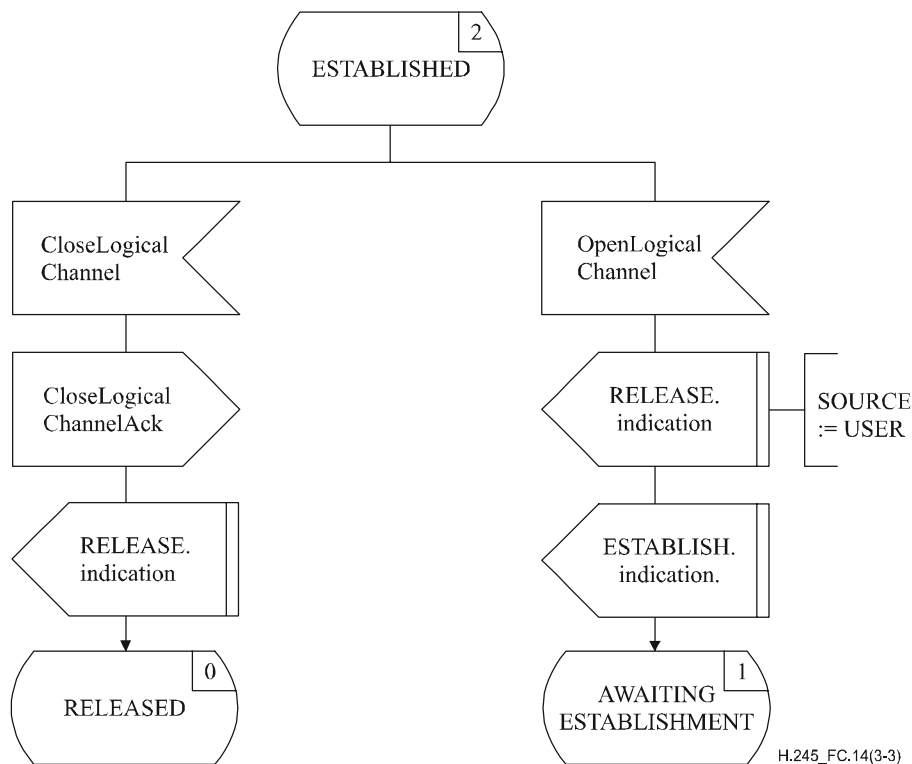


Figure C.14/H.245 – Diagramme SDL de l'entité LCSE entrante (feuille 3 de 3)

C.5 Procédures de signalisation de voie logique bidirectionnelle

C.5.1 Introduction

Le protocole spécifié ici permet d'obtenir une ouverture et une fermeture fiable des voies logiques bidirectionnelles au moyen de procédures avec acquittement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de voie logique bidirectionnelle (B-LCSE, *bidirectional logical channel signalling entity*). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE, de même qu'en termes d'états d'entité B-LCSE. Les informations de protocole sont transférées à l'entité B-LCSE homologue au moyen des messages appropriés définis dans l'Annexe A.

Il y a une entité B-LCSE sortante et une entité B-LCSE entrante. A chacune des extrémités sortante et entrante, il y a une instance d'entité B-LCSE pour chaque voie logique bidirectionnelle. Il n'y a pas de relation entre une entité B-LCSE entrante et une entité B-LCSE sortante situées d'un côté, autre que celle obtenue au moyen des primitives provenant de l'utilisateur de l'entité B-LCSE et allant vers ce même utilisateur. Les conditions d'erreur des entités B-LCSE sont signalées.

Une voie logique bidirectionnelle comprend un couple de voies logiques unidirectionnelles associées. Le terme "direct" (côté sortant) est utilisé pour faire référence à la transmission dans le sens suivant: du terminal présentant la demande de voie logique bidirectionnelle vers l'autre terminal et le terme "inverse" (côté entrant) est utilisé pour faire référence au sens opposé de transmission.

Les données ne devront être envoyées que sur une voie logique bidirectionnelle dans l'état ESTABLISHED (voie établie). Cependant les données peuvent être reçues sur la voie dans le sens direct quand l'entité B-LCSE entrante est dans l'état AWAITING CONFIRMATION (attente de confirmation). Les données qui sont reçues dans d'autres états que l'état ESTABLISHED (voie établie) et l'état AWAITING CONFIRMATION (attente de la confirmation) devront être ignorées et il faudra considérer qu'aucune défaillance ne s'est produite.

Un terminal peut refuser une demande d'ouverture de voie logique bidirectionnelle uniquement parce qu'il ne peut pas utiliser les paramètres demandés pour la voie inverse. Dans ce cas, il devra refuser la demande avec le paramètre cause valant `unsuitableReverseParameters` et immédiatement déclencher les procédures d'établissement de voie logique bidirectionnelle comme cela a été demandé par le terminal distant, dans lesquels les paramètres dans le sens inverse sont identiques aux paramètres dans le sens direct de la demande présentée par le terminal distant qui s'est soldé par un échec, et avec les paramètres dans le sens direct que le terminal peut utiliser et dont on sait qu'ils pourront être utilisés par le terminal distant.

La commutation de mode devrait être effectuée en fermant et en ouvrant les voies logiques existantes, ou en ouvrant de nouvelles voies logiques.

NOTE – Certaines Recommandations qui utilisent la présente Recommandation peuvent définir certaines voies logiques par défaut. Ces voies seront considérées comme étant dans l'état ESTABLISHED dès le début de la communication et ne devront pas être ouvertes en utilisant ces procédures. Elles peuvent cependant être fermées par ces mêmes procédures et être ultérieurement réouvertes pour répondre au même objectif ou pour répondre à un objectif différent.

Un terminal qui ne peut plus traiter les signaux sur une voie logique devrait prendre des mesures appropriées pour la fermeture de la voie logique et la transmission au terminal distant des informations appropriées relatives aux capacités (modifiées).

Le texte ci-après est un aperçu général du fonctionnement du protocole des entités B-LCSE. En cas de divergence entre cette spécification et la spécification formelle, cette dernière sera applicable.

C.5.1.1 Aperçu général du protocole

L'ouverture d'une voie logique est déclenchée quand la primitive de demande ESTABLISH est présentée par l'utilisateur dans l'entité B-LCSE sortante. Un message `OpenLogicalChannel`, contenant les paramètres de voie logique dans les sens direct et inverse, est envoyé à l'entité B-LCSE entrante homologue et le temporisateur T103 démarre. Si un message `OpenLogicalChannelAck` est reçu en réponse au message `OpenLogicalChannel`, alors le temporisateur T103 est arrêté, un message `OpenLogicalChannelConfirm` est envoyé à l'entité B-LCSE entrante homologue et l'utilisateur est informé par la primitive de confirmation ESTABLISH que la voie logique a bien été ouverte. La voie logique peut alors être utilisée pour émettre et recevoir les informations de l'utilisateur. Si, cependant, un message `OpenLogicalChannelReject` est reçu en réponse au message `OpenLogicalChannel`, le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité B-LCSE homologue a refusé l'établissement de la voie logique.

Si le temporisateur T103 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE et un message `CloseLogicalChannel` est envoyé à l'entité B-LCSE entrante homologue.

Une voie logique dont l'établissement a abouti peut être fermée quand la primitive de demande RELEASE est présentée par l'utilisateur de l'entité B-LCSE sortante. Un message `CloseLogicalChannel` est envoyé à l'entité B-LCSE entrante homologue et le temporisateur T103 démarre. Quand un message `CloseLogicalChannelAck` est reçu, le temporisateur T103 est arrêté et l'utilisateur est informé par la primitive de confirmation RELEASE que la fermeture de la voie logique a abouti.

Si le temporisateur T103 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE.

Avant que l'un des messages `OpenLogicalChannelAck` et `OpenLogicalChannelReject` ait été reçu en réponse à un message `OpenLogicalChannel` préalablement envoyé, l'utilisateur de l'entité B-LCSE sortante peut fermer la voie logique en utilisant la primitive de demande RELEASE.

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

C.5.1.2 Aperçu général du protocole – Entité B-LCSE entrante

Quand un message `OpenLogicalChannel` est reçu dans l'entité B-LCSE entrante, l'utilisateur est informé de la demande d'ouverture d'une nouvelle voie logique par la primitive d'indication `ESTABLISH`. L'utilisateur de l'entité B-LCSE entrante signale l'acceptation de la demande d'ouverture de voie logique en présentant une primitive de réponse `ESTABLISH`, et un message `OpenLogicalChannelAck` est envoyé à l'entité B-LCSE sortante homologue. La voie directe de la voie bidirectionnelle peut alors être utilisée pour recevoir les informations de l'utilisateur. L'utilisateur de l'entité B-LCSE entrante signale le refus de la demande d'ouverture de voie logique en présentant la primitive de demande `RELEASE`, et un message `OpenLogicalChannelReject` est envoyé à l'entité B-LCSE sortante homologue.

Quand un message `OpenLogicalChannelConfirm` est reçu dans l'entité B-LCSE entrante, l'utilisateur est informé par la primitive de confirmation `ESTABLISH` que la voie logique bidirectionnelle est établie. La voie inverse de la voie logique bidirectionnelle peut alors être utilisée pour transmettre les informations de l'utilisateur.

Une voie logique dont l'établissement a abouti peut être fermée quand le message `CloseLogicalChannel` est reçu dans l'entité B-LCSE entrante. L'utilisateur de l'entité B-LCSE entrante en est informé par la primitive d'indication `RELEASE` et le message `CloseLogicalChannelAck` est envoyé à l'entité B-LCSE sortante homologue.

C.5.1.3 Résolution des conflits

Des conflits peuvent survenir quand des demandes d'ouverture de voies logiques sont lancées en même temps. Il est possible de déterminer qu'il y a un conflit à partir de la connaissance des capacités échangées. Par ailleurs, les deux terminaux peuvent déclencher l'ouverture d'une voie logique bidirectionnelle pour répondre au même objectif, bien que les paramètres exacts demandés puissent être différents, et ces terminaux ont des capacités suffisantes pour satisfaire les deux demandes. Les terminaux doivent pouvoir détecter le moment où les deux situations surviendront et devront réagir de la façon suivante.

Avant que les voies logiques puissent être ouvertes, l'un des terminaux sera désigné comme étant le terminal maître et l'autre comme le terminal asservi. Le protocole défini au § C.2 est un moyen de faire cette désignation. Le terminal maître doit refuser sans délai toute demande présentée par le terminal asservi qu'il identifie comme étant une demande conflictuelle. Le terminal asservi peut détecter de tels conflits, mais doit répondre à la demande présentée par le terminal maître, tout en sachant que sa demande antérieure sera refusée.

En ce qui concerne le second type de conflit défini ci-dessus, il est impossible d'établir une distinction entre le cas où deux voies bidirectionnelles sont demandées et le cas où une seule de ces voies est effectivement demandée. Les terminaux devront réagir en supposant qu'une seule voie est demandée, mais un terminal pourra ultérieurement réitérer sa demande si l'hypothèse est incorrecte.

Pour éviter autant que possible que les extrémités tentent d'ouvrir des voies logiques conflictuelles lorsque le point d'extrémité asservi est soumis à des limitations de capacités symétriques, il est recommandé de procéder comme suit. Lorsque le terminal maître et le terminal asservi ont indiqué leurs choix de capacités de réception pour un type de média particulier, le terminal asservi doit tenter d'ouvrir une voie logique pour la capacité préférée du terminal maître dont il assure la mise en œuvre conformément à l'ordre de priorité indiqué par le terminal maître pour ses capacités. Le terminal maître, quant à lui, doit tenter d'ouvrir une voie logique pour sa capacité préférée dont le terminal asservi assure la mise en œuvre, conformément à l'ordre de priorité qu'il a indiqué pour ses capacités.

Par exemple, si le terminal maître a déclaré être à même d'assurer la mise en œuvre des capacités G.723.1, G.729 et G.711, et que le terminal asservi ait indiqué être à même d'assurer la mise en œuvre des capacités G.711 et G.729, la capacité préférable étant citée en premier dans les deux cas, le terminal maître et le terminal asservi doivent tenter d'ouvrir des voies logiques pour la capacité G.729.

Après que la demande d'ouverture d'une voie logique a été refusée par le terminal maître pour cause de conflit masterSlaveConflict ou équivalente, il incombe au terminal asservi d'ouvrir une voie non conflictuelle.

Lorsque le terminal asservi détecte un conflit et que le terminal maître ne refuse pas une voie logique ouverte conflictuelle, le terminal asservi doit fermer la voie conflictuelle. Dans le cas de voies logiques conflictuelles dues à des limitations de capacités symétriques, le terminal asservi doit ouvrir une voie logique appropriée en appliquant la procédure de remplacement et, en temps utile, fermer la voie logique conflictuelle.

C.5.2 Communication entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE

C.5.2.1 Primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE

Les communications entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE s'effectuent en utilisant les primitives indiquées dans le Tableau C.15.

Tableau C.15/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
ESTABLISH (établissement)	FORWARD_PARAM REVERSE_PARAM	FORWARD_PARAM REVERSE_PARAM	REVERSE_DATA	REVERSE_DATA
RELEASE (libération)	CAUSE (raison)	SOURCE (origine) CAUSE (raison)	Non défini (Note 2)	– (Note 1)
ERROR (erreur)	Non défini	ERRCODE	Non défini	Non défini
NOTE 1 – "-" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'existe pas.				

C.5.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- les primitives ESTABLISH sont utilisées pour établir une voie logique pour les communications audiovisuelles et les transmissions de données;
- les primitives RELEASE sont utilisées pour libérer une voie logique;
- la primitive ERROR indique les erreurs d'entité B-LCSE à une entité de gestion.

C.5.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués au Tableau C.15 est la suivante:

- le paramètre FORWARD_PARAM spécifie les paramètres associés à la voie dans le sens direct, c'est-à-dire du terminal contenant l'entité B-LCSE sortante vers le terminal contenant l'entité B-LCSE entrante. Ce paramètre est mappé au champ forwardLogicalChannelParameters du message OpenLogicalChannel et transmis de façon transparente à l'utilisateur de l'entité LCSE homologue;

- b) le paramètre REVERSE_PARAM spécifie les paramètres associés à la voie inverse, c'est-à-dire du terminal contenant l'entité B-LCSE entrante vers le terminal contenant l'entité B-LCSE sortante. Ce paramètre est mappé au champ reverseLogicalChannelParameters du message OpenLogicalChannel et transmis de façon transparente à l'utilisateur de l'entité LCSE homologue;
- c) le paramètre REVERSE_DATA spécifie certains paramètres associés à la voie inverse, c'est-à-dire du terminal contenant l'entité B-LCSE entrante vers le terminal contenant l'entité B-LCSE sortante. Ce paramètre est mappé au champ reverseLogicalChannelParameters du message OpenLogicalChannelAck et transmis de façon transparente à 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. Le paramètre SOURCE a la valeur de "USER" ou "B-LCSE", indiquant soit l'utilisateur de l'entité B-LCSE soit l'entité B-LCSE elle-même. Le deuxième cas peut être le résultat d'une erreur de protocole;
- e) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité B-LCSE homologue a refusé une demande d'établissement de voie logique. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "B-LCSE";
- f) le paramètre ERRCODE indique le type d'erreur de l'entité B-LCSE. Le Tableau C.19 indique les valeurs autorisées pour le paramètre ERRCODE.

C.5.2.4 Etats d'entité B-LCSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE, de même que l'échange de messages entre les entités B-LCSE homologues. Les états sont spécifiés distinctement pour chacune des entités B-LCSE entrante et sortante. Les états correspondant à l'entité B-LCSE sortante sont:

Etat 0: RELEASED (libre)

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

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

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

Etat 2: ESTABLISHED (voie logique établie)

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

Etat 3: AWAITING RELEASE (attente de libération)

L'entité B-LCSE sortante attend de libérer une voie logique avec l'entité B-LCSE entrante homologue. La voie logique ne devra pas être utilisée pour envoyer des données, mais des données peuvent toujours être reçues.

Les états correspondant à l'entité B-LCSE entrante sont:

Etat 0: RELEASED (libre)

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

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

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

Etat 2: AWAITING CONFIRMATION (attente de confirmation)

L'entité B-LCSE entrante attend la confirmation que la voie logique est établie avec une entité B-LCSE sortante homologue. La voie logique ne devra pas être utilisée pour envoyer des données, mais des données peuvent être reçues.

Etat 3: ESTABLISHED (voie logique établie)

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

C.5.2.5 Diagramme de changement d'état

La séquence autorisée de primitives entre l'entité B-LCSE et l'utilisateur de l'entité B-LCSE est définie dans le présent paragraphe. La séquence autorisée des primitives fait référence à des états de l'entité B-LCSE du point de vue de l'utilisateur de l'entité B-LCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités B-LCSE sortante et B-LCSE entrante, comme cela est indiqué à la Figure C.15 et à la Figure C.16 respectivement.

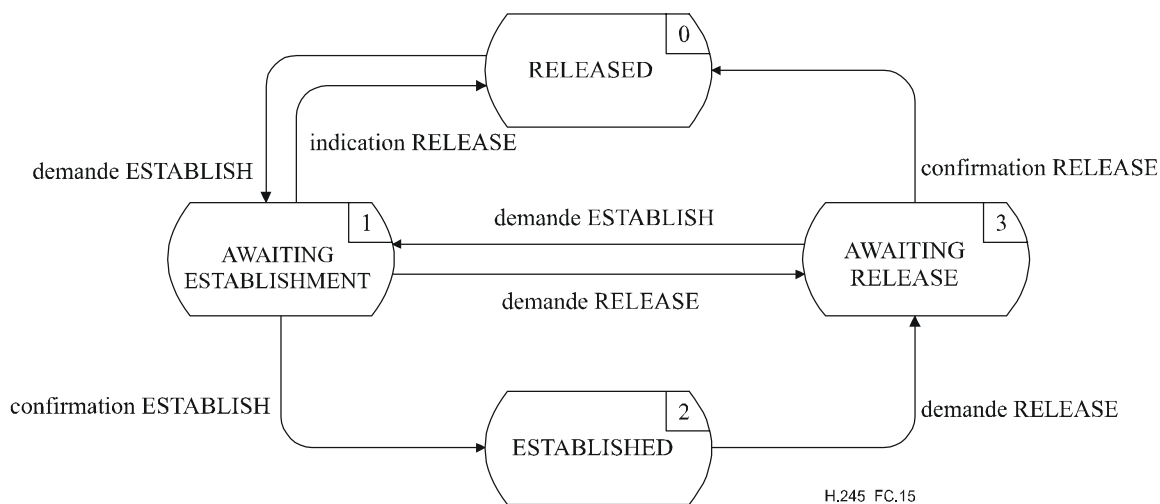


Figure C.15/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité B-LCSE sortante

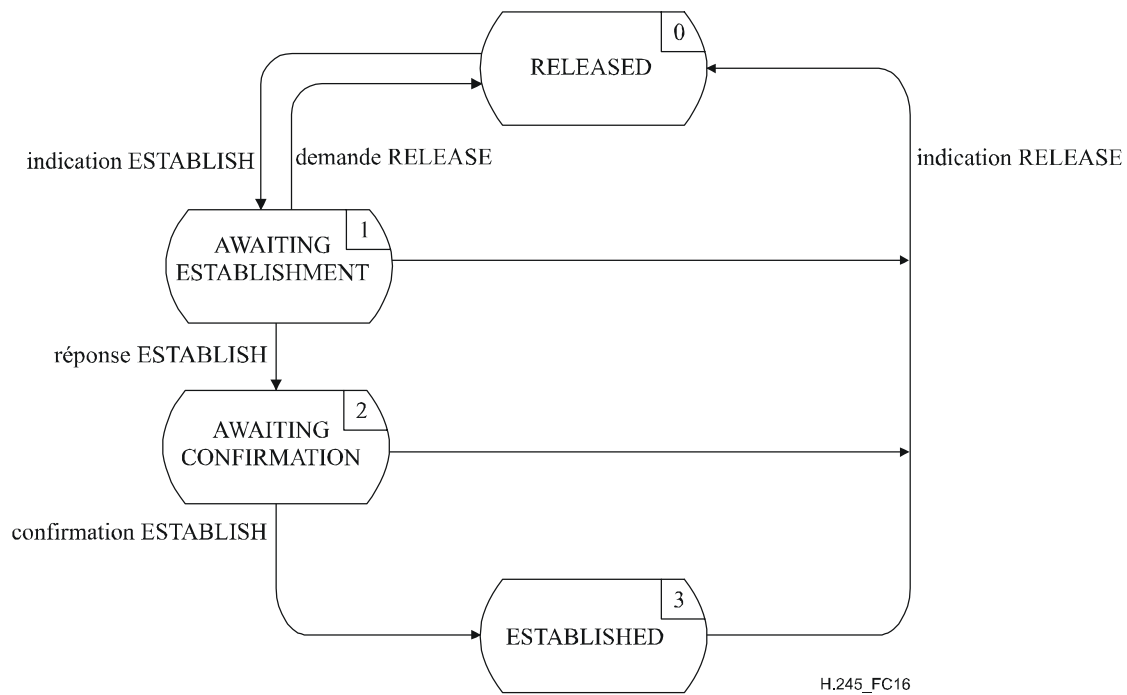


Figure C.16/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité B-LCSE entrante

C.5.3 Communication entre les entités B-LCSE homologues

C.5.3.1 Messages d'entité B-LCSE

Le Tableau C.16 indique les messages et champs des entités B-LCSE, définis dans l'Annexe A, qui concernent le protocole des entités B-LCSE.

Tableau C.16/H.245 – Noms et champs de message des entités B-LCSE

Fonction	Message	Sens	Champ
établissement	OpenLogicalChannel	S → E (Note)	forwardLogicalChannelNumber forwardLogicalChannelParameters reverseLogicalChannelParameters
	OpenLogicalChannelAck	S ← E	forwardLogicalChannelNumber reverseLogicalChannelParameters
	OpenLogicalChannelReject	S ← E	forwardLogicalChannelNumber cause
	OpenLogicalChannelConfirm	S → E	forwardLogicalChannelNumber
libération	CloseLogicalChannel	S → E	forwardLogicalChannelNumbersource
	CloseLogicalChannelAck	S ← E	forwardLogicalChannelNumber
NOTE – Sens: S – sortant, E – entrant.			

C.5.3.2 Variables d'état d'entité B-LCSE

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

out_LCN

Cette variable d'état distingue entre elles les entités B-LCSE sortantes. Elle est initialisée pendant l'initialisation de l'entité B-LCSE sortante. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés à partir d'une entité B-LCSE sortante. En ce qui concerne les messages reçus dans une entité B-LCSE sortante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de out_LCN.

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

in_LCN

Cette variable d'état distingue entre elles les entités B-LCSE entrantes. Elle est initialisée lors de l'initialisation de l'entité B-LCSE entrante. La valeur de in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés à partir d'une entité B-LCSE entrante. En ce qui concerne les messages B-LCSE reçus dans l'entité B-LCSE entrante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de in_LCN.

C.5.3.3 Temporisateurs d'entité B-LCSE

Le temporisateur suivant est spécifié pour l'entité B-LCSE entrante et l'entité B-LCSE sortante:

T103

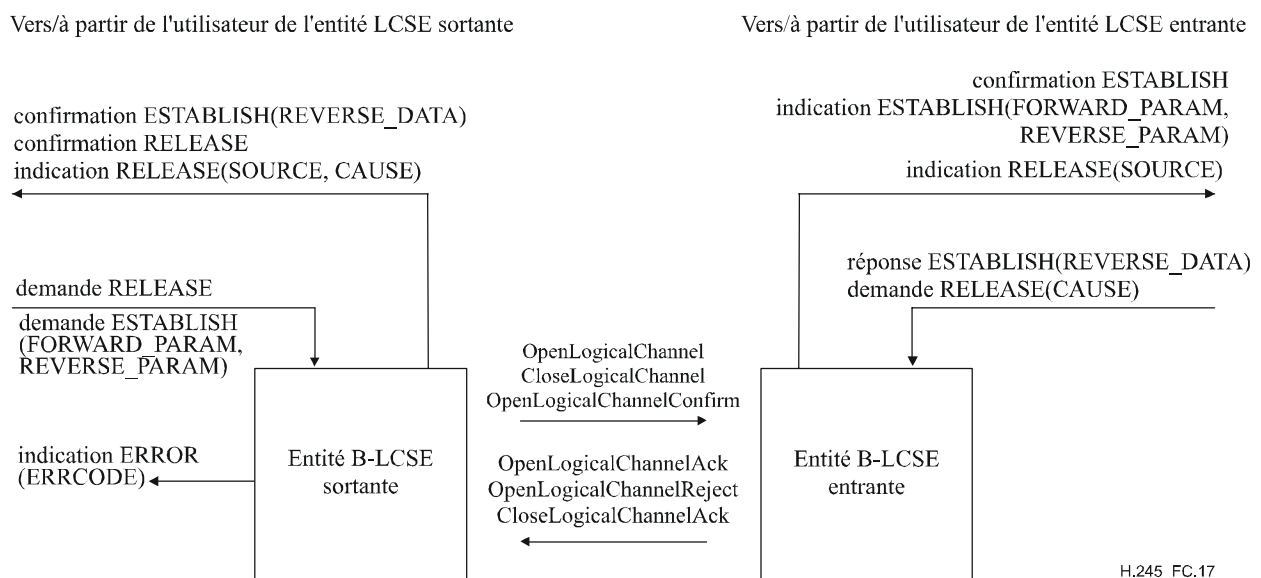
Dans l'entité B-LCSE sortante, ce temporisateur est utilisé pendant les états AWAITING ESTABLISHMENT (attente d'établissement) et AWAITING RELEASE (attente de libération). Il spécifie le temps maximal pendant lequel aucun des messages OpenLogicalChannelAck, OpenLogicalChannelReject et CloseLogicalChannelAck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

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

C.5.4 Procédures d'entité B-LCSE

C.5.4.1 Introduction

La Figure C.17 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités B-LCSE entrante et sortante.



H.245_FC.17

Figure C.17/H.245 – Primitives et messages dans l'entité de signalisation de voie logique bidirectionnelle

C.5.4.2 Valeurs par défaut des paramètres des primitives

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les paramètres des primitives d'indication et de confirmation prennent les valeurs indiquées dans le Tableau C.17.

Tableau C.17/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut (Note)
indication ESTABLISH	FORWARD_PARAM	OpenLogicalChannel.forwardLogicalChannelParameters
	REVERSE_PARAM	OpenLogicalChannel.reverseLogicalChannelParameters
confirmation ESTABLISH	REVERSE_DATA	OpenLogicalChannelAck.reverseLogicalChannelParameters
indication RELEASE	SOURCE	CloseLogicalChannel.source
	CAUSE	null
NOTE – Un paramètre de primitive devra être codé comme valant "null", si un champ de message indiqué n'est pas présent dans le message.		

C.5.4.3 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement indiqué dans les diagrammes SDL, les champs de messages prennent les valeurs indiquées dans le Tableau C.18.

Tableau C.18/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 devra pas être codé si le paramètre de primitive correspondant vaut "null", c'est-à-dire est absent.		

C.5.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition particulière d'erreur. Le Tableau C.19 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE sortante et le Tableau C.20 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité B-LCSE entrante.

Tableau C.19/H.245 – Valeurs du paramètre ERRCODE dans l'entité B-LCSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
Message inapproprié	A	OpenLogicalChannelAck	RELEASED
	B	OpenLogicalChannelReject	RELEASED ESTABLISHED
	C	CloseLogicalChannelAck	ESTABLISHED
Pas de réponse provenant de l'entité B-LCSE homologue	D	Expiration de la temporisation T103	AWAITING ESTABLISHMENT AWAITING RELEASE

Tableau C.20/H.245 – Valeurs du paramètre ERRCODE dans l'entité B-LCSE entrante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
Message inapproprié	E	OpenLogicalChannelConfirm	AWAITING ESTABLISHMENT
Pas de réponse provenant de l'entité B-LCSE homologue	F	Expiration du temporisateur T103	AWAITING CONFIRMATION

C.5.4.5 Diagramme SDL

Les procédures de l'entité B-LCSE sortante et de l'entité B-LCSE entrante sont décrites sous forme de diagrammes SDL dans les Figures C.18 et C.19 respectivement.

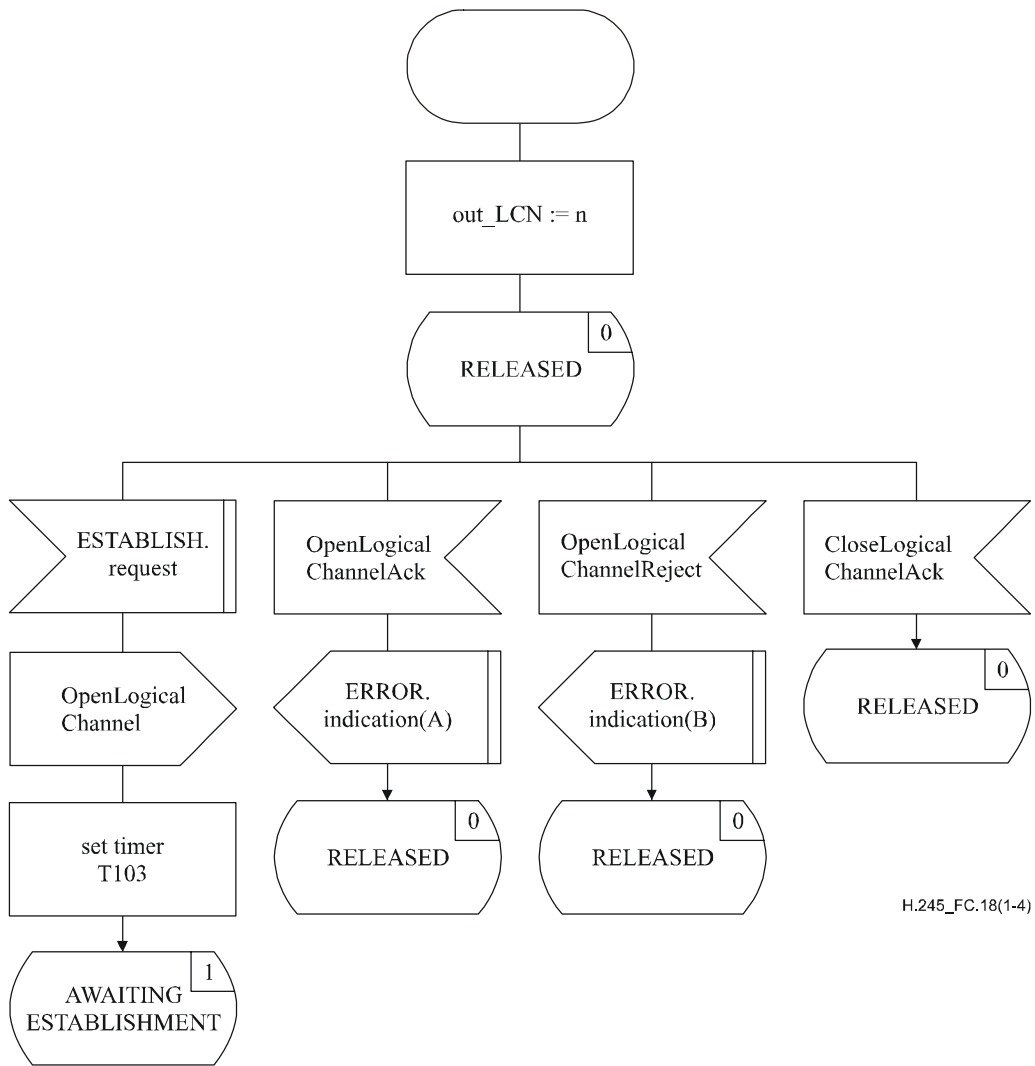


Figure C.18/H.245 – Diagramme SDL de l'entité B-LCSE sortante (feuille 1 de 4)

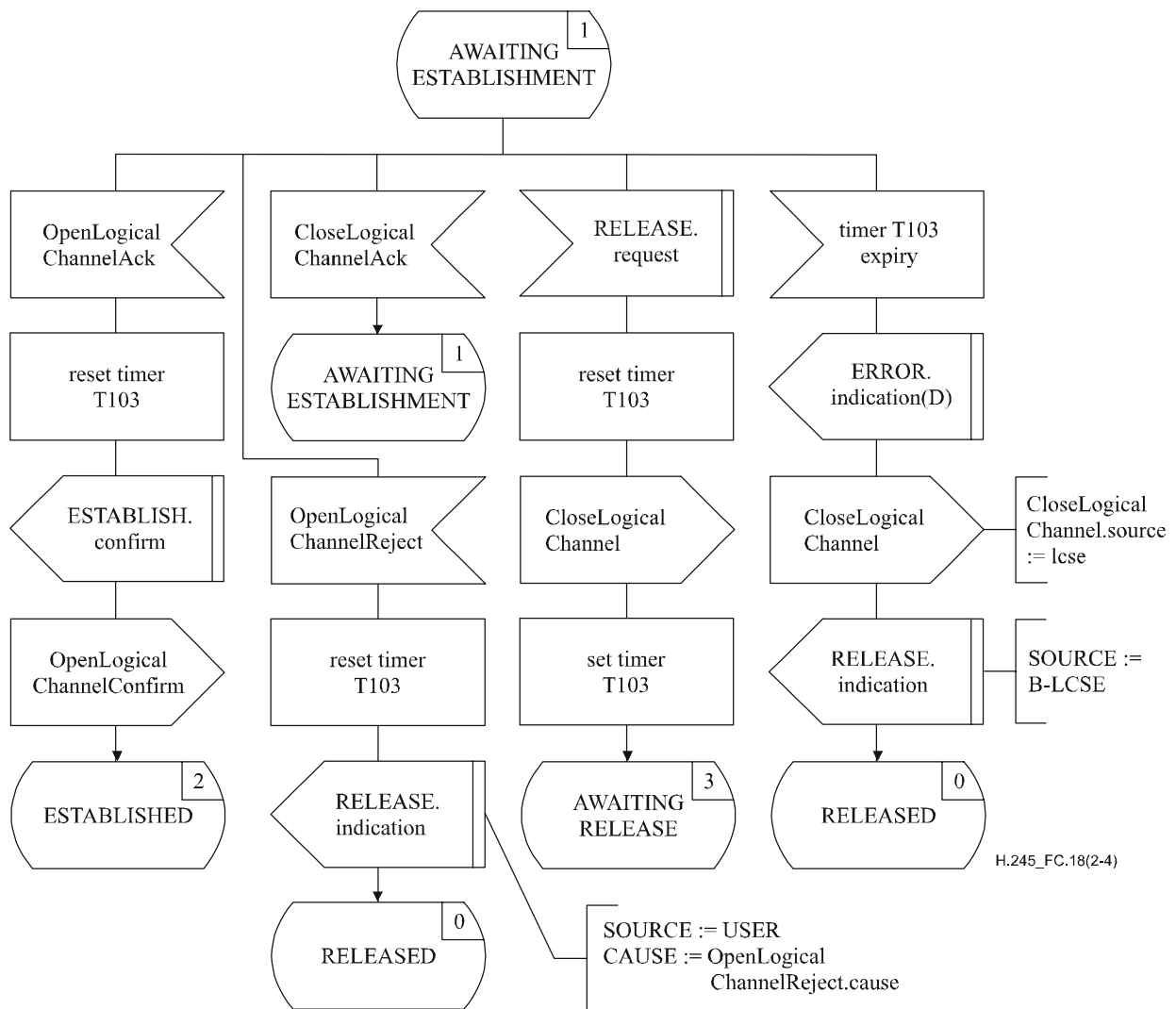


Figure C.18/H.245 – Diagramme SDL de l'entité B-LCSE sortante (feuille 2 de 4)

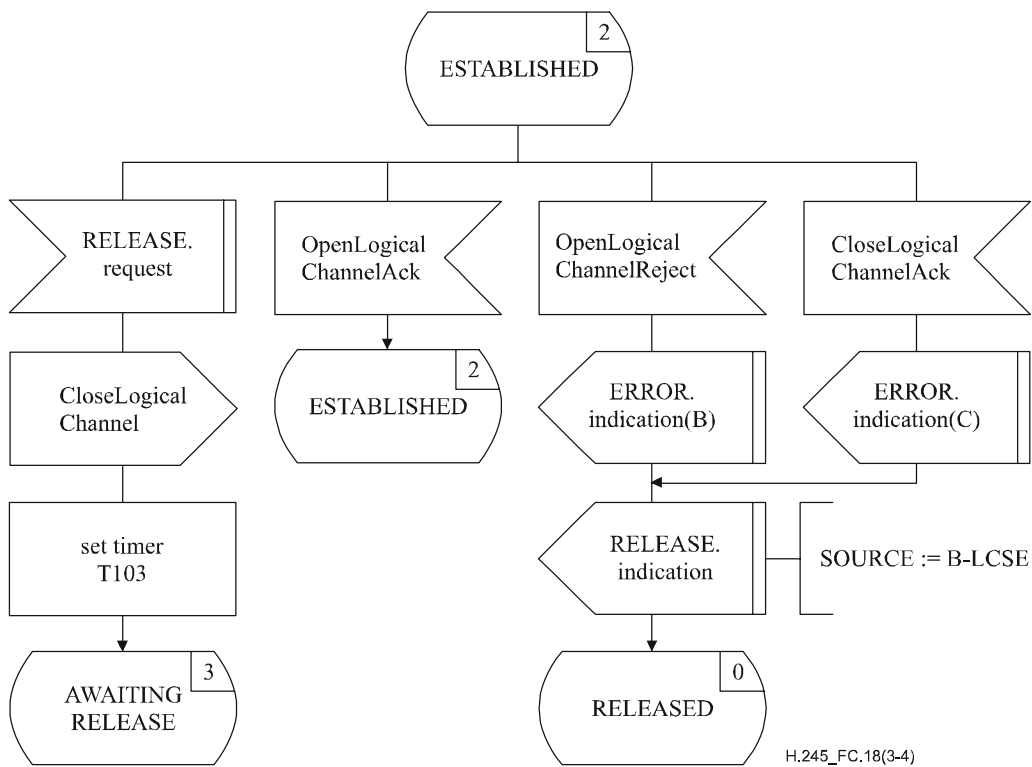


Figure C.18/H.245 – Diagramme SDL de l'entité B-LCSE sortante (feuille 3 de 4)

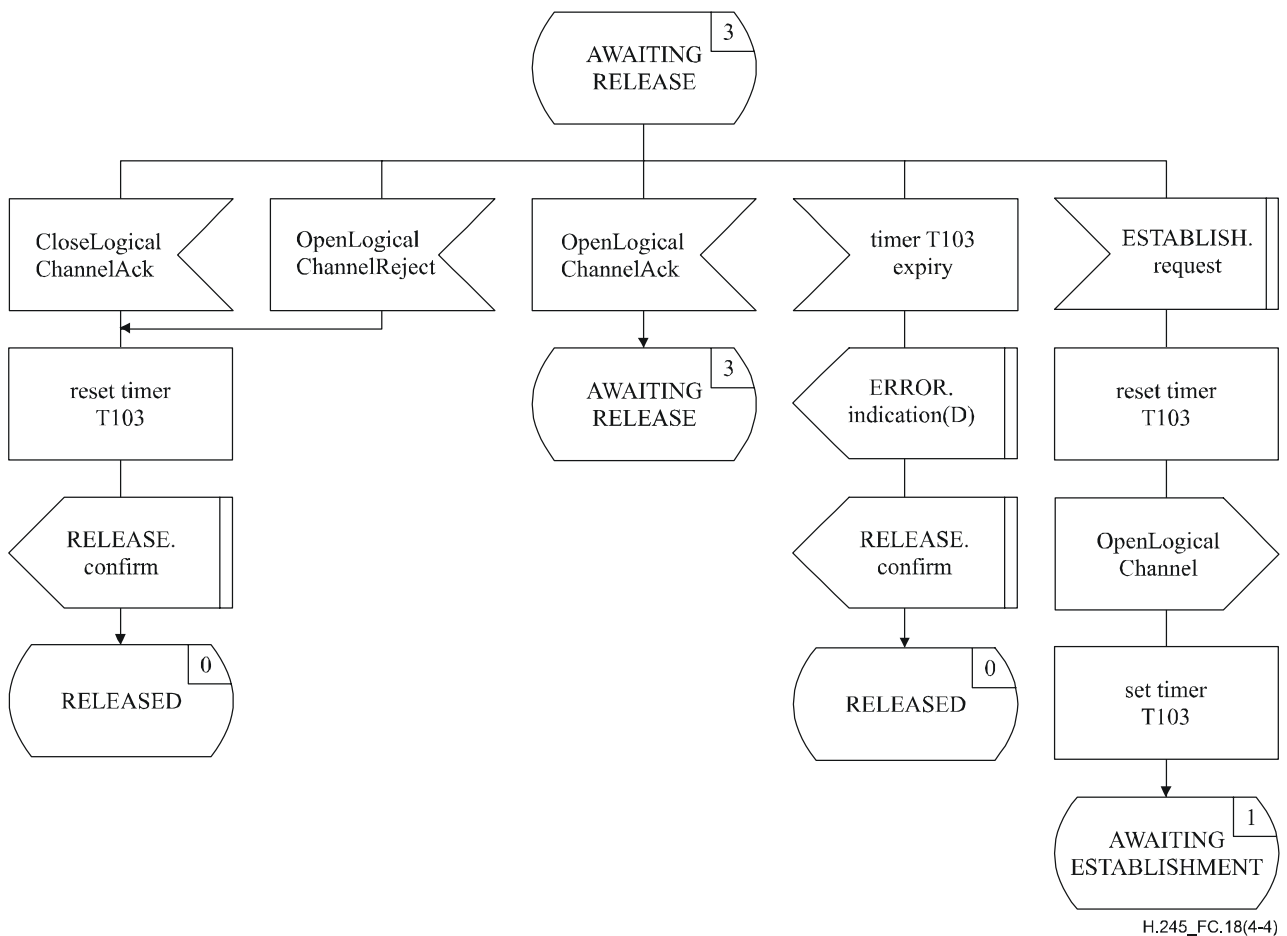


Figure C.18/H.245 – Diagramme SDL de l'entité B-LCSE sortante (feuille 4 de 4)

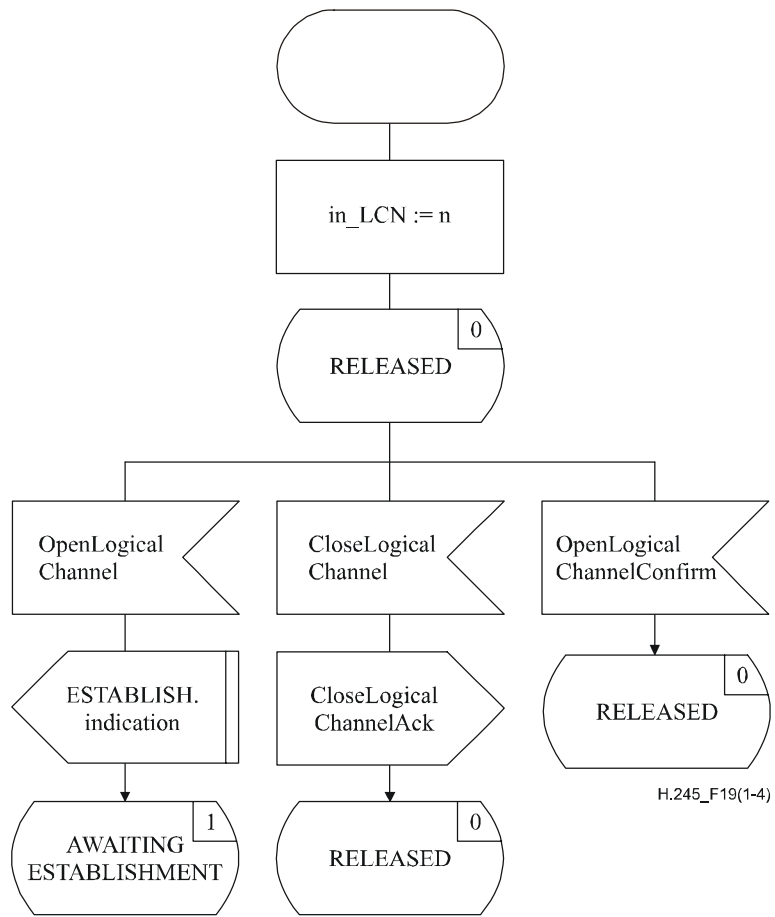
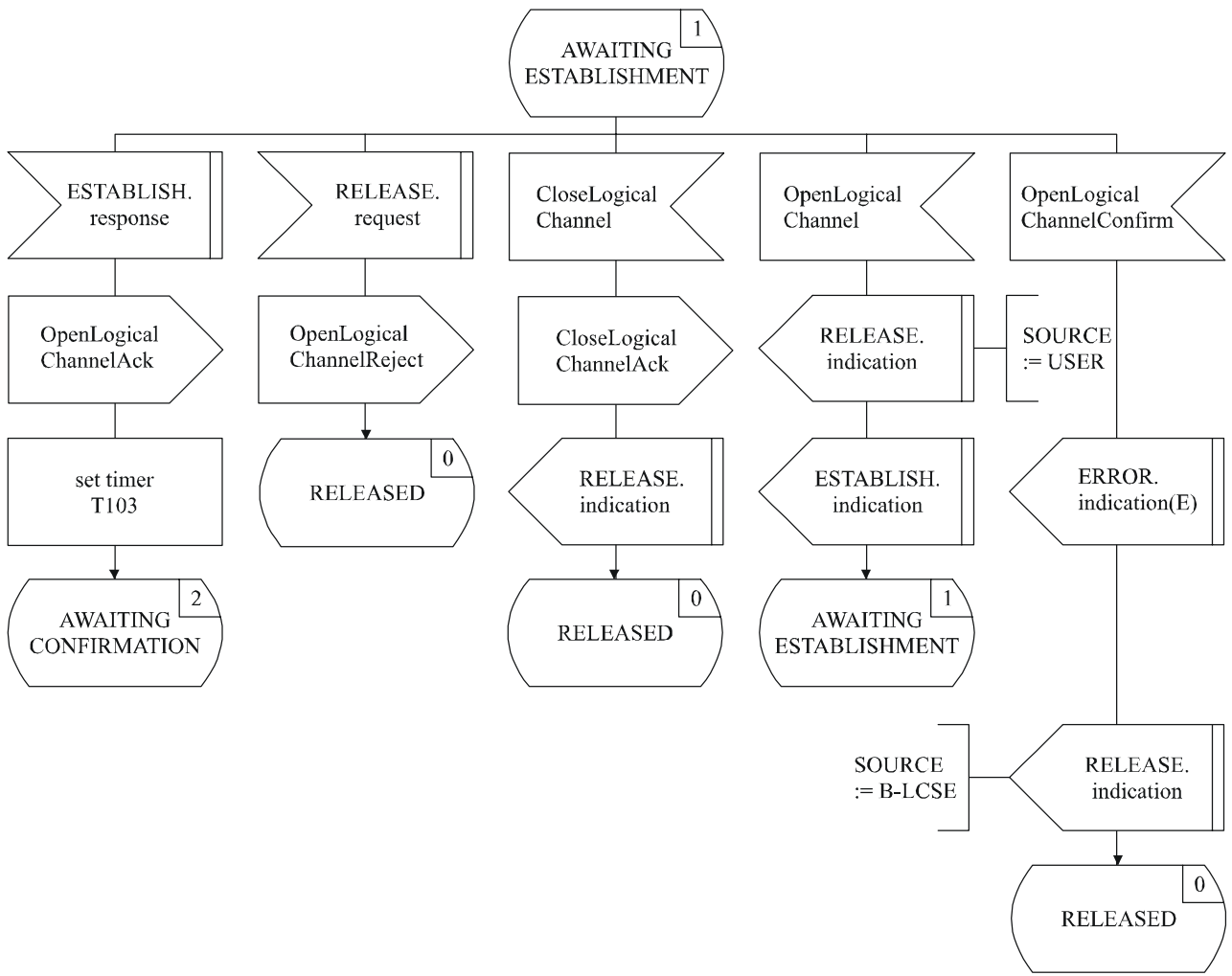


Figure C.19/H.245 – Diagramme SDL de l'entité B-LCSE entrante (feuille 1 de 4)



H.245_FC.19(2-4)

Figure C.19/H.245 – Diagramme SDL de l'entité B-LCSE entrante (feuille 2 de 4)

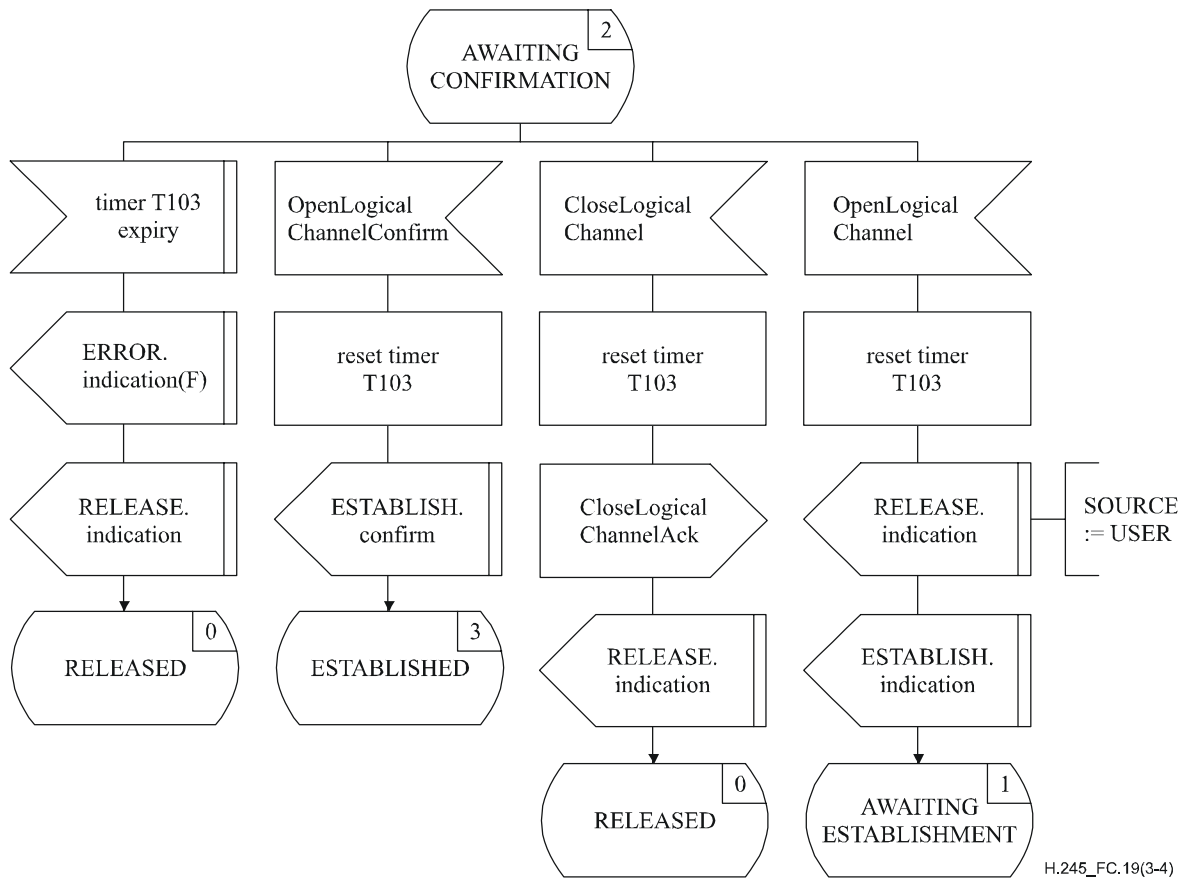


Figure C.19/H.245 – Diagramme SDL de l'entité B-LCSE entrante (feuille 3 de 4)

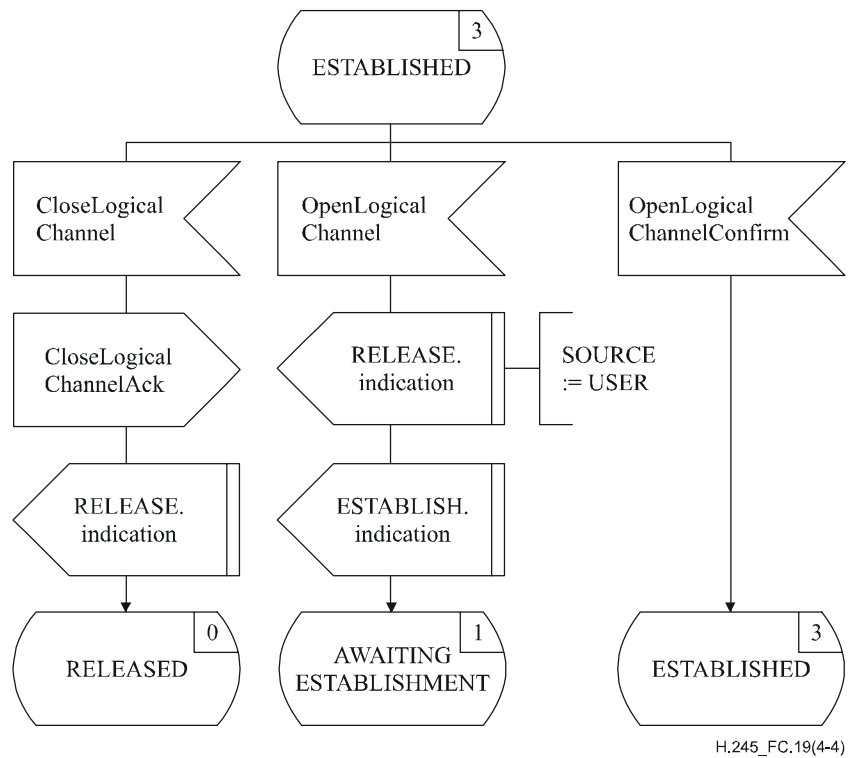


Figure C.19/H.245 – Diagramme SDL de l'entité B-LCSE entrante (feuille 4 de 4)

C.6 Procédures de fermeture de voie logique

C.6.1 Introduction

Ces procédures sont utilisées par un terminal pour demander à un terminal distant de fermer une voie logique. Il convient de noter que ces procédures sont uniquement des procédures de demande de fermeture; la voie logique est effectivement fermée en utilisant les procédures d'entité LCSE et d'entité B-LCSE. Ces procédures sont désignées ici comme l'entité de signalisation de fermeture de voie logique (CLCSE, *close logical channel signalling entity*). Elles sont spécifiées en termes de primitives et d'états à l'interface entre l'entité CLCSE et l'utilisateur de l'entité CLCSE. Les informations de protocole sont transférées vers l'entité CLCSE homologue au moyen des messages appropriés définis dans l'Annexe A. Il y a une entité CLCSE sortante et une entité CLCSE entrante. A chacune des extrémités entrante et sortante, il y a une instance d'entité CLCSE pour chaque voie logique.

Si un terminal ne peut pas traiter les signaux entrants, il peut utiliser ces procédures pour demander la fermeture des voies logiques correspondantes.

Un terminal qui répond de façon positive, c'est-à-dire en présentant une primitive de réponse CLOSE, devra déclencher la fermeture de la voie logique en envoyant la primitive de demande RELEASE à l'entité LCSE ou B-LCSE appropriée dès que cela est possible.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

C.6.1.1 Aperçu général du protocole – Entité CLCSE sortante

Une procédure de demande de fermeture de voie logique est déclenchée quand la primitive de demande CLOSE est présentée par l'utilisateur de l'entité CLCSE sortante. Un message RequestChannelClose est envoyé à l'entité CLCSE entrante homologue et le temporisateur T108 est déclenché. Si un message RequestChannelCloseAck est reçu en réponse au message RequestChannelClose, le temporisateur T108 est arrêté et l'utilisateur est informé, par une primitive de confirmation CLOSE, de l'aboutissement de la procédure de demande de fermeture de voie logique. Si, cependant, un 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 la fermeture de la voie logique.

Si la temporisation T108 arrive à expiration, l'utilisateur de l'entité CLCSE sortante en est informé par la primitive d'indication REJECT et un message RequestChannelCloseRelease est envoyé.

C.6.1.2 Aperçu général du protocole – Entité CLCSE entrante

Quand un message RequestChannelClose est reçu dans l'entité CLCSE entrante, l'utilisateur est informé de la demande de fermeture de la voie logique par la primitive d'indication CLOSE. L'utilisateur de l'entité CLCSE entrante signale l'acceptation de demande de fermeture de la voie logique en présentant une primitive de réponse CLOSE, et un message RequestChannelCloseAck est envoyé à l'entité CLCSE sortante homologue. L'utilisateur de l'entité CLCSE entrante signale le refus de la demande de voie logique en présentant la primitive de demande REJECT, et un message RequestChannelCloseReject est envoyé à l'entité CLCSE sortante homologue.

C.6.2 Communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE

C.6.2.1 Primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE

La communication entre l'entité CLCSE et l'utilisateur de l'entité CLCSE s'effectue en utilisant les primitives indiquées dans le Tableau C.21.

Tableau C.21/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
CLOSE	– (Note 1)	–	–	–
REJECT	CAUSE	SOURCE CAUSE	Non défini (Note 2)	Non défini
NOTE 1 – "–" signifie l'absence de paramètres.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

C.6.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives CLOSE sont utilisées pour demander la fermeture d'une voie logique;
- b) les primitives REJECT sont utilisées pour refuser la fermeture d'une voie logique.

C.6.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.21 est la suivante:

- a) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE prend la valeur de "USER" ou "PROTOCOL". Ce dernier cas peut résulter d'une fin de temporisation;
- b) le paramètre CAUSE indique le motif de refus de la fermeture d'une voie logique. Le paramètre CAUSE est absent quand le paramètre SOURCE indique "PROTOCOL".

C.6.2.4 Etats d'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 correspondant à une entité CLCSE sortante sont:

Etat 0: IDLE

L'entité CLCSE est au repos.

Etat 1: AWAITING RESPONSE

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

Les états correspondant à une entité CLCSE entrante sont:

Etat 0: IDLE

L'entité CLCSE est au repos.

Etat 1: AWAITING RESPONSE

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

C.6.2.5 Diagramme de changement d'état

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité CLCSE et l'utilisateur de l'entité CLCSE. Les séquences autorisées sont spécifiées de façon distincte pour chacune des entités CLCSE entrante et sortante, comme cela est indiqué aux Figures C.20 et C.21 respectivement.

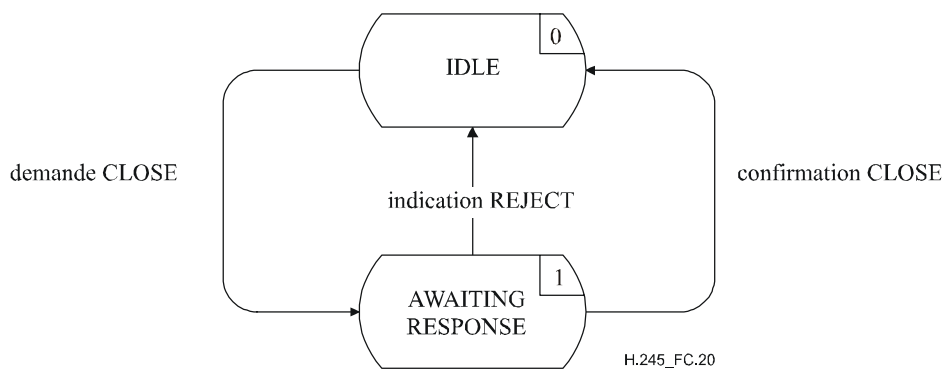


Figure C.20/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité CLCSE sortante

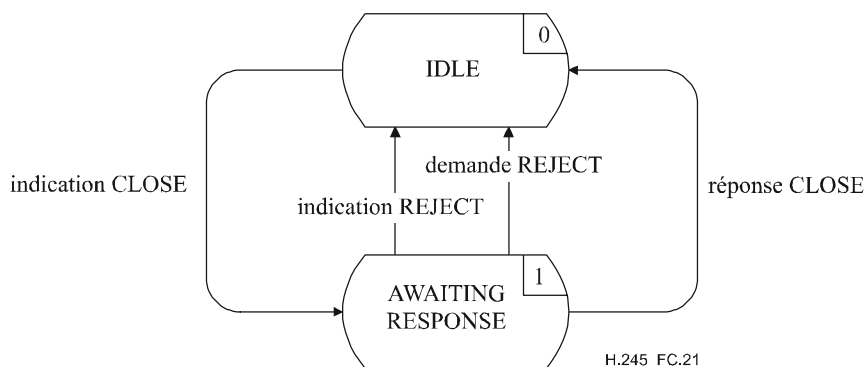


Figure C.21/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité CLCSE entrante

C.6.3 Communication entre les entités CLCSE homologues

C.6.3.1 Messages

Le Tableau C.22 indique les messages et les champs des entités CLCSE, définis dans l'Annexe A, qui se rapportent au protocole d'entité CLCSE.

Tableau C.22/H.245 – Noms et champs des messages des entités CLCSE

Fonction	Message	Sens	Champ
transfert	RequestChannelClose	S → E (Note)	forwardLogicalChannelNumber
	RequestChannelCloseAck	S ← E	forwardLogicalChannelNumber
	RequestChannelCloseReject	S ← E	forwardLogicalChannelNumber
réinitialisation	RequestChannelCloseRelease	S → E	forwardLogicalChannelNumber
NOTE – Sens: S – sortant, E – entrant.			

C.6.3.2 Variables d'état d'entité CLCSE

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

out_LCN

Cette variable d'état établit une distinction entre les entités CLCSE sortantes. Elle est initialisée lors de l'initialisation des entités CLCSE sortantes. La valeur de out_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés à partir d'une entité CLCSE sortante. En ce qui concerne les messages reçus dans l'entité CLCSE sortante, la valeur du champ forwardLogicalChannelNumber des messages est identique à celle de la valeur de out_LCN.

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

in_LCN

Cette variable d'état établit une distinction entre les entités CLCSE entrantes. Elle est initialisée lors de l'initialisation de l'entité CLCSE entrante. La valeur de in_LCN est utilisée pour définir le champ forwardLogicalChannelNumber des messages envoyés à partir d'une entité CLCSE entrante. En ce qui concerne les messages reçus dans l'entité CLCSE entrante, la valeur du champ forwardLogicalChannelNumber des messages est identique à la valeur de in_LCN.

C.6.3.3 Temporiseurs d'entité CLCSE

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

T108

Ce temporisateur est utilisé dans l'état AWAITING RESPONSE (attente de réponse). Il spécifie le temps maximal pendant lequel aucun des messages RequestChannelCloseAck et RequestChannelCloseReject ne peut être reçu.

C.6.4 Procédures d'entité CLCSE

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

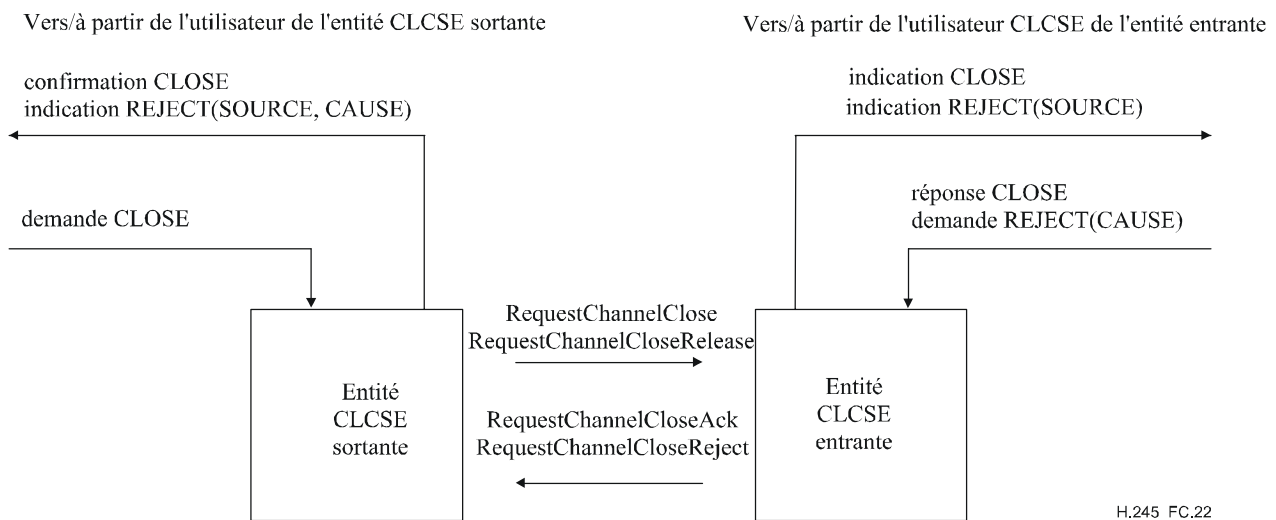


Figure C.22/H.245 – Primitives et messages dans l'entité de signalisation de fermeture de voie logique (CLCSE)

C.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 les valeurs indiquées dans le Tableau C.23.

Tableau C.23/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
indication REJECT	SOURCE CAUSE	USER null

C.6.4.2 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.24.

Tableau C.24/H.245 – Valeurs par défaut des champs des messages

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

C.6.4.3 Diagrammes SDL

Les procédures de l'entité CLCSE sortante et de l'entité CLCSE entrante sont décrites sous forme de diagrammes SDL aux Figures C.23 et C.24 respectivement.

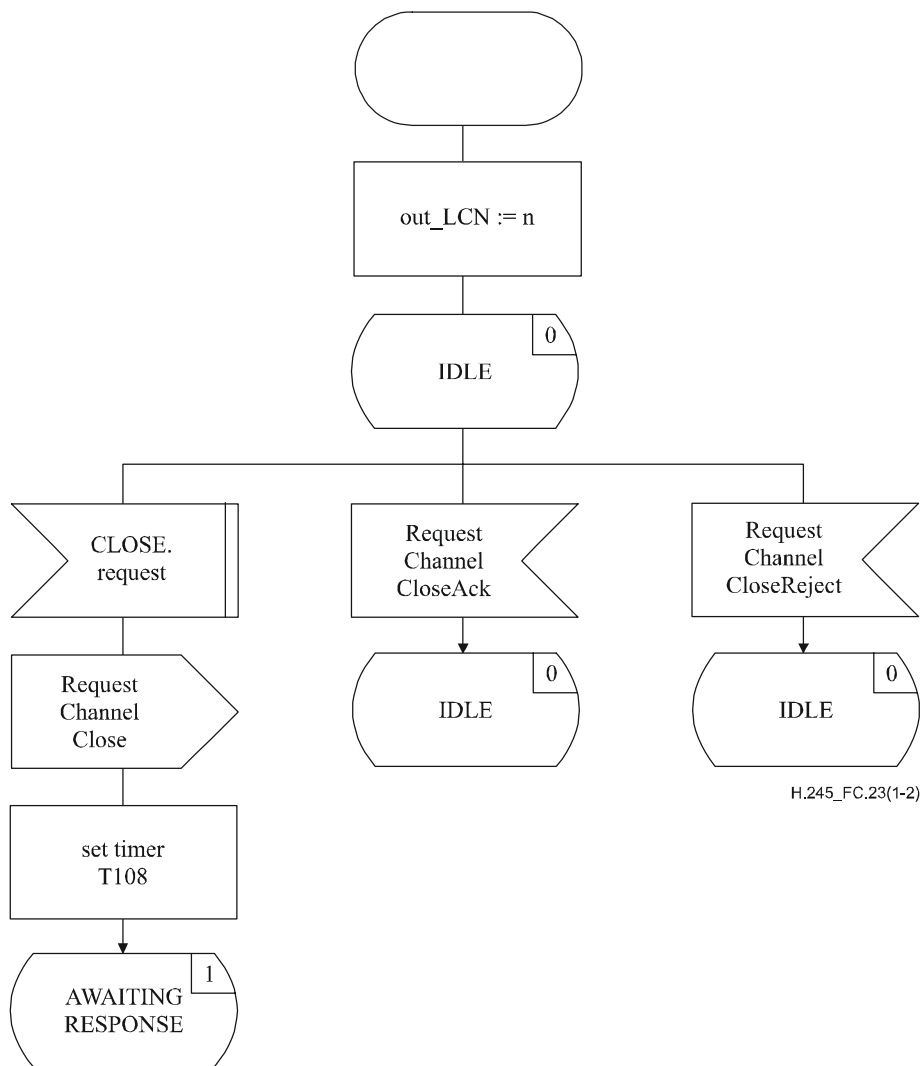


Figure C.23/H.245 – Diagramme SDL de l'entité CLCSE sortante (feuille 1 de 2)

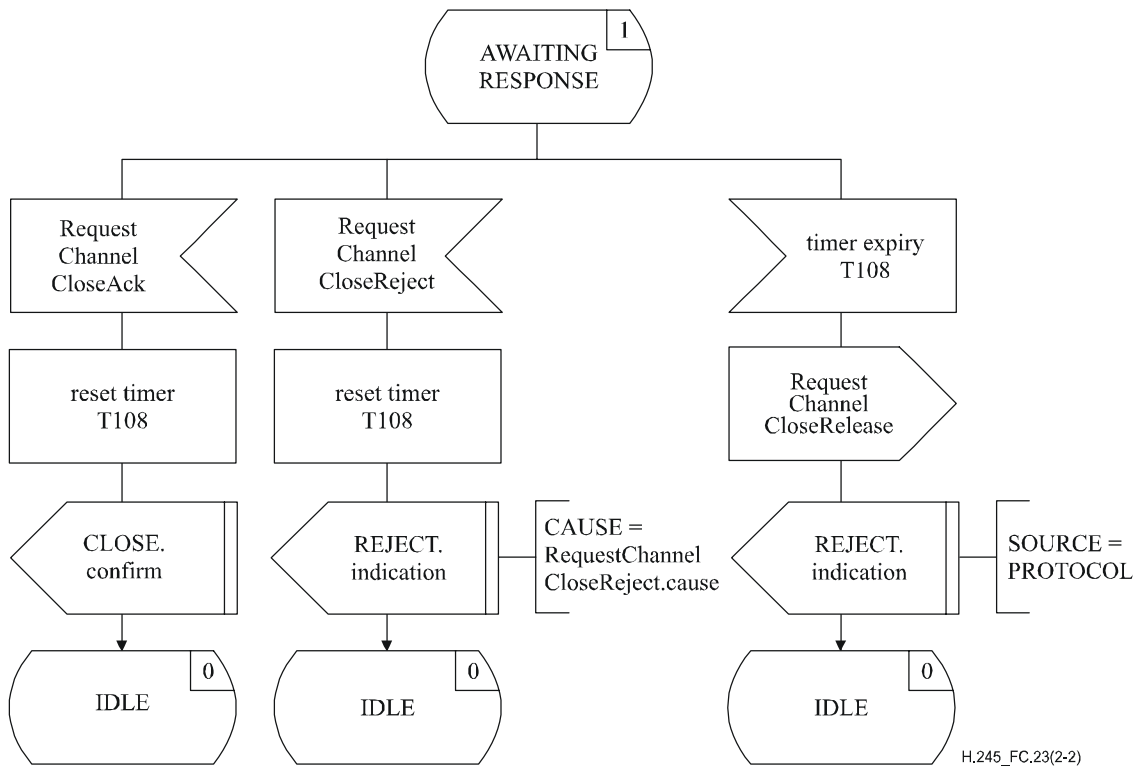


Figure C.23/H.245 – Diagramme SDL de l'entité CLCSE sortante (feuille 2 de 2)

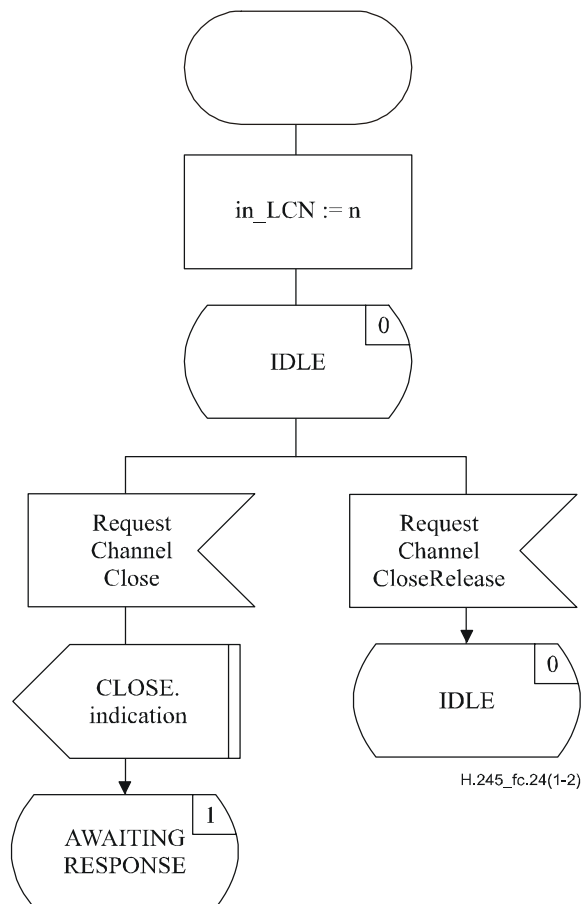


Figure C.24/H.245 – Diagramme SDL de l'entité CLCSE entrante (feuille 1 de 2)

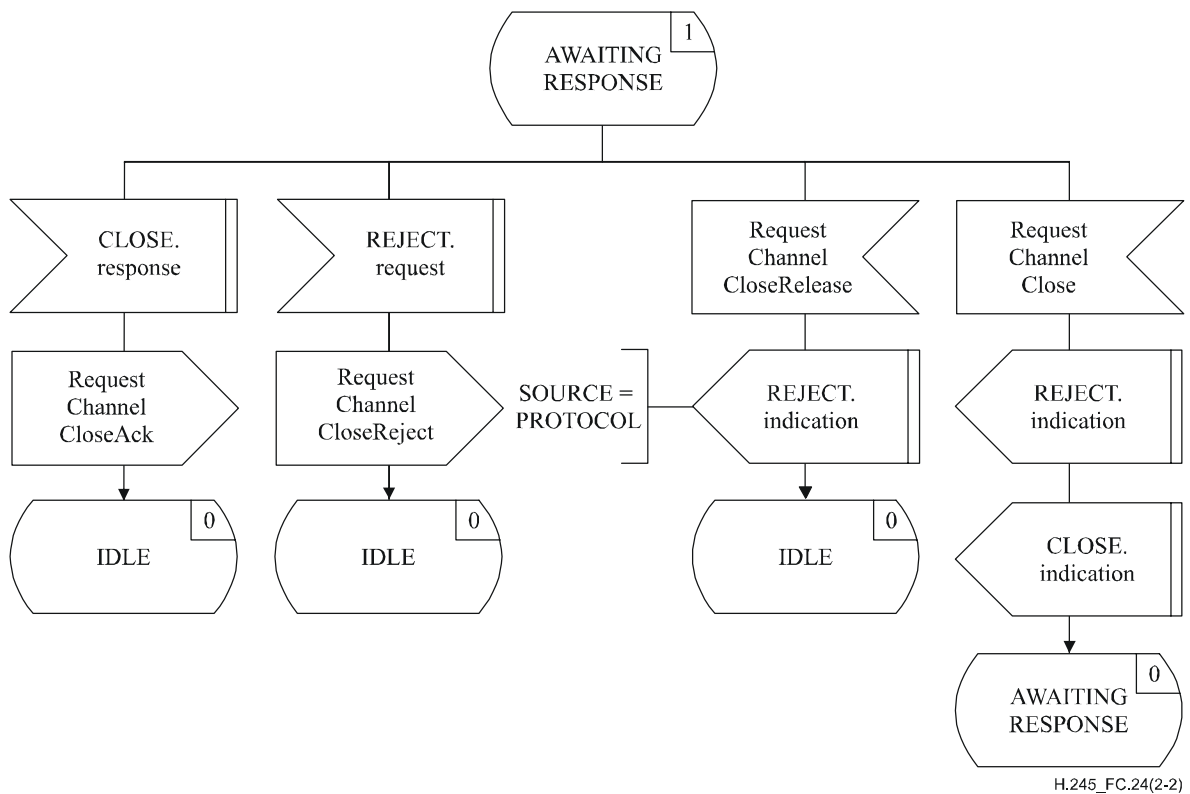


Figure C.24/H.245 – Diagramme SDL de l'entité CLCSE entrante (feuille 2 de 2)

C.7 Procédures relatives au tableau de multiplexage H.223

C.7.1 Introduction

Le tableau de multiplexage permet d'associer chaque octet d'une unité MUX-PDU selon la Rec. UIT-T H.223 [10] à un numéro de voie logique donné. Le tableau de multiplexage selon la Rec. UIT-T H.223 peut contenir jusqu'à 16 entrées, numérotées de 0 à 15. Les entrées de tableau comprises entre 1 et 15 devront être envoyées par les émetteurs aux récepteurs comme cela est spécifié dans les procédures suivantes.

Les procédures décrites ici sont désignées comme l'entité de signalisation du tableau de multiplexage (MTSE, *multiplex table signalling entity*). Elles sont spécifiées en termes de primitives et d'états à l'interface entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les informations de protocole sont transmises à l'entité MTSE homologue au moyen des messages appropriés définis dans l'Annexe A.

On distingue une entité MTSE sortante et une entité MTSE entrante. Il existe une instance d'entité MTSE pour chaque entrée du tableau de multiplexage.

Un terminal émetteur utilise ce protocole pour indiquer à un terminal distant une ou plusieurs nouvelles entrées du tableau de multiplexage. Le terminal distant peut accepter ou refuser ces nouvelles entrées. Si le terminal distant accepte une entrée du tableau de multiplexage, l'entrée précédente ayant un 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 elementList. L'émetteur ne devra aucunement utiliser une entrée de tableau de multiplexage qui est désactivée. Avant d'envoyer un message MultiplexEntrySend, l'émetteur devra cesser d'utiliser les entrées qui sont décrites. Il ne devra pas recommencer à utiliser ces entrées avant d'avoir reçu un message MultiplexEntrySendAck. Cette procédure est utilisée

parce que si l'utilisation des entrées du tableau de multiplexage ne cesse pas avant l'envoi du message MultiplexEntrySend, les erreurs peuvent susciter une ambiguïté dans le récepteur.

L'émetteur devra cesser d'utiliser des entrées désactivées avant d'envoyer le message MultiplexEntrySend indiquant que ces entrées ont été désactivées. Des entrées désactivées peuvent être de nouveau utilisées à tout moment en envoyant le message MultiplexEntrySend en vue d'activer cette entrée. La désactivation des entrées dont l'émetteur n'a plus besoin peut augmenter la probabilité de détecter des erreurs dans le champ des codes de multiplexage H.223.

NOTE – Alors que certaines entrées du tableau de multiplexage sont mises à jour, il est possible de continuer à utiliser les autres entrées (actives). En outre, une entrée de tableau de multiplexage peut être effacée dans le message MultiplexEntrySend qui est utilisé pour modifier d'autres entrées du tableau de multiplexage.

Au début de la communication, sauf spécification contraire dans une Recommandation appropriée, seule l'entrée de tableau 0 est disponible pour la transmission, les entrées 1 à 15 étant désactivées.

Une procédure de demande d'entrée de multiplexage peut être utilisée à tout moment pour obtenir la retransmission des entrées spécifiées dans les tableaux de multiplexage à partir du terminal distant, par exemple par suite d'une interruption ou de toute autre motif d'incertitude.

Le texte suivant donne un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

C.7.1.1 Aperçu général du protocole – Entité MTSE sortante

Une procédure de demande d'envoi d'entrée de tableau de multiplexage est déclenchée quand la primitive de demande TRANSFER est présentée par l'utilisateur de l'entité MTSE sortante. Un message MultiplexEntrySend est envoyé à l'entité MTSE entrante homologue, et le temporisateur T104 démarre. Si un message MultiplexEntrySendAck est reçu en réponse au message MultiplexEntrySend, alors le temporisateur T104 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER que la demande d'envoi de l'entrée du tableau de multiplexage a abouti. Si, cependant, 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 du tableau de multiplexage.

Si la temporisation T104 arrive à expiration, l'utilisateur de l'entité MTSE sortante en est informé par la primitive d'indication REJECT et un message MultiplexEntrySendRelease est envoyé.

Seuls les messages MultiplexEntrySendAck et MultiplexEntrySendReject répondant au message MultiplexEntrySend le plus récent sont acceptés. Les réponses à des messages MultiplexEntrySend précédents sont ignorés.

Une nouvelle procédure de demande d'envoi de l'entrée du tableau de multiplexage peut être déclenchée par la primitive de demande TRANSFER par l'utilisateur de l'entité MTSE sortante avant qu'un message MultiplexEntrySendAck ou MultiplexEntrySendReject ait été reçu.

C.7.1.2 Aperçu général du protocole – Entité MTSE entrante

Quand un message MultiplexEntrySend est reçu dans l'entité MTSE entrante, l'utilisateur est informé de la demande d'envoi de l'entrée du tableau de multiplexage par la primitive d'indication TRANSFER. L'utilisateur de l'entité MTSE entrante signale l'acceptation de l'entrée du tableau de multiplexage en présentant la primitive de réponse TRANSFER et un message MultiplexEntrySendAck est envoyé à l'entité MTSE sortante homologue. L'utilisateur de l'entité MTSE entrante signale le refus de l'entrée du tableau de multiplexage en présentant la primitive de demande REJECT et un message MultiplexEntrySendReject est envoyé à l'entité MTSE sortante homologue.

Un nouveau message MultiplexEntrySend peut être reçu avant que l'utilisateur de l'entité MTSE entrante ait répondu à un message MultiplexEntrySend précédent. L'utilisateur de l'entité MTSE

entrante est informé par la primitive d'indication REJECT, suivie de la primitive d'indication TRANSFER, et l'utilisateur de l'entité MTSE entrante répond à la nouvelle entrée du tableau de multiplexage.

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

C.7.2 Communication entre l'entité MTSE et l'utilisateur de l'entité MTSE

C.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 C.25.

Tableau C.25/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	MUX-DESCRIPTOR	MUX-DESCRIPTOR	– (Note 1)	–
REJECT	CAUSE	SOURCE CAUSE	Non défini (Note 2)	Non défini
NOTE 1 – "–" signifie aucun paramètre.				
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.				

C.7.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives TRANSFER sont utilisées pour le transfert des entrées du tableau de multiplexage;
- b) les primitives REJECT sont utilisées pour refuser une entrée du tableau de multiplexage et pour mettre fin à un transfert d'entrée de tableau de multiplexage.

C.7.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.25 est la suivante:

- a) le paramètre MUX-DESCRIPTOR est une entrée du tableau de multiplexage. Ce paramètre est mappé au champ MultiplexEntryDescriptor du message multiplexEntrySend et transmis de façon transparente de l'utilisateur de l'entité MTSE sortante vers l'utilisateur de l'entité MTSE entrante. Il peut y avoir plusieurs descripteurs MUX-DESCRIPTOR associés à la primitive TRANSFER;
- b) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE prend la valeur "USER" ou "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;
- c) le paramètre CAUSE indique le motif du refus d'une entrée de tableau de multiplexage. Le paramètre CAUSE est absent quand le paramètre SOURCE indique "PROTOCOL".

C.7.2.4 Etats d'entité MTSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les états sont spécifiés de façon distincte pour chacune des entités MTSE entrante et sortante. Les états correspondant à une entité MTSE sortante sont:

Etat 0: IDLE

Il n'y a pas de transfert de l'entité MTSE en cours. L'entrée du tableau de multiplexage peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'utilisateur de l'entité MTSE a demandé le transfert d'une entrée du tableau de multiplexage, et une réponse provenant de l'entité MTSE homologue est attendue. L'entrée du tableau de multiplexage ne devra pas être utilisée par l'émetteur.

Les états correspondant à une entité MTSE entrante sont:

Etat 0: IDLE

Il n'y a pas de transfert de l'entité MTSE en cours. L'entrée du tableau de multiplexage peut être utilisée par l'émetteur.

Etat 1: AWAITING RESPONSE

L'entité MTSE homologue a transféré une entrée du tableau de multiplexage, et une réponse provenant de l'utilisateur de l'entité MTSE est attendue. L'entrée du tableau de multiplexage peut ne pas être utilisée par l'émetteur.

C.7.2.5 Diagramme de changement d'état

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité MTSE et l'utilisateur de l'entité MTSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités MTSE sortante et entrante, comme cela est indiqué dans les Figures C.25 et C.26 respectivement.

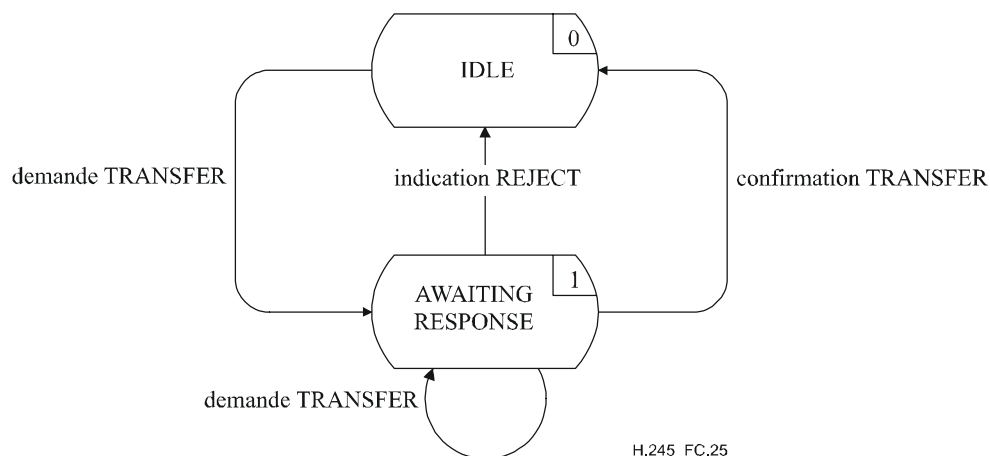


Figure C.25/H.245 – Diagramme de changement d'état pour la séquence de primitives dans l'entité MTSE sortante

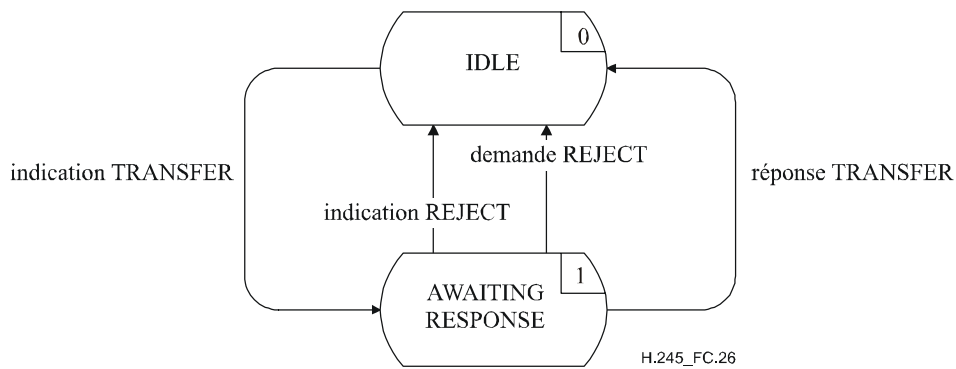


Figure C.26/H.245 – Diagramme de changement d'état pour la séquence de primitives dans l'entité MTSE entrante

C.7.3 Communication entre les entités MTSE homologues

C.7.3.1 Messages

Le Tableau C.26 indique les messages et les champs de l'entité MTSE, définis dans l'Annexe A, qui se rapportent au protocole d'entité MTSE.

Tableau C.26/H.245 – Noms et champs de message de l'entité MTSE

Fonction	Message	Sens	Champ
transfert	MultiplexEntrySend	S → E (Note)	sequenceNumber multiplexEntryDescriptors.multiplexTable EntryNumber multiplexEntryDescriptors.elementList
	MultiplexEntrySendAck	S ← E	sequenceNumber multiplexTableEntryNumber
refus	MultiplexEntrySendReject	S ← E	sequenceNumber multiplexTableEntryNumber rejectionDescriptions.cause
réinitialisation	MultiplexEntrySendRelease	S → E	multiplexTableEntryNumber

NOTE – Sens: S – sortant, E – entrant.

C.7.3.2 Variables d'état d'entité MTSE

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

out_ENUM

Cette variable d'état établit une distinction entre les entités MTSE sortantes. Elle est initialisée pendant l'initialisation de l'entité MTSE sortante. La valeur de out_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages envoyés à partir d'une entité MTSE sortante. En ce qui concerne les messages reçus dans une entité MTSE sortante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de out_ENUM;

out_SQ

Cette variable d'état est utilisée pour indiquer le message MultiplexEntrySend qui a été envoyé le plus récemment. Elle est incrémentée de un et mappée au champ sequenceNumber du message MultiplexEntrySend avant la transmission d'un message MultiplexEntrySend. L'opération arithmétique effectuée sur la variable out_SQ est modulo 256;

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

in_ENUM

Cette variable d'état est utilisée pour établir une distinction entre les entités MTSE entrantes. Elle est initialisée lors de l'initialisation de l'entité MTSE entrante. La valeur de in_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages envoyés à partir d'une entité MTSE entrante. En ce qui concerne les messages reçus dans une entité MTSE entrante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de in_ENUM;

in_SQ

Cette variable d'état est utilisée pour enregistrer la valeur du champ sequenceNumber du message MultiplexEntrySend qui a été reçu en dernier. Les messages MultiplexEntrySendAck et MultiplexEntrySendReject ont leurs champs sequenceNumber positionnés sur la valeur de in_SQ, avant d'être envoyés à l'entité MTSE homologue.

C.7.3.3 Temporisateurs d'entité MTSE

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

T104

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

C.7.4 Procédures d'entité MTSE

C.7.4.1 Introduction

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

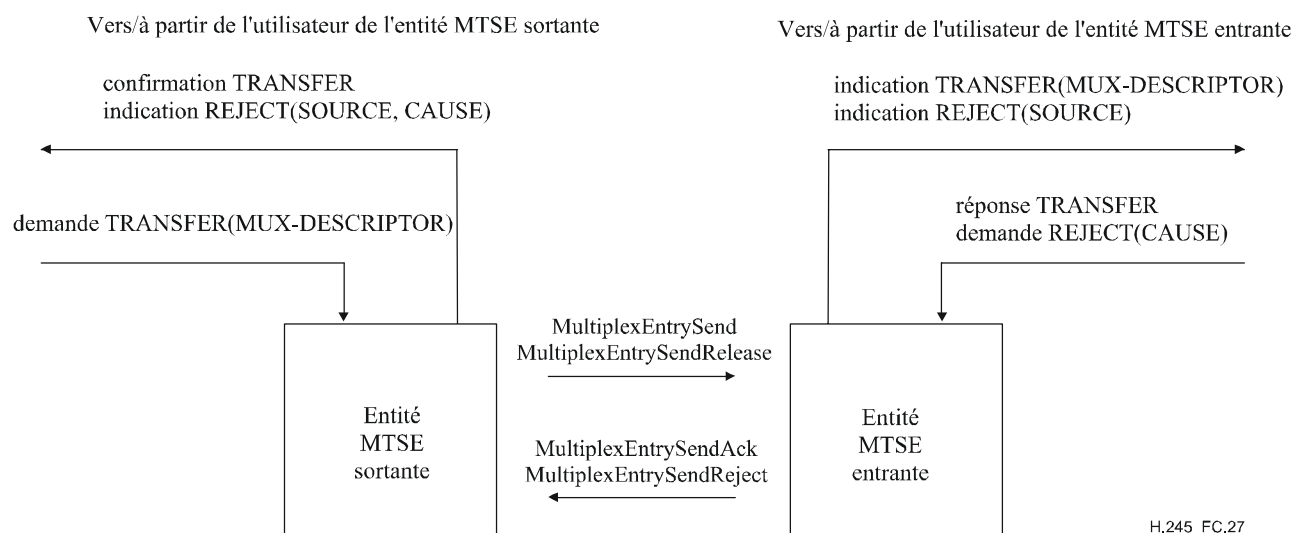


Figure C.27/H.245 – Primitives et messages dans l'entité de signalisation de tableau de multiplexage (MTSE)

C.7.4.2 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 les valeurs indiquées dans le Tableau C.27.

Tableau C.27/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
indication TRANSFER	MUX-DESCRIPTOR	MultiplexEntrySend.multiplexEntryDescriptors.elementList
indication REJECT	SOURCE CAUSE	USER null

C.7.4.3 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.28.

Tableau C.28/H.245 – Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut (Note)
MultiplexEntrySend	sequenceNumber	out_SQ
	multiplexEntryDescriptors.multiplexTableEntryNumber	out_ENUM
	multiplexEntryDescriptors.elementList	demande TRANSFER(MUX-DESCRIPTOR)
MultiplexEntrySendAck	sequenceNumber	in_SQ
	multiplexTableEntryNumber	in_ENUM
MultiplexEntrySendReject	sequenceNumber	in_SQ
	rejectionDescriptions.multiplexTableEntryNumber	in_ENUM
	rejectionDescriptions.cause	demande REJECT(CAUSE)
MultiplexEntrySendRelease	multiplexTableEntryNumber	out_ENUM
NOTE – Un champ de message ne doit pas être codé si le paramètre de primitive correspondant vaut "null", c'est-à-dire n'est pas présent.		

C.7.4.4 Diagrammes SDL

Les procédures de l'entité MTSE sortante et de l'entité MTSE entrante sont décrites sous forme de diagrammes SDL dans les Figures C.28 et C.29 respectivement.

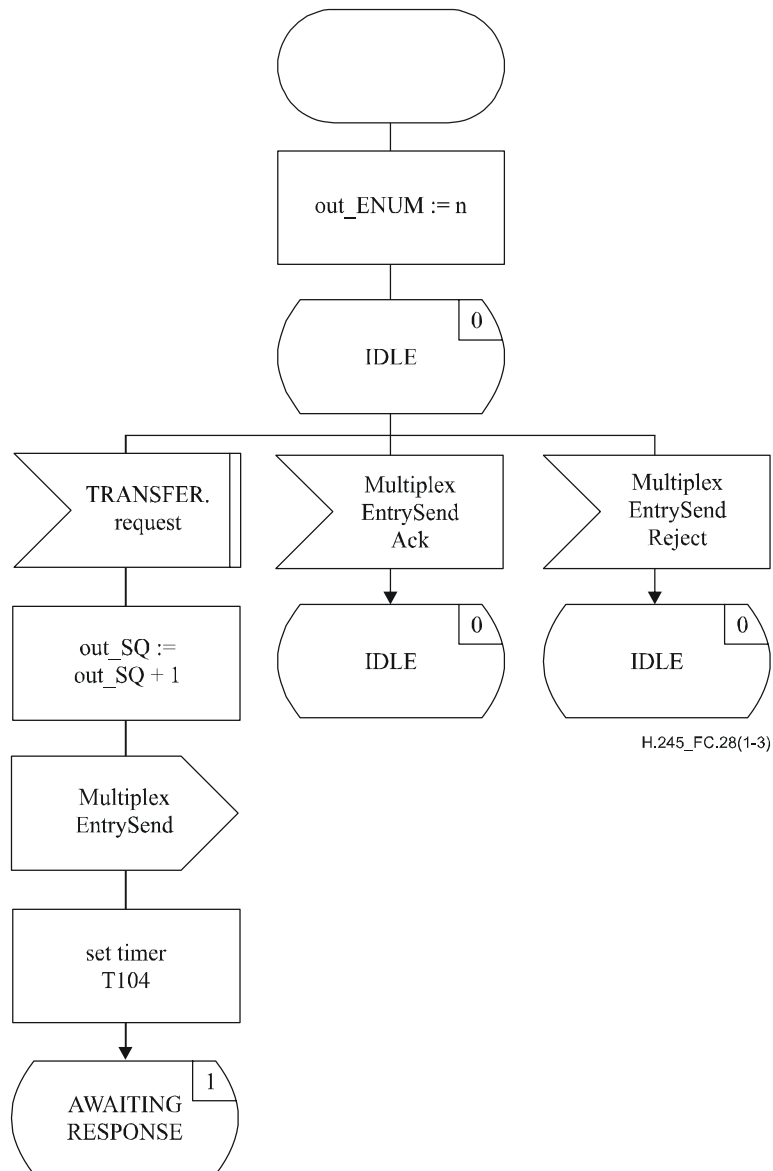
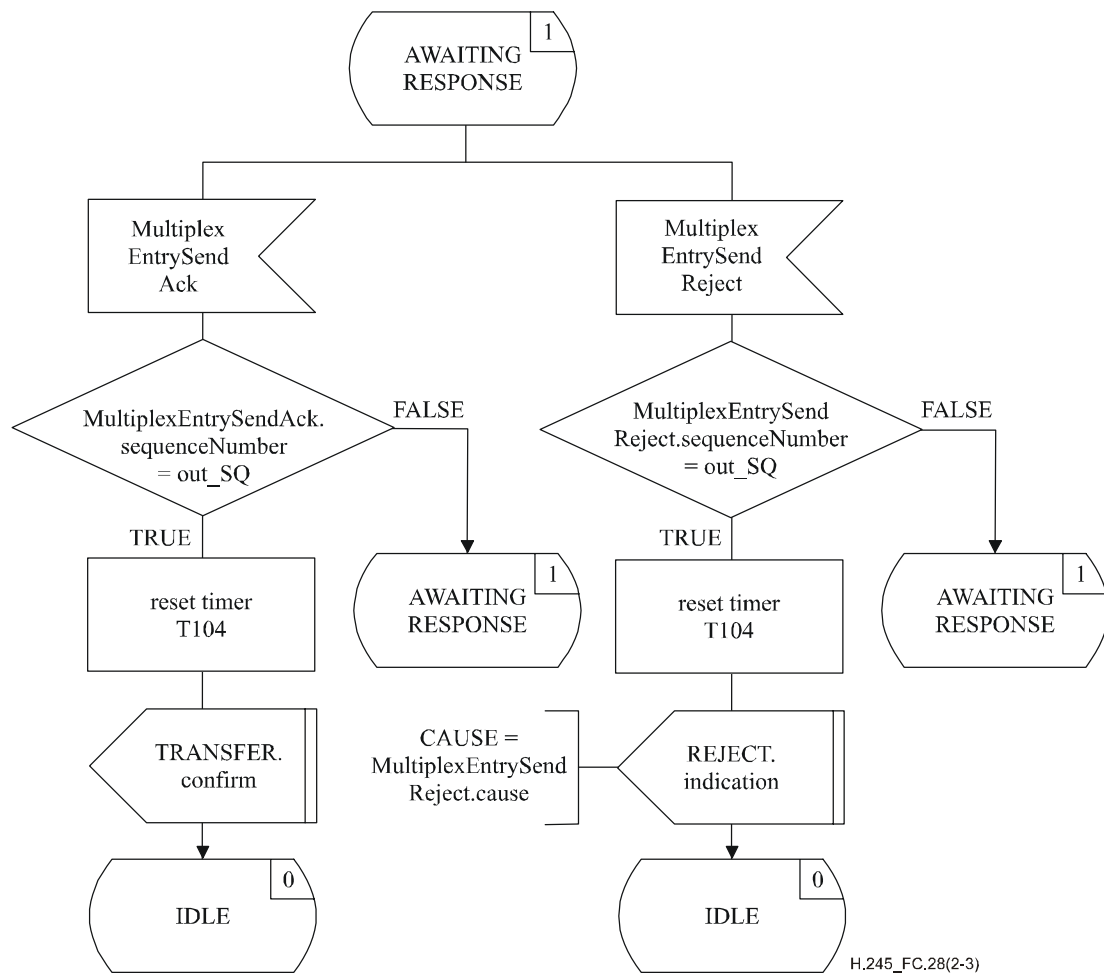


Figure C.28/H.245 – Diagramme SDL dans l'entité MTSE sortante (feuille 1 de 3)



H.245_FC.28(2-3)

Figure C.28/H.245 – Diagramme SDL dans l'entité MTSE sortante (feuille 2 de 3)

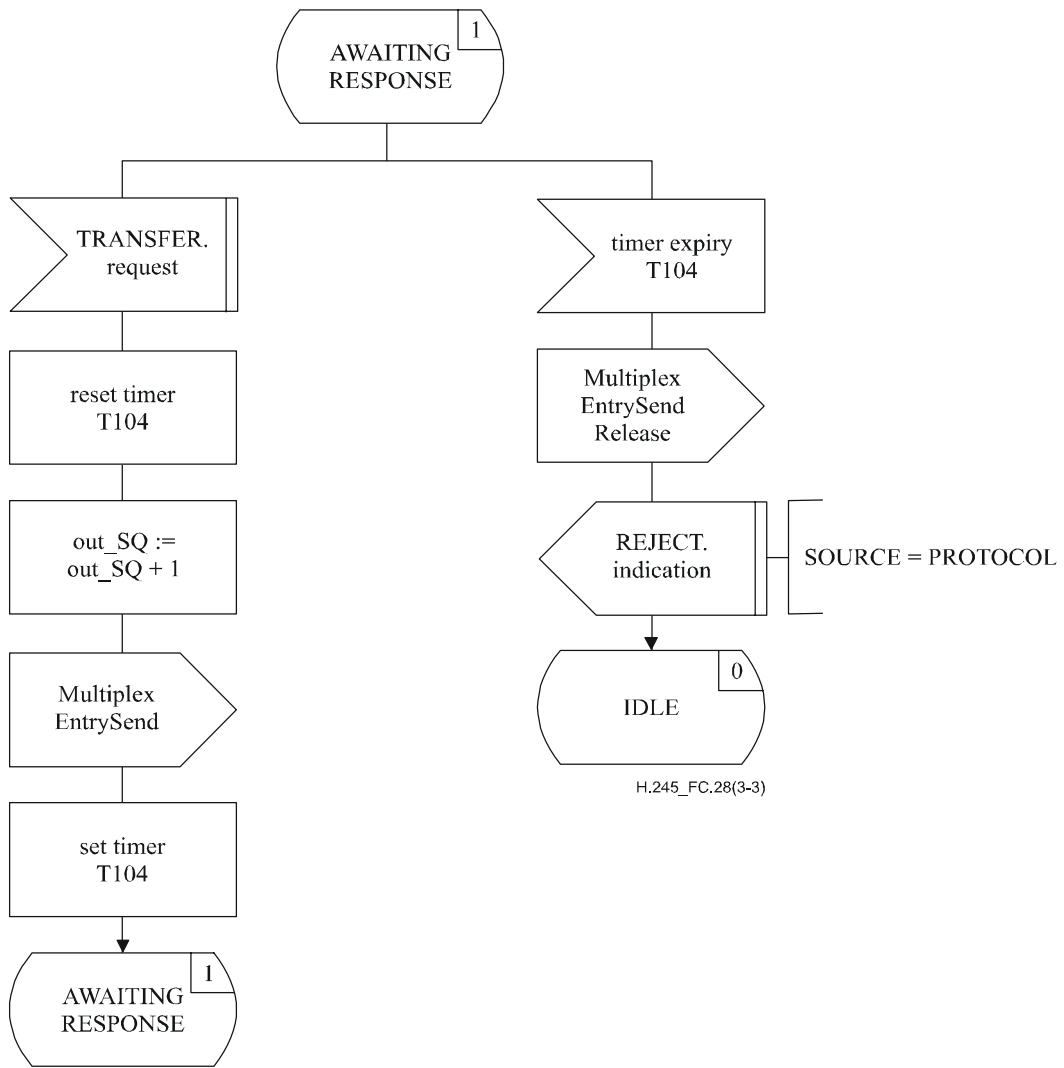


Figure C.28/H.245 – Diagramme SDL dans l'entité MTSE sortante (feuille 3 de 3)

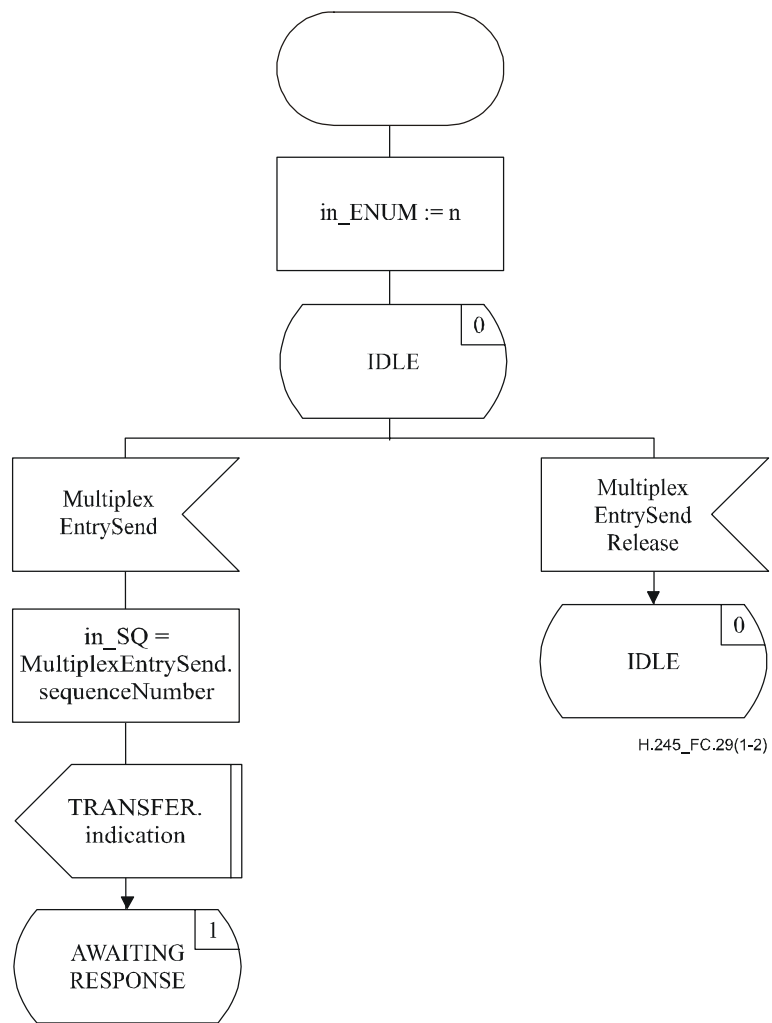
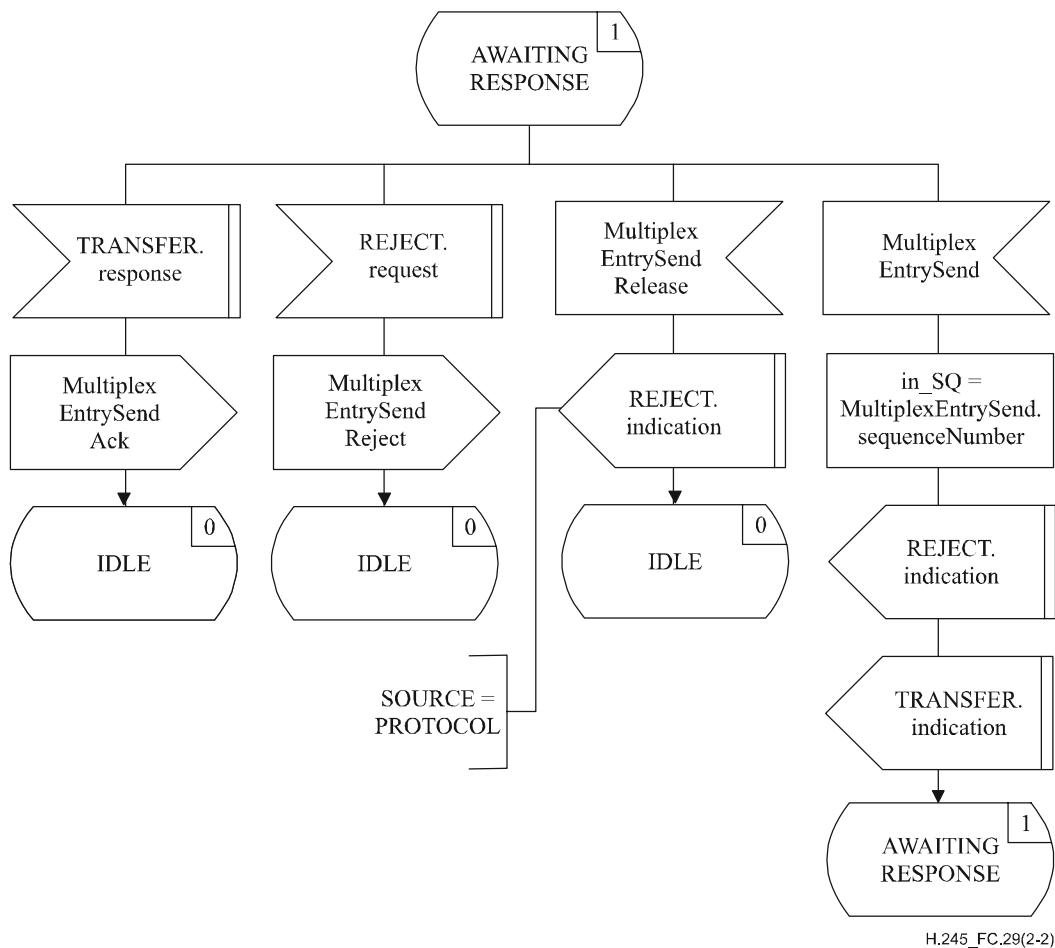


Figure C.29/H.245 – Diagramme SDL dans l'entité MTSE entrante (feuille 1 de 2)



H.245_FC.29(2-2)

Figure C.29/H.245 – Diagramme SDL dans l'entité MTSE entrante (feuille 2 de 2)

C.8 Procédures relatives à la demande d'entrée de multiplexage

C.8.1 Introduction

Ces procédures sont utilisées par un terminal pour demander la retransmission d'un ou plusieurs descripteurs MultiplexEntryDescriptor. Ces procédures sont désignées comme entité de signalisation de demande d'entrée de multiplexage (RMESE, *request multiplex entry signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité RMESE et l'utilisateur de l'entité RMESE. Les informations de protocole sont transmises à l'entité RMESE homologue au moyen des messages appropriés définis dans l'Annexe A. Il y a une entité RMESE sortante et une entité RMESE entrante. Il existe une instance d'entité RMESE pour chaque entrée de tableau de multiplexage.

Un terminal qui apporte une telle réponse positive, c'est-à-dire en présentant une primitive de réponse SEND, doit déclencher les procédures du tableau de multiplexage pour envoyer l'entrée de tableau de multiplexage dès que possible.

Le texte suivant présente un aperçu général du fonctionnement du protocole. En cas de divergence avec la spécification formelle du protocole qui suit, la spécification formelle sera applicable.

NOTE – Le présent protocole a été défini de sorte qu'il y ait une entité RMESE indépendante pour chaque entrée du tableau de multiplexage et la syntaxe a été définie pour permettre à un message unique de transmettre les informations relatives à une ou plusieurs entrées du tableau de multiplexage. La façon dont les messages sont élaborés est décidée au moment de la mise en œuvre: un terminal peut par exemple répondre à un message RequestMultiplexEntry demandant l'envoi de trois entrées avec un, deux ou trois messages de réponse.

C.8.1.1 Aperçu général du protocole – Entité RMESE sortante

Une procédure de demande d'entrée de multiplex est lancée quand la primitive de demande SEND est présentée par l'utilisateur dans l'entité RMESE sortante. Un message RequestMultiplexEntry est envoyé à l'entité RMESE entrante homologue, et le temporisateur T107 démarre. Si un message RequestMultiplexEntryAck est reçu en réponse au message RequestMultiplexEntry, alors le temporisateur T107 est arrêté, et l'utilisateur est informé par la primitive de confirmation SEND que la procédure de demande d'entrée de multiplexage a abouti. Si, cependant, un message RequestMultiplexEntryReject est reçu en réponse au message RequestMultiplexEntry, alors le temporisateur T107 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité RMESE homologue a refusé d'envoyer l'entrée de multiplexage.

Si la temporisation T107 arrive à expiration, alors l'utilisateur de l'entité RMESE sortante en est informé par la primitive d'indication REJECT et un message RequestMultiplexEntryRelease est envoyé.

C.8.1.2 Aperçu général du protocole – Entité RMESE entrante

Quand un message RequestMultiplexEntry est reçu dans l'entité RMESE entrante, l'utilisateur est informé de la demande d'entrée de multiplexage par la primitive d'indication SEND. L'utilisateur de l'entité RMESE entrante signale l'acceptation de la demande d'entrée de multiplexage en présentant 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 refus de la demande d'entrée de multiplexage en présentant la primitive de demande REJECT et un message RequestMultiplexEntryReject est envoyé à l'entité RMESE sortante homologue.

C.8.2 Communication entre l'entité RMESE et l'utilisateur de l'entité RMESE

C.8.2.1 Primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE

La communication entre l'entité RMESE et l'utilisateur de l'entité RMESE est effectuée en utilisant les primitives indiquées dans le Tableau C.29.

Tableau C.29/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
SEND	– (Note 1)	–	–	–
REJECT	CAUSE	SOURCE CAUSE	Non défini (Note 2)	Non défini

NOTE 1 – "–" signifie aucun paramètre.
NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.

C.8.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- 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.

C.8.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués au Tableau C.29 est la suivante:

- a) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur de "USER" ou de "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;
- b) le paramètre CAUSE indique le motif de refus d'envoi d'une entrée de tableau de multiplexage. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

C.8.2.4 Etats d'entité RMESE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE.

Les états correspondant à une entité RMESE sortante sont:

Etat 0: IDLE

L'entité RMESE est au repos.

Etat 1: AWAITING RESPONSE

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

Les états correspondant à une entité RMESE entrante sont:

Etat 0: IDLE

L'entité RMESE est au repos.

Etat 1: AWAITING RESPONSE

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

C.8.2.5 Diagramme de changement d'état

La séquence autorisée de primitives entre l'entité RMESE et l'utilisateur de l'entité RMESE est définie ici. Les séquences autorisées sont spécifiées séparément pour chacune des entité RMESE entrante et sortante, comme l'indiquent respectivement les Figures C.30 et C.31.

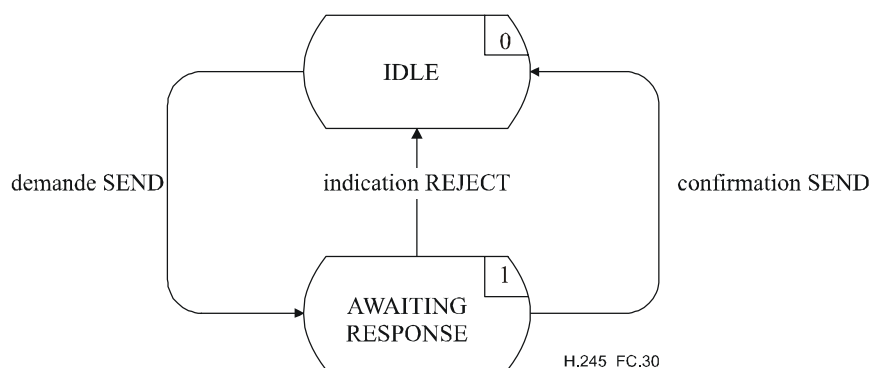


Figure C.30/H.245 – Diagramme de changement d'état pour la séquence de primitives dans l'entité RMESE sortante

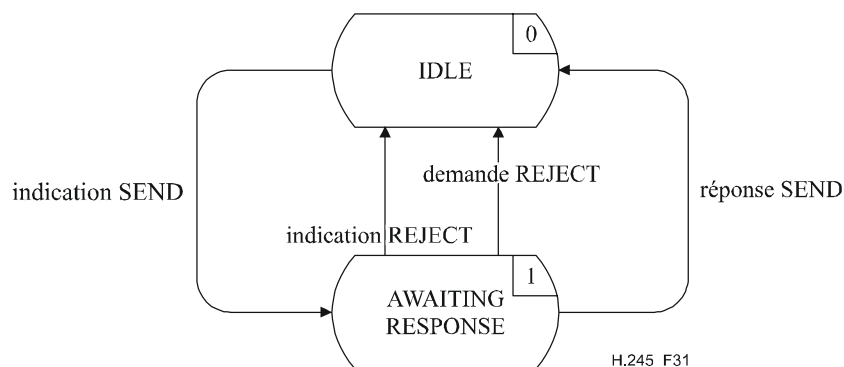


Figure C.31/H.245 – Diagramme de changement d'état pour la séquence de primitives dans l'entité RMESE entrante

C.8.3 Communication entre les entités RMESE homologues

C.8.3.1 Messages

Le Tableau C.30 indique les messages et les champs de l'entité RMESE, définis dans l'Annexe A, qui se rapportent au protocole d'entité RMESE.

Tableau C.30/H.245 – Noms et champs de message de l'entité RMESE

Fonction	Message	Sens	Champ
transfert	RequestMultiplexEntry	S → E (Note)	multiplexTableEntryNumber
	RequestMultiplexEntryAck	S ← E	multiplexTableEntryNumber
	RequestMultiplexEntryReject	S ← E	multiplexTableEntryNumber rejectionDescriptions.cause
réinitialisation	RequestMultiplexEntryRelease	S → E	

NOTE – Sens: S – sortant, E – entrant.

C.8.3.2 Variables d'état d'entité RMESE

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

out_ENUM

Cette variable d'état permet d'établir une distinction entre les entités RMESE sortantes. Elle est initialisée lors de l'initialisation de l'entité RMESE sortante. La valeur de la variable out_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages envoyés à partir d'une entité RMESE sortante. En ce qui concerne les messages reçus par une entité RMESE sortante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de out_ENUM.

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

in_ENUM

Cette variable d'état permet d'établir une distinction entre les entités RMESE entrantes. Elle est initialisée lors de l'initialisation de l'entité RMESE entrante. La valeur de la variable in_ENUM est utilisée pour définir le champ multiplexTableEntryNumber des messages envoyés à partir d'une entité RMESE entrante. En ce qui concerne les messages reçus par une entité RMESE entrante, la valeur du champ multiplexTableEntryNumber des messages est identique à la valeur de in_ENUM.

C.8.3.3 Temporisateurs d'entité RMESE

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

T107

Ce temporisateur est utilisé pendant l'état AWAITING RESPONSE (attente de la réponse). Il spécifie le temps maximal pendant lequel aucun message RequestMultiplexEntryAck ou RequestMultiplexEntryReject ne peut être reçu.

C.8.4 Procédures d'entité RMESE

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

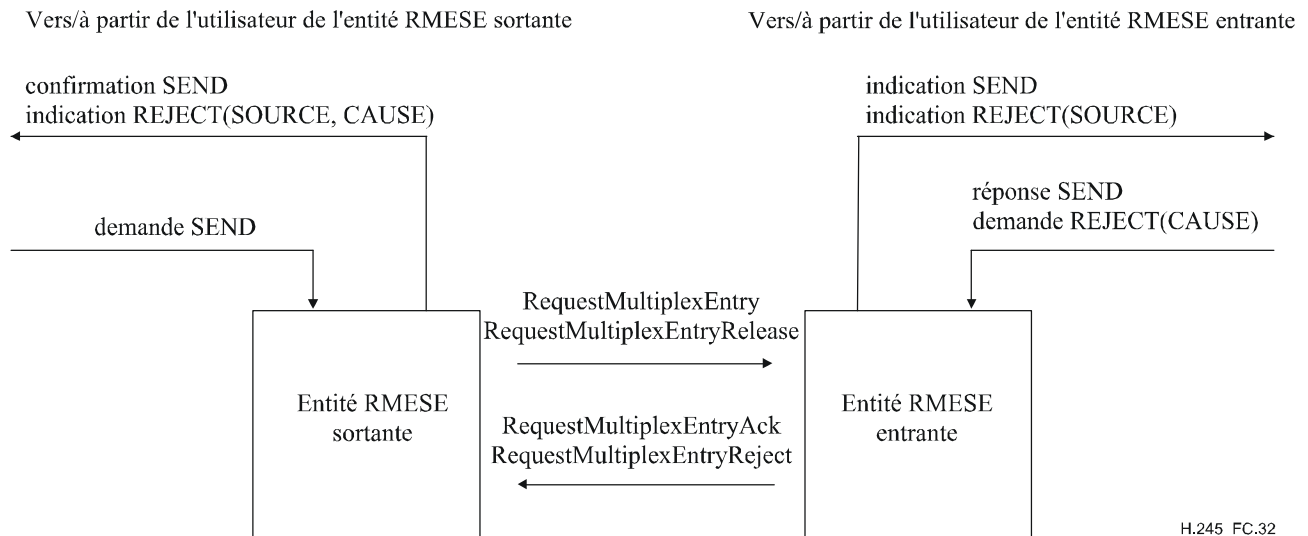


Figure C.32/H.245 – Primitives et messages de l'entité de signalisation de demande d'entrée de multiplexage

C.8.4.1 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 C.31, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau C.31/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
indication REJECT	SOURCE	USER
	CAUSE	null

C.8.4.2 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement décrit dans les diagrammes SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.32.

Tableau C.32/H.245 – Valeurs par défaut des champs des messages

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

C.8.4.3 Diagrammes SDL

Les procédures des entités RMESE entrante et sortante sont décrites sous la forme de diagrammes SDL dans les Figures C.33 et C.34 respectivement.

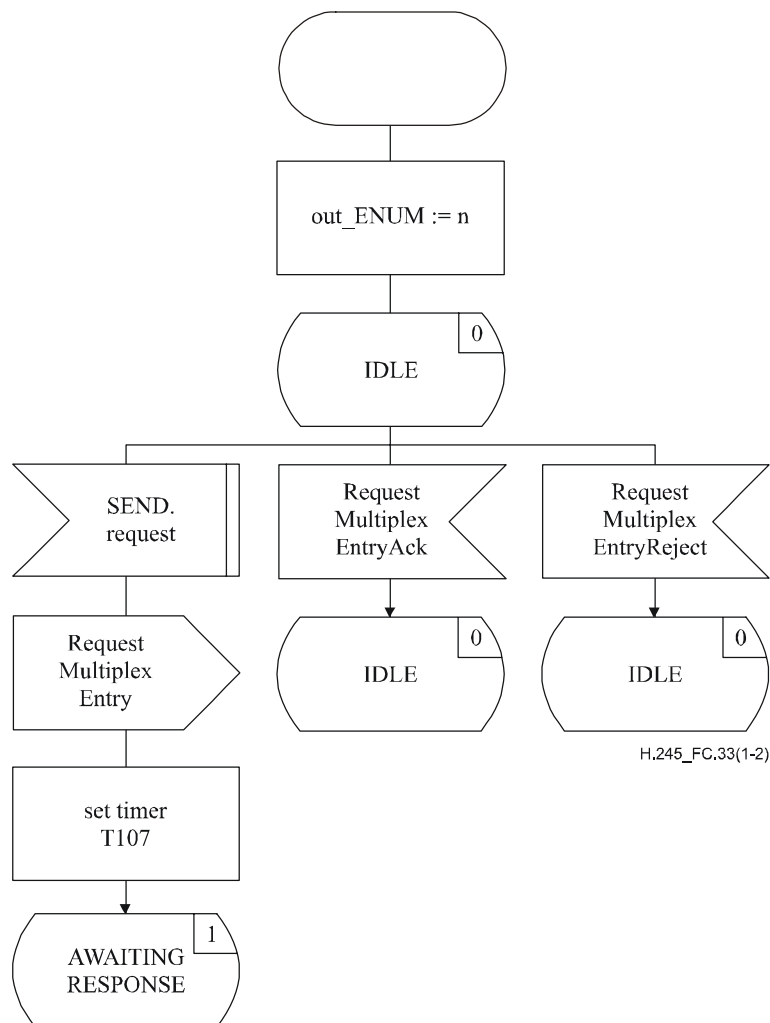


Figure C.33/H.245 – Diagramme SDL de l'entité RMESE sortante (feuille 1 de 2)

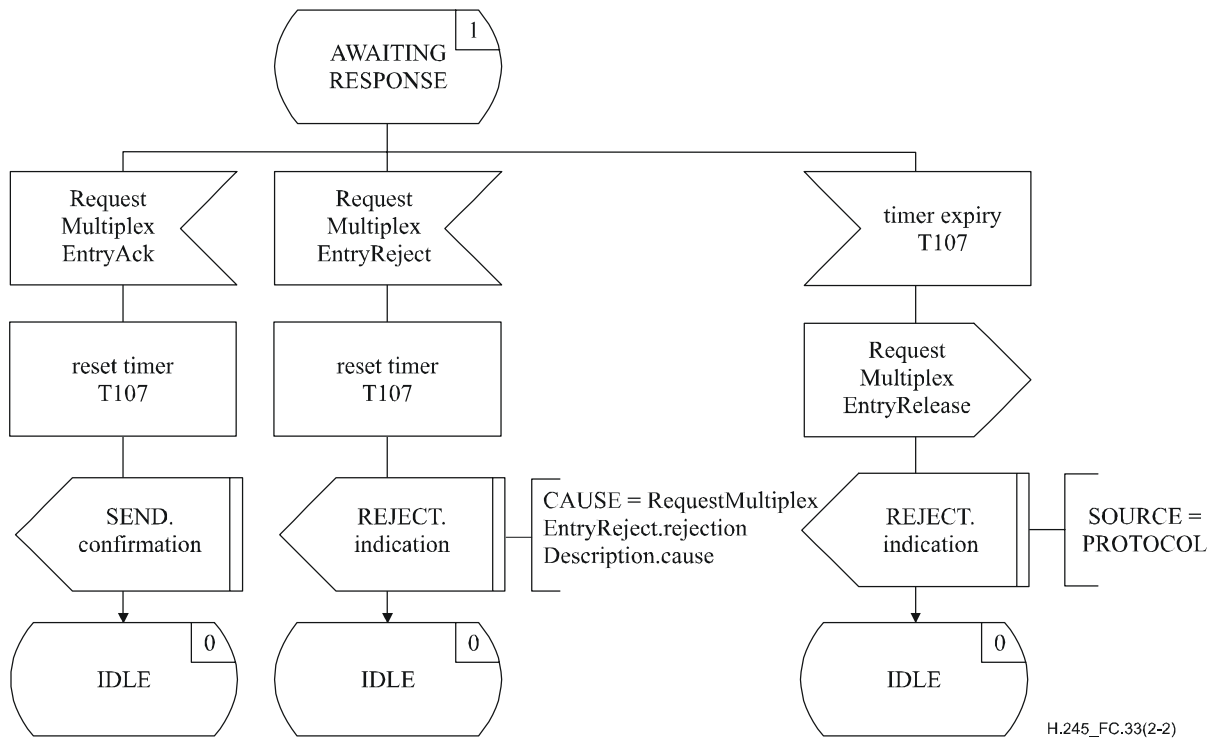


Figure C.33/H.245 – Diagramme SDL de l'entité RMESE sortante (feuille 2 de 2)

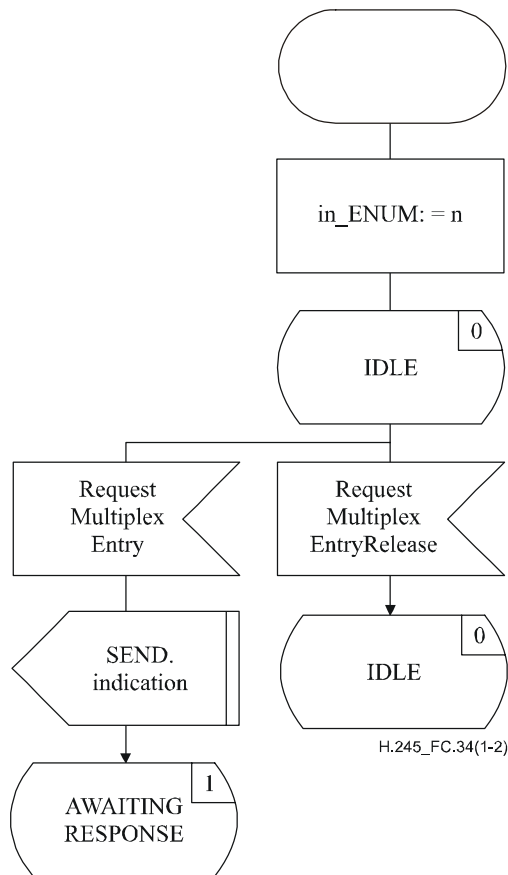


Figure C.34/H.245 – Diagramme SDL de l'entité RMESE entrante (feuille 1 de 2)

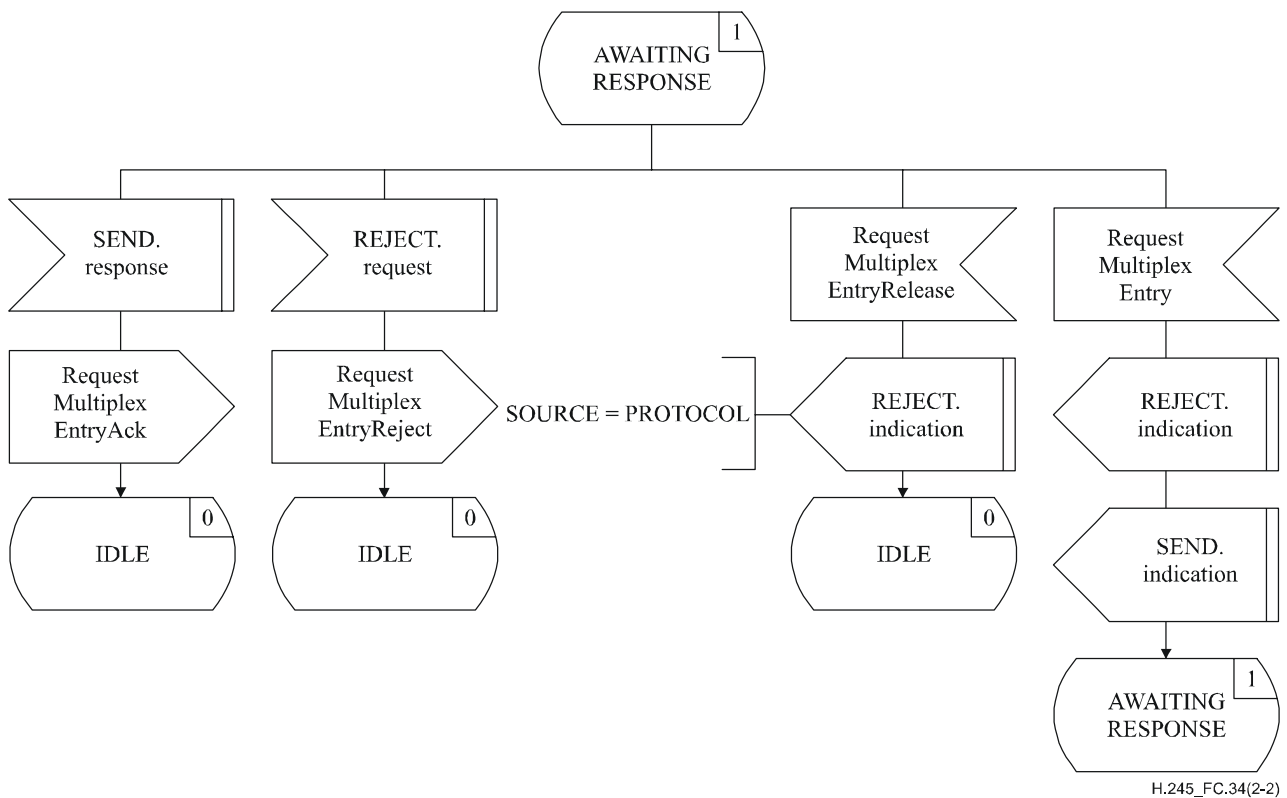


Figure C.34/H.245 – Diagramme SDL de l'entité RMESE entrante (feuille 2 de 2)

C.9 Procédures de demande de mode

C.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. On fait référence ici à ces procédures sous le nom de l'entité de signalisation de demande de mode (MRSE). Les procédures 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 à l'entité MRSE homologue par l'intermédiaire de messages appropriés définis dans l'Annexe A. Il y a une entité MRSE sortante et une entité MRSE entrante. A chacune des extrémités entrante et sortante, il y a une instance de l'entité MRSE par appel.

Un terminal qui apporte une telle réponse positive, c'est-à-dire en présentant la primitive de réponse TRANSFER, devra déclencher les procédures de signalisation de voie logique pour définir le mode approprié de transmission dès que possible.

Si les capacités actuellement valables reçues du terminal distant contiennent une ou plusieurs capacités d'émission, un terminal peut choisir le mode qu'il préfère en appliquant les procédures de demande de mode. Un terminal dont les capacités actuellement valables contiennent une ou plusieurs capacités d'émission et qui reçoit une telle demande doit satisfaire la demande.

Une demande de mode ne doit pas être envoyée à un terminal dont les capacités actuellement valables ne contiennent pas de capacités d'émission, c'est-à-dire que le terminal ne souhaite pas être contrôlé à distance et ne doit pas non plus être contrôlé de cette façon. Si ce terminal reçoit cependant une demande de mode, il peut satisfaire cette demande.

Un terminal qui reçoit la commande multipointModeCommand doit satisfaire toutes les demandes de mode reçues, jusqu'à ce que cette commande soit annulée par la réception de la commande cancelMultipointModeCommand. Une demande de mode peut être envoyée à un terminal dont les

capacités actuellement valables ne contiennent pas de capacités d'émission quand une commande multipointModeCommand a été préalablement envoyée.

La demande de mode peut inclure des voies qui sont déjà ouvertes. Par exemple, dans le cas où une voie pour G.723.1 serait actuellement ouverte et un terminal souhaiterait recevoir une voie G.728 supplémentaire, il enverrait une demande de mode contenant à la fois les voies G.723.1 et G.728. L'absence de la demande de voie G.723.1 indiquerait que le mode G.723.1 n'est plus souhaité.

Quand le paramètre logicalChannelNumber est présent, la demande se rapporte seulement à la voie logique indiquée (qui doit être dans l'état ouvert) et requiert que le mode de la voie logique indiquée soit remplacé par le mode spécifié.

NOTE – A moins que le paramètre logicalChannelNumber ne soit présent, la description de la demande de mode spécifie un mode complet. Si, par exemple, des signaux vidéo sont en cours de transmission et si une demande de mode n'incluant pas de spécification pour la vidéo est reçue, il faut alors interrompre la transmission vidéo.

Si une source alimente plusieurs récepteurs, elle peut être incapable de répondre à des signaux reçus tels que des demandes d'émission dans un mode particulier.

Le texte suivant donne un aperçu général du fonctionnement du protocole d'entité MRSE. En cas de divergence avec la spécification formelle, la spécification formelle sera applicable.

C.9.1.1 Aperçu général du protocole – Entité MRSE sortante

Une procédure de demande de mode est déclenchée quand la primitive de demande TRANSFER est présentée par l'utilisateur dans l'entité MRSE sortante. Un message RequestMode est envoyé à l'entité MRSE entrante homologue, et le temporisateur T109 démarre. Si un message RequestModeAck est reçu en réponse au message RequestMode, alors le temporisateur T109 est arrêté et l'utilisateur est informé par la primitive de confirmation TRANSFER que la demande de mode a abouti. Si, cependant, un message RequestModeReject est reçu en réponse au message RequestMode, le temporisateur T109 est arrêté et l'utilisateur est informé par la primitive d'indication REJECT que l'utilisateur de l'entité MRSE homologue a refusé d'accepter la demande de mode.

Si la temporisation T109 arrive à expiration, l'entité MRSE sortante en est informée par la primitive d'indication REJECT et un message RequestModeRelease est envoyé.

Seuls les messages RequestModeAck et RequestModeReject répondant au message RequestMode le plus récent sont acceptés. Les messages envoyés en réponse à des messages RequestMode précédents sont ignorés.

Une nouvelle procédure de demande de mode peut être déclenchée par la primitive de demande TRANSFER par l'utilisateur de l'entité MRSE sortante avant qu'un message RequestModeAck ou RequestModeReject ait été reçu.

C.9.1.2 Aperçu général du protocole – Entité MRSE entrante

Quand un message RequestMode est reçu dans l'entité MRSE entrante, l'utilisateur est informé de la demande de mode par la primitive d'indication TRANSFER. L'utilisateur de l'entité MRSE entrante signale l'acceptation de la demande de mode en présentant une primitive de réponse TRANSFER et un message RequestModeAck est envoyé à l'entité MRSE sortante homologue. L'utilisateur de l'entité MRSE entrante signale le refus de la demande de mode en présentant une primitive de demande REJECT et un message RequestModeReject est envoyé à l'entité MRSE sortante homologue.

Un nouveau message RequestMode peut être reçu avant que l'utilisateur de l'entité MRSE entrante ait répondu à un message RequestMode antérieur. L'utilisateur de l'entité MRSE entrante est informé par la primitive d'indication REJECT, suivie de la primitive d'indication TRANSFER, et l'utilisateur de l'entité MRSE entrante répond à la nouvelle entrée du tableau de multiplexage.

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

C.9.2 Communication entre l'entité MRSE et l'utilisateur de l'entité MRSE

C.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 C.33.

Tableau C.33/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.				

C.9.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) les primitives TRANSFER sont utilisées pour le transfert de la demande de mode;
- b) les primitives REJECT sont utilisées pour refuser une demande de mode.

C.9.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.33 est la suivante:

- a) le paramètre MODE-ELEMENT spécifie un élément de mode. Ce paramètre est mappé au champ requestedModes du message RequestMode et transmis de façon transparente de l'utilisateur de l'entité MRSE sortante vers l'utilisateur de l'entité MRSE entrante. Ce paramètre est obligatoire. Il peut y avoir plusieurs éléments MODE-ELEMENT associés aux primitives TRANSFER;
- b) le paramètre MODE-PREF informe l'utilisateur de l'utilisation ou de la non-utilisation du mode préféré. Il est mappé au champ response du message RequestModeAck et transmis de façon transparente de l'utilisateur de l'entité RMSE entrante vers l'utilisateur de l'entité RMSE sortante. Il a deux valeurs "MOST-PREFERRED" et "LESS-PREFERRED";
- c) le paramètre SOURCE indique l'origine de la primitive d'indication REJECT. Le paramètre SOURCE a la valeur de "USER" ou de "PROTOCOL". Ce dernier cas peut être le résultat d'une fin de temporisation;
- d) le paramètre CAUSE indique le motif du refus d'une demande de mode. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "PROTOCOL".

C.9.2.4 Etats d'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 au repos.

Etat 1: AWAITING RESPONSE

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

Les états correspondant à une entité MRSE entrante sont:

Etat 0: IDLE

L'entité MRSE est au repos.

Etat 1: AWAITING RESPONSE

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

C.9.2.5 Diagramme de changement d'état

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

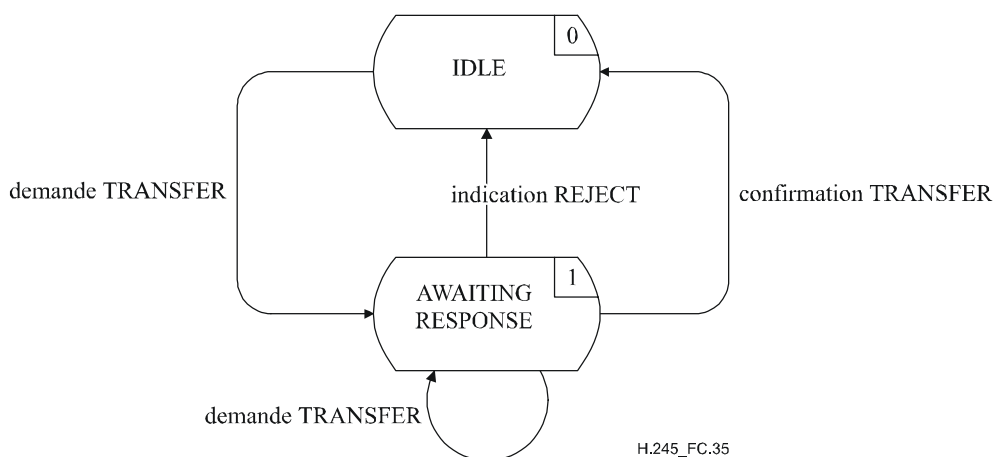


Figure C.35/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité MRSE sortante

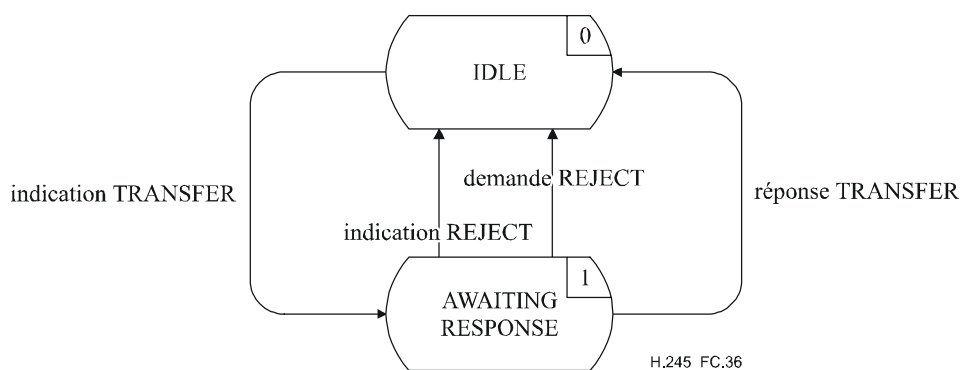


Figure C.36/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité MRSE entrante

C.9.3 Communication entre les entités MRSE homologues

C.9.3.1 Messages

Le Tableau C.34 indique les messages et les champs de l'entité MRSE, définis dans l'Annexe A, qui se rapportent au protocole de l'entité MRSE.

Tableau C.34/H.245 – Noms et champs des messages de l'entité MRSE

Fonction	Message	Sens	Champ
demande de mode	RequestMode	S → E (Note)	sequenceNumber requestedModes
	RequestModeAck	S ← E	sequenceNumber response
	RequestModeReject	S ← E	sequenceNumber cause
réinitialisation	RequestModeRelease	S → E	–
NOTE – Sens: S – sortant, E – entrant.			

C.9.3.2 Variables d'état d'entité MRSE

La variable d'état suivante est définie dans l'entité MRSE sortante:

out_SQ

Cette variable d'état est utilisée pour indiquer le message RequestMode le plus récent. Elle est incrémentée de un et mappée au champ sequenceNumber du message RequestMode avant la transmission du message RequestMode. L'opération arithmétique exécutée sur la variable out_SQ est modulo 256.

La variable d'état suivante est définie 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 à l'entité MRSE homologue.

C.9.3.3 Temporisateurs d'entité MRSE

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

T109

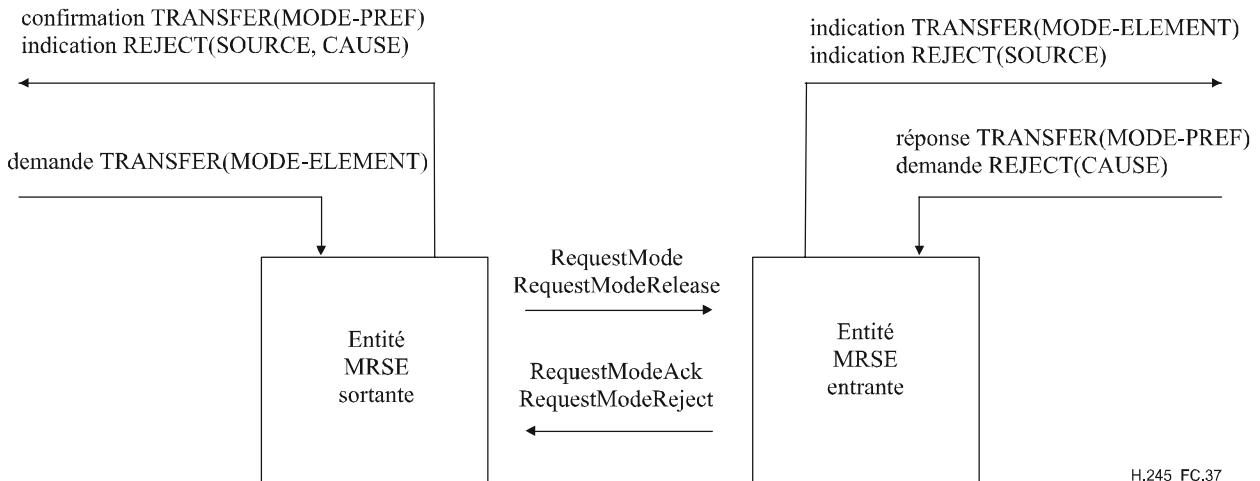
Ce temporisateur est utilisé pendant l'état AWAITING RESPONSE. Il spécifie le temps maximal pendant lequel aucun message RequestModeAck ou RequestModeReject ne peut être reçu.

C.9.4 Procédures d'entité MRSE

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

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

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



H.245_FC.37

Figure C.37/H.245 – Primitives et messages dans l'entité de signalisation de demande de mode

C.9.4.1 Valeurs par défaut des paramètres des primitives

Les paramètres des primitives d'indication et de confirmation prennent les valeurs indiquées dans le Tableau C.35, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau C.35/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
indication TRANSFER	MODE-ELEMENT	RequestMode.requestedModes
confirmation TRANSFER	MODE-PREF	RequestModeACK.response
indication REJECT	SOURCE	USER
	CAUSE	null

C.9.4.2 Valeurs par défaut des champs des messages

Les champs de message prennent les valeurs indiquées dans le Tableau C.36, lorsque rien n'est décrit explicitement dans les diagrammes SDL.

Tableau C.36/H.245 – Valeurs par défaut des champs des 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	in_SQ
	cause	demande REJECT(CAUSE)
RequestModeRelease	–	–

C.9.4.3 Diagramme SDL

Les procédures de l'entité MRSE sortante et de l'entité MRSE entrante sont décrites sous forme de diagrammes SDL dans les Figures C.38 et C.39 respectivement.

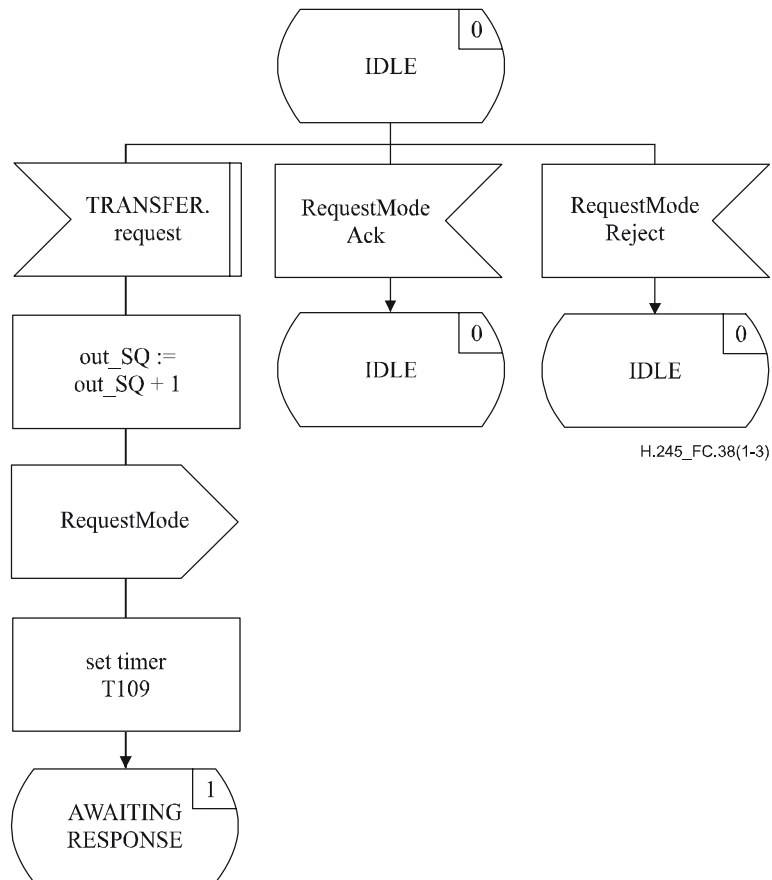


Figure C.38/H.245 – Diagramme SDL de l'entité MRSE sortante (feuille 1 de 3)

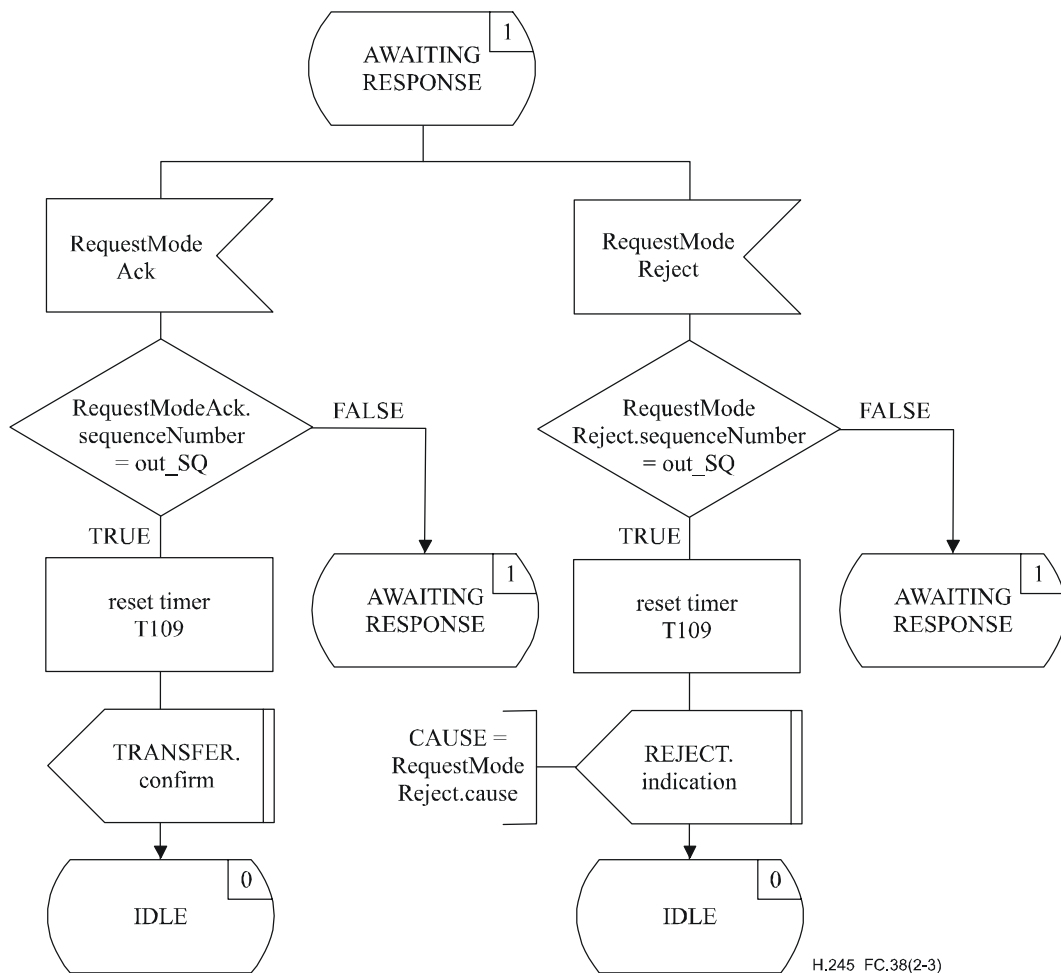


Figure C.38/H.245 – Diagramme SDL de l'entité MRSE sortante (feuille 2 de 3)

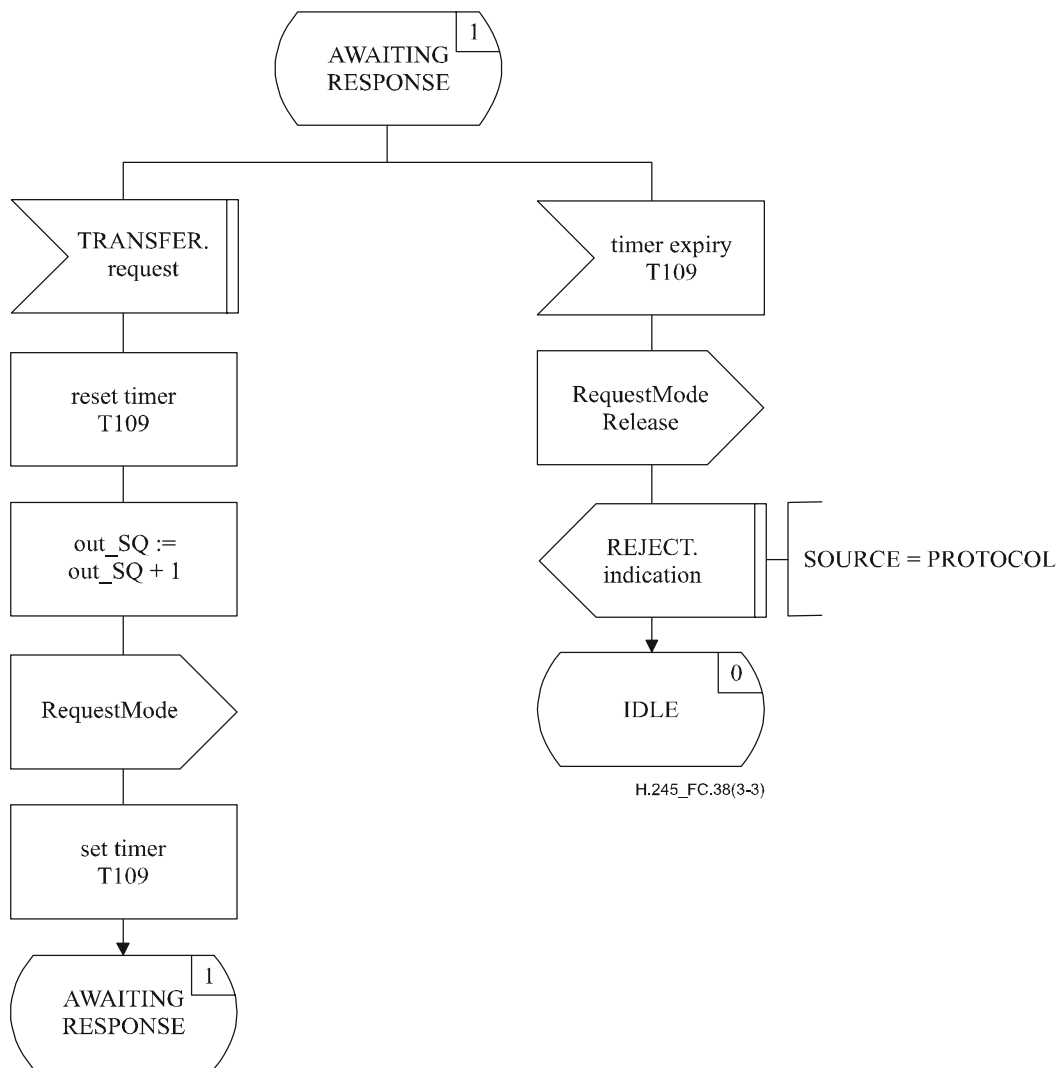


Figure C.38/H.245 – Diagramme SDL de l'entité MRSE sortante (feuille 3 de 3)

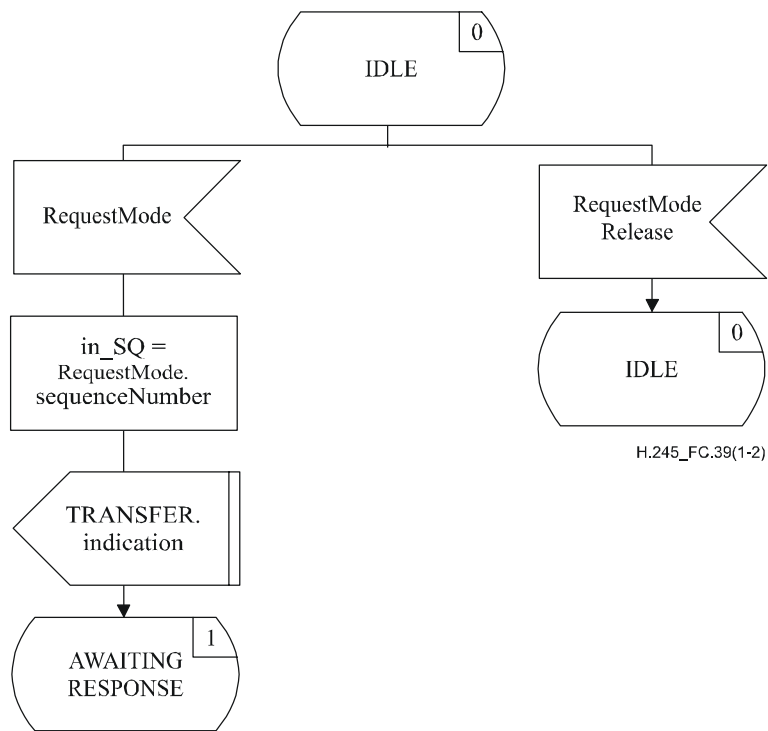


Figure C.39/H.245 – Diagramme SDL de l'entité MRSE entrante (feuille 1 de 2)

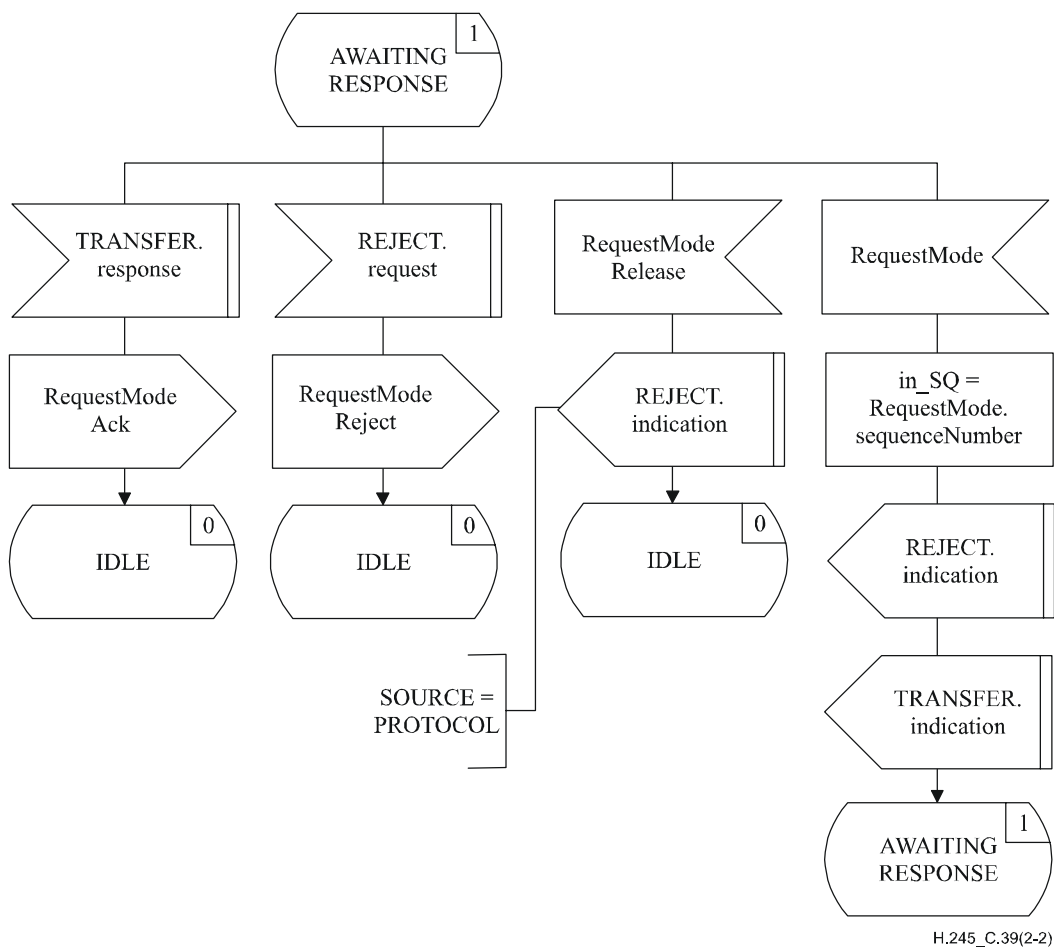


Figure C.39/H.245 – Diagramme SDL de l'entité MRSE entrante (feuille 2 de 2)

C.10 Procédures liées au temps de propagation aller et retour

C.10.1 Introduction

Les procédures décrites dans ces 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 est toujours active.

Il est fait référence à la fonction décrite ici sous le nom d'entité de signalisation du temps de propagation aller et retour (RTDSE, *round-trip delay signalling entity*). Les procédures sont spécifiées en termes de primitives et d'états à l'interface entre l'entité RTDSE et l'utilisateur de l'entité RTDSE. Il y a une instance de l'entité RTDSE dans chaque terminal. N'importe quel terminal peut procéder à la détermination du temps de propagation aller et retour.

Le texte suivant donne un aperçu général du fonctionnement du protocole dans les entités RTDSE. En cas de divergence avec la spécification formelle, la spécification formelle sera applicable.

C.10.1.1 Aperçu général du protocole – Entités RTDSE

Une procédure de détermination du temps de propagation aller et retour est déclenchée quand la primitive de demande TRANSFER est présentée par l'utilisateur de l'entité RTDSE. Un message RoundTripDelayRequest est envoyé à l'entité RTDSE homologue, et le temporisateur T105 démarre. Si un message RoundTripDelayResponse est reçu en réponse au message RoundTripDelayRequest, le temporisateur T105 est arrêté et l'utilisateur est informé du temps de propagation aller et retour par la primitive de confirmation TRANSFER, ce temps correspondant à la valeur du temporisateur T105.

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

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

Seul le message RoundTripDelayResponse répondant au message RoundTripDelayRequest le plus récent est accepté. Les messages envoyés en réponse à des messages RoundTripDelayRequest antérieurs sont ignorés.

Une nouvelle procédure de détermination du temps de propagation aller et retour peut être déclenchée au moyen de la primitive de demande TRANSFER par l'utilisateur de l'entité RTDSE avant qu'un message RoundTripDelayResponse ait été reçu.

C.10.2 Communication entre l'entité RTDSE et l'utilisateur de l'entité RTDSE

C.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 C.37. Ces primitives sont destinées à définir les procédures d'entité RTDSE et ne sont pas supposées spécifier une mise en œuvre ou apporter des contraintes.

Tableau C.37/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
TRANSFER	– (Note 1)	Non défini (Note 2)	Non défini	DELAY
EXPIRY	Non défini	–	Non défini	Non défini

NOTE 1 – "-" signifie aucun paramètre.
 NOTE 2 – "non défini" signifie que cette primitive n'est pas définie.

C.10.2.2 Définition des primitives

La définition de ces primitives est la suivante:

- a) la primitive TRANSFER est utilisée pour demander la détermination du temps de propagation aller et retour et pour en donner un compte rendu;
- b) la primitive EXPIRY indique qu'aucune réponse n'a été reçue du terminal homologue.

C.10.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués dans le Tableau C.37 est la suivante:

- a) le paramètre DELAY renvoie le temps de propagation aller et retour mesuré.

C.10.2.4 Etats d'entité RTDSE

Les états suivants sont utilisés pour spécifier la séquence autorisée de primitives entre l'entité RTDSE et l'utilisateur de l'entité RTDSE.

Etat 0: IDLE

Il n'y a pas de transfert en cours entre entités RTDSE.

Etat 1: AWAITING RESPONSE

L'utilisateur de l'entité RTDSE a demandé la mesure du temps de propagation aller et retour. Une réponse de l'entité RTDSE homologue est attendue.

C.10.2.5 Diagramme de changement d'état

Le présent paragraphe définit la séquence autorisée de primitives entre l'entité RTDSE et l'utilisateur de l'entité RTDSE. Les séquences autorisées sont indiquées à la Figure C.40.

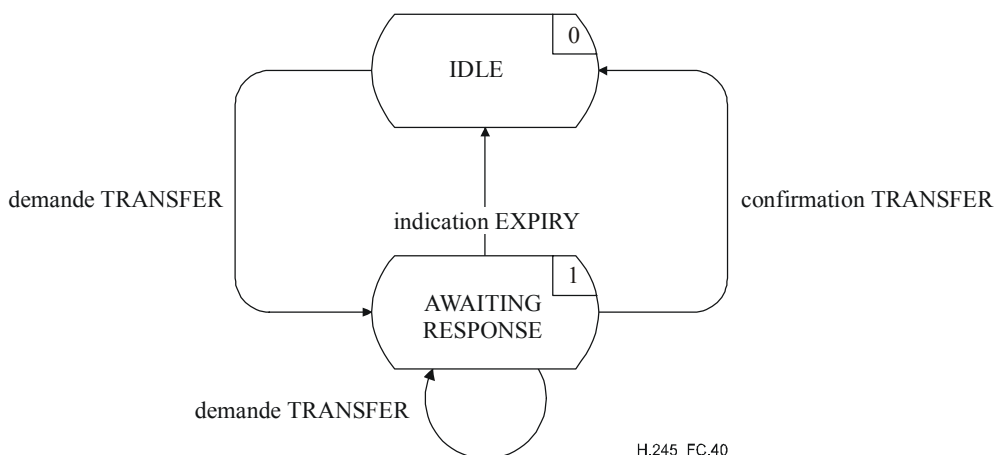


Figure C.40/H.245 – Diagramme de changement d'état pour la séquence de primitives de l'entité RTDSE

C.10.3 Communication entre les entités RTDSE homologues

C.10.3.1 Messages

Le Tableau C.38 indique les messages et les champs de l'entité RTDSE, définis dans l'Annexe A, qui se rapportent au protocole d'entité RTDSE.

Tableau C.38/H.245 – Noms et champs des messages d'entité RTDSE

Fonction	Message	Champ
transfert	RoundTripDelayRequest	sequenceNumber
	RoundTripDelayResponse	sequenceNumber

C.10.3.2 Variables d'état d'entité RTDSE

La variable d'état suivante de l'entité RTDSE est définie:

out_SQ

Cette variable d'état est utilisée pour indiquer le message RoundTripDelayRequest le plus récent. Elle est incrémentée de un et mappée au champ sequenceNumber du message RoundTripDelayRequest avant la transmission d'un message RoundTripDelayRequest. L'opération arithmétique effectuée sur la variable d'état out_SQ est modulo 256.

C.10.3.3 Temporisateurs d'entité RTDSE

Le temporisateur suivant est spécifié pour l'entité RTDSE:

T105

Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie le temps maximal pendant lequel aucun message RoundTripDelayResponse ne peut être reçu.

C.10.4 Procédures d'entité RTDSE

C.10.4.1 Introduction

La Figure C.41 récapitule les primitives d'entité RTDSE et leurs paramètres, de même que les messages.

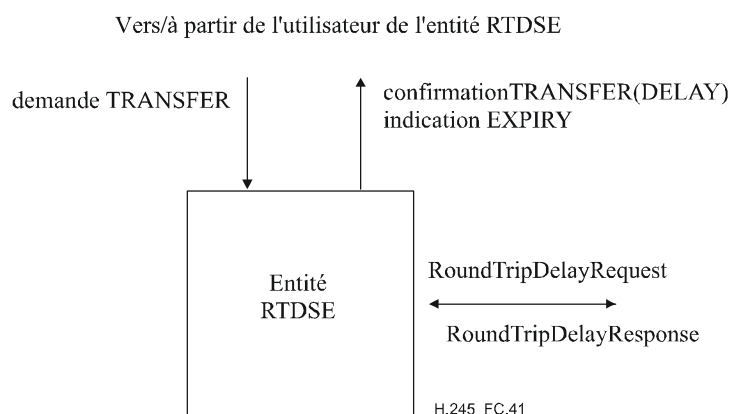


Figure C.41/H.245 – Primitives et messages de l'entité RTDSE

C.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 les valeurs indiquées dans le Tableau C.39.

Tableau C.39/H.245 – Valeurs par défaut des paramètres des primitives

Primitive	Paramètre	Valeur par défaut
confirmation TRANSFER	DELAY	Valeur initiale du temporisateur T105 minorée de la valeur du temporisateur T105
indication EXPIRY	–	–
NOTE – Par définition, les temporisateurs utilisent un compte à rebours jusqu'à zéro. Le paramètre DELAY indique le temps de fonctionnement du temporisateur, correspondant à la différence entre la valeur initiale et la valeur gardée par le temporisateur.		

C.10.4.3 Valeurs par défaut des champs des messages

Quand cela n'est pas explicitement décrit dans les messages SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.40.

Tableau C.40/H.245 – Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut
RoundTripDelayRequest	sequenceNumber	out_SQ
RoundTripDelayResponse	sequenceNumber	RoundTripDelayRequest.sequenceNumber

C.10.4.4 Diagrammes SDL

Les procédures des entités RTDSE sont décrites sous forme de diagrammes SDL dans la Figure C.42.

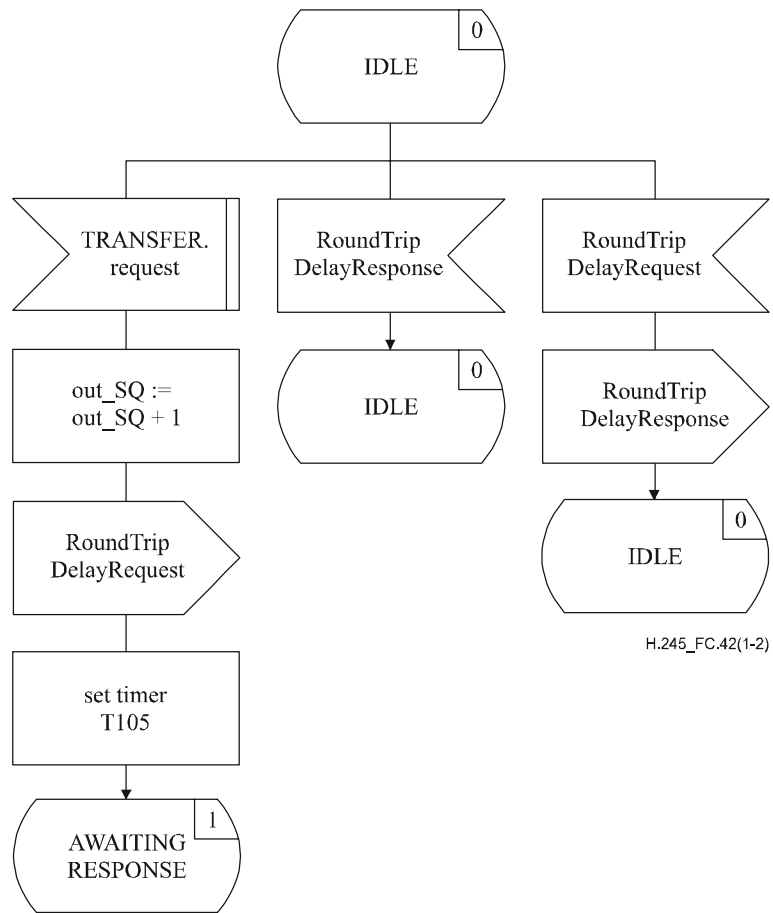


Figure C.42/H.245 – Diagramme SDL de l'entité RTDSE (feuille 1 de 2)

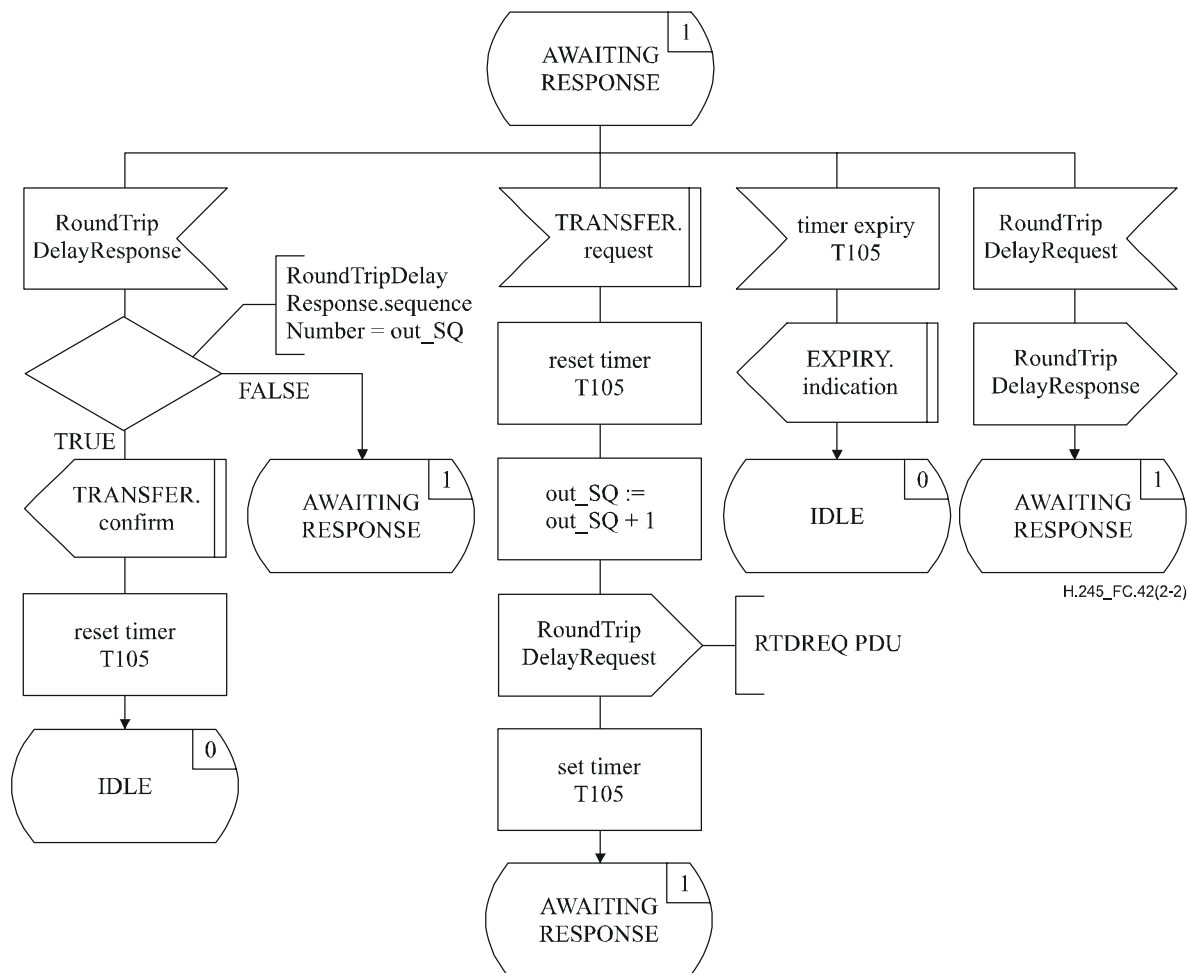


Figure C.42/H.245 – Diagramme SDL de l'entité RTDSE (feuille 2 de 2)

C.11 Procédures relatives à la boucle de maintenance

C.11.1 Introduction

Le protocole spécifié ici assure la fiabilité de fonctionnement des boucles de maintenance en utilisant des procédures avec acquittement.

Le protocole spécifié ici est désigné comme l'entité de signalisation de boucle de maintenance (MLSE, *maintenance loop signalling entity*). Les procédures sont spécifiées en termes de primitives à l'interface entre l'entité MLSE et l'utilisateur de l'entité MLSE, de même qu'en termes d'états d'entité MLSE. Les informations relatives au protocole sont transmises à l'entité MLSE homologue au moyen des messages appropriés définis dans l'Annexe A.

Il y a une entité MLSE sortante et une entité MLSE entrante. A chacune des extrémités entrante et sortante, il y a une instance d'entité MLSE pour chaque voie logique bidirectionnelle, et une instance pour la boucle du système. Il n'y a pas de liaison entre une entité MLSE entrante et une entité MLSE sortante située du même côté, autre que celle obtenue au moyen de primitives vers l'utilisateur de l'entité MLSE et à partir de ce même utilisateur. Les conditions d'erreur des entités MLSE sont indiquées.

Le terminal contenant l'entité MLSE entrante devra mettre en boucle les données appropriées pendant qu'il est dans l'état LOOPED, et non à un autre moment. Le terminal contenant l'entité MLSE sortante devra être capable de recevoir les données en boucle pendant qu'il est dans un état quelconque, mais pendant qu'il est dans l'état LOOPED, il devrait recevoir des données en boucle uniquement.

NOTE – Le message MaintenanceLoopOffCommand s'applique à toutes les entités MLSE. Il est toujours utilisé pour arrêter toutes les boucles de maintenance.

Le texte suivant est un aperçu général du fonctionnement du protocole d'entité MLSE. En cas de divergence avec la spécification formelle, cette dernière sera applicable.

C.11.1.1 Aperçu général du protocole – Sortant

L'établissement d'une boucle de maintenance est déclenchée quand la primitive de demande LOOP est présentée par l'utilisateur dans l'entité MLSE sortante. Un message MaintenanceLoopRequest est envoyé à l'entité MLSE entrante homologue, et le temporisateur T102 démarre. Si un message MaintenanceLoopAck est reçu en réponse à un message MaintenanceLoopRequest, le temporisateur T102 est arrêté et l'utilisateur est informé par la primitive de confirmation LOOP que la boucle de maintenance a bien été établie. Si, cependant, un message MaintenanceLoopReject est reçu en réponse au message MaintenanceLoopRequest, le temporisateur T102 est arrêté et l'utilisateur est informé par la primitive d'indication RELEASE que l'utilisateur de l'entité MLSE homologue a refusé l'établissement de la boucle de maintenance.

Si la temporisation T102 arrive à expiration pendant cette période, l'utilisateur est informé par la primitive d'indication RELEASE, et un message MaintenanceLoopOffCommand est envoyé à l'entité MLSE entrante homologue. Ceci annulera toutes les boucles de maintenance, et pas simplement celle relative à l'entité MLSE considérée.

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

Avant qu'un message MaintenanceLoopAck ou MaintenanceLoopReject ait été reçu en réponse à un message MaintenanceLoopRequest préalablement envoyé, l'utilisateur de l'entité MLSE sortante peut annuler la boucle de maintenance en utilisant la primitive de demande RELEASE.

C.11.1.2 Aperçu général du protocole – Entrant

Quand un message MaintenanceLoopRequest est reçu dans l'entité MLSE entrante, l'utilisateur est informé de la demande d'établissement d'une boucle de maintenance par la primitive d'indication LOOP. L'utilisateur de l'entité MLSE entrante signale l'acceptation de la demande d'établissement de la boucle de maintenance en présentant la primitive de réponse LOOP, et un message MaintenanceLoopAck est envoyé à l'entité MLSE sortante homologue. La boucle de maintenance doit alors être exécutée. L'utilisateur de l'entité MLSE entrante signale le rejet de la demande d'établissement de la boucle de maintenance en présentant la primitive de demande RELEASE, et un message MaintenanceLoopReject est envoyé à l'entité MLSE sortante homologue.

Une boucle de maintenance qui a été établie avec succès peut être annulée quand le message MaintenanceLoopOffCommand est reçu dans l'entité MLSE entrante. L'utilisateur de l'entité MLSE entrante en est informé par la primitive d'indication RELEASE.

C.11.2 Communication entre l'entité MLSE et l'utilisateur de l'entité MLSE

C.11.2.1 Primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE

La communication entre l'entité MLSE et l'utilisateur de l'entité MLSE est effectuée au moyen des primitives indiquées dans le Tableau C.41.

Tableau C.41/H.245 – Primitives et paramètres

Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
LOOP	LOOP_TYPE	LOOP_TYPE	– (Note 1)	–
RELEASE	CAUSE	SOURCE CAUSE	Non défini (Note 2)	Non défini
ERROR	Non défini	ERRCODE	Non défini	Non défini
NOTE 1 – "-" signifie l'absence de paramètres.				
NOTE 2 – "Non défini" signifie que cette primitive n'existe pas.				

C.11.2.2 Définition des primitives

La définition de ces primitives est la suivante:

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

C.11.2.3 Définition des paramètres

La définition des paramètres de primitive indiqués au Tableau C.41 est la suivante:

- a) le paramètre LOOP_TYPE spécifie les paramètres associés à la boucle de maintenance. Il a les valeurs de "SYSTEM", "MEDIA" et "LOGICAL_CHANNEL". Ce paramètre, accompagné du numéro de voie logique, détermine la valeur du champ type du message MaintenanceLoopRequest qui est alors transmis de façon permanente 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 a la valeur de "USER" ou "MLSE", indiquant l'utilisateur de l'entité MLSE ou l'entité MLSE elle-même. Ce dernier cas peut résulter d'une erreur de protocole;
- c) le paramètre CAUSE indique la raison pour laquelle l'utilisateur de l'entité MLSE homologue a rejeté une demande d'établissement de boucle de maintenance. Le paramètre CAUSE n'est pas présent quand le paramètre SOURCE indique "MLSE";
- d) le paramètre ERRCODE indique le type d'erreur d'entité MLSE. Le Tableau C.45 indique les valeurs autorisées du paramètre ERRCODE.

C.11.2.4 Etats d'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 les messages échangés entre les entités MLSE homologues. Les états sont spécifiés séparément pour chacune des entités MLSE entrante et sortante. Les états correspondant à une entité MLSE sortante sont:

Etat 0: NOT LOOPED

Il n'y a pas de boucle de maintenance.

Etat 1: AWAITING RESPONSE

L'entité MLSE sortante attend l'établissement d'une boucle de maintenance avec une entité MLSE entrante homologue.

Etat 2: LOOPED

La boucle de maintenance entre les entités MLSE homologues a été établie. Toutes les données reçues sur le canal approprié devraient être des données en boucle.

Les états correspondant à une entité MLSE entrante sont:

Etat 0: NOT LOOPED

Il n'y a pas de boucle de maintenance.

Etat 1: AWAITING RESPONSE

L'entité MLSE entrante attend l'établissement d'une boucle de maintenance avec une entité MLSE sortante homologue. Les données appropriées ne devront pas être mises en boucle.

Etat 2: LOOPED

Une boucle de maintenance a été établie entre des entités MLSE homologues. Toutes les données reçues sur le canal approprié devront être mises en boucle.

C.11.2.5 Diagramme de changement d'état

La séquence autorisée de primitives entre l'entité MLSE et l'utilisateur de l'entité MLSE est définie ici. La séquence autorisée de primitives concerne les états de l'entité MLSE du point de vue de l'utilisateur de l'entité MLSE. Les séquences autorisées sont spécifiées séparément pour chacune des entités MLSE entrante et sortante, comme cela est indiqué dans les Figures C.43 et C.44 respectivement.

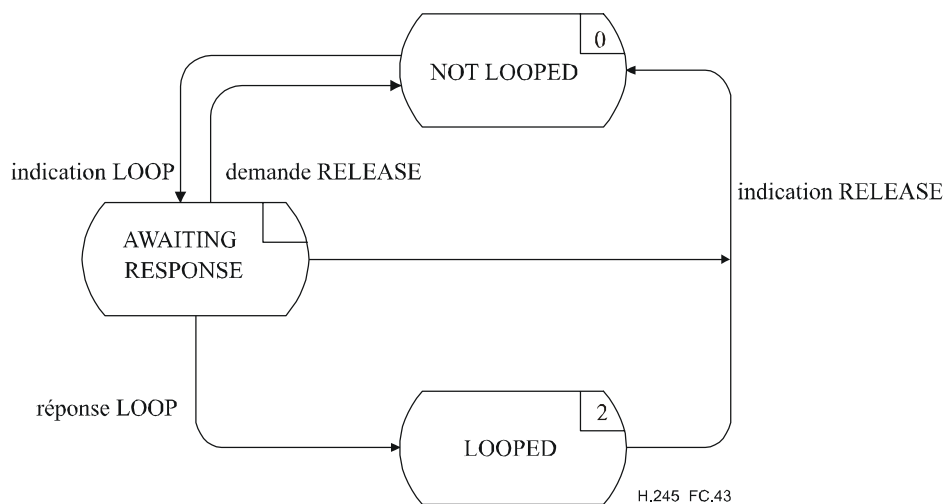


Figure C.43/H.245 – Diagramme de changement d'état pour une séquence de primitives de l'entité MLSE sortante

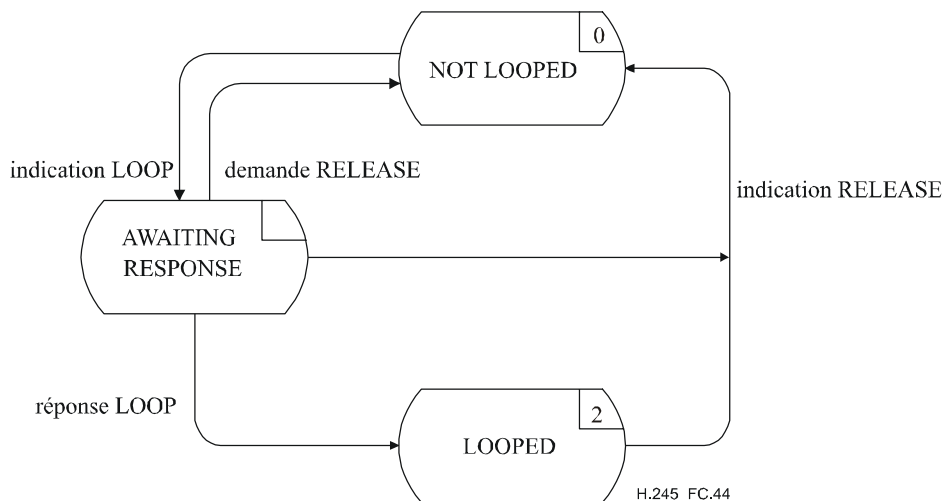


Figure C.44/H.245 – Diagramme de transition d'état pour une séquence de primitives de l'entité MLSE entrante

C.11.3 Communication entre les entités MLSE homologues

C.11.3.1 Messages d'entité MLSE

Le Tableau C.42 indique les messages et champs des entités MLSE, définis dans l'Annexe A, qui se rapportent au protocole d'entité MLSE.

Tableau C.42/H.245 – Noms et champs de messages d'entité MLSE

Fonction	Message	Sens	Champ
établissement	MaintenanceLoopRequest	S → E (Note)	type
	MaintenanceLoopAck	S ← E	type
	MaintenanceLoopReject	S ← E	type cause
libération	MaintenanceLoopOffCommand	S → E	–
NOTE – Sens: S – sortant, E – entrant.			

C.11.3.2 Variables d'état d'entité MLSE

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

out_MLN

Cette variable d'état est utilisée pour établir une distinction entre les entités MLSE sortantes. Elle est initialisée lors de l'initialisation de l'entité MLSE sortante. La valeur de out_MLN est utilisée pour définir le champ type des messages MaintenanceLoopRequest envoyés à partir d'une entité MLSE sortante.

Les variables d'état suivantes sont définies dans l'entité MLSE entrante:

in_MLN

Cette variable d'état établit une distinction entre les entités MLSE entrantes. Elle est initialisée lors de l'initialisation des entités MLSE entrantes. En ce qui concerne les messages MaintenanceLoopRequest reçus dans une entité MLSE entrante, la valeur du champ type des messages est compatible avec la valeur de in_MLN.

in_TYPE

Cette variable d'état enregistre la valeur du paramètre LOOP_TYPE quand le message MaintenanceLoopRequest est reçu. Cette variable d'état est utilisée pour positionner la valeur du champ type du message MaintenanceLoopAck.

C.11.3.3 Temporisateurs d'entité MLSE

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

T102

Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie le temps maximal durant lequel aucun message MaintenanceLoopAck ou MaintenanceLoopReject ne peut être reçu.

C.11.4 Procédures d'entité MLSE

C.11.4.1 Introduction

La Figure C.45 récapitule les primitives et leurs paramètres, de même que les messages, pour chacune des entités MLSE entrante et sortante.

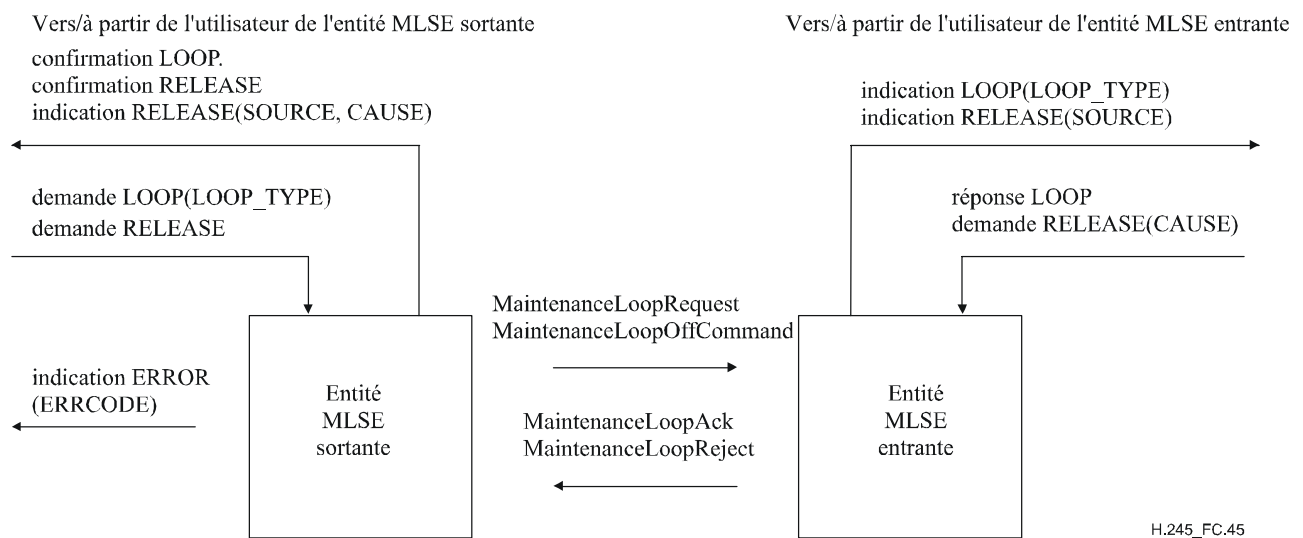


Figure C.45/H.245 – Primitives et messages dans l'entité de signalisation de boucle de maintenance

C.11.4.2 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 les valeurs indiquées dans le Tableau C.43.

Tableau C.43/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 – Un paramètre par défaut devra être codé comme mis à la valeur "null" (néant) si un champ de message indiqué n'est pas présent dans le message.		

C.11.4.3 Valeurs par défaut des champs des messages

Quand cela n'a pas été explicitement décrit dans les diagrammes SDL, les champs des messages prennent les valeurs indiquées dans le Tableau C.44.

Tableau C.44/H.245 – Valeurs par défaut des champs des messages

Message	Champ	Valeur par défaut (Note 1)
MaintenanceLoopRequest	type	demande LOOP(LOOP_TYPE) and out_MLN (Note 2)
MaintenanceLoopAck	type	in_LOOP and in_MLN (Note 3)
MaintenanceLoopReject	type cause	in_LOOP and in_MLN (Note 3) demande RELEASE(CAUSE)
MaintenanceLoopOffCommand	–	–

NOTE 1 – Un champ de message ne devra pas être codé si le paramètre de primitive correspondant vaut "null", c'est-à-dire est 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.

C.11.4.4 Valeurs du paramètre ERRCODE

Le paramètre ERRCODE de la primitive d'indication ERROR indique une condition d'erreur particulière. Le Tableau C.45 indique les valeurs que le paramètre ERRCODE peut prendre dans l'entité MLSE sortante. Il n'y a pas de primitive d'indication ERROR associée à l'entité MLSE entrante.

Tableau C.45/H.245 – Valeurs du paramètre ERRCODE dans l'entité MLSE sortante

Type d'erreur	Code d'erreur	Condition d'erreur	Etat
Message inapproprié	A	MaintenanceLoopAck	LOOPED
Pas de réponse de l'entité MLSE homologue	B	Expiration de la temporisation T102	AWAITING RESPONSE

C.11.4.5 Diagrammes SDL

Les procédures de l'entité MLSE entrante et de l'entité MLSE sortante sont décrites sous forme de diagrammes SDL dans les Figures C.46 et C.47 respectivement.

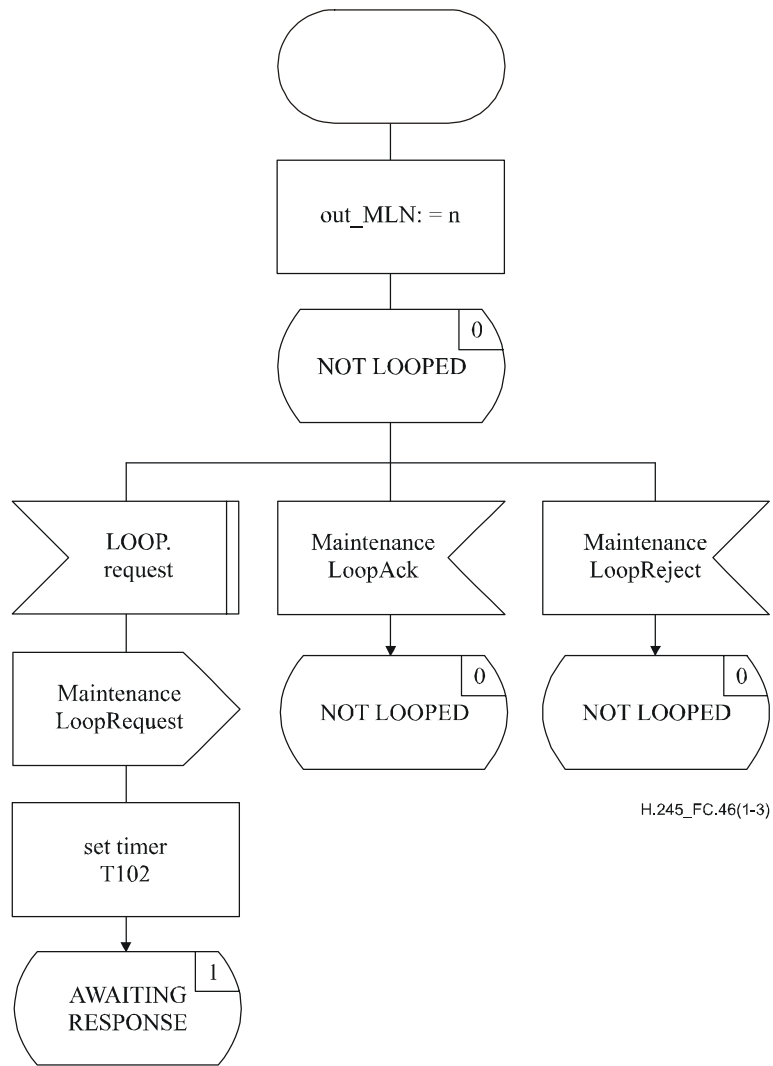


Figure C.46/H.245 – Diagramme SDL de l'entité MLSE sortante (feuille 1 de 3)

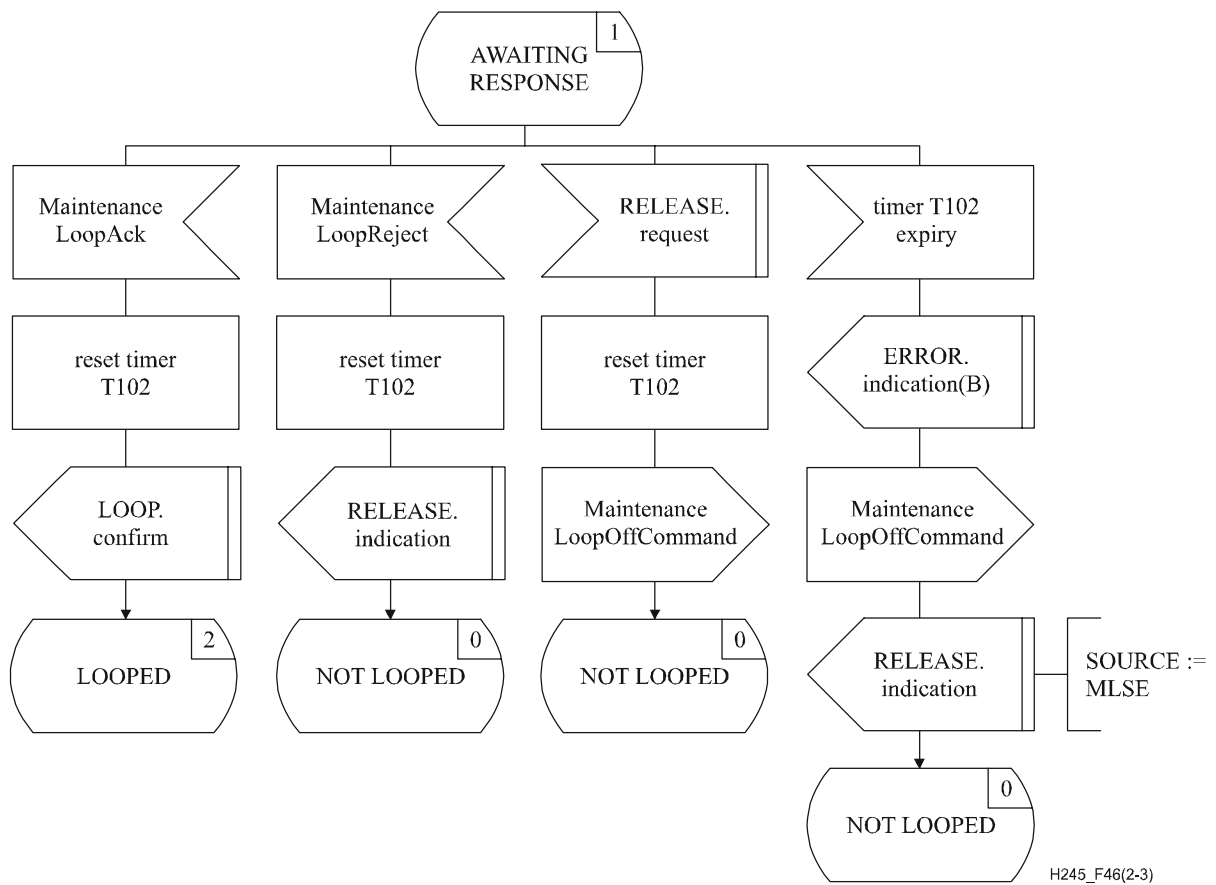


Figure C.46/H.245 – Diagramme SDL de l'entité MLSE sortante (feuille 2 de 3)

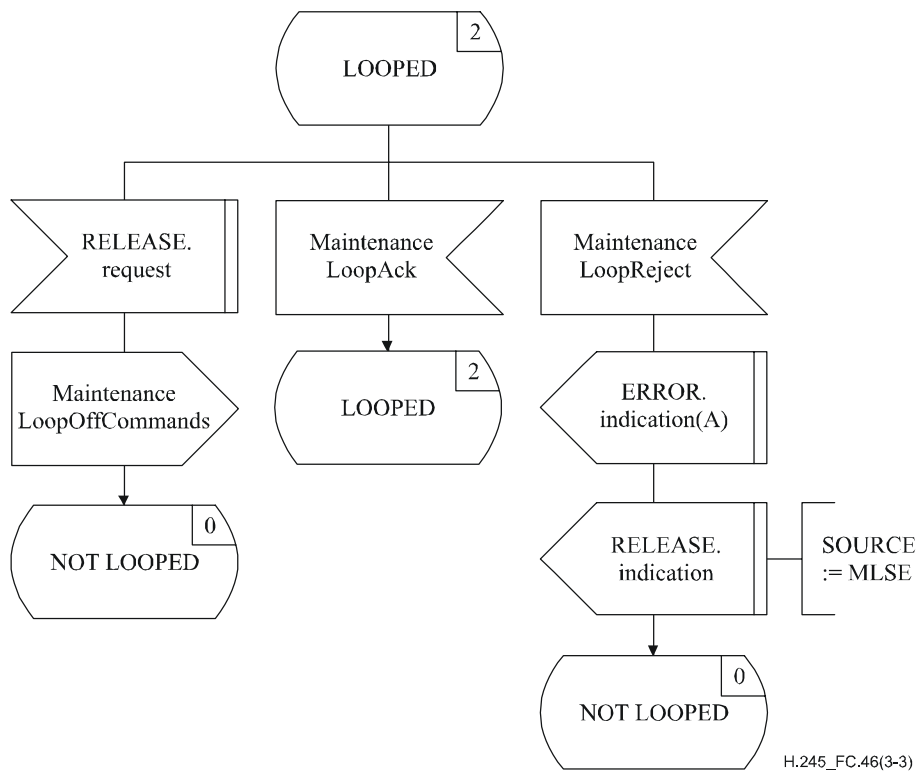


Figure C.46/H.245 – Diagramme SDL de l'entité MLSE sortante (feuille 3 de 3)

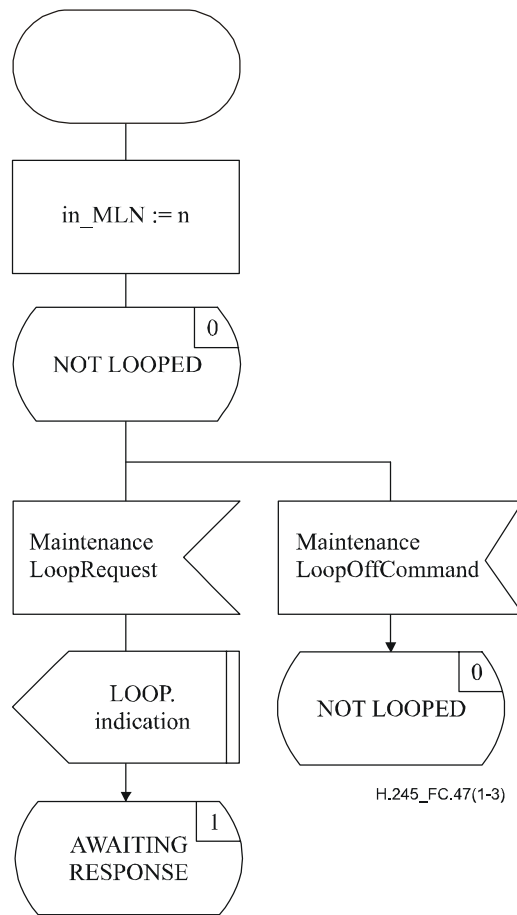


Figure C.47/H.245 – Diagramme SDL de l'entité MLSE entrante (feuille 1 de 3)

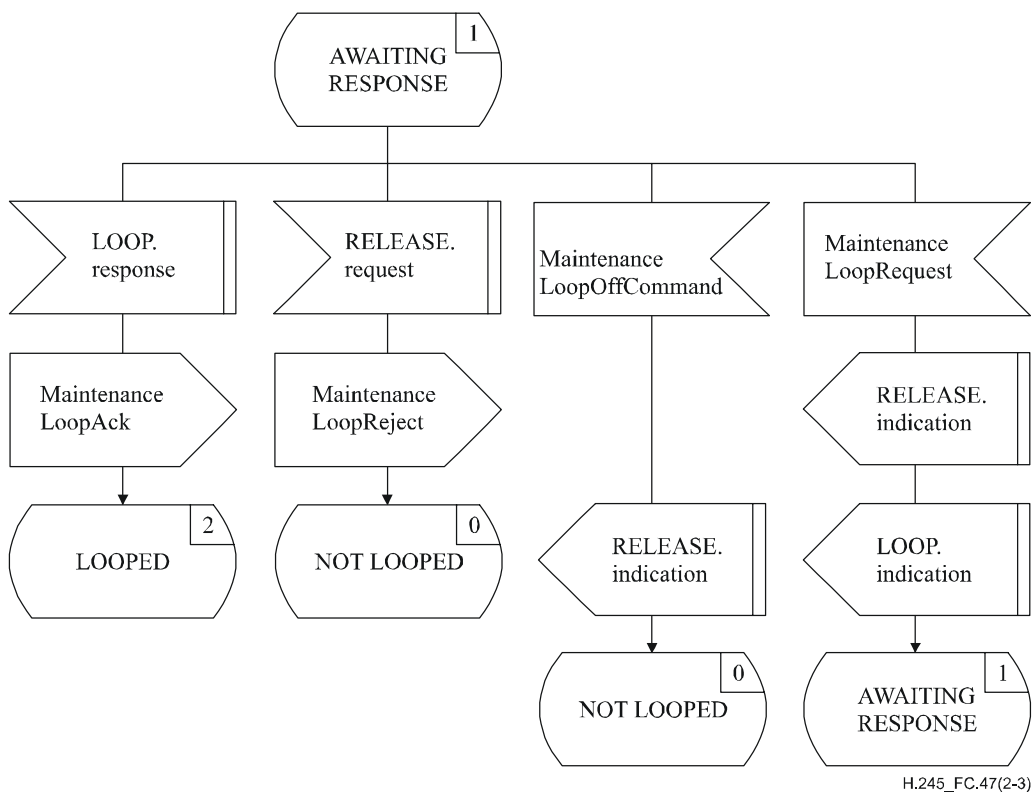


Figure C.47/H.245 – Diagramme SDL de l'entité MLSE entrante (feuille 2 de 3)

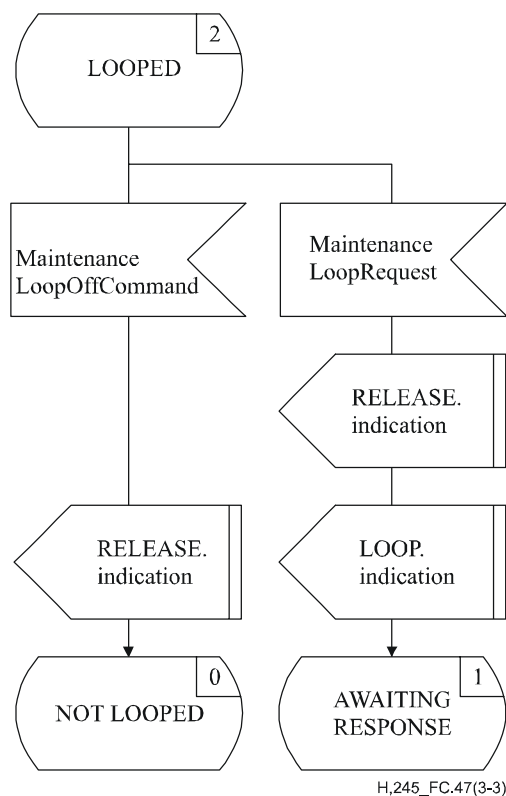


Figure C.47/H.245 – Description SDL de l'entité MLSE entrante (feuille 3 de 3)

Annexe D

Affectations d'identificateur d'objet

Le Tableau D.1 indique l'affectation des identificateurs d'objet définis en vue de leur utilisation dans la présente Recommandation.

Tableau D.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 cette Recommandation utilisée comme protocole de commande du système multimédia. Ceci indique la première version de la présente Recommandation.
{itu-t (0) recommendation (0) h (8) 245 version (0) 2}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la deuxième version de la présente Recommandation.

Tableau D.1/H.245

Valeur d'identificateur d'objet	Description
{itu-t (0) recommandation (0) h (8) 245 version (0) 3}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la troisième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 4}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la quatrième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 5}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la cinquième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 6}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la sixième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 7}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la septième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 8}	Cet identificateur d'objet est utilisé pour indiquer la version de cette Recommandation utilisée comme protocole de commande du système multimédia. A ce jour, onze versions normalisées ont été définies. Ceci indique la huitième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 9}	Cet identificateur d'objet est utilisé pour indiquer la version de la présente Recommandation qui sert de protocole de commande du système multimédia. Onze versions normalisées sont actuellement définies. Celle-ci indique la neuvième version de la présente Recommandation.
{itu-t (0) recommandation (0) h (8) 245 version (0) 10}	Cet identificateur d'objet est utilisé pour indiquer la version de la présente Recommandation qui sert de protocole de commande du système multimédia. Onze versions normalisées sont actuellement définies. Celle-ci indique la dixième version de la présente Recommandation.

Tableau D.1/H.245

Valeur d'identificateur d'objet	Description
{itu-t (0) recommendation (0) h (8) 245 version (0) 11}	Cet identificateur d'objet est utilisé pour indiquer la version de la présente Recommandation qui sert de protocole de commande du système multimédia. Onze versions normalisées sont actuellement définies. Celle-ci indique la onzième version de la présente Recommandation.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) video (0) ISO/IEC 14496-2 (0)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique ISO/CEI 14496-2. Cette capacité est définie dans l'Annexe E.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (0) ISO/IEC 14496-3 (0)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique ISO/CEI 14496-3. Cette capacité est définie dans l'Annexe H.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) amr (1)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique du codec vocal multidébit adaptatif GSM. Cette capacité est définie dans l'Annexe I.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) acelp (2)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique du codec vocal TIA/EIA/ANSI IS-136 ACELP. Cette capacité est définie dans l'Annexe J.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) us1 (3)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique du codec vocal TIA/EIA/ANSI IS-136 US1. Cette capacité est définie dans l'Annexe K.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) is127evrc (4)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique du codec à débit variable amélioré TIA/EIA IS-127. Cette capacité est définie dans l'Annexe L.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique pour l'ISO/CEI 13818-7. Cette capacité est définie dans l'Annexe M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) rfc3389 (6)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique pour le bruit de confort de signalisation spécifié dans RFC 3389. Cette capacité est définie dans l'Annexe N.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) L-16 (7)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique pour le codec 16 bits linéaire à débit variable de base d'échantillons tel que défini dans IETF RFC 1890. Cette capacité est définie dans l'Annexe O.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) bounded-audio-stream (8)}	Cet identificateur d'objet est utilisé pour indiquer la capacité de flux audio borné en tant que capacité générique. Cette capacité est définie dans l'Annexe P.

Tableau D.1/H.245

Valeur d'identificateur d'objet	Description
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) dataq (2) ISO/IEC 14496-1 (0)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique ISO/CEI 14496-1. Cette capacité est définie dans l'Annexe G.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) control (3) logical-channel-bit-rate-management (0)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique de gestion de débit de voie logique. Cette capacité est définie dans l'Annexe F.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique ISO/CEI 13818-7. Cette capacité est définie dans l'Annexe M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) itu-r bs.1196 (6)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique UIT-R BS.1196. Cette capacité est définie dans l'Annexe M.
{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) Nx64 (1)}	Cet identificateur d'objet est utilisé pour indiquer la capacité générique de transmission de données non codées dans une voie à Nx64, comme décrit dans l'Annexe Q.

Annexe E

Définitions relatives à la capacité ISO/CEI 14496-2

Le Tableau E.1 définit l'identificateur de la capacité ISO/CEI 14496-2 [49]. Les paramètres définis ne seront inclus que comme **genericVideoCapability** dans la structure **VideoCapability** et comme **genericVideoMode** dans la structure **VideoMode**. Les Tableaux E.2 à E.6 définissent les paramètres de capacité associés.

Une même instanciation **profileAndLevel** ISO/CEI 14496-2 peut prendre en charge plusieurs objets vidéo. Chaque objet vidéo est acheminé sous forme de flux élémentaire dans sa propre voie logique séparée. Etant donné qu'il est possible de transmettre activement plusieurs environnements vidéo ISO/CEI 14496-2 simultanément et que chacun d'eux peut être construit à partir de plusieurs flux d'objet, il est nécessaire d'avoir un mécanisme permettant d'indiquer les flux d'objet qui sont associés les uns aux autres dans un même environnement vidéo ISO/CEI 14496-2. Cette association sera exécutée au moyen du mécanisme **forwardLogicalChannelDependency** de **OpenLogicalChannel** chaque fois que plusieurs objets vidéo sont utilisés pour le même environnement vidéo ISO/CEI 14496-2. Tous les objets vidéo associés les uns aux autres dans un environnement vidéo ISO/CEI 14496-2 auront les mêmes profil et niveau grâce à l'indication de la même valeur **profileAndLevel** au moment de l'ouverture des voies logiques. Si une voie logique a été ouverte avec une indication de dépendance vis-à-vis d'une autre voie logique de cette manière et si cette dernière voie logique est fermée, les autres voies logiques ouvertes qui avaient été regroupées auparavant par une chaîne de liens **forwardLogicalChannelDependency** resteront regroupées sur le plan logique dans un même environnement vidéo ISO/CEI 14496-2.

Tableau E.1/H.245 – Identificateur pour la capacité ISO/CEI 14496-2

Nom de la capacité	ISO/CEI 14496-2
Classe de la capacité	Codec vidéo
Type de l'identificateur de la capacité	Standard
Valeur de l'identificateur de la capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) video (0) ISO/IEC 14496-2 (0)}
maxBitRate	Le champ maxBitRate sera toujours inclus
nonCollapsingRaw	Ce champ ne sera pas inclus
transport	Ce champ ne sera pas inclus

Tableau E.2/H.245 – Profil et niveau pour la capacité ISO/CEI 14496-2

Nom du paramètre	profileAndLevel
Description du paramètre	C'est un GenericParameter de type nonCollapsing. profileAndLevel indique la capacité de traitement des différents profils en combinaison avec le niveau (voir Tableau G-.1 'Tableau FLC pour indication de profil et de niveau' de l'ISO/CEI 14496-2).
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Obligatoire
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau E.3/H.245 – Paramètre d'objet pour la capacité ISO/CEI 14496-2

Nom du paramètre	object
Description du paramètre	C'est un GenericParameter de type nonCollapsing. object indique l'ensemble des outils devant être utilisés par le décodeur du flux binaire contenu dans la voie logique (voir Tableau 6-10 'Tableau FLC pour indication du type d'objet vidéo' de l'ISO/CEI 14496-2).
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif. Ne sera pas présent pour l'échange de capacités. Sera présent pour la signalisation de voie logique. Pourra être présent pour la demande de mode.
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau E.4/H.245 – Information relative à la configuration du décodeur pour la capacité ISO/CEI 14496-2

Nom du paramètre	decoderConfigurationInformation
Description du paramètre	C'est un GenericParameter de type nonCollapsing. decoderConfigurationInformation indique comment configurer le décodeur pour un objet donné (flux binaire) (voir § 6.2.1 'Codes de démarrage' et les § K.3.1 'VideoObject' à K.3.4 'FaceObject' de l'ISO/CEI 14496-2).
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif. Ne sera pas présent pour l'échange de capacités et la demande de mode. Pourra être présent pour la signalisation de voie logique.
Type du paramètre	octetString
Remplace	–

Tableau E.5/H.245 – Rang de dessin pour la capacité ISO/CEI 14496-2

Nom du paramètre	drawingOrder
Description du paramètre	C'est un GenericParameter de type nonCollapsing. drawingOrder indique le rang pour le dessin d'un objet vidéo dans une composition d'objets vidéo (éventuellement recouverts). L'objet vidéo ayant la plus petite valeur de drawingOrder sera dessiné en premier. Si plusieurs objets vidéo ont la même valeur de drawingOrder, l'objet correspondant à la voie logique ayant le plus petit numéro de voie logique sera dessiné en premier. Si le paramètre drawingOrder est absent pendant la signalisation de voie logique, on suppose qu'il a la valeur 32768.
Valeur de l'identificateur du paramètre	3
Statut du paramètre	Facultatif. Ne sera pas présent pour l'échange de capacités et la demande de mode. Pourra être présent pour la signalisation de voie logique.
Type du paramètre	unsignedMax. Doit être compris entre 0..65535.
Remplace	–

Tableau E.6/H.245 – Paramètre visualBackChannelHandle pour capacité ISO/CEI 14496-2

Nom du paramètre	visualBackChannelHandle
Description du paramètre	Il s'agit d'un paramètre générique de type Collapsing. La présence de ce paramètre indique que l'émetteur reçoit sur la voie de retour ou que le récepteur émet sur la voie de retour des messages de type ISO/CEI 14496-2.
Valeur de l'identificateur du paramètre	4
Statut du paramètre	Peut être présent pour l'échange de capacités, la signalisation de voie logique ou la demande de mode.
Type du paramètre	Logique
Remplace	–

Annexe F

Définitions relatives à la capacité de gestion du débit de voie logique

Le Tableau F.1 définit l'identificateur de la capacité de gestion de débit. Les paramètres définis, qui renseignent sur les messages de gestion de débit que le terminal prend en charge, ne seront inclus que comme **genericControlCapability** dans la structure **Capability**. Les Tableaux F.2 à F.4 définissent les paramètres de capacité associés.

Tableau F.1/H.245 – Identificateur de la capacité de gestion du débit de voie logique

Nom de la capacité	Gestion du débit de voie logique H.245
Classe de la capacité	Commande
Type de l'identificateur de la capacité	Standard
Valeur de l'identificateur de la capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) control (3) logical-channel-bitrate-management (0)}
maxBitRate	Le champ maxBitRate ne sera pas inclus
nonCollapsingRaw	Ce champ ne sera pas inclus
transport	Ce champ ne sera pas inclus

Tableau F.2/H.245 – Paramètre de capacité de commande de débit pour la gestion de débit binaire

Nom du paramètre	Flow Control Capability
Description du paramètre	Paramètre générique de type collapsing. La présence de ce paramètre indique la capacité de prise en charge du message FlowControlIndication.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau F.3/H.245 – Paramètre de capacité de modification du débit de voie logique pour la gestion de débit

Nom du paramètre	Logical Channel Bit-Rate Change Capability
Description du paramètre	Paramètre générique de type collapsing. La présence de ce paramètre indique la capacité de prise en charge de la procédure de modification du débit de voie logique, qui utilise les messages LogicalChannelRateRequest, LogicalChannelRateAcknowledge, LogicalChannelRateReject et LogicalChannelRateRelease.
Valeur de l'identificateur du paramètre	1

Tableau F.3/H.245 – Paramètre de capacité de modification du débit de voie logique pour la gestion de débit

Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau F.4/H.245 – Paramètre de fréquence RTCP pour la gestion de débit

Nom du paramètre	RTCP Frequency Capability
Description du paramètre	Paramètre générique de type collapsing. Ce paramètre indique la fréquence à laquelle le terminal peut envoyer des rapports RTCP.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	unsigned32Min
Remplace	–

Annexe G

Définitions relatives à la capacité ISO/CEI 14496-1

Le Tableau G.1 définit l'identificateur de la capacité ISO/CEI 14496-1 [48]. Les Tableaux G.2 à G.6 définissent les paramètres de capacité associés. Les paramètres définis ne seront inclus que comme **genericDataCapability** dans la structure **DataCapability** et comme **genericDataMode** dans la structure **DataMode**. Pour l'échange de capacités, le type de flux streamType et l'indication de profil et de niveau profileAndLevel seront spécifiés et le type d'objet objectType pourra être spécifié. Au moment de l'ouverture d'une voie logique (vers l'avant ou vers l'arrière) l'identificateur ES_ID ou le descripteur d'objet objectDescriptor sera spécifié.

Des précisions sur l'utilisation de la capacité générique ISO/CEI 14496-1 sont données dans l'Annexe F/H.324.

G.1 Identificateur de capacité

Tableau G.1/H.245 – Identificateur de capacité ISO/CEI 14496-1

Nom de la capacité	ISO/CEI 14496-1 Capability
Classe de la capacité	Application de données
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) ISO/IEC 14496-1 (0)}
maxBitRate	Ce champ doit être inclus pour indiquer le débit maximal d'un flux de données simple ISO/CEI 14496

Tableau G.1/H.245 – Identificateur de capacité ISO/CEI 14496-1

nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ doit être inclus pour indiquer le protocole de protection contre les erreurs pour un flux de données ISO/CEI 14496 spécifique transporté sur une voie logique

G.2 Paramètres de capacité utilisés pour les négociations de capacité et pour la signalisation de voie logique

Tableau G.2/H.245 – Paramètre de capacité type de flux

Nom du paramètre	streamType
Description du paramètre	C'est un paramètre générique de type nonCollapsing. StreamType indique le type d'ISO/CEI 14496 auquel se réfère une instance spécifique de la capacité générique ISO/CEI 14496-1 telle qu'elle apparaît dans le Tableau 9 ("streamType Values") de l'ISO/CEI 14496-1.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Facultatif. Doit être présent pour l'échange de capacités mais pas pour la signalisation de voie logique ou la demande de mode.
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau G.3/H.245 – Paramètre de capacité profil et niveau

Nom du paramètre	profileAndLevel
Description du paramètre	C'est un paramètre générique de type nonCollapsing. Profil et niveau indiquent la capacité de traiter les profils particuliers en combinaison avec le niveau comme indiqué dans le: <ul style="list-style-type: none"> • Tableau 3 de l'ISO/CEI 14496-1 ("ODProfileLevelIndication Values") pour le paramètre streamType = 0x01 • Tableau 4 de l'ISO/CEI 14496-1 ("sceneProfileLevelIndication Values") pour le paramètre streamType = 0x03 • Tableau 5 de l'ISO/CEI 14496-1 ("audioProfileLevelIndication Values") pour le paramètre streamType = 0x05 • Tableau 6 de l'ISO/CEI 14496-1 ("visualProfileLevelIndication Values") pour le paramètre streamType = 0x04
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif. Doit être présent pour l'échange de capacités mais pas pour la signalisation de voie logique ou la demande de mode.
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau G.4/H.245 – Paramètre de capacité objectType

Nom du paramètre	objectType
Description du paramètre	<p>Il s'agit d'un GenericParameter de type nonCollapsing.</p> <p>objectType indique l'ensemble des outils devant être utilisés par le décodeur du flux binaire contenu dans la voie logique, comme indiqué dans:</p> <ul style="list-style-type: none"> • le Tableau 8 de l'ISO/CEI 14496-1 ("Valeurs de l'indication objectTypeIndication") pour le paramètre streamType = 0x04 ou 0x05 • le Tableau 7 de l'ISO/CEI 14496-1 ("Valeurs de l'indication graphicsProfileLevelIndication") pour le paramètre streamType = 0x03 <p>Pour toutes les autres valeurs de streamType, objectType n'est pas défini et ne sera donc pas utilisé.</p>
Valeur de l'identificateur du paramètre	2
Statut du paramètre	<p>Facultatif.</p> <p>Pour streamType = 0x04 ou 0x05, ne sera pas présent pour l'échange de capacités. Sera présent pour la signalisation de voie logique. Pourra être présent pour la demande de mode.</p> <p>Pour streamType = 0x03, sera présent pour l'échange de capacités et pour la signalisation de voie logique. Pourra être présent pour la demande de mode.</p> <p>Pour les autres valeurs de streamType, ne sera pas présent.</p>
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

G.3 Paramètres de capacité utilisés pour la signalisation de voie logique seulement

Tableau G.5/H.245 – Paramètre de capacité descripteur d'objet

Nom du paramètre	objectDescriptor
Description du paramètre	<p>C'est un paramètre générique de type nonCollapsing.</p> <p>objectDescriptor contient une chaîne d'octets qui fournit toutes les informations nécessaires pour configurer le décodeur pour un flux binaire donné dans une voie logique (voir l'ISO/CEI 14496-1). Il ne doit contenir des informations que pour un seul flux élémentaire.</p>
Valeur de l'identificateur du paramètre	3
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Peut être présent pour la signalisation de voie logique.
Type du paramètre	octetString
Remplace	–

Tableau G.6/H.245 – Paramètre de capacité ES_ID

Nom du paramètre	ES_ID
Description du paramètre	C'est un paramètre générique de type nonCollapsing. ES_ID indique l'identificateur ID de flux élémentaire contenu dans une voie logique spécifique auquel d'autres flux binaires ISO/CEI 14496 peuvent faire référence. Pour le descripteur InitialObjectDescriptor, ES_ID doit être mis à "0" (zéro).
Valeur de l'identificateur du paramètre	4
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités. Peut être présent pour la signalisation de voie logique. Doit être présent pour la demande de mode.
Type du paramètre	unsignedMax. Doit être compris entre 0..65535.
Remplace	–

Annexe H

Définition des capacités ISO/CEI 14496-3

Le Tableau H.1 définit l'identificateur des capacités ISO/CEI 14496-3 [50] et ISO/CEI 14496-3/Amd.1 [51]. Les Tableaux H.2 à H.11 définissent les paramètres de capacité associés pour l'ISO/CEI 14496-3, qu'il ne faudra inclure qu'en tant que genericAudioCapability dans la structure AudioCapability et en tant que genericAudioMode dans la structure AudioMode. Pour l'échange de capacités, les paramètres profileAndLevel, formatType et maxA1-sduAudioFrames doivent être présents, audioObjectType et maxAudioObjects peuvent être présents et tous les autres doivent être absents. Si formatType indique le format de flux de transport ISO/CEI 14496-3, maxAudioObjects doit être présent pour l'échange de capacités. A l'ouverture d'une voie logique (directe ou inverse), profileAndLevel, formatType et audioObjectType doivent être présents et tous les autres paramètres peuvent être spécifiés. Pour la demande de mode, profileAndLevel et formatType doivent être présents et audioObjectType peut être spécifié.

Le paramètre profileAndLevel de l'ISO/CEI 14496-3 et de l'ISO/CEI 14496-3/Amd.1 peut prendre en charge plusieurs types d'objets audio. L'objet audio doit être transporté sous l'un des deux formats de flux binaire: le format de données brutes ou le format de flux de transport l'ISO/CEI 14496-3. formatType indique le choix du type de format. Dans les applications utilisant la transmission multidébit ou la transmission échelonnée, il est utile de permettre des modifications de la structure des objets audio dans une voie logique. On peut le faire au moyen du format MPEG-4/Audio qui permet de changer la configuration du flux trame par trame. En transmission à faible débit, le format de données brutes permet de diminuer la redondance due à la transmission de la configuration du flux avec chaque trame.

Tableau H.1/H.245 – Identificateur de capacité ISO/CEI 14496-3

Nom de la capacité	ISO/CEI 14496-3
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 14496-3 (0)}
maxBitRate	Ce champ doit toujours être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau H.2/H.245 – Profil et niveau pour capacité ISO/CEI 14496-3

Nom du paramètre	profileAndLevel
Description du paramètre	C'est un paramètre générique de type nonCollapsing. profileAndLevel indique la capacité de traiter les profils particuliers en combinaison avec le niveau comme indiqué dans l'ISO/CEI 14496-1 et l'ISO/CEI 14496-1/Amd.1.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Obligatoire
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau H.3/H.245 – Type de format pour capacité ISO/CEI 14496-3

Nom du paramètre	formatType
Description du paramètre	C'est un paramètre générique de type nonCollapsing. formatType indique, de la manière suivante, le choix du type de format de flux binaire d'un objet audio entre le format de données brutes et le format audio: <ul style="list-style-type: none"> • 0: format de données brutes (ISO/CEI 14496-3 et ISO/CEI 14496-3/Amd.1) • 1: le format défini en tant que Low-overhead MPEG-4 Audio Transport Multiplex (LATM) dans l'ISO/CEI 14496-3/Amd.1.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Obligatoire
Type du paramètre	Logique
Remplace	–

Tableau H.4/H.245 – Nombre maximal de trames audio par A1-sdu pour capacité ISO/CEI 14496-3

Nom du paramètre	maxA1-sduAudioFrames
Description du paramètre	C'est un paramètre générique de type collapsing (fédérable) qui indique le nombre maximal de trames audio par paquet ou unité AL-SDU.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif. Doit être présent pour l'échange de capacités. Doit être présent pour la signalisation de voie logique. Ne doit pas être présent pour la demande de mode.
Type du paramètre	unsignedMin. Doit être compris entre 1..256.
Remplace	–

Tableau H.5/H.245 – Type d'objet audio pour capacité ISO/CEI 14496-3

Nom du paramètre	audioObjectType
Description du paramètre	C'est un paramètre générique de type nonCollapsing (non fédérable). audioObjectType indique l'ensemble d'outils que doit utiliser le décodeur du flux binaire contenu dans la voie logique comme indiqué dans l'ISO/CEI 14496-3/Amd.1. Il peut être utilisé pour limiter la capacité au sein du profileAndLevel spécifié dans l'échange de capacités.
Valeur de l'identificateur du paramètre	3
Statut du paramètre	Facultatif. Peut être présent pour l'échange de capacités. Doit être présent pour la signalisation de voie logique. Peut être présent pour la demande de mode.
Type du paramètre	unsignedMax. Doit être compris entre 0..31.
Remplace	–

Tableau H.6/H.245 – Configuration audio spécifique pour capacité ISO/CEI 14496-3

Nom du paramètre	audioSpecificConfig
Description du paramètre	C'est un paramètre générique de type nonCollapsing. audioSpecificConfig indique la manière de configurer le décodeur pour un objet particulier (voir l'ISO/CEI 14496-3/Amd.1).
Valeur de l'identificateur du paramètre	4
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Doit être présent pour la signalisation de voie logique si formatType est zéro (format de données brutes). Sinon, il ne doit pas être présent pour la signalisation de voie logique.
Type du paramètre	octetString
Remplace	–

Tableau H.7/H.245 – Nombre maximal d'objets audio pour capacité ISO/CEI 14496-3

Nom du paramètre	maxAudioObjects
Description du paramètre	C'est un paramètre générique de type Collapsing qui indique le nombre maximal d'objets audio multiplexés dans la charge utile audio.
Valeur de l'identificateur du paramètre	5
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la signalisation de voie logique si formatType est zéro (format de données brutes). Sinon, il doit être présent pour l'échange de capacités et la signalisation de voie logique. Ne doit pas être présent pour la demande de mode.
Type du paramètre	unsignedMin. Doit être compris entre 1..16.
Remplace	–

Tableau H.8/H.245 – Configuration multiplex présente pour capacité ISO/CEI 14496-3

Nom du paramètre	muxConfigPresent
Description du paramètre	C'est un paramètre générique de type nonCollapsing. muxConfigPresent indique si la configuration est multiplexée dans la charge utile audio elle-même comme indiqué dans l'ISO/CEI 14496-3/Amd.1: <ul style="list-style-type: none"> • 0: la configuration de la charge utile audio (streamMuxConfig) n'est pas multiplexée dans la charge utile audio; • 1: streamMuxConfig est multiplexé dans la charge utile audio.
Valeur de l'identificateur du paramètre	6
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Doit être présent pour la signalisation de voie logique si formatType est 1 (format LATM). Sinon, il ne doit pas être présent pour la signalisation de voie logique.
Type du paramètre	Logique
Remplace	–

Tableau H.9/H.245 – Données EP présentes pour capacité ISO/CEI 14496-3

Nom du paramètre	EP_DataPresent
Description du paramètre	C'est un paramètre générique de type nonCollapsing. EP_DataPresent indique si la charge utile audio a une résilience d'erreur pour l'erreur sur les bits (pas pour la perte de paquets) comme indiqué dans l'ISO/CEI 14496-3/Amd.1: <ul style="list-style-type: none"> • 0: la charge utile audio n'a pas de résilience des erreurs; • 1: la charge utile a une résilience des erreurs. La configuration pour l'outil EP de l'ISO/CEI 14496-3/Amd.1 (ErrorProtection_SpecificConfig) peut être présente pour la signalisation de voie logique.

Tableau H.9/H.245 – Données EP présentes pour capacité ISO/CEI 14496-3

Nom du paramètre	EP_DataPresent
Valeur de l'identificateur du paramètre	7
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Doit être présent pour la signalisation de voie logique si formatType est 1 (format LATM). Sinon, il ne doit pas être présent pour la signalisation de voie logique.
Type du paramètre	Logique
Remplace	–

Tableau H.10/H.245 – Configuration multiplex de flux pour capacité ISO/CEI 14496-3

Nom du paramètre	streamMuxConfig
Description du paramètre	C'est un paramètre générique de type nonCollapsing. streamMuxConfig indique la configuration de la charge utile audio comme indiqué dans l'ISO/CEI 14496-3/Amd.1.
Valeur de l'identificateur du paramètre	8
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Doit être présent pour la signalisation de voie logique si formatType est 1 (format LATM). Sinon, il ne doit pas être présent pour la signalisation de voie logique.
Type du paramètre	octetString
Remplace	–

Tableau H.11/H.245 – Configuration spécifique de protection contre les erreurs pour capacité ISO/CEI 14496-3

Nom du paramètre	errorProtection_SpecificConfig
Description du paramètre	C'est un paramètre générique de type nonCollapsing. errorProtection_SpecificConfig indique la manière de configurer l'outil EP de l'ISO/CEI 14496-3/Amd.1 comme indiqué dans la description LATM EP_MuxElement() de l'ISO/CEI 14496-3/Amd.1.
Valeur de l'identificateur du paramètre	9
Statut du paramètre	Facultatif. Ne doit pas être présent pour l'échange de capacités et la demande de mode. Doit être présent pour la signalisation de voie logique si formatType est 1 (format LATM). Sinon, il ne doit pas être présent pour la signalisation de voie logique.
Type du paramètre	octetString
Remplace	–

Annexe I

Définition des capacités multidébit adaptatif GSM

Le Tableau I.1 définit l'identificateur de capacités multidébit adaptatif (AMR, *adaptive multi-rate*) GSM. Les Tableaux I.2 à I.7 définissent les paramètres de capacité associés. Les spécifications qui s'appliquent figurent dans les références [58], [69], [70], [71], [72], [73], [74] et [75].

Le paragraphe I.1 définit la signalisation du mode et le groupage par paquets des trames vocales dans la structure d'octet.

Tableau I.1/H.245 – Identificateur de capacité GSM AMR

Nom de la capacité	AMR
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) amr (1)}
maxBitRate	Doit être 122
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau I.2/H.245 – Paramètre de capacité GSM AMR – Nombre max. de trames audio par Al-sdu

Nom du paramètre	maxAl-sduAudioFrames
Description du paramètre	C'est un paramètre générique de type collapsing qui spécifie le nombre maximal de trames audio par unité AL-SDU.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Doit être présent pour l'échange de capacités et la signalisation de canal logique. Ne doit pas être présent pour la demande de mode.
Type du paramètre	unsignedMin
Remplace	–

Tableau I.3/H.245 – Paramètre de capacité GSM AMR – Débit

Nom du paramètre	bitRate
Description du paramètre	C'est un paramètre générique de type nonCollapsing qui spécifie le débit AMR. Il ne doit être utilisé que dans les demandes de mode. 0 = 4.75, 1 = 5.15, 2 = 5.90, 3 = 6.70, 4 = 7.40, 5 = 7.95, 6 = 10.2, 7 = 12.2
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif
Type du paramètre	unsignedMin
Remplace	–

Tableau I.4/H.245 – Paramètre de capacité GSM AMR – Bruit de confort GSM AMR

Nom du paramètre	gsmAmrComfortNoise
Description du paramètre	C'est un paramètre générique de type collapsing qui spécifie qu'il faut utiliser le bruit de confort GSM AMR dans la demande de mode. Ce paramètre ne sera utilisé que dans les demandes de mode et non dans les capacités étant donné que celle-ci est obligatoire.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau I.5/H.245 – Paramètre de capacité GSM AMR – Bruit de confort GSM EFR

Nom du paramètre	gsmEfrComfortNoise
Description du paramètre	C'est un paramètre générique de type collapsing. Dans une capacité, ce paramètre spécifie s'il y a une capacité de bruit de confort GSM EFR ou non. Dans une demande de mode, le paramètre spécifie que le bruit de confort GSM EFR est requis.
Valeur de l'identificateur du paramètre	3
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau I.6/H.245 – Paramètre de capacité GSM AMR – Bruit de confort IS-641

Nom du paramètre	is-641ComfortNoise
Description du paramètre	C'est un paramètre générique de type collapsing. Dans une capacité, ce paramètre spécifie s'il y a une capacité de bruit de confort IS-641 ou non. Dans une demande de mode, le paramètre spécifie que le bruit de confort IS-641 est requis.
Valeur de l'identificateur du paramètre	4
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau I.7/H.245 – Paramètre de capacité GSM AMR – Bruit de confort PDC EFR

Nom du paramètre	pdceFRComfortNoise
Description du paramètre	C'est un paramètre générique de type collapsing. Dans une capacité, ce paramètre spécifie s'il y a une capacité de bruit de confort PDC EFR ou non. Le PDC EFR est un vocodeur ACELP à 6,7 kbit/s spécifié au § 5.4 du [74]. Dans une demande de mode, le paramètre spécifie que le bruit de confort PDC EFR est requis.

Tableau I.7/H.245 – Paramètre de capacité GSM AMR – Bruit de confort PDC EFR

Nom du paramètre	pdceFRComfortNoise
Valeur de l'identificateur du paramètre	5
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

I.1 Définition de la signalisation de mode et du bourrage de bits pour réaliser l'alignement des octets

Le mode de signalisation AMR utilisé dans les systèmes mobiles est partiellement fondé sur la signalisation extérieure qui n'est pas définie dans la spécification du codec vocal AMR. Pour les besoins de compatibilité avec les systèmes mobiles, le présent paragraphe définit la signalisation de mode nécessaire pour utiliser le mode AMR dans le cadre des Recommandations UIT-T de la série H. La taille des trames vocales du codec AMR dans les différents modes n'est pas un multiple de huit, raison pour laquelle il faut recourir au bourrage de bits pour obtenir une structure d'octet.

Il convient de noter qu'à l'avenir, le contenu du présent paragraphe pourra être modifié de manière à se référer à la documentation de l'ETSI ou à d'autres documents de normalisation appropriés.

Le Tableau I.10 montre le mappage entre tous les modes AMR et les indices de mode spécifiques $mi(k)$. Les indices de mode sont également réservés aux trames de suppression des silences utilisées dans différents systèmes. Les Tableaux I.11 à I.14 spécifient les formats qui s'appliquent à ceux-ci. Le Tableau I.15 spécifie une trame de non-transmission.

Les bits que fournit le codeur vocal AMR, $\{s(1),s(2),\dots,s(K_s)\}$, seront réarrangés en fonction de leur importance subjective avant d'être alignés en octets. Les Tableaux I.16 à I.23 définissent le réarrangement correct pour les modes de codec vocal 12,2 kbit/s, 10,2 kbit/s, 7,95 kbit/s, 7,40 kbit/s, 6,70 kbit/s, 5,90 kbit/s, 5,15 kbit/s et 4,75 kbit/s, respectivement. Dans les tableaux, les paramètres des codecs vocaux sont numérotés dans l'ordre de leur remise par le codeur vocal correspondant suivant GSM 06.90. Les bits réarrangés sont étiquetés $\{d(0),d(1),\dots,d(K_d - 1)\}$, et sont définis dans l'ordre d'importance décroissant. L'indice K_d se rapporte au nombre de bits produits par le codeur vocal; voir le Tableau I.8.

Tableau I.8/H.245 – Nombre de bits vocaux dans différents modes AMR

Mode du Codec	Nombre de bits vocaux remis par bloc (K_d)
AMR12.2	244
AMR10.2	204
AMR7.95	159
AMR7.4	148
AMR6.7	134
AMR5.9	118
AMR5.15	103
AMR4.75	95

L'algorithme de classement est en pseudo-code tel que:

pour $j =$ de 0 à $K_d - 1$
 $d(j) = s(\text{table}(j) + 1)$

où $\text{table}(j)$ est lu ligne par ligne, de gauche à droite.

Dès lors la structure d'octet $b_n(k)$ pour chaque mode de codec AMR est définie de la manière suivante:

Nombre de bits de bourrage: $K_s = 8 * N - K_d - K_i$, où K_i est le nombre de bits d'indice de mode

Octet[0]: $b_0(k) = m_i(k)$, pour $k = 0, 1, 2, 3$ (indice de mode)

$b_0(k) = d(k - 4)$, pour $k = 4, 5, 6, 7$

Octet[m]: $b_m(k) = d(8 * m - 4 + k)$, pour $k = 0, \dots, 7$ et $0 < m < N - 1$

Octet[N - 1]: $b_{N-1}(k) = d(8 * (N - 1) - 4 + k)$, pour $k = 0, \dots, 7 - K_s$

If $K_s > 0$

$b_{N-1}(k) = \text{UB}$, pour $k = 8 - K_s, \dots, 7$

Tableau I.9/H.245 – Exemple de mappage entre le mode codage vocal AMR à 6,7 kbit/s et les indices de mode

Octet	Structure d'octet							MSB	LSB
b_0	d(3)	d(2)	d(1)	d(0)	0	0	1	1	
b_1	d(11)	d(10)	d(9)	d(8)	d(7)	d(6)	d(5)	d(4)	
b_2	d(12)	
b_{17}	UB	UB	UB	UB	UB	UB	d(133)	d(132)	

Tableau I.10/H.245 – Mappage entre les modes de codage vocal AMR définis dans GSM 06.90 et les bits d'indice de mode dans les octets transmis

Mode_index (4 bits)	Attribution de nom dans GSM 06.90 et GSM 06.92
0 (Amr4-75k)	Mode 4,75 kbit/s
1 (Amr5-15k)	Mode 5,15 kbit/s
2 (Amr5-90k)	Mode 5,90 kbit/s
3 (Amr6-70k)	Mode 6,70 kbit/s (PDC-EFR)
4 (Amr7-40k)	Mode 7,40 kbit/s (IS-641)
5 (Amr7-95k)	Mode 7,95 kbit/s
6 (Amr10-2k)	Mode 10,2 kbit/s
7 (Amr12-2k)	Mode 12,2 kbit/s (GSM EFR)
8	Trame de bruit de confort GSM AMR (obligatoire)
9	Trame de bruit de confort GSM EFR (facultatif)
10	Trame de bruit de confort IS-641 (facultatif)
11	Trame de bruit de confort PDC EFR (facultatif)
12-14	Pour utilisation future
15	Pas de transmission

**Tableau I.11/H.245 – Mappage entre les bits descripteurs de
bruit de confort GSM 06.92 et les octets pour l'indice de mode 8**
(Les bits s1 à s35 se réfèrent à GSM 06.92)

Octets transmis	MSB	Mappage des bits						LSB
1	Indice du 1 ^{er} sous-vecteur LSF s4	Indice de vecteur de référence de LSF s3 s2 s1			Indice de mode mi(3) mi(2) mi(1) mi(0)			
2	Indice du 2 ^e sous-vecteur LSF s12	Indice du 1 ^{er} sous-vecteur LSF s11 s10 s9 s8 s7 s6 s5						
3	Indice du 2 ^e sous-vecteur LSF s20 s19 s18 s17 s16 s15 s14 s13							
4	Indice du 3 ^e sous-vecteur LSF s28 s27 s26 s25 s24 s23 s22 s21							
5	Bit de type SID t1	Energie de trame s35 s34 s33 s32 s31 s30						Indice du 3 ^e sous-vecteur LSF s29
6	Bits de bourrage UB UB UB UB				Indication du mode vocal smi(2) smi(1) smi(0)			

Définition de bits descripteurs additionnels nécessaires pour le descripteur de silence du Tableau I.11:

SID-type (t1) est {0=SID_FIRST, 1=SID_UPDATE}

Speech Mode Indication (smi(0)-smi(2)) est le mode vocal conformément aux huit premières entrées du tableau Mode_Index.

Tableau I.12/H.245 – Mappage entre les bits descripteurs d'insertion de silence GSM 06.60 (paramètres également décrits dans GSM 06.62) et les octets pour l'indice de mode 9
(Les bits s1 à s91 se réfèrent à GSM 06.60)

Octets transmis	MSB	Mappage des bits						LSB
1	Indice de 1 ^{er} sous-matrice LSF s4 s3 s2 s1				Indice de mode mi(3) mi(2) mi(1) mi(0)			
2	Indice de 2 ^e sous-matrice LSF s12 s11 s10 s9 s8					Indice de 1 ^{er} sous-matrice LSF s7 s6 s5		
3	Indice de 3 ^e sous-matrice LSF s20 s19 s18 s17 s16					Indice de 2 ^e sous-matrice LSF s15 s14 s13		
4	Indice de 4 ^e sous-matrice LSF s28 s27 s26 s25				Signe de 3 ^e sous-matrice LSF s24	Indice de 3 ^e sous-matrice LSF s23 s22 s21		
5	Indice de 5 ^e sous-matrice LSF s36 s35 s34 s33				Indice de 4 ^e sous-matrice LSF s32 s31 s30 s29			
6	Bits de bourrage	Gain de répertoire fixe					Indice de 5 ^e sous-matrice LSF	
	UB	s91	s90	s89	s88	s87	s38	s37

Tableau I.13/H.245 – Mappage entre les bits descripteurs d'insertion de silence TIA IS-641-A et les octets pour l'indice de mode 10
(Les bits cn0 à cn37 se réfèrent à TIA IS-641-A)

Octets transmis	MSB	Mappage des bits						LSB
1	Indice de 1 ^{er} sous-vecteur LSF cn3 cn2 cn1 cn0				Indice de mode mi(3) mi(2) mi(1) mi(0)			
2	Indice de 2 ^e sous-vecteur LSF cn11 cn10 cn9 cn8				Indice de 1 ^{er} sous-vecteur LSF cn7 cn6 cn5 cn4			
3	Indice de 3 ^e sous-vecteur LSF cn19 cn18 cn17			cn16	Indice de 2 ^e sous-vecteur LSF cn15 cn14 cn13 cn12			
4	Gain d'excitation aléatoire cn27 cn26		Indice de 3 ^e sous-vecteur LSF cn25 cn24 cn23 cn22 cn21 cn20					
5	Indice de 1 ^{er} paramètre RESC cn35 cn34		Gain d'excitation aléatoire cn33 cn32 cn31 cn30 cn29 cn28					
6	Bits de bourrage						Indice de 2 ^e paramètre RESC	
	UB	UB	UB	UB	UB	UB	cn37	cn36

Tableau I.14/H.245 – Mappage entre les bits descripteurs d'insertion de silence RCR STD-27H et les octets pour l'indice de mode 11
(Les bits s1 à s35 se réfèrent à RCR STD-27H)

Octets transmis	MSB	Mappage des bits						LSB
1	Indice de 1 ^{er} sous-vecteur LSF s4	Indice de vecteur de référence LSF s3 s2 s1			Indice de mode mi(3) mi(2) mi(1) mi(0)			
2	Indice de 2 ^e sous-vecteur LSF s12	Indice de 1 ^{er} sous-vecteur LSF s11 s10 s9 s8 s7 s6 s5						
3		Indice de 2 ^e sous-vecteur LSF s20 s19 s18 s17 s16 s15 s14 s13						
4		Indice de 3 ^e sous-vecteur LSF s28 s27 s26 s25 s24 s23 s22 s21						
5	Type de SID t1	Energie de trame s35 s34 s33 s32 s31 s30						Indice de 3 ^e sous-vecteur LSF s29
6		Bits de bourrage UB UB UB UB UB UB UB						Type de SID t2

Définition de bits descripteurs additionnels nécessaires pour le Tableau I.14 PDC-EFR:

SID-type est {0=POST0, 1=POST1(SID_UPDATE), 2=PRE, 3=POST1_BAD}, où LSB de SID-type est t1, et le MSB de SID-type est t2.

Tableau I.15/H.245 – Définition de la trame de non-transmission pour l'indice de mode 15

Octets transmis	MSB	Contenu de trame						LSB
1		Bits de bourrage UB UB UB			Indice de mode mi(3) mi(2) mi(1) mi(0)			

Tableau I.16/H.245 – Importance subjective des bits à codage vocal pour AMR12.2

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	23	15	16	17	18
19	20	21	22	24	25	26	27	28	38
141	39	142	40	143	41	144	42	145	43
146	44	147	45	148	46	149	47	97	150
200	48	98	151	201	49	99	152	202	86
136	189	239	87	137	190	240	88	138	191
241	91	194	92	195	93	196	94	197	95
198	29	30	31	32	33	34	35	50	100
153	203	89	139	192	242	51	101	154	204
55	105	158	208	90	140	193	243	59	109
162	212	63	113	166	216	67	117	170	220
36	37	54	53	52	58	57	56	62	61
60	66	65	64	70	69	68	104	103	102
108	107	106	112	111	110	116	115	114	120
119	118	157	156	155	161	160	159	165	164
163	169	168	167	173	172	171	207	206	205
211	210	209	215	214	213	219	218	217	223
222	221	73	72	71	76	75	74	79	78
77	82	81	80	85	84	83	123	122	121
126	125	124	129	128	127	132	131	130	135
134	133	176	175	174	179	178	177	182	181
180	185	184	183	188	187	186	226	225	224
229	228	227	232	231	230	235	234	233	238
237	236	96	199						

Tableau I.17/H.245 – Importance subjective des bits à codage vocal pour AMR10.2

7	6	5	4	3	2	1	0	16	15
14	13	12	11	10	9	8	26	27	28
29	30	31	115	116	117	118	119	120	72
73	161	162	65	68	69	108	111	112	154
157	158	197	200	201	32	33	121	122	74
75	163	164	66	109	155	198	19	23	21
22	18	17	20	24	25	37	36	35	34
80	79	78	77	126	125	124	123	169	168
167	166	70	67	71	113	110	114	159	156
160	202	199	203	76	165	81	82	92	91
93	83	95	85	84	94	101	102	96	104
86	103	87	97	127	128	138	137	139	129
141	131	130	140	147	148	142	150	132	149
133	143	170	171	181	180	182	172	184	174
173	183	190	191	185	193	175	192	176	186
38	39	49	48	50	40	52	42	41	51
58	59	53	61	43	60	44	54	194	179
189	196	177	195	178	187	188	151	136	146
153	134	152	135	144	145	105	90	100	107
88	106	89	98	99	62	47	57	64	45
63	46	55	56						

Tableau I.18/H.245 – Importance subjective des bits à codage vocal pour AMR7.95

8	7	6	5	4	3	2	14	16	9
10	12	13	15	11	17	20	22	24	23
19	18	21	56	88	122	154	57	89	123
155	58	90	124	156	52	84	118	150	53
85	119	151	27	93	28	94	29	95	30
96	31	97	61	127	62	128	63	129	59
91	125	157	32	98	64	130	1	0	25
26	33	99	34	100	65	131	66	132	54
86	120	152	60	92	126	158	55	87	121
153	117	116	115	46	78	112	144	43	75
109	141	40	72	106	138	36	68	102	134
114	149	148	147	146	83	82	81	80	51
50	49	48	47	45	44	42	39	35	79
77	76	74	71	67	113	111	110	108	105
101	145	143	142	140	137	133	41	73	107
139	37	69	103	135	38	70	104	136	

Tableau I.19/H.245 – Importance subjective des bits à codage vocal pour AMR7.4

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	26	87	27
88	28	89	29	90	30	91	51	80	112
141	52	81	113	142	54	83	115	144	55
84	116	145	58	119	59	120	21	22	23
17	18	19	31	60	92	121	56	85	117
146	20	24	25	50	79	111	140	57	86
118	147	49	78	110	139	48	77	53	82
114	143	109	138	47	76	108	137	32	33
61	62	93	94	122	123	41	42	43	44
45	46	70	71	72	73	74	75	102	103
104	105	106	107	131	132	133	134	135	136
34	63	95	124	35	64	96	125	36	65
97	126	37	66	98	127	38	67	99	128
39	68	100	129	40	69	101	130		

Tableau I.20/H.245 – Importance subjective des bits à codage vocal pour AMR6.7

0	1	4	3	5	6	13	7	2	8
9	11	15	12	14	10	28	82	29	83
27	81	26	80	30	84	16	55	109	56
110	31	85	57	111	48	73	102	127	32
86	51	76	105	130	52	77	106	131	58
112	33	87	19	23	53	78	107	132	21
22	18	17	20	24	25	50	75	104	129
47	72	101	126	54	79	108	133	46	71
100	125	128	103	74	49	45	70	99	124
42	67	96	121	39	64	93	118	38	63
92	117	35	60	89	114	34	59	88	113
44	69	98	123	43	68	97	122	41	66
95	120	40	65	94	119	37	62	91	116
36	61	90	115						

Tableau I.21/H.245 – Importance subjective des bits à codage vocal pour AMR5.9

0	1	4	5	3	6	7	2	13	15
8	9	11	12	14	10	16	28	74	29
75	27	73	26	72	30	76	51	97	50
71	96	117	31	77	52	98	49	70	95
116	53	99	32	78	33	79	48	69	94
115	47	68	93	114	46	67	92	113	19
21	23	22	18	17	20	24	111	43	89
110	64	65	44	90	25	45	66	91	112
54	100	40	61	86	107	39	60	85	106
36	57	82	103	35	56	81	102	34	55
80	101	42	63	88	109	41	62	87	108
38	59	84	105	37	58	83	104		

Tableau I.22/H.245 – Importance subjective des bits à codage vocal pour AMR5.15

7	6	5	4	3	2	1	0	15	14
13	12	11	10	9	8	23	24	25	26
27	46	65	84	45	44	43	64	63	62
83	82	81	102	101	100	42	61	80	99
28	47	66	85	18	41	60	79	98	29
48	67	17	20	22	40	59	78	97	21
30	49	68	86	19	16	87	39	38	58
57	77	35	54	73	92	76	96	95	36
55	74	93	32	51	33	52	70	71	89
90	31	50	69	88	37	56	75	94	34
53	72	91							

Tableau I.23/H.245 – Importance subjective des bits à codage vocal pour AMR4.75

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	23	24	25	26
27	28	48	49	61	62	82	83	47	46
45	44	81	80	79	78	17	18	20	22
77	76	75	74	29	30	43	42	41	40
38	39	16	19	21	50	51	59	60	63
64	72	73	84	85	93	94	32	33	35
36	53	54	56	57	66	67	69	70	87
88	90	91	34	55	68	89	37	58	71
92	31	52	65	86					

Annexe J

Définitions relatives au codec vocal TDMA ACELP

Le Tableau J.1 définit l'identificateur de capacités du codec vocal TIA/EIA 136 ACELP [75]. Les Tableaux J.2 à J.4 définissent les paramètres de capacité associés. Ce codec est utilisé dans les stations de base et dans les téléphones mobiles des systèmes cellulaires TDMA et des services PCS. Les caractéristiques techniques de ce codec figurent dans la norme TIA/EIA 136 Part 410. Celle-ci est publiée par l'Association des industries de télécommunication (TIA, *North American*) (*Telecommunications Industry Association*) et par l'Institut national américain de normalisation (ANSI, *American National Standards Institute*).

Tableau J.1/H.245 – Identificateur de capacité TIA/EIA 136 ACELP

Nom de la capacité	TIA/EIA 136 ACELP Vocoder
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) acelp (2)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau J.2/H.245 – Paramètre de capacité TIA/EIA 136 ACELP – Nombre maximal de trames audio par Al-sdu

Nom du paramètre	maxAl-sduAudioFrames
Description du paramètre	Paramètre générique de type collapsing. Spécifie le nombre maximal de trames audio par unité AL-SDU
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Ce paramètre sera présent pour l'échange de capacités et pour la signalisation de voies logiques. Il ne sera pas présent pour la demande de mode.
Type du paramètre	unsignedMin
Remplace	–

Tableau J.3/H.245 – Paramètre de capacité TIA/EIA 136 ACELP – Bruit de confort

Nom du paramètre	comfortNoise
Description du paramètre	Paramètre générique de type collapsing. Spécifie qu'il faut utiliser le bruit de confort TIA/EIA 136 (IS-641) dans la demande de mode. Ce paramètre doit être utilisé dans les demandes de mode seulement et non dans les capacités car celle-ci est obligatoire.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau J.4/H.245 – Paramètre de capacité TIA/EIA 136 – Embrouillage

Nom du paramètre	scrambled
Description du paramètre	Paramètre générique de type collapsing. Spécifie qu'il y a lieu d'utiliser l'embrouillage dans la demande de mode. Ce paramètre ne doit être utilisé que dans les demandes de mode seulement et non dans les capacités car celle-ci est obligatoire.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Annexe K

Définitions relatives au codec vocal TDMA US1

Le Tableau K.1 définit l'identificateur de capacités du codec vocal TIA/EIA 136 US1 [76]. Les Tableaux K.2 à K.4 définissent les paramètres de capacité associés. Ce codec est utilisé dans les stations de base et les téléphones mobiles des systèmes cellulaires TDMA et des services PCS. Les caractéristiques techniques de ce codec figurent dans la norme TIA/EIA 136 Part 430. Celle-ci est publiée par l'Association des industries de télécommunication (TIA, *North American Telecommunications Industry Association*) et par l'Institut national américain de normalisation (ANSI, *American National Standards Institute*).

Tableau K.1/H.245 – Identificateur de capacité TIA/EIA 136 US1

Nom de la capacité	TIA/EIA 136 US1 Vocoder
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) us1 (3)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau K.2/H.245 – Paramètre de capacité TIA/EIA 136 US1 – Nombre maximal de trames audio par Al-sdu

Nom du paramètre	maxAl-sduAudioFrames
Description du paramètre	Paramètre générique de type collapsing. Spécifie le nombre maximal de trames audio par unité AL-SDU
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Ce paramètre sera présent pour l'échange de capacités et pour la signalisation de voies logiques. Il ne sera pas présent pour la demande de mode.
Type du paramètre	unsignedMin
Remplace	–

Tableau K.3/H.245 – Paramètre de capacité TIA/EIA 136 US1 – Bruit de confort

Nom du paramètre	comfortNoise
Description du paramètre	Paramètre générique de type collapsing. Spécifie qu'il faut utiliser le bruit de confort dans la demande de mode. Ce paramètre ne doit être utilisé que dans les demandes de mode et non dans les capacités car celle-ci est obligatoire.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Tableau K.4/H.245 – TIA/EIA 136 US1 Capability Parameter – Embrouillage

Nom du paramètre	scrambled
Description du paramètre	Paramètre générique de type collapsing. Spécifie qu'il y a lieu d'utiliser l'embrouillage dans la demande de mode. Ce paramètre ne doit être utilisé que dans les demandes de mode et non dans les capacités car celle-ci est obligatoire.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Annexe L

Définitions relatives au codec vocal CDMA EVRC

Le Tableau L.1 définit l'identificateur de capacités du codec à débit variable amélioré (EVRC, *enhanced variable rate codec*) utilisé dans les stations de base et les téléphones mobiles des systèmes cellulaires CDMA TIA/EIA IS-95 et des services PCS. La description technique complète et les caractéristiques détaillées de ce codec figurent dans la norme TIA/EIA IS-127 publiée par l'Association des industries de télécommunication (TIA). Les Tableaux L.2 à L.4 définissent les paramètres de capacité associés.

Tableau L.1/H.245 – Identificateur de capacité CDMA EVRC

Nom de la capacité	TIA/EIA IS-127 CDMA EVRC
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) is127evrc (4)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

**Tableau L.2/H.245 – Paramètre de capacité TIA/EIA IS-127 CDMA EVRC –
Nombre maximal de trames audio par Al-sdu**

Nom du paramètre	maxAl-sduAudioFrames
Description du paramètre	Paramètre générique de type collapsing. Spécifie le nombre maximal de trames audio par unité AL-SDU.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Ce paramètre sera présent pour l'échange de capacités et pour la signalisation de voies logiques. Il ne sera pas présent pour la demande de mode.
Type du paramètre	unsignedMin
Remplace	–

Tableau L.3/H.245 – Paramètre de capacité CDMA EVRC – Débit EVRC

Nom du paramètre	EVRCRate
Description du paramètre	Paramètre générique de type nonCollapsing qui spécifie le mode du débit de sortie du vocodeur. Il convient de l'utiliser dans les demandes de mode: 1 = plein débit; 2 = demi-débit; 3 = huitième de débit; 4 = mode neutralisé.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif
Type du paramètre	unsignedMin
Remplace	–

Tableau L.4/H.245 – Paramètre de capacité CDMA EVRC – Embrouillage

Nom du paramètre	scrambled
Description du paramètre	Paramètre générique de type collapsing. Spécifie qu'il y a lieu d'utiliser l'embrouillage dans la demande de mode.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Annexe M

Définitions relatives à l'ISO/CEI 13818-7 et à l'UIT-R BS.1196

Le Tableau M.1 définit l'identificateur de capacité pour l'ISO/CEI 13818-7. Le Tableau M.2 définit les paramètres de capacité associés.

Le Tableau M.3 définit l'identificateur de capacité pour l'UIT-R BS.1196. Il n'y a pas de paramètres de capacité associés.

Tableau M.1/H.245 – Identificateur de capacités pour l'ISO/CEI 13818-7

Nom de la capacité	ISO/CEI 13818-7
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) ISO/IEC 13818-7 (5)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau M.2/H.245 – Profil et niveau de la capacité pour l'ISO/CEI 13818-7

Nom du paramètre	profileAndLevel
Description du paramètre	Paramètre générique de type nonCollapsing. Spécifie la capacité de traiter les profils particuliers en combinaison avec le niveau comme indiqué dans l'ISO/CEI 13818-7.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Obligatoire
Type du paramètre	unsignedMax. Doit être compris entre 0..255.
Remplace	–

Tableau M.3/H.245 – Identificateur de capacités UIT-R BS.1196

Nom de la capacité	UIT-R BS.1196
Classe de la capacité	Codec audio
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) iut-r-bs.1196 (6)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Annexe N

RFC 3389 – Charge utile du protocole RTP concernant le bruit de confort

Le Tableau N.1 définit l'identificateur de capacité relatif au document RFC 3389. Il n'existe pas de paramètres de capacité associés.

Le document RFC 3389 du groupe IETF spécifie un moyen de signaler un bruit de confort mélangé avec des paquets audio normaux. Il est fondé sur l'Appendice II/G.711. Le principal objet de ce RFC est de fournir une signalisation de bruit de confort aux codecs qui ne la définissent pas en propre. Ce protocole est généralement accepté comme étant le moyen de signaler un bruit de confort dans le cadre des systèmes fondés sur le protocole RTP.

La capacité de bruit de confort peut faire l'objet d'une signalisation comme toute autre capacité. Les médias de ce type peuvent être ouverts conjointement avec les protocoles G.711, G.726, etc. par spécification de cette capacité dans le cadre d'une voie en flux à multiples charges utiles (MPS, *multiple payload stream*).

Tableau N.1/H.245 – Bruit de confort selon RFC 3389

Nom de la capacité	Bruit de confort RFC 3389
Classe de la capacité	Codec audio
Type d'identificateur de capacité	Standard
Valeur d'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) rfc3389 (6)}
maxBitRate	Ce champ doit être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Annexe O

Identificateur de capacité L16

Le Tableau O.1 définit l'identificateur de capacité pour le protocole L16. Le codec L16 est un codec de données audio non comprimées, utilisant une représentation signée sur 16 bits avec 65535 échelons équidistants entre le niveau minimal et le niveau maximal du signal, dans l'étendue de -32768 à 32767. La valeur est représentée en notation de complément à deux et dans l'ordre des octets du réseau. Ce codec est utilisé pour la vérification des performances acoustiques et éventuellement pour les applications à faible coût de réseau local à large bande. Il est défini dans le RFC 1890 du groupe IETF, au § 4.4.8.

Le Tableau O.2 définit le paramètre de capacité associé.

Tableau O.1/H.245 – Identificateur de capacité L16

Nom de la capacité	Codec audio L16
Classe de la capacité	Codec audio
Type d'identificateur de capacité	Standard
Valeur d'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) L-16 (7)}
maxBitRate	Ce champ doit être inclus. Le débit définit le nombre d'échantillons par seconde; c'est également le rythme du marqueur temporel RTP. Les valeurs recommandées pour le débit sont: 8000, 11025, 16000, 22050, 24000, 32000, 44100, et 48000 échantillons par seconde. D'autres valeurs sont admissibles.
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau O.2/H.245 – Paramètre de capacité L16 – Voies

Nom du paramètre	channels
Description du paramètre	Il s'agit d'un paramètre générique de type "collapsing" (fédérable). Il spécifie le nombre de flux audio entrelacés. La valeur par défaut est 1, la valeur de stéréophonie est 2. L'entrelacement intervient entre échantillons individuels de 2 octets.
Valeur d'identificateur du paramètre	0
Statut du paramètre	Doit être présent pour l'échange de capacités et la signalisation de voie logique. Ne doit pas être présent pour la demande de mode.
Type du paramètre	unsignedMin
Remplace	–

Annexe P

Capacité de flux audio borné

Cette capacité générique peut être utilisée comme capacité audio générique dans le paramètre AudioCapability contenu dans les messages TerminalCapabilitySet, OpenLogicalChannel et CommunicationModeCommand, et comme mode genericAudioMode à l'intérieur d'un message RequestMode.

Deux paramètres sont définis pour la capacité de flux audio borné:

- la longueur minimale de paquet (trames par paquet) en tant que paramètre de type "collapsing" (fédérable);
- la capacité audio définie par la Rec. UIT-T H.245 pour laquelle une longueur minimale de paquet est requise. Cette valeur définit le codec ainsi que la longueur maximale de paquet pour le flux audio recherché.

Tableau P.1/H.245 – Identificateur de capacité pour flux audio borné

Nom de la capacité	Flux audio borné
Classe de la capacité	Codec audio
Type d'identificateur de capacité	Standard
Valeur d'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) audio (1) bounded-audio-stream (8)}
maxBitRate	Le champ maxBitRate doit toujours être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Tableau P.2/H.245 – Paramètre de longueur minimale de paquet

Nom du paramètre	minimumPacketSize
Description du paramètre	Il s'agit d'un paramètre générique de type "collapsing" (fédérable). Il spécifie le nombre minimal de trames par paquet.
Valeur d'identificateur du paramètre	0
Statut du paramètre	Obligatoire
Type du paramètre	unsignedMax. Doit être compris entre 0..256.
Remplace	–

Tableau P.3/H.245 – Audio capability

Nom du paramètre	audioCapability
Description du paramètre	Il s'agit d'un paramètre générique de type "nonCollapsing" (non fédérable). Il contient une capacité audio valide par codage conformément à l'Annexe B.
Valeur d'identificateur du paramètre	1
Statut du paramètre	Obligatoire
Type du paramètre	octetString
Remplace	–

Annexe Q

Capacité générique de relais sur IP de N circuits à 64 K

Q.1 Introduction

La présente annexe décrit l'utilisation d'une capacité générique pour le relais de Nx64 circuits à structure verrouillée dans un réseau IP. Aucune hypothèse n'est effectuée en ce qui concerne le contenu ou la structure des informations à relayer. Plusieurs paramètres sont définis afin de simplifier la négociation d'une voie satisfaisante.

Cette capacité vise à répondre aux exigences suivantes:

- 1) le transport doit toujours être en mesure d'acheminer des informations numériques avec ou sans restriction au débit de 64 kbit/s ou 56 kbit/s, comme indiqué au moyen de messages de la Rec. UIT-T Q.931. Noter que seules les informations du support vont être transportées ici, à l'exclusion des signaux de verrouillage de trames de réseau ou des voies de commande;
- 2) la transmission doit toujours être transparente en terme d'octets (dite "transport à structure verrouillée");
- 3) aucun traitement de signal de quelque sorte que ce soit (bourrage, annulation d'écho, détection de tonalité, suppression de silence, ou codage, etc.) ne devrait être effectué à l'interface TDM/IP;
- 4) le transport par protocole RTP (ou SRTP) doit toujours être pris en charge;
- 5) la longueur de paquet doit toujours être négociable;
- 6) des mécanismes de reprise sur perte/corruption de paquet (et d'ordre de paquet), doivent toujours être négociables:
 - a) correction d'erreur directe conformément à RFC 2733, dans laquelle chaque paquet de "parité" de correction FEC implique R paquets successifs – ce qui double la largeur de bande de la voie;
 - b) transmission redondante conformément à RFC 2198 dans laquelle des copies supplémentaires de chaque bloc de média sont transmises;
- 7) la largeur de bande devrait être spécifiée sous la forme du produit du nombre par la capacité des voies ($N \times 64$ kbit/s);
- 8) le nombre de voies-circuits ne changera pas pendant la durée de vie d'une session de média.

Cette capacité fait appel à un format de paquet défini dans la Rec. UIT-T Y.1413 [80]. Une certaine attention a été prêtée aux travaux effectués dans le groupe de travail PWE3 de l'IETF, mais ces études semblent plus orientées vers le transport efficace de circuits audio en bouquet que vers celui de données numériques (voir p. ex. [PWE3]).

Q.2 Description

Q.2.1 Terminologie

Trame: agrégation d'un même échantillon de 8 bits par voie, dans l'ordre de 1 à N voies. Dans les voies à 64 kbit/s échantillonnées à 8 kHz, la trame est composée de N octets. La longueur de trame d'un seul échantillon par voie a été choisie de façon à offrir une certaine flexibilité en terme de longueur totale de paquet pour une étendue de N comprise entre 1 et 31 ou plus, ainsi qu'un nombre d'échantillons par voie compris entre 1 et (par exemple) 2047.

- **M:** nombre de trames par charge utile de paquet.
- **N:** nombre de voies TDM à 64 kbit/s par trame.

Q.2.2 Identificateur de capacité

L'élément de service proposé de capacité générique de transmission de données non codées sur voies $N \times 64K$ sera identifié par la structure **GenericCapability.capabilityIdentifier.standard** mise à:

{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) data (2) $N \times 64$ (10)}

Pour utilisation dans la capacité générique.

Tableau Q.1/H.245 – Identificateur de capacité Nx64

Nom de la capacité	Relais de circuits Nx64
Classe de la capacité	Codec de données
Type d'identificateur de capacité	Standard
Valeur d'identificateur de capacité	itu-t (0) recommandation (0) h (8) 245 generic-capabilities (1) data (2) Nx64 (10)
maxBitRate	Le champ maxBitRate doit toujours être inclus
nonCollapsingRaw	Ce champ ne doit pas être inclus
transport	Ce champ ne doit pas être inclus

Q.2.3 Paramètres pour l'échange de capacités

Les paramètres définis pour la capacité Nx64 capacité doivent être comme décrit dans les Tableaux Q.2 et Q.3:

Tableau Q.2/H.245 – Paramètre de nombre de voies Nx64

Nom du paramètre	Nombre de voies
Description du paramètre	Il s'agit d'un paramètre générique de type "Collapsing". Le nombre de voies spécifie le nombre de voies de support à 64 kbit/s à acheminer dans le flux.
Valeur d'identificateur du paramètre	1
Statut du paramètre	Facultatif, la valeur par défaut est 1
Type du paramètre	unsignedMax. Doit être compris entre 1..255. unsignedMin (seulement pendant l'échange de capacités).
Remplace	–

Tableau Q.3/H.245 – Paramètre de longueur de charge utile Nx64

Nom du paramètre	Longueur de charge utile
Description du paramètre	Il s'agit d'un paramètre générique de type "Collapsing". La longueur de charge utile spécifie le nombre de trames à acheminer dans un même paquet du flux.
Valeur d'identificateur du paramètre	2
Statut du paramètre	unsignedMax est obligatoire unsignedMin est facultatif
Type du paramètre	unsignedMax. Doit être compris entre 1..2047. unsignedMin (seulement pendant l'échange de capacités).
Remplace	–

La longueur maximale de bloc, exprimée en trames, est le seul paramètre qui doit toujours être spécifié, et de toute façon, la valeur minimale reviendra par défaut à la valeur maximale si aucun minimum n'est spécifié. Cela permet la spécification d'une large étendue de combinaisons avec une même capacité. La structure **GenericCapability.maxBitRate** peut être mise à la valeur du nombre maximal de voies multiplié par 640.

Q.2.4 Paramètres de capacités d'établissement de voie

Q.2.4.1 Type H.245

Lorsqu'elle est utilisée en tant que type de données dans un message **H.245 OpenLogicalChannel** ou **OpenLogicalChannelAck**, ou en tant que **mode** dans un message **RequestMode**, la capacité générique Nx64 ne devrait pas contenir de valeurs minimales; c'est-à-dire que tous les paramètres spécifiés devraient contenir les valeurs recherchées pour la voie de données spécifiée. Ces valeurs doivent toujours, évidemment, être compatibles avec les capacités déjà échangées.

Q.2.4.2 Type "Fast Connect" ou "Extended Fast Connect"

Lorsqu'elle est utilisée en tant que type de données de média dans une proposition H.245 **OpenLogicalChannel** contenue à l'intérieur d'un élément H.225.0 **fastStart** [12], la capacité générique Nx64 ne doit contenir que les valeurs maximales des paramètres afin d'indiquer les valeurs effectivement demandées. Tout paramètre omis est implicitement offert avec la valeur par défaut seulement.

Q.2.5 Format de paquet

Le format de paquet doit être celui du protocole RTP [83], avec la charge utile composée d'une ou de plusieurs trames dans l'ordre des octets du réseau, de concert avec les éléments de redondance négociés, s'ils existent. Une trame est définie en tant qu'agrégation d'un même échantillon de 8 bits issu de chacune des N voies TDM dans un ordre déterminé (par le réseau à commutation de circuits). Ce format de trame est identique au format d'encapsulation à structure verrouillée (sans signalisation) spécifié au § 9.2.1/Y.1413, Figure Q.1, reproduite ci-dessous. Ce verrouillage de trames évite d'avoir à connaître l'éventuelle structure interne des données, tout en conservant la synchronisation du flux de paquets avec le circuit d'origine. La longueur de trame (M) doit rester constante pendant toute la durée de vie de la connexion.

8	7	6	5	4	3	2	1	Trame
Bits appartenant à l'intervalle de temps 1								1
Bits appartenant à l'intervalle de temps 2								
...								
Bits appartenant à l'intervalle de temps N								2
Bits appartenant à l'intervalle de temps 1								
Bits appartenant à l'intervalle de temps 2								
...								
Bits appartenant à l'intervalle de temps N								...
...								
Bits appartenant à l'intervalle de temps 1								
Bits appartenant à l'intervalle de temps 2								M
...								
Bits appartenant à l'intervalle de temps N								

NOTE 1 – Le bit 8 est le bit de poids fort.

NOTE 2 – Le paquet contient M trames TDM avec N intervalles de temps par trame.

Figure Q.1/H.245 – Format de charge utile pour encapsulation à structure verrouillée sans protocole CAS (d'après la Rec. UIT-T Y.1413)

Q.2.6 Restrictions relatives à l'en-tête RTP

Les contraintes suivantes sont adaptées à partir du § 8.4/Y.1413. Elles doivent toujours être observées lors du formatage de paquets conformément à cette capacité.

- 1) Le bourrage, les extensions d'en-tête, les sources de synchronisation multiples, et les marqueurs ne sont pas utilisés.
- 2) Le(s) type(s) de charge utile doi(ven)t être choisi(s) en fonction de l'étendue dynamique.
- 3) Les numéros de séquence doivent être consécutifs pour des paquets consécutifs; en association avec la longueur fixe de charge utile, cela permet à un récepteur de calculer le nombre exact de trames perdues par un paquet manquant.
- 4) Le marqueur temporel RTP, en association avec la longueur et le débit des paquets, peut être utilisé afin d'acheminer des informations de rythme dans le Réseau IP; la fréquence d'horloge utilisée pour produire les marqueurs temporels devrait être un entier multiple de 8 kHz. Des directives relatives au choix approprié de cette fréquence d'horloge sont données dans l'Appendice V/Y.1413.
- 5) Le champ de source de synchronisation, contenu dans l'en-tête RTP, peut être utilisé pour la détection des connexions erronées.

Q.2.7 Formatage de redondance

Cette capacité permet plusieurs méthodes facultatives de correction d'erreur par redondance ou par correction d'erreur directe. Dans le cadre de l'échange de capacités, une ou plusieurs méthodes peu(ven)t être spécifiée(s) au moyen des capacités H.245 normales.

Q.2.8 Considérations relatives au rythme

Le transport d'informations synchrones dans le réseau IP est exposé à plusieurs types de problèmes. Le transport par protocole IP souffre de gigue dans les temps de propagation de paquet. Cette gigue peut être compensée par introduction d'un retard supplémentaire (avec mise en mémoire tampon au niveau du récepteur.) Des mécanismes de correction d'erreur peuvent assurer la redondance.

Le pointage temporel au niveau de l'origine et de la destination TDM affecte également la performance de bout en bout de la connexion. Si les systèmes TDM de source et de destination sont calés sur des horloges différentes qui dérivent l'une par rapport à l'autre, des défauts ou des excès de remplissage des tampons de données sont inévitables. L'apparition de tels événements peut être quelque peu limitée par la mise en mémoire tampon, mais il en résultera tôt ou tard une perte d'informations, ou l'insertion de données dépourvues de sens. Plusieurs situations d'horloge différentes peuvent être prises en considération.

Q.2.9 Horloge commune

Aussi bien l'origine que la destination sont calées sur une horloge de réseau commune. Dans ces circonstances, des défauts ou des excès de remplissage ne devraient pas se produire, tant que le transport de paquets s'effectue de façon satisfaisante.

Q.2.9.1 Horloges indépendantes

Dans ce cas, l'émetteur et le récepteur fonctionnent à des vitesses différentes, mais la dérive d'une des horloges par rapport à l'autre peut être compensée temporellement par mise en mémoire tampon. L'utilisation d'horloges précises et étalonnées peut réduire l'apparition des glissements jusqu'à un niveau satisfaisant, selon l'application utilisant le transport.

Q.2.9.2 Capacité support

Ce format peut être utilisé afin d'acheminer des données à 64 kbit/s ou 56 kbit/s avec ou sans restriction, par encapsulage dans une voie à 64 kbit/s, comme indiqué par la capacité support Q.931. Voir le § 7.2.2.1/H.225.0 en ce qui concerne le codage de la capacité support dans le message d'établissement.

Appendice I

Aperçu général de la syntaxe ASN.1

I.1 Introduction à la syntaxe ASN.1

La notation de syntaxe abstraite numéro un (ASN.1, *abstract syntax notation one*) est un langage de spécification de données. Ce langage a été initialement normalisé par la Rec. UIT-T X.409 faisant partie de la série des Recommandations de messagerie électronique X.400. La Rec. UIT-T X.208 et, plus récemment, la Rec. UIT-T X.680 sont issues de la Recommandation initiale. La syntaxe ASN.1 permet de spécifier sans ambiguïté des structures de données complexes, y compris celles contenant des champs de longueur variable, des champs facultatifs, de même que des récurrences.

Les Recommandations mentionnées ci-dessus ne traitent que de la syntaxe et de la sémantique des spécifications ASN.1. Le codage binaire des structures de données est traité dans d'autres Recommandations, notamment la Rec. UIT-T X.690 (règles de codage de base ou BER) et la Rec. UIT-T X.691 (règles de codage compact ou PER). Les règles de codage de base (BER, *basic encoding rule*) 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 avec les valeurs de données. Les règles de codage compact (PER, *packing encoding rule*) 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 connue à la fois de l'émetteur et du récepteur.

Les mises en œuvre conformes à la présente Recommandation utilisent les règles de codage compact (PER). Étant donné que les deux correspondants savent que les messages seront conformes à la spécification H.245, il n'est pas nécessaire de coder cette spécification dans les messages. À des fins de simplification, la variante "avec alignement" 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 utilisent un nombre entier d'octets. L'alignement est effectué en ajoutant des zéros aux données à l'avant des champs de grande longueur.

I.2 Types de données de base de la syntaxe ASN.1

Le type de données le plus simple est BOOLEAN (booléen) qui prend les valeurs "Faux" et "Vrai". Ces valeurs sont codées sur un seul bit prenant respectivement les valeurs 0 et 1. Par exemple, le paramètre `segmentableFlag` BOOLEAN est codé:

Valeur	Codage
"Faux"	0
"Vrai"	1

Le type de données essentiel est INTEGER (nombre entier), qui représente des valeurs de nombres entiers. Les nombres du type INTEGER peuvent être illimités comme dans l'expression:

```
bitRate                INTEGER
```

ou ils peuvent être limités à un intervalle de valeurs, comme:

```
maximumA12SDUSize     INTEGER (0..65535)
```

Les nombres entiers limités sont codés différemment selon la taille de l'intervalle. Supposons que N soit le nombre de valeurs de l'intervalle, c'est-à-dire la limite supérieure moins la limite inférieure plus un. Le nombre entier limité sera codé de l'une des cinq façons suivantes selon la valeur de N:

N	Codage
1	Pas de bit nécessaire
2-255	Un champ compris entre 1 et 8 bits sans alignement
256	Un champ de 8 bits avec alignement
257-65536	Un champ de 16 bits avec alignement
Supérieur	Comme le nombre minimal d'octets avec alignement précédé par le codage du nombre d'octets selon la règle ci-dessus

Dans tous les cas, le nombre qui est effectivement utilisé est la valeur devant être codée moins la limite inférieure de l'intervalle. Dans ces exemples, les "bits de remplissage" représentent de zéro à sept bits 0 qui sont ajoutés à la chaîne codée de sorte que le champ suivant puisse commencer à la frontière d'un octet.

firstGOB

INTEGER (0..17)

Valeur	Codage
0	00000
3	00011

h233IVResponseTime

INTEGER (0..255)

Valeur	Codage
3	bits de remplissage 00000011
254	bits de remplissage 11111110

skew

INTEGER (0..4095)

Valeur	Codage
3	bits de remplissage 00000000 00000011
4095	bits de remplissage 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) sous la forme d'un octet avec alignement qui précède le nombre lui-même. Par exemple,

-1 bits de remplissage 00000001 11111111
0 bits de remplissage 00000001 00000000
128 bits de remplissage 00000010 00000000 10000000
1000000 bits de remplissage 00000011 00001111 01000010 01000000

La notation ASN.1 accepte différents types de chaînes de données. Ces chaînes sont constituées, par exemple, de suites de longueur variable contenant des bits, des octets, ou d'autres types de données courts. Elles sont généralement codées sous forme d'une longueur suivie par les données. La longueur peut être codée sous la forme d'un nombre entier illimité ou d'un nombre entier limité si la taille (SIZE) de la chaîne est spécifiée. 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 alignement. 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 le plus significatif (MSB) mis à 1, suivie par les données
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 un octet de zéros indiquant une chaîne de longueur nulle.

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 effectif est codé comme un nombre entier limité. L'index est suivi par le codage du choix effectif 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 effectif 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, -- unité 1/29,97 Hz
    cifMPI           INTEGER (1..4) OPTIONAL, -- unité 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 de remplissage pour l'alignement.

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 (SIZE) ou un nombre illimité d'éléments. Si ce nombre est connu *a priori* et est inférieur à 64K, il n'est pas codé. Sinon, le nombre effectif de composantes est codé sous la forme d'une longueur limitée ou semi-limitée. Il 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 `protocolIdentifier` contient:

```
protocolIdentifier OBJECT IDENTIFIER,  
    -- devra être mis à la valeur  
    -- {itu-t (0) recommendation (0) h (8) 245 version (0) 1}
```

Tous les nombres entiers figurant entre accolades `{}` sont codés, qu'ils soient ou non entre parenthèses `()`. Dans l'exemple considéré ici, les nombres entiers 0, 0, 8, 245, 0, 1 doivent être codés.

Pour le codage de ce paramètre, les données sont codées avec les règles de codage de base (BER) (X.690) et sont précédées par la longueur de ce codage en octets. La longueur est codée comme un ensemble de nombres entiers semi-limités (voir l'exemple OCTET STRING ci-dessus). Ce qui suit illustre comment ce codage est effectué.

Le premier octet indique la longueur du codage qui suit.

Les deux premières composantes de l'identificateur d'objet sont associées comme $40 \times \text{première composante} + \text{seconde composante}$; dans ce cas, $40 \times 0 + 0 = 0$. Les autres composantes sont codées comme elles sont. Chacune est codée en une série d'octets, le premier bit de chaque octet indiquant s'il y en a davantage. Ainsi:

0 → 0000 0000

8 → 0000 1000

tandis que 245, qui est supérieur à 127, devient 1000 0001 0111 0101

ainsi, l'ensemble du codage en numérotation hexadécimale comprend les sept octets 06000881 750001.

Appendice II

Exemples de procédures H.245

II.1 Introduction

Le présent appendice donne des exemples des procédures définies dans l'Annexe C. La Figure II.1-1 montre les symboles utilisés dans les diagrammes contenus dans le présent appendice.

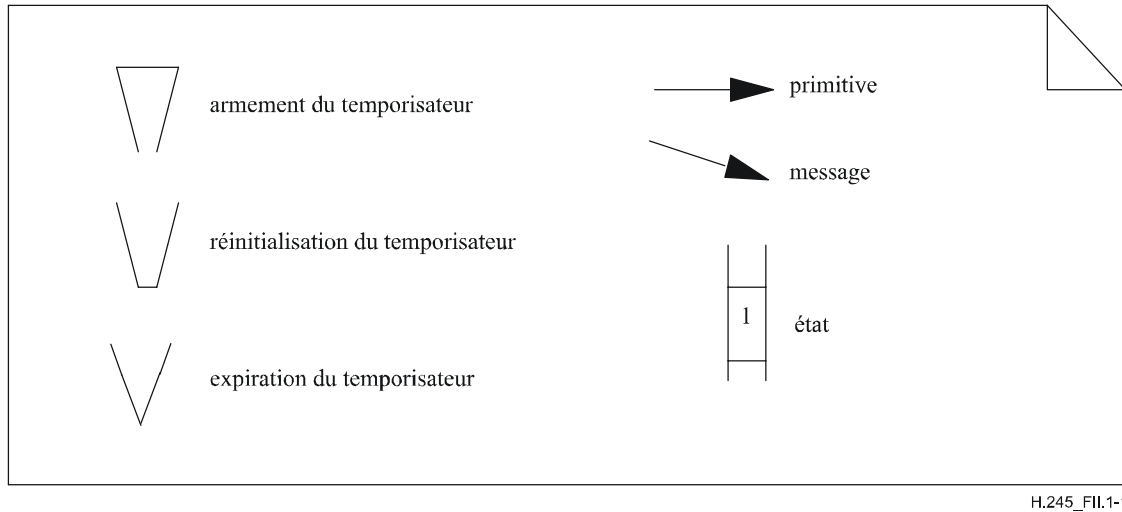


Figure II.1-1/H.245 – Symboles utilisés dans les figures

II.2 Entité de signalisation de désignation maître-esclave

Dans les Figures II.2-1 à II.2-10, les messages sont représentés par les noms abrégés indiqués dans le Tableau II.2-1.

Tableau II.2-1/H.245 – Noms abrégés pour la désignation maître-esclave

Message	Nom utilisé dans les exemples
MasterSlaveDetermination	MSD
MasterSlaveDeterminationAck	MSDAck
MasterSlaveDeterminationReject	MSDReject
MasterSlaveDeterminationRelease	MSDRelease

Dans les Figures II.2-1 à II.2-10, les états IDLE, OUTGOING AWAITING RESPONSE et INCOMING AWAITING RESPONSE sont respectivement désignés par "0", "1" et "2".

Dans les figures suivantes, la valeur de paramètre associée aux primitives d'indication DETERMINE et de confirmation DETERMINE est celle du paramètre TYPE. La valeur de champ associée au message MasterSlaveDeterminationAck est celle du champ décision.

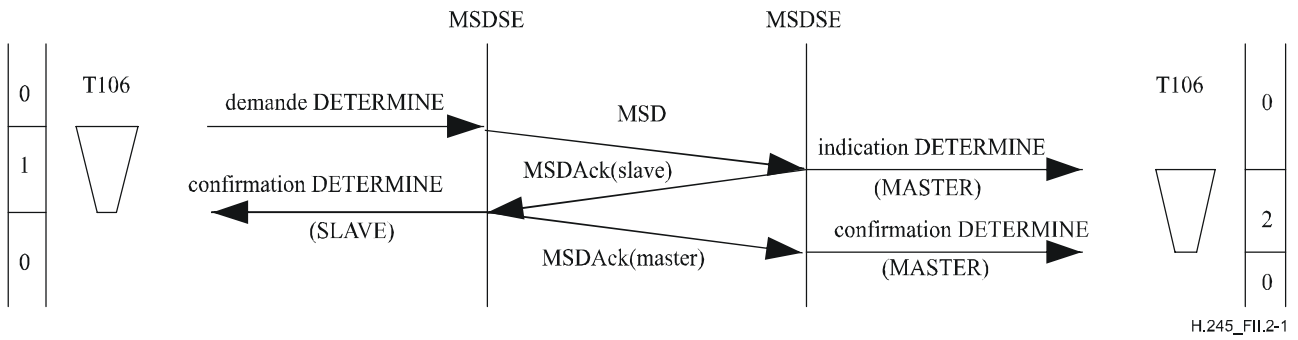


Figure II.2-1/H.245 – Désignation maître-esclave – Terminal maître dans l'entité MSDSE distante

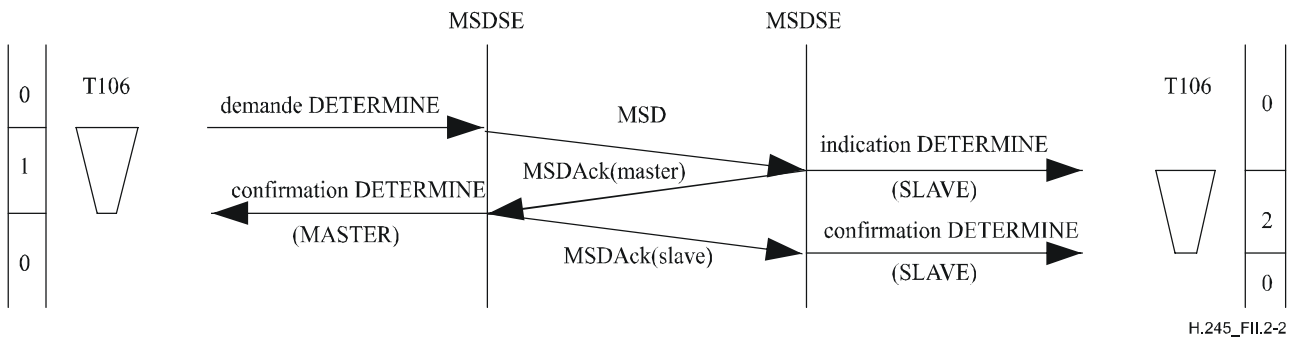


Figure II.2-2/H.245 – Désignation maître-esclave – Terminal asservi dans l'entité MSDSE distante

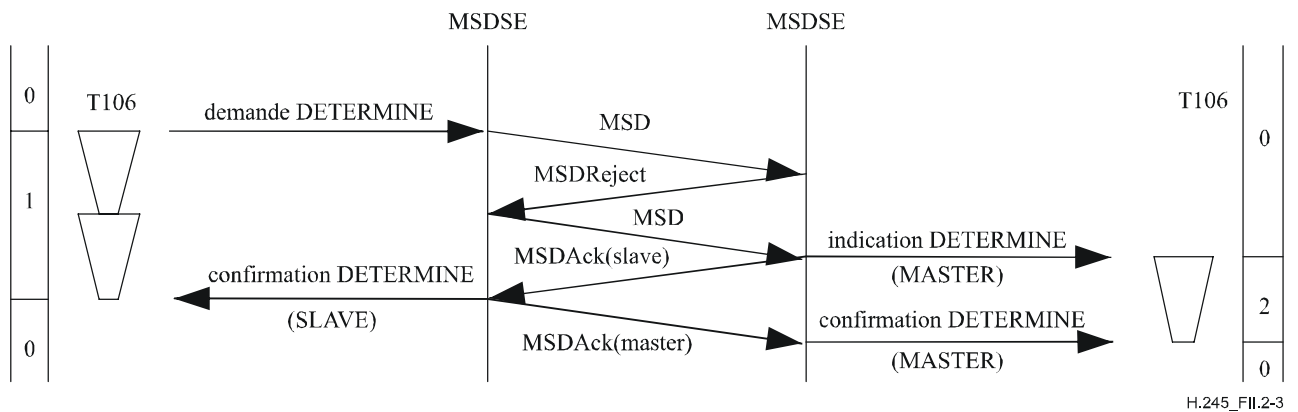


Figure II.2-3/H.245 – Désignation maître-esclave – La première tentative a donné un résultat indéterminé. La deuxième tentative a abouti

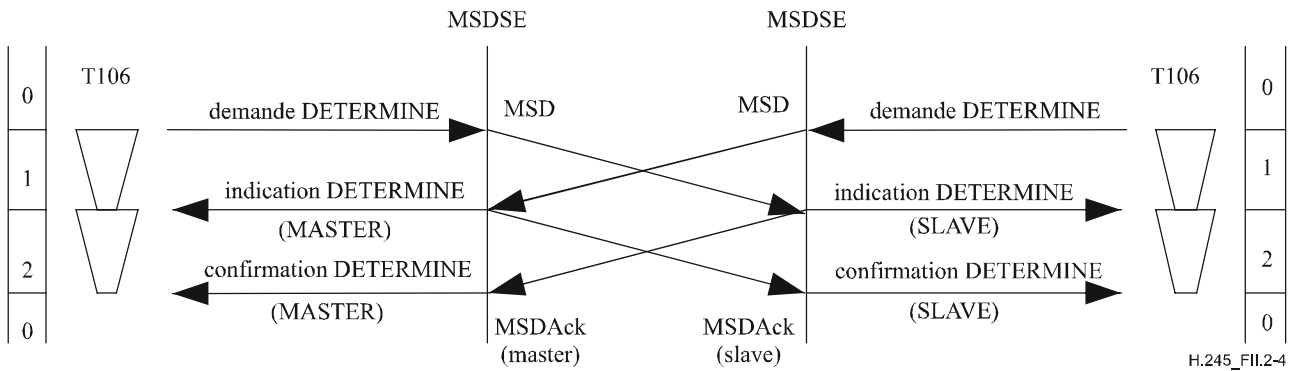


Figure II.2-4/H.245 – Désignation maître-esclave – Désignation simultanée

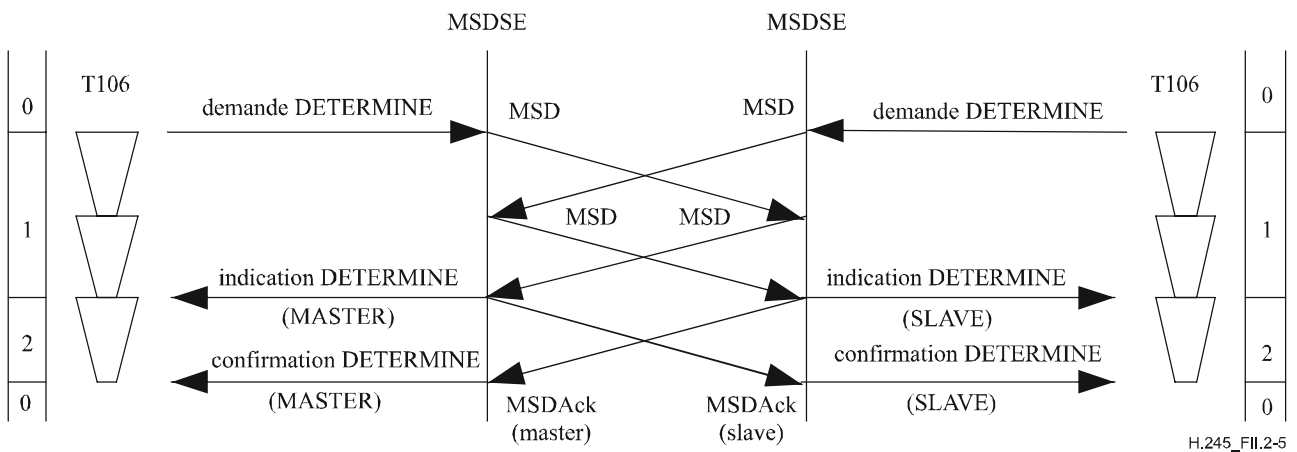


Figure II.2-5/H.245 – Désignation maître-esclave – Désignation simultanée mais avec retour de résultat indéterminé à la première tentative

Dans la Figure II.2-6, la temporisation locale T106 est arrivée à expiration. Seul le terminal de droite connaît son statut. Ce terminal est en mesure de recevoir de nouvelles commandes mais ne peut rien demander à l'autre terminal qui dépend de la connaissance du résultat de la désignation. Le terminal de gauche ne peut ni accepter ni lancer de nouvelles procédures. Une deuxième procédure de désignation doit être lancée.

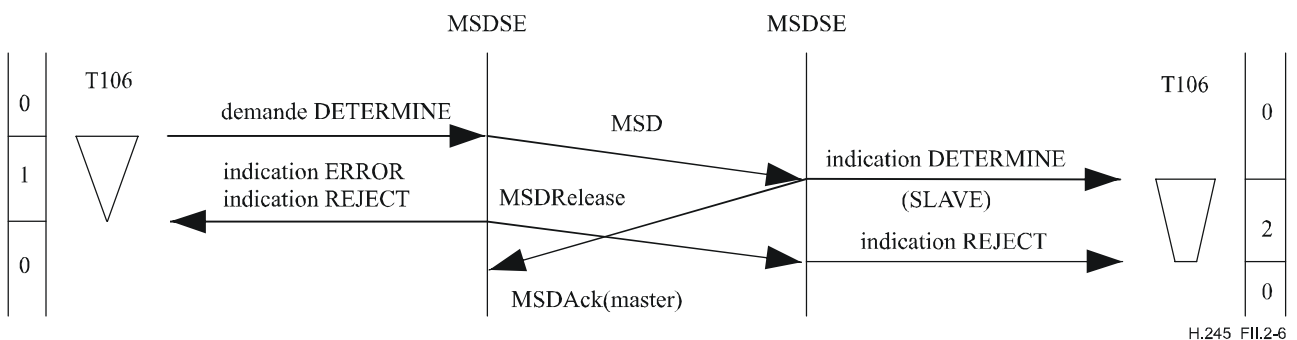


Figure II.2-6/H.245 – Désignation maître-esclave – Temporisation locale T106 à expiration avec terminal asservi du côté distant

Dans la Figure II.2-7, la temporisation distante T106 est venue à expiration pendant l'état INCOMING AWAITING ACKNOWLEDGEMENT. Les deux terminaux connaissent leur statut. Le terminal de gauche peut recevoir et émettre des commandes. Le terminal distant ne sait cependant pas si le terminal local est prêt à recevoir, et il ne peut pas émettre des commandes dépendant de la connaissance du résultat de la désignation. Une deuxième procédure de désignation doit être lancée.

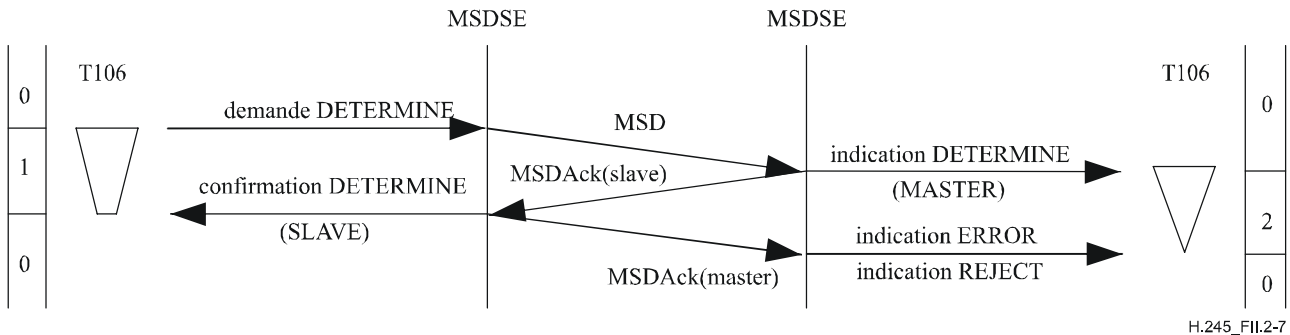


Figure II.2-7/H.245 – Désignation maître-esclave – Temporisation distante T106 à expiration avec terminal maître du côté distant

Dans la Figure II.2-8, la temporisation distante T106 est arrivée à expiration pendant l'état OUTGOING AWAITING ACKNOWLEDGEMENT d'une procédure de désignation simultanée. Les deux terminaux connaissent leur statut. Le terminal de droite peut recevoir et émettre des commandes. Cependant, le terminal de gauche ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre de commandes qui dépendent de la connaissance du résultat de la désignation. Le terminal de gauche peut recevoir de telles commandes. Une deuxième procédure de désignation doit être lancée.

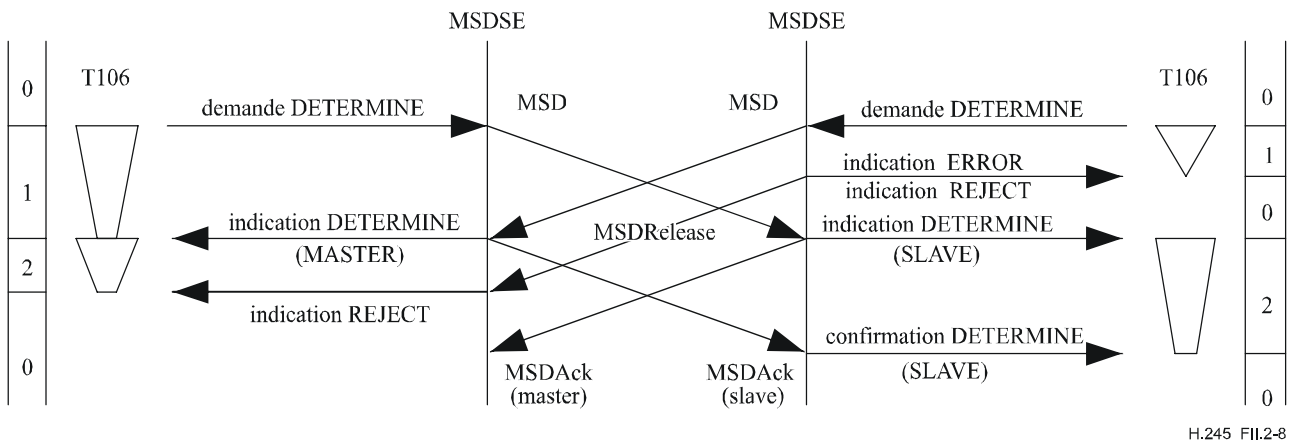


Figure II.2-8/H.245 – Désignation maître-esclave – Procédures de désignation simultanées avec expiration de la temporisation T106 côté terminal asservi

Dans la Figure II.2-9, la temporisation distante T106 est arrivée à expiration au cours de l'état INCOMING AWAITING ACKNOWLEDGEMENT d'une procédure de désignation simultanée. Les deux terminaux connaissent leur statut. Le terminal de gauche peut recevoir et émettre des commandes. Cependant, le terminal de droite ne sait pas si l'autre terminal est prêt à recevoir, et il ne peut pas émettre de commandes qui dépendent de la connaissance du résultat de la désignation. Le terminal de droite peut recevoir de telles commandes. Une deuxième procédure de désignation doit être lancée.

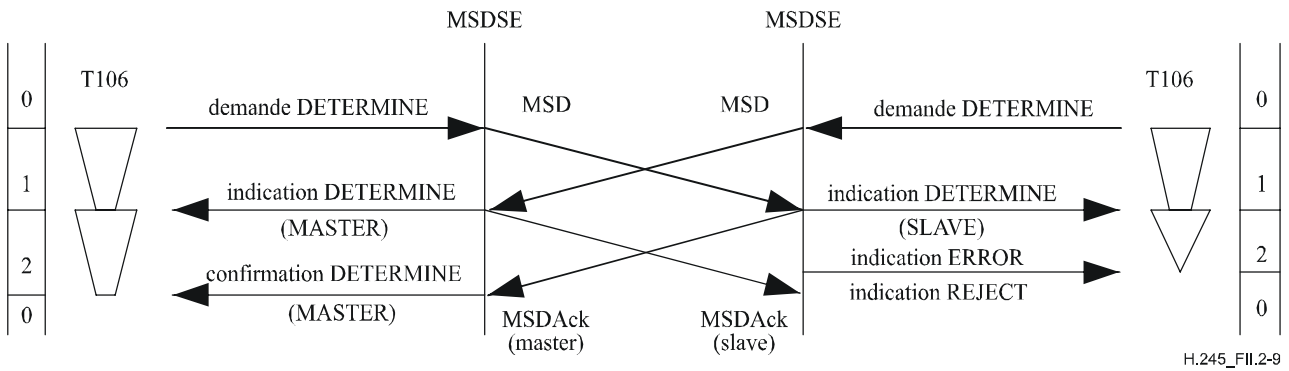


Figure II.2-9/H.245 – Désignation maître-esclave – Procédures de désignation simultanées avec expiration de la temporisation T106 au cours de l'état INCOMING AWAITING ACKNOWLEDGEMENT

Dans la Figure II.2-10, un résultat indéterminé a été obtenu N100 fois. Dans ce cas, N100 = 3.

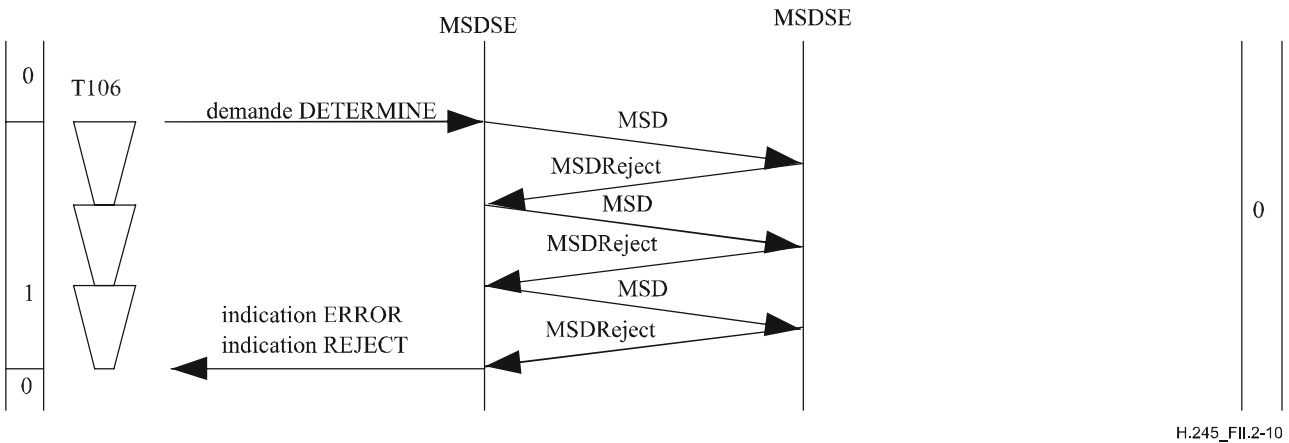


Figure II.2-10/H.245 – Désignation maître-esclave – Résultat indéterminé avec N100 = 3

II.3 Entité de signalisation d'échange de capacités (CESE)

Les Figures II.3-1 à II.3-4 illustrent les procédures relatives à l'entité CESE. Les états IDLE et AWAITING RESPONSE sont respectivement désignés par "0" et "1".

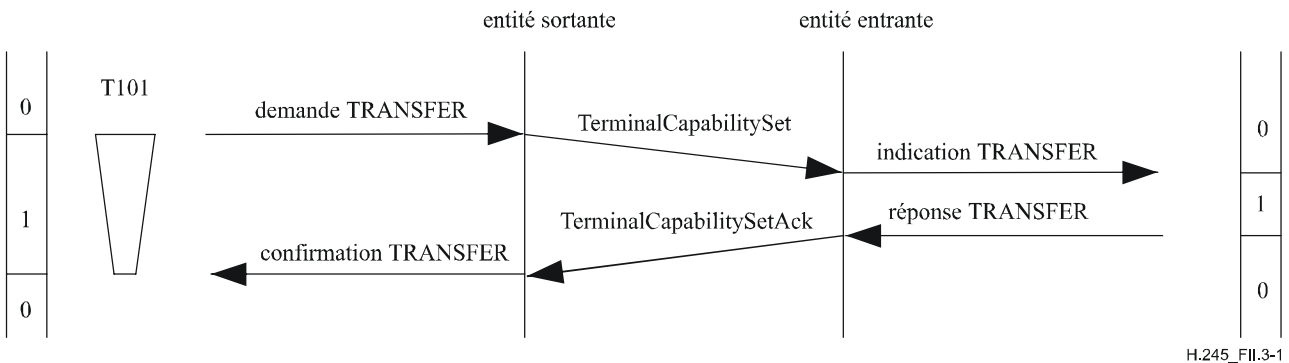


Figure II.3-1/H.245 – Echange de capacités avec acceptation provenant de l'utilisateur de l'entité CESE homologue entrante

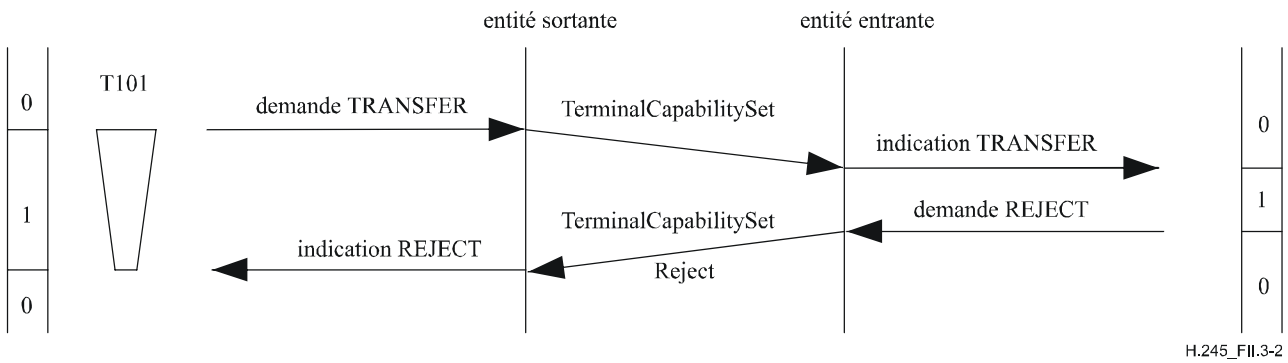
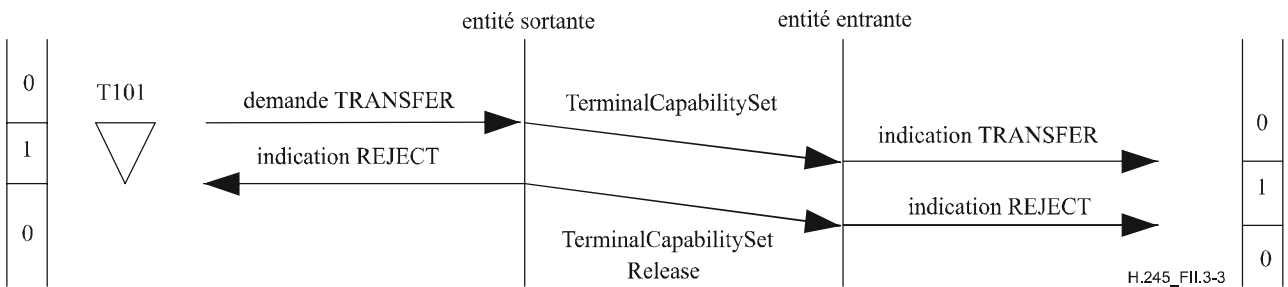
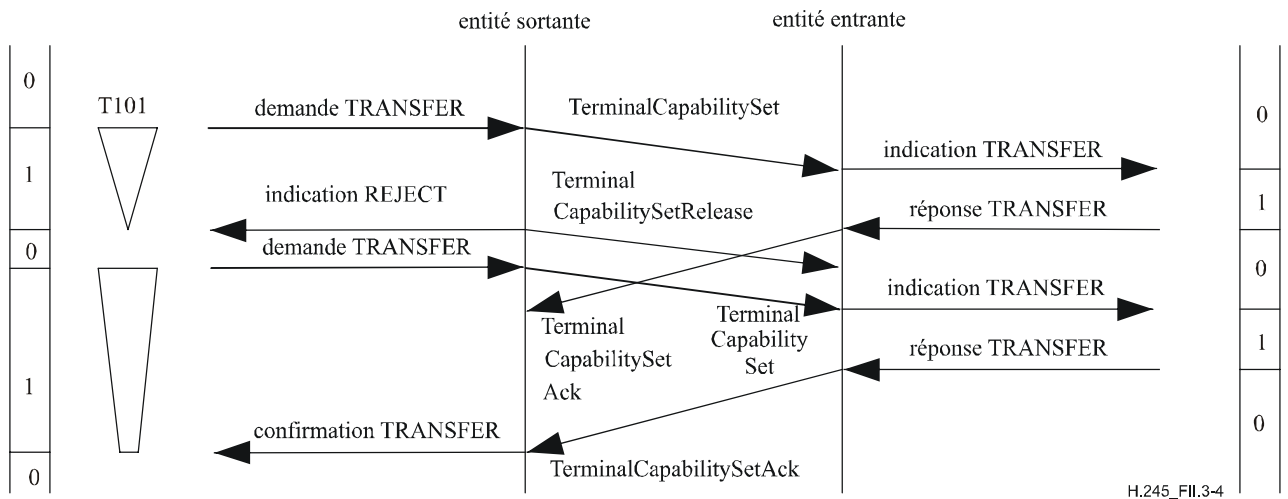


Figure II.3-2/H.245 – Echange de capacités avec rejet provenant de l'utilisateur de l'entité CESE homologue entrante



NOTE – Le message TerminalCapabilitySetRelease arrive dans l'entité CESE entrante avant la réponse provenant de l'utilisateur de l'entité CESE entrante.

Figure II.3-3/H.245 – Echange de capacités avec expiration de la temporisation T101



NOTE – Le message TerminalCapabilitySetRelease arrive dans l'entité CESE entrante après la réponse provenant de l'utilisateur de l'entité CESE entrante. Dans l'entité CESE sortante, le message TerminalCapabilitySetAck, envoyé en réponse au premier message TerminalCapabilitySet, est ignoré. Seul le second échange de capacités a abouti.

Figure II.3-4/H.245 – Echange de capacités avec expiration de la temporisation T101 suivi d'un deuxième échange de capacités

II.4 Entité de signalisation de voie logique (LCSE)

Les Figures II.4-1 à II.4-7 illustrent les procédures relatives à l'entité LCSE. Les états suivants d'une entité LCSE sortante: RELEASED, AWAITING ESTABLISHMENT, ESTABLISHED et AWAITING RELEASE sont respectivement désignés par "0", "1", "2" et "3". Les états suivants d'entité LCSE entrante: RELEASED, AWAITING ESTABLISHMENT et ESTABLISHED sont respectivement désignés par "0", "1" et "2".

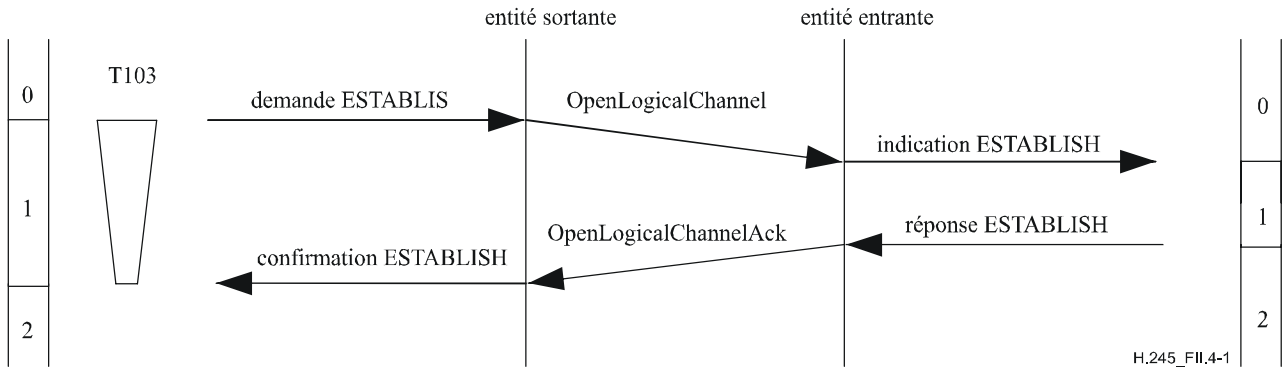


Figure II.4-1/H.245 – Etablissement de voie logique

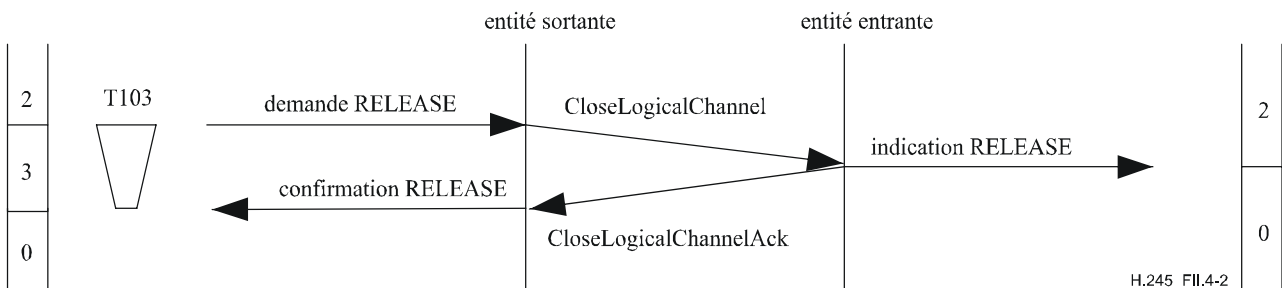


Figure II.4-2/H.245 – Libération de voie logique

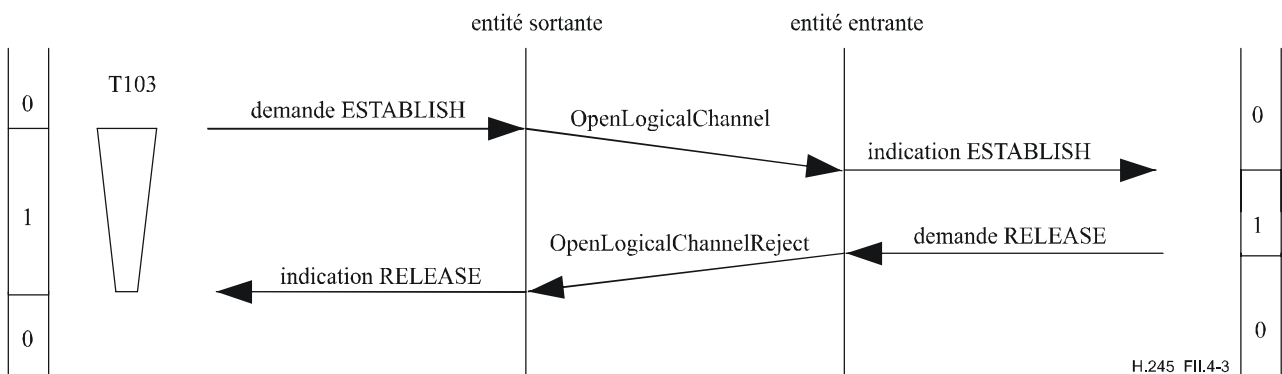


Figure II.4-3/H.245 – Rejet d'établissement de voie logique par l'utilisateur de l'entité LCSE homologue

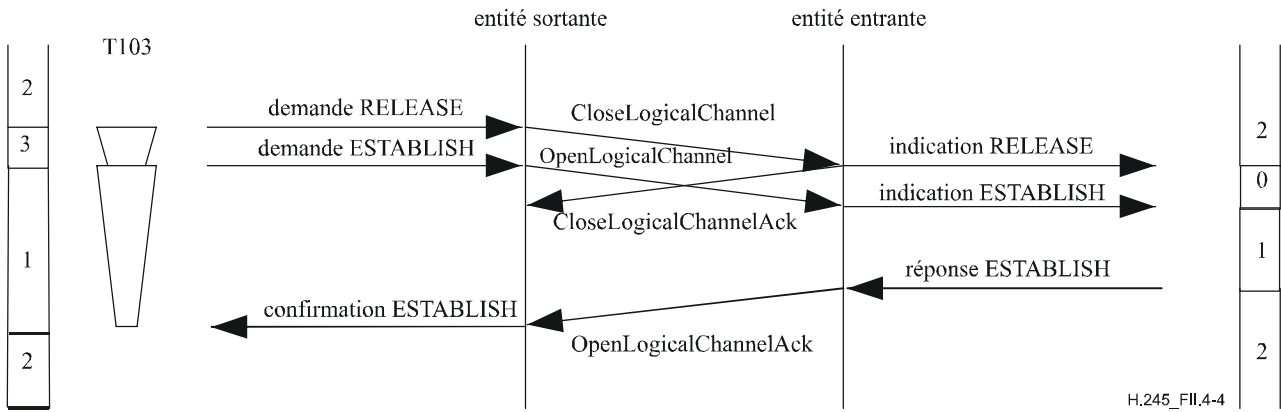


Figure II.4-4/H.245 – Libération de voie logique suivie du rétablissement immédiat

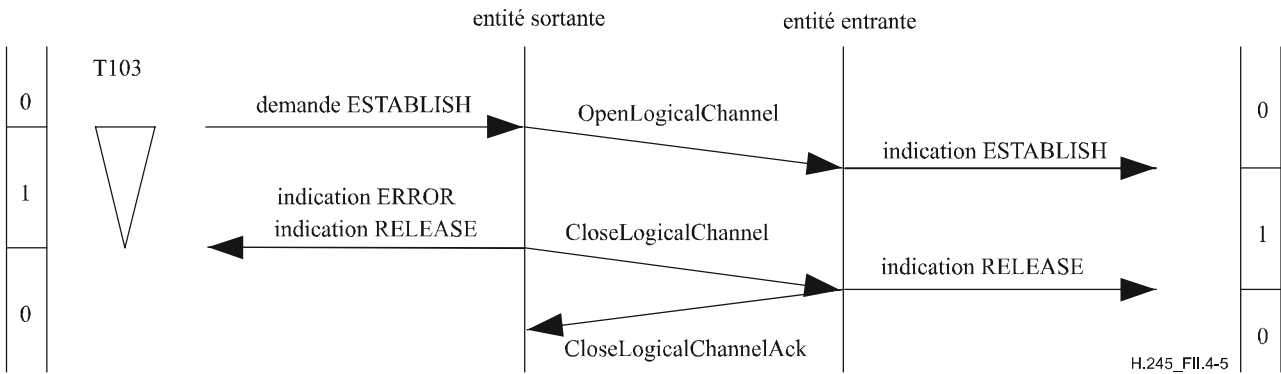
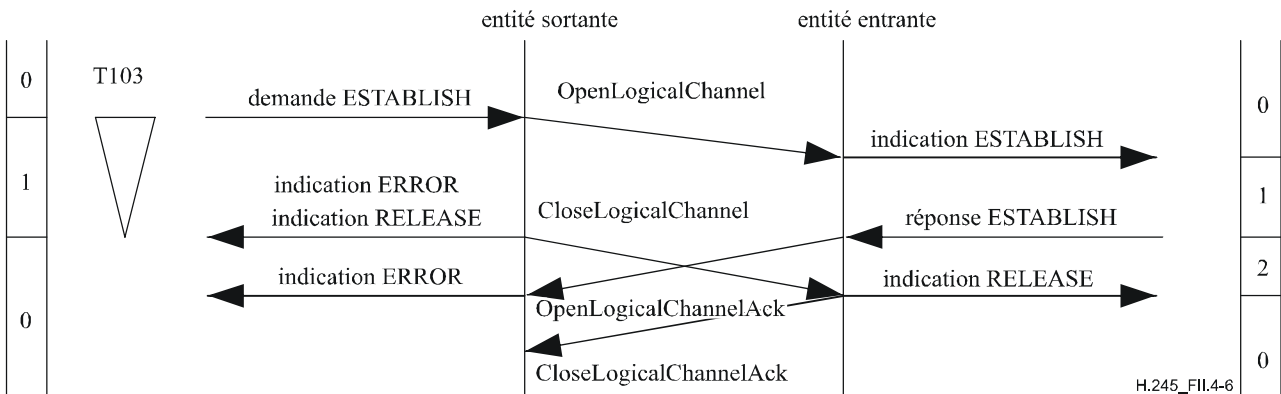


Figure II.4-5/H.245 – Demande d'établissement de voie logique avec expiration de la temporisation T103 en raison d'un retard de la réponse provenant de l'utilisateur d'entité LCSE homologue entrante



NOTE – L'expiration de la temporisation T103 a lieu après la transmission du message OpenLogicalChannelAck dans l'entité LCSE entrante, mais avant la réception du message OpenLogicalChannelAck dans l'entité LCSE sortante.

Figure II.4-6/H.245 – Demande d'établissement de voie logique avec expiration de la temporisation T103

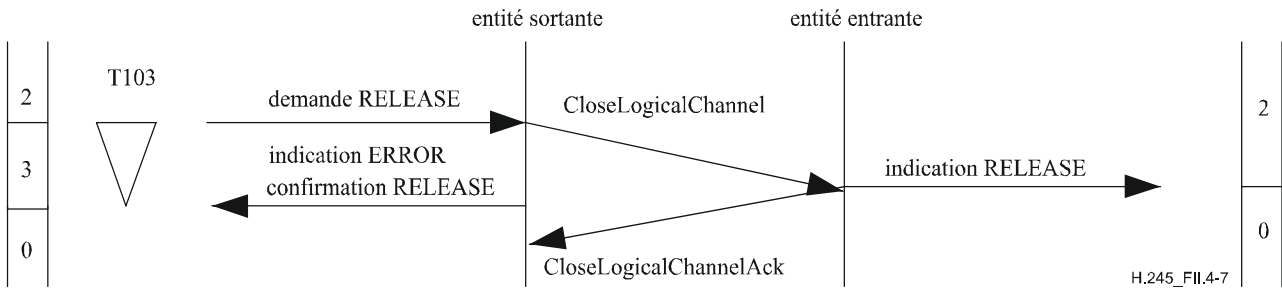


Figure II.4-7/H.245 – Demande de libération de voie logique avec expiration de la temporisation T103

II.5 Entité de signalisation de fermeture de voie logique (CLCSE)

Les Figures II.5-1 à II.5-4 illustrent les procédures relatives à l'entité CLCSE. Les états IDLE et AWAITING RESPONSE sont désignés par "0" et "1" respectivement.

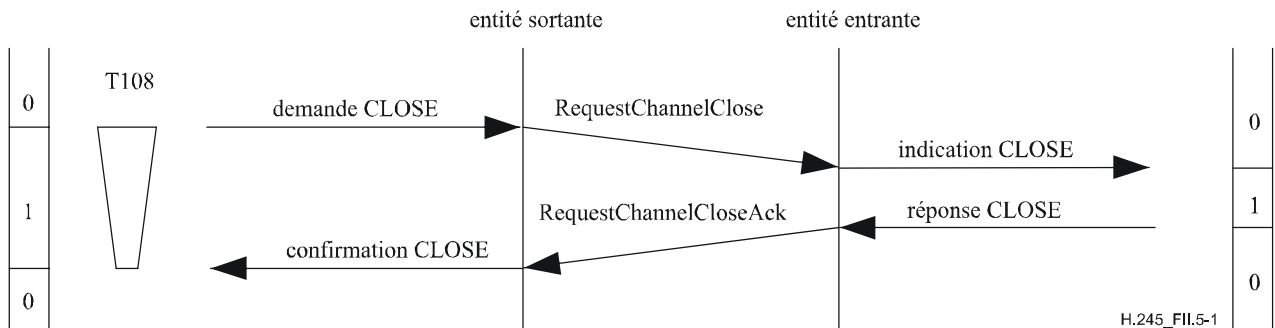


Figure II.5-1/H.245 – Demande de fermeture de voie logique

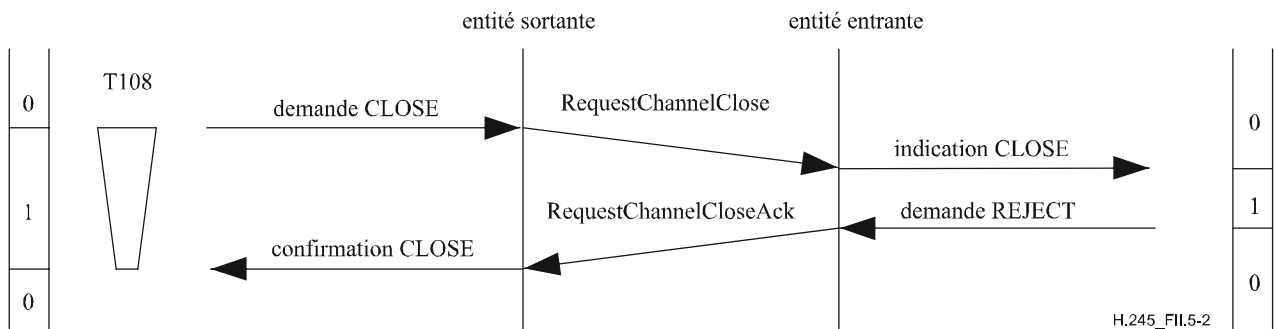
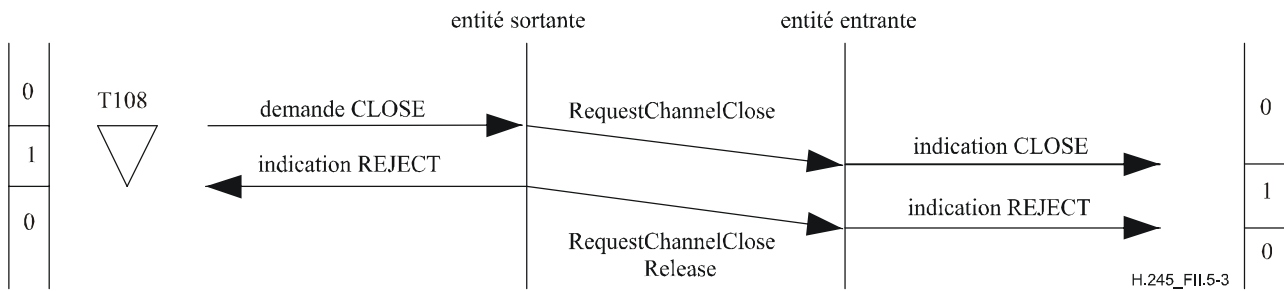
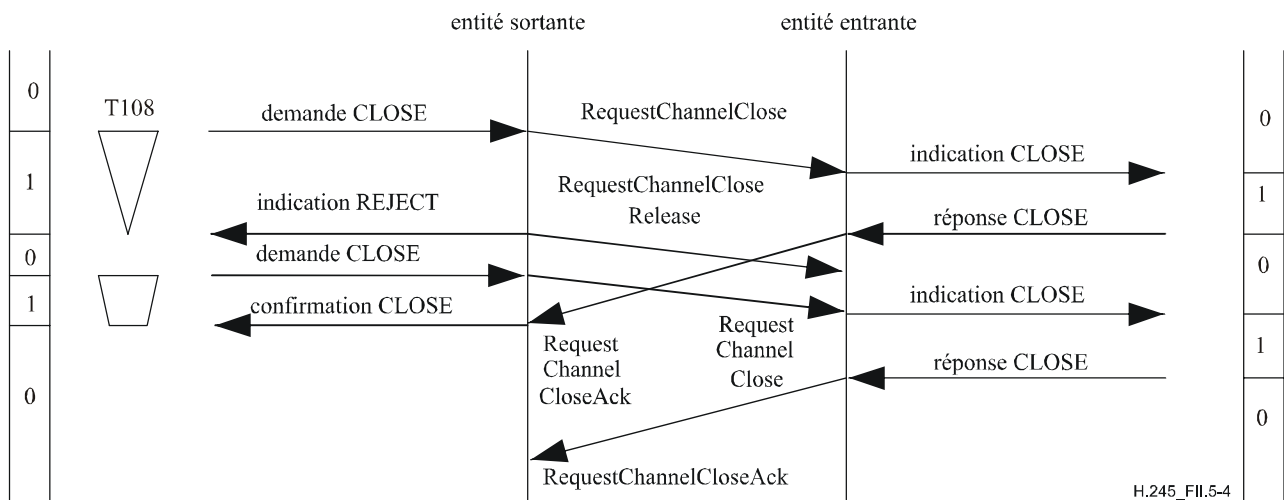


Figure II.5-2/H.245 – Demande de fermeture de voie logique avec rejet provenant de l'utilisateur de l'entité CLCSE homologue entrante



NOTE – Le message RequestChannelCloseRelease arrive dans l'entité CLCSE entrante avant la réponse provenant de l'utilisateur de l'entité CLCSE entrante.

Figure II.5-3/H.245 – Demande de fermeture de voie logique avec expiration de la temporisation T108



NOTE – La demande de fermeture de voie est confirmée dès la réception du premier message RequestChannelClose.

Figure II.5-4/H.245 – Demande de fermeture de voie logique avec expiration de la temporisation T108, suivie d'une deuxième demande de fermeture de voie logique

II.6 Entité de signalisation du tableau de multiplexage (MTSE)

Les Figures II.6-1 à II.6-5 illustrent les procédures relatives à l'entité MTSE. Les états IDLE et AWAITING RESPONSE sont respectivement désignés par "0" et "1".

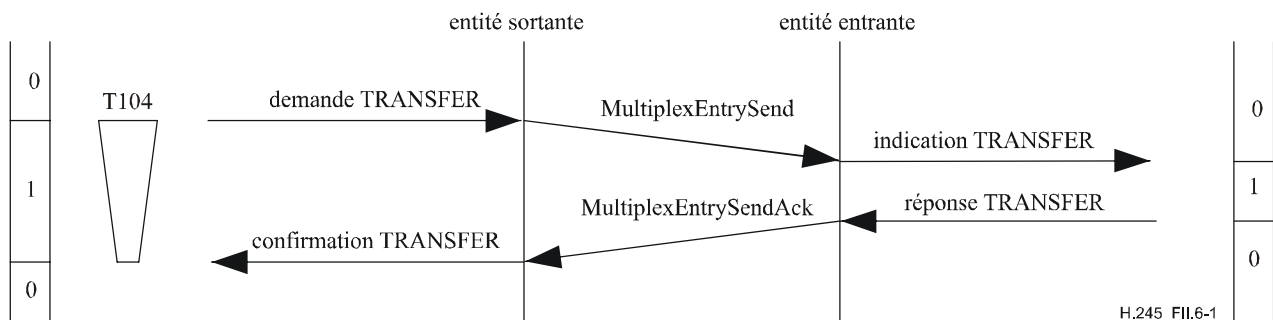


Figure II.6-1/H.245 – Aboutissement de la demande d'envoi du tableau de multiplexage

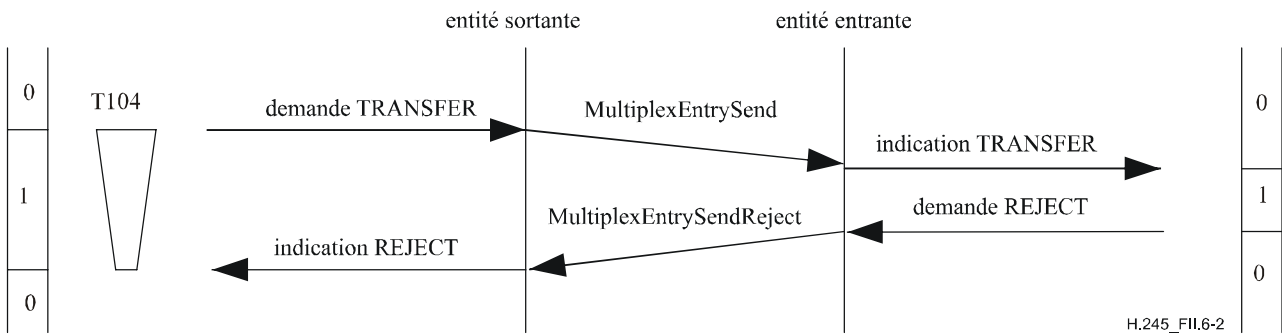
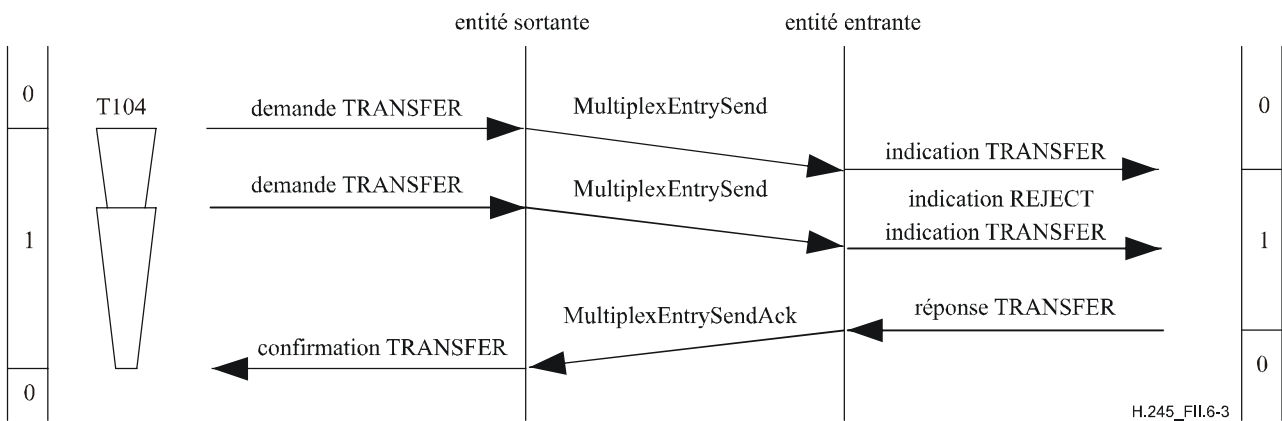


Figure II.6-2/H.245 – Demande d'envoi du tableau de multiplexage avec rejet provenant de l'utilisateur de l'entité MTSE homologue



NOTE – La première demande n'a pas abouti.

Figure II.6-3/H.245 – Demande d'envoi du tableau de multiplexage avec une deuxième demande d'envoi de tableau de multiplexage avant l'acquittement de la première demande

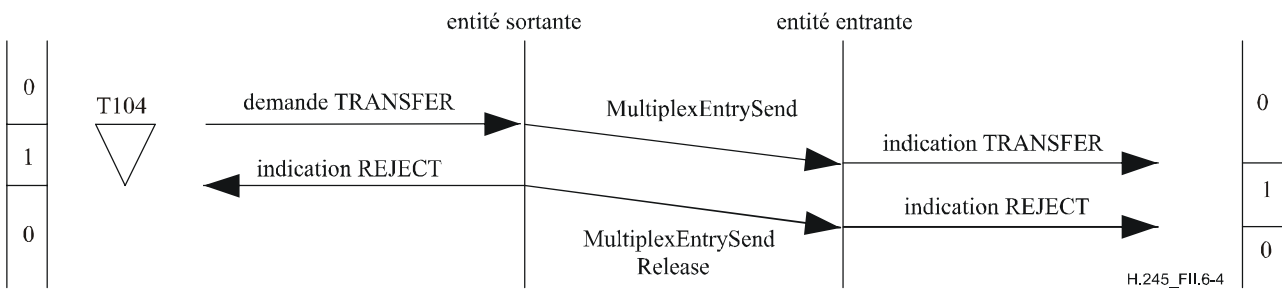
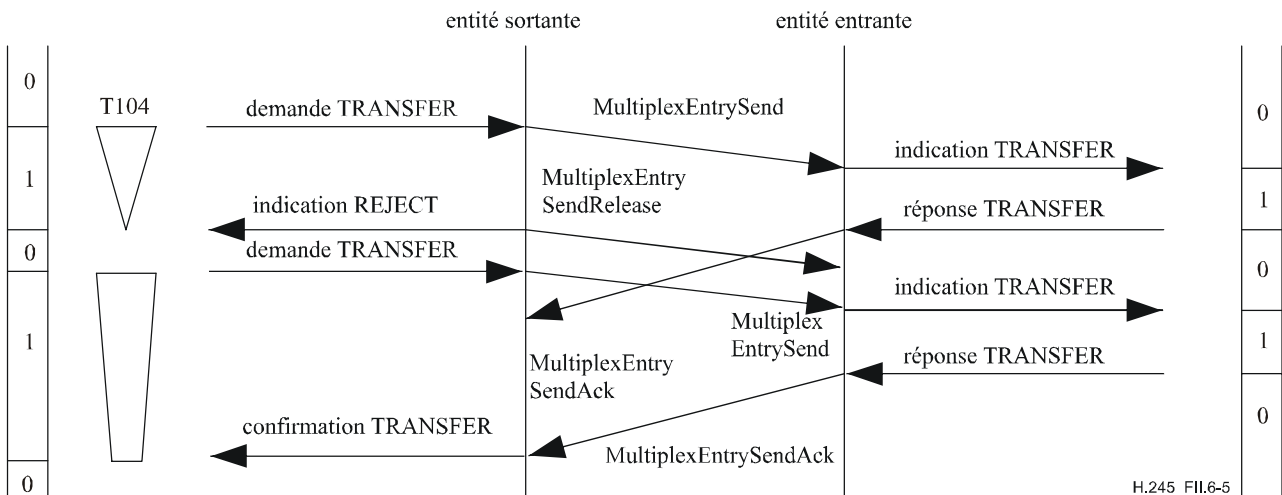


Figure II.6-4/H.245 – Demande d'envoi de tableau de multiplexage avec expiration de la temporisation T104



NOTE – Le premier message MultiplexEntrySendAck est ignoré dans l'entité MTSE sortante. Seule la deuxième demande a abouti.

Figure II.6-5/H.245 – Demande d'envoi de tableau de multiplexage avec expiration de la temporisation T104 suivie d'une deuxième demande d'envoi de tableau de multiplexage

II.7 Entité de signalisation de demande de mode (MRSE, mode request signalling entity)

Les Figures II.7-1 à II.7-5 illustrent les échanges entre entités MRSE. Les états IDLE et AWAITING RESPONSE sont désignés par "0" et "1" respectivement.

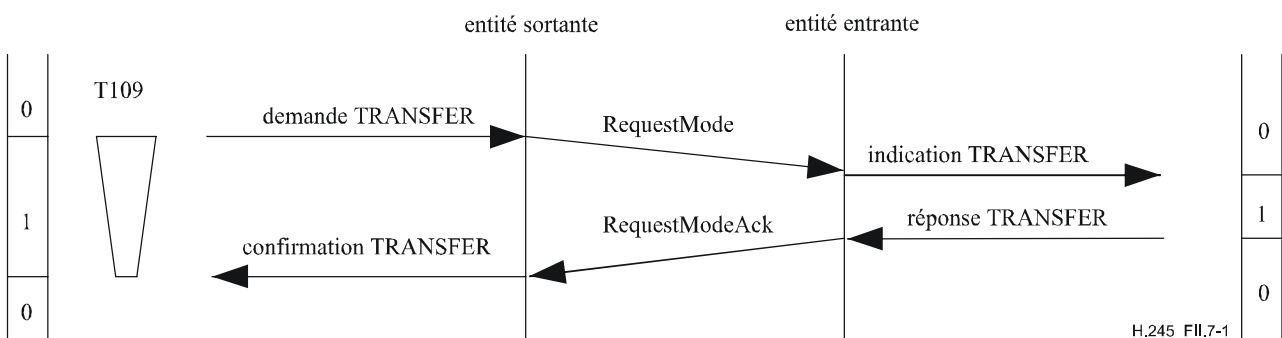


Figure II.7-1/H.245 – Aboutissement de demande de mode

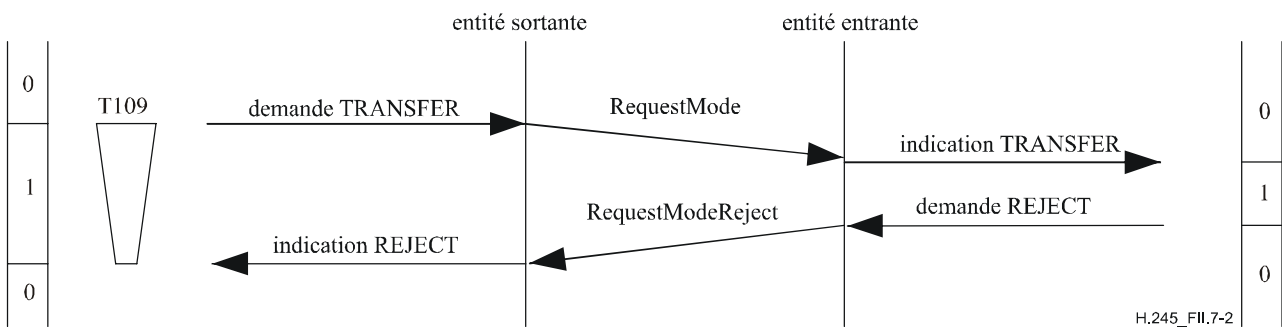
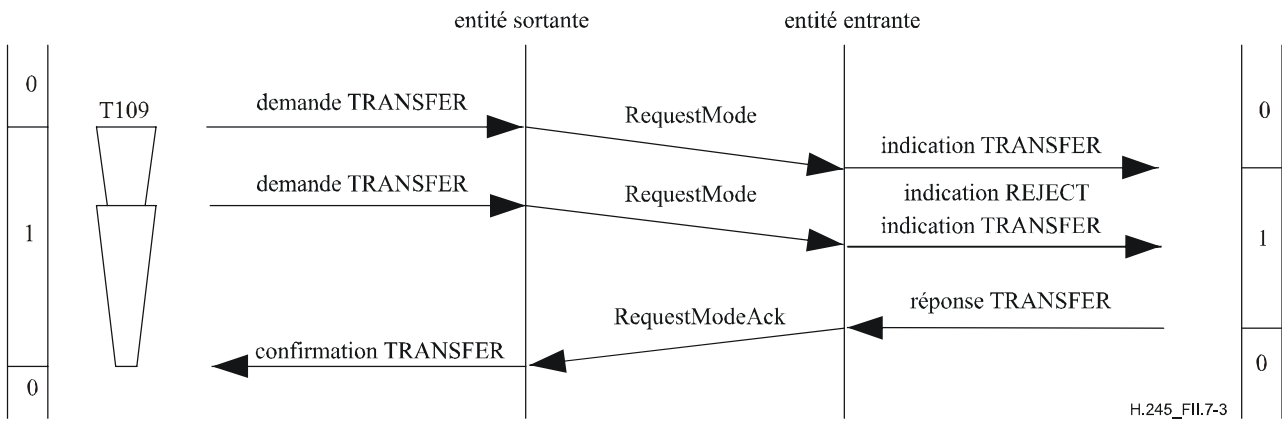
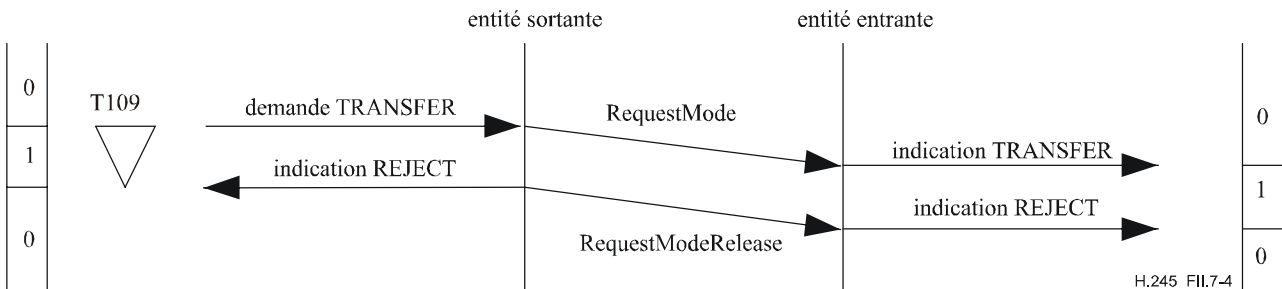


Figure II.7-2/H.245 – Demande de mode avec rejet provenant de l'utilisateur de l'entité MRSE homologue



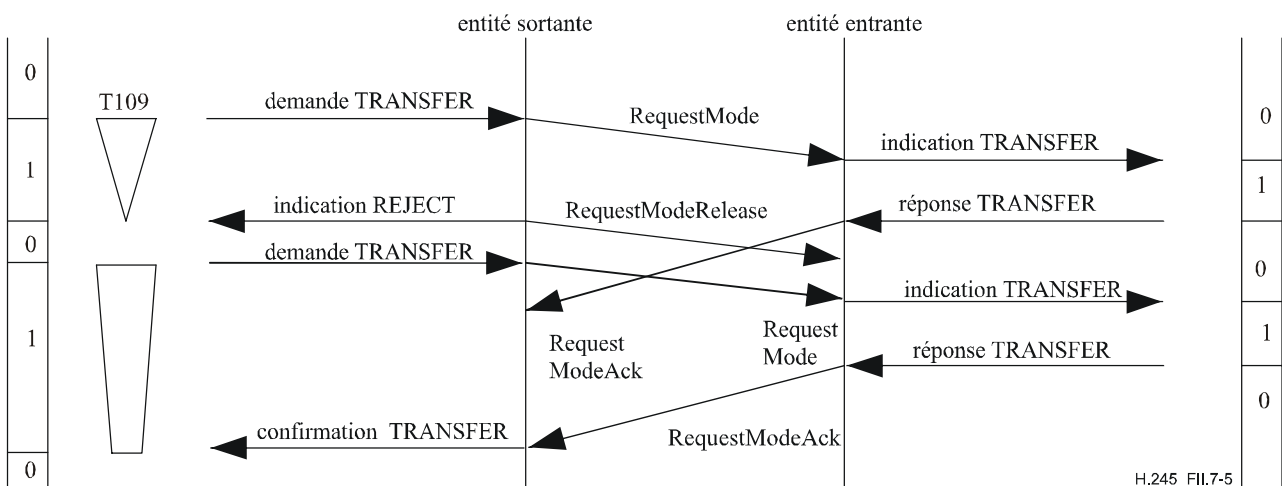
NOTE – La première demande a échoué.

Figure II.7-3/H.245 – Demande de mode avec deuxième demande de mode avant acquittement de la première demande



NOTE – La demande de mode a échoué.

Figure II.7-4/H.245 – Demande de mode avec expiration de la temporisation T109



NOTE – Le premier message RequestModeAck est ignoré dans l'entité MRSE sortante. Seule la deuxième demande a abouti.

Figure II.7-5/H.245 – Demande de mode avec expiration de la temporisation T109, suivie d'une deuxième demande de mode

II.8 Entité de signalisation de temps de propagation aller et retour (RTDSE)

Les Figures II.8-1 à II.8-4 illustrent les procédures d'entité RTDSE. Les états IDLE et AWAITING RESPONSE de l'entité sont désignés respectivement par "0" et "1".

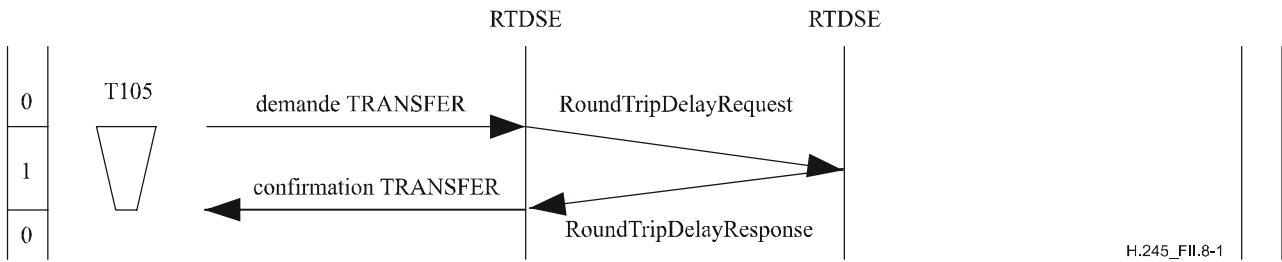


Figure II.8-1/H.245 – Procédure de détermination du temps de propagation aller et retour

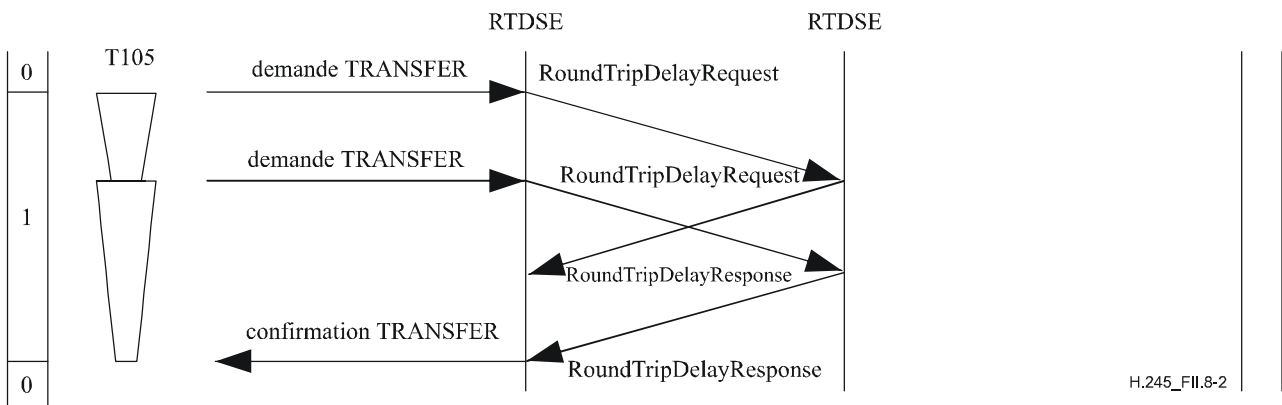


Figure II.8-2/H.245 – Procédure de détermination du temps de propagation aller et retour avec procédure de détermination du temps de propagation aller et retour antérieure non acquittée et en attente

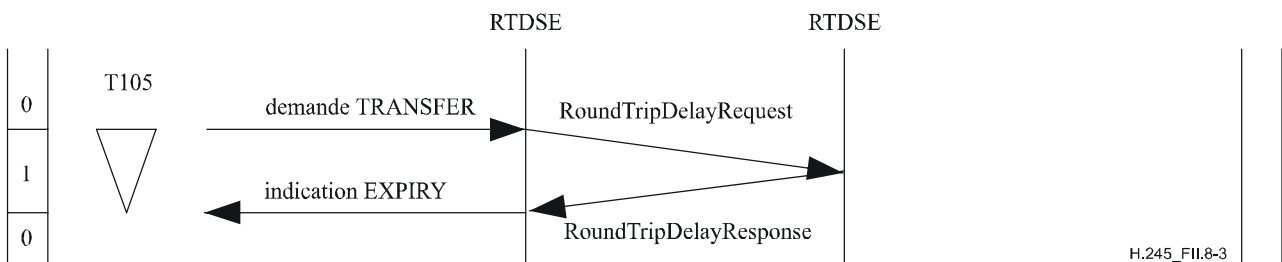
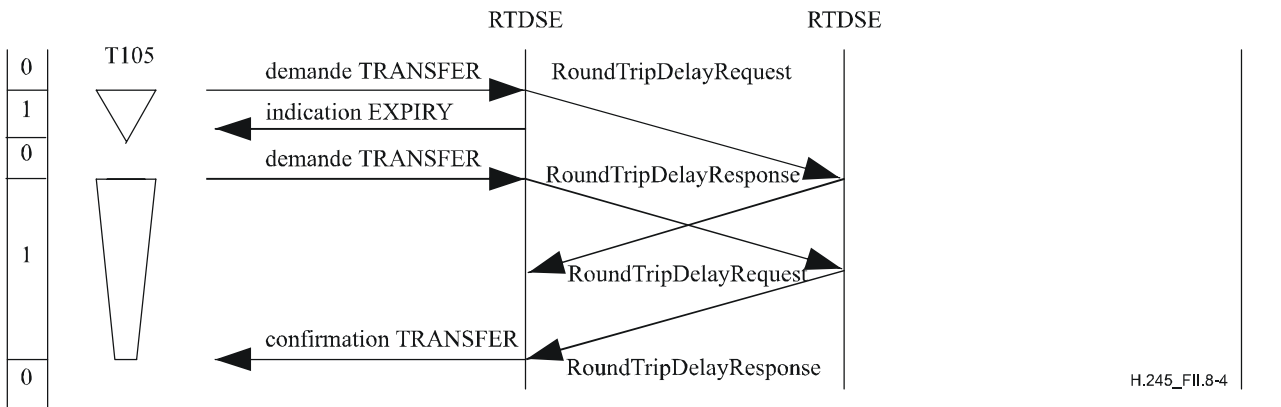


Figure II.8-3/H.245 – Procédure de détermination du temps de propagation aller et retour avec expiration de la temporisation T105



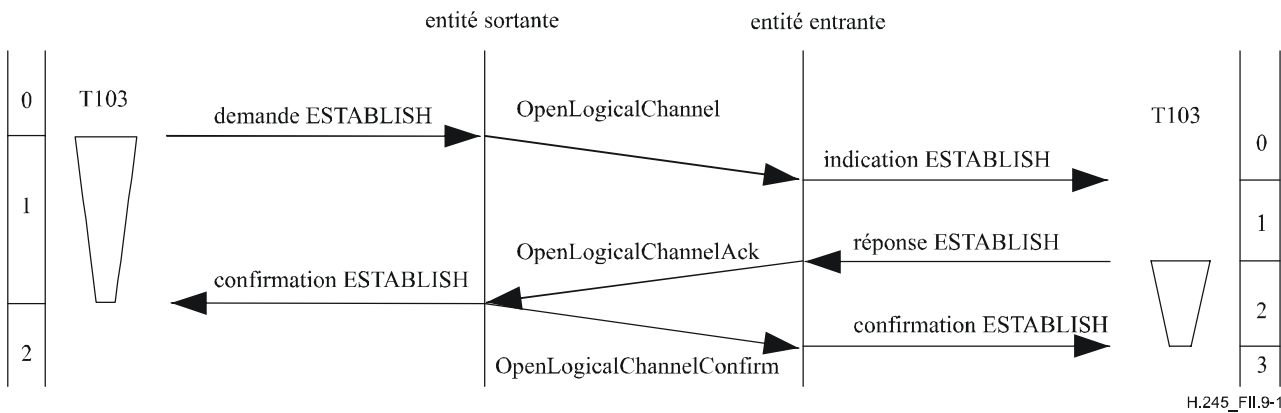
H.245_FII.8-4

NOTE – Le message RoundTripDelayResponse issu de la première procédure arrive pendant la deuxième procédure et est ignoré.

Figure II.8-4/H.245 – Procédure de détermination du temps de propagation aller et retour avec expiration de la temporisation T105, suivie d'une deuxième procédure de détermination du temps de propagation aller et retour

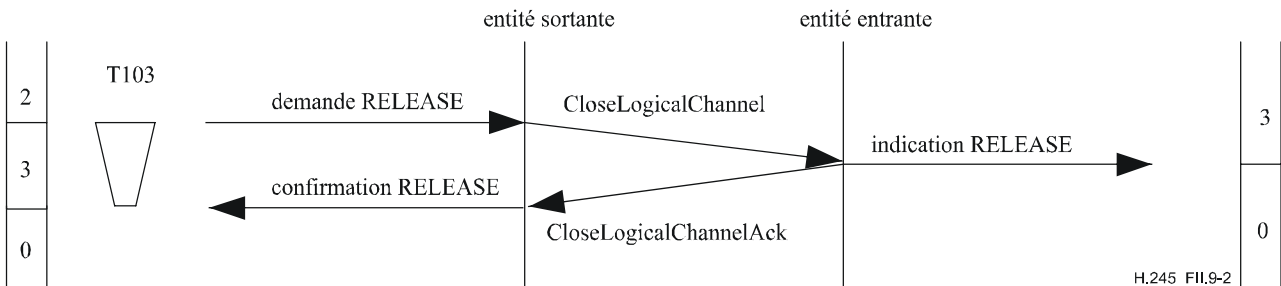
II.9 Entité de signalisation de voie logique bidirectionnelle (B-LCSE)

Les Figures II.9-1 à II.9-7 illustrent les procédures des entités B-LCSE. Les états RELEASED, AWAITING ESTABLISHMENT, ESTABLISHED et AWAITING RELEASE de l'entité B-LCSE sortante sont désignés par "0", "1", "2" et "3" respectivement. Les états RELEASED, AWAITING ESTABLISHMENT, AWAITING CONFIRMATION et ESTABLISHED de l'entité B-LCSE entrante sont désignés par "0", "1", "2" et "3" respectivement.



H.245_FII.9-1

Figure II.9-1/H.245 – Etablissement de voie logique bidirectionnelle



H.245_FII.9-2

Figure II.9-2/H.245 – Libération de voie logique bidirectionnelle

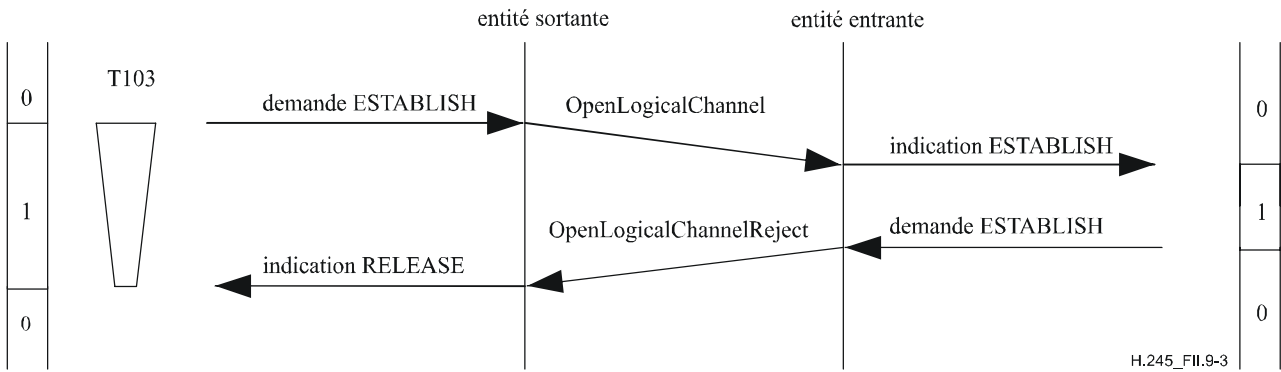


Figure II.9-3/H.245 – Rejet d'établissement de voie logique bidirectionnelle par l'utilisateur de l'entité B-LCSE homologue

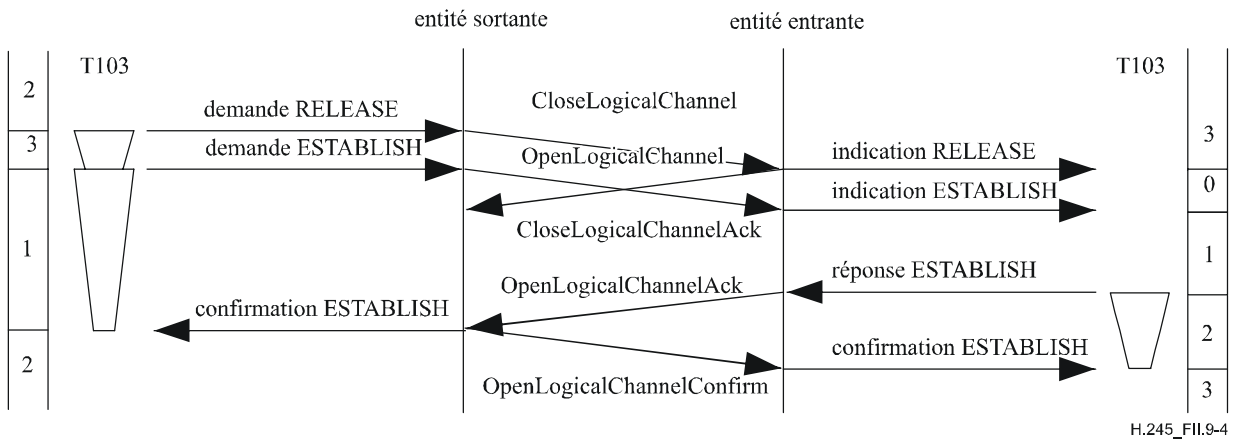


Figure II.9-4/H.245 – Libération de voie logique bidirectionnelle suivie par un rétablissement immédiat

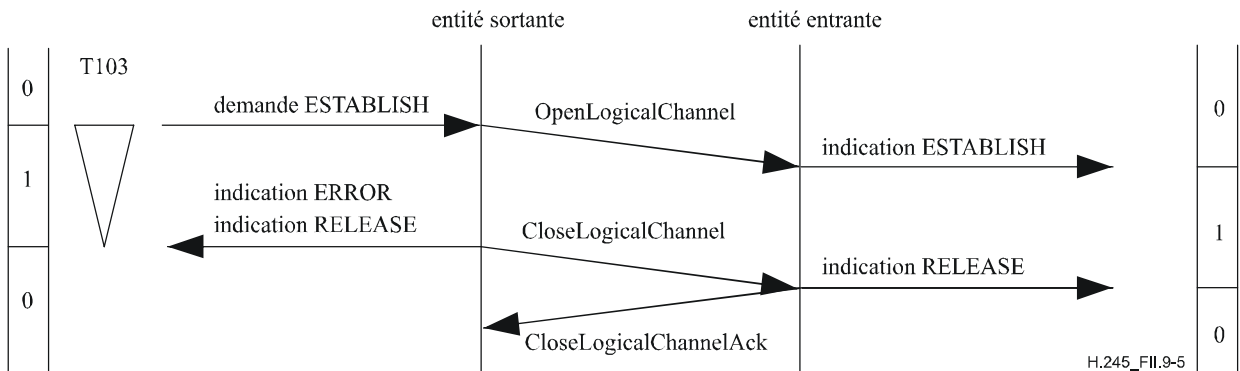
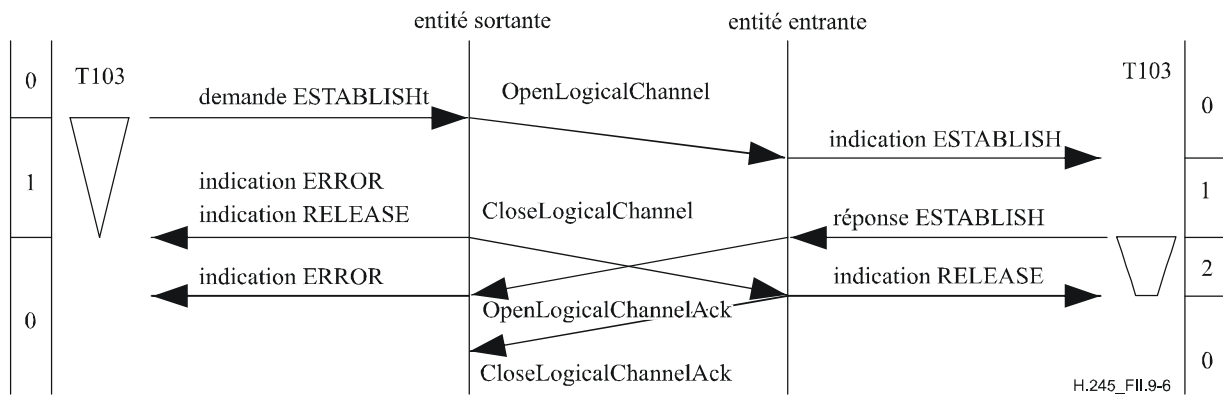


Figure II.9-5/H.245 – Demande d'établissement de voie logique bidirectionnelle avec fin de temporisation T103 du côté sortant par suite d'une réaction lente de l'utilisateur de l'entité B-LCSE entrante homologue



NOTE – La temporisation T103 du côté sortant est arrivée à expiration après la transmission du message OpenLogicalChannelAck dans l'entité B-LCSE entrante, mais avant la réception du message OpenLogicalChannelAck dans l'entité B-LCSE sortante.

Figure II.9-6/H.245 – Demande d'établissement de voie logique bidirectionnelle avec fin de temporisation T103 du côté sortant

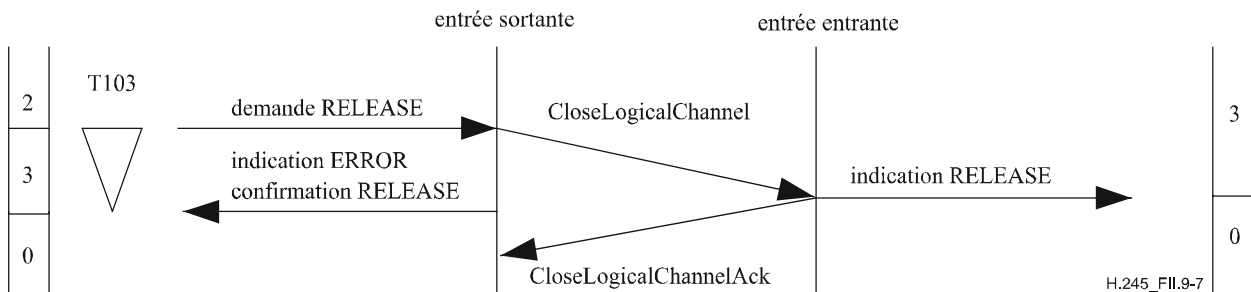


Figure II.9-7/H.245 – Demande de libération de voie logique bidirectionnelle avec fin de temporisation T103 du côté sortant

Appendice III

Récapitulatif des temporisateurs et des compteurs utilisés dans les procédures

Le présent appendice énumère les temporisateurs et les compteurs spécifiés dans l'Annexe C.

La présente Recommandation ne définit pas les valeurs à mettre dans ces temporisateurs. Les valeurs peuvent être définies dans d'autres Recommandations telles que les Recommandations UIT-T H.310, H.323 et H.324.

III.1 Temporisateurs

Le Tableau III.1 montre les temporisateurs spécifiés dans la présente Recommandation.

Tableau III.1/H.245 – Temporisateurs utilisés dans les procédures

Temporisateur	Procédure	Définition
T106	Désignation maître-esclave	Ce temporisateur est utilisé dans les états OUTGOING AWAITING RESPONSE et INCOMING AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message d'acquiescement 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 voie logique bidirectionnelle	Ce temporisateur est utilisé dans les états AWAITING ESTABLISHMENT et AWAITING RELEASE. Il spécifie la durée maximale pendant laquelle aucun message OpenLogicalChannelAck, OpenLogicalChannelReject ou CloseLogicalChannelAck ne peut être reçu.
T108	Fermeture de voie logique	Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RequestChannelCloseAck ou RequestChannelCloseReject ne peut être reçu.
T104	Tableau 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	Temps de propagation aller et retour	Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RoundTripDelayResponse ne peut être reçu.
T107	Demande d'entrée de multiplexage	Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message RequestMultiplexEntryAck ou RequestMultiplexEntryReject ne peut être reçu.
T102	Boucle de maintenance	Ce temporisateur est utilisé dans l'état AWAITING RESPONSE. Il spécifie la durée maximale pendant laquelle aucun message MaintenanceLoopAck ou MaintenanceLoopReject ne peut être reçu.

III.2 Compteurs

Le Tableau III.2 indique les compteurs spécifiés dans la présente Recommandation.

Tableau III.2/H.245 – Compteurs utilisés dans les procédures

Compteur	Procédure	Définition
N100	Désignation maître-esclave	Ce compteur spécifie le nombre maximal d'envois de messages MasterSlaveDetermination au cours de l'état OUTGOING AWAITING RESPONSE.

Appendice IV

Extensions à la Recommandation H.245

La présente Recommandation est un "document évolutif" utilisé par de nombreuses Recommandations sur les systèmes, notamment H.310, H.323, H.324 et V.70. Il est prévu que des extensions autorisant une compatibilité amont lui soient fréquemment apportées, probablement lors de chaque réunion de la Commission d'études 16 de l'UIT-T. Le présent appendice explique les procédures qui doivent être utilisées pour apporter des extensions à la présente Recommandation.

Pour une période considérée, une seule syntaxe H.245 est en vigueur. Aucune autre Recommandation de l'UIT-T ne doit inclure des variantes de syntaxe H.245 de façon normative.

Les demandes d'extensions à la présente Recommandation doivent être soumises, sous forme de contribution blanche ou de document de liaison, à la Commission d'études 16 de l'UIT-T, et une copie doit être envoyée dès que possible au rapporteur et au rédacteur de la Rec. H.245. De telles demandes doivent inclure:

- 1) des spécifications générales de la syntaxe qui doit être élaborée par le rédacteur de la H.245 ou une proposition de syntaxe fondée sur la version actuellement approuvée de la présente Recommandation;
- 2) une proposition de sémantique pour l'Annexe B;
- 3) une proposition de procédures pour l'Annexe C si de nouvelles procédures sont demandées.

Toutes les extensions à la présente Recommandation doivent permettre une compatibilité amont avec ses versions antérieures. La syntaxe, la sémantique et les procédures existantes ne peuvent pas être modifiées. La signification de la syntaxe existante ne peut pas être modifiée. En l'occurrence, en cas d'extension d'une capacité H.245, une telle extension ne doit pas modifier la signification de la capacité initiale d'une manière telle qu'un terminal qui ne comprend pas l'extension devrait modifier son fonctionnement afin d'utiliser la capacité sans l'extension. Tous les éléments d'extension ASN.1 doivent être limités.

Les demandes doivent être soumises dès que possible pour laisser le temps aux experts de la Rec. H.245 dans la Commission d'études 16 d'examiner les extensions. Il est entendu que la syntaxe exacte demandée peut être modifiée pour les raisons suivantes:

- 1) vérification de la syntaxe ASN.1 correcte;
- 2) harmonisation avec d'autres demandes concomitantes pour des extensions à H.245;
- 3) compatibilité amont avec les versions existantes de H.245;
- 4) avis des experts pour l'ajout de nouvelles fonctions dans la structure existante de H.245;
- 5) dénomination incompatible avec la syntaxe existante;
- 6) éléments ASN.1 illimités ou ambigus.

Les abréviations et les sigles doivent être évités, en particulier lorsqu'un mot ou une expression n'est pas abrégé ou ne figure pas sous forme de sigle dans la syntaxe existante. Le mot "Parameter", par exemple, ne devrait pas être abrégé en "Param". Si un mot a été utilisé dans la syntaxe existante, il convient de ne pas utiliser un autre mot ayant la même signification. Ainsi, si le terme "Entry" a été régulièrement utilisé pour décrire des éléments d'un type de structure de données, il convient de l'employer au lieu du terme "Item", par exemple. L'homogénéité est de rigueur.

Tous les éléments ASN.1 doivent être limités et la manière de limiter les types les plus courants est décrite ci-après.

Il convient de limiter les éléments SET OF et SEQUENCE OF du langage ASN.1 en définissant une taille minimale et une taille maximale ou une taille fixe. Si aucune limite maximale n'est

inhérente à la sémantique d'un élément, il faut choisir une limite maximale raisonnable, bien qu'arbitraire, telle que la valeur 256, par exemple. Si un élément SET OF ou SEQUENCE OF est OPTIONAL (facultatif), une valeur minimale différente de zéro doit être spécifiée sauf s'il existe une différence de sémantique entre les cas "présent mais vide" et "non présent", auquel cas la différence de sémantique doit être décrite. Si une demande d'ajout contient des éléments SET OF ou SEQUENCE OF illimités, le rédacteur peut utiliser SIZE (1..256) comme limite par défaut.

Il est nécessaire de limiter les chaînes de caractères ASN.1 en définissant une taille minimale et une taille maximale ou une taille fixe. Si une demande d'ajout contient des éléments chaînes de caractères illimités, le rédacteur peut utiliser SIZE (0..255) comme limite par défaut.

Il convient de limiter les éléments INTEGER (nombre entier) en définissant un intervalle de valeurs. Si aucun intervalle de valeurs n'est inhérent à la sémantique d'un élément, il est nécessaire de choisir un intervalle raisonnable, bien qu'arbitraire, dont la valeur maximale est sélectionnée parmi les valeurs suivantes:

255	$(2^8 - 1)$
65535	$(2^{16} - 1)$
16777215	$(2^{24} - 1)$
4294967295	$(2^{32} - 1)$

Si une demande d'ajout contient des éléments INTEGER illimités, le rédacteur peut utiliser, par défaut, INTEGER (0..4294967295).

Le rédacteur de la Rec. UIT-T H.245 étudiera toutes les demandes d'ajout et proposera le texte définitif des versions étendues de la Rec. UIT-T H.245 en vue de l'approbation par la Commission d'études 16. A chaque approbation d'une nouvelle version de la présente Recommandation par la Commission d'études, le numéro de version H.245 contenu dans **protocolIdentifier** sera incrémenté de un pour identifier la nouvelle version.

Il convient de remarquer que la Commission d'études 16 a l'intention de n'accepter que des extensions harmonisées de la Rec. UIT-T H.245 qui émanent du rédacteur de la Rec. UIT-T H.245.

Appendice V

Procédure **replacementFor**

La procédure **replacementFor** H.245 permet de procéder à un changement imperceptible de mode d'un codec à un autre sans recourir à deux décodeurs de médias. Elle peut être utilisée uniquement si le terminal récepteur a indiqué la capacité **maxPendingReplacementFor**.

L'ouverture et la fermeture des voies logiques H.245 n'étant pas synchronisées avec le contenu des médias, ceux-ci peuvent disparaître entre la fermeture d'une voie logique et l'ouverture de la voie de remplacement. Le paramètre **replacementFor** permet d'éviter une telle disparition.

Exemple

Supposons que la voie logique 723 soit ouverte pour l'acheminement de données audio G.723.1 et que l'on souhaite passer au mode G.711 (sur la voie logique 711), mais que le récepteur ne puisse accepter qu'une seule voie audio. La procédure **replacementFor** peut être utilisée par l'émetteur pour effectuer un changement imperceptible de mode de la manière suivante:

- 1) *uniquement dans le cas d'un système H.323 utilisant le protocole RSVP*, étant donné que la nouvelle voie demandera une plus grande largeur de bande (64 kbit/s) que la voie existante (6,4 kbit/s), l'émetteur et le récepteur réservent une largeur de bande RSVP plus grande;
- 2) l'émetteur envoie le message `OpenLogicalChannel` pour la nouvelle voie logique 711, y compris le paramètre `replacementFor`, qui se rapporte à la voie logique 723 existante.
Ce paramètre indique au récepteur que la voie logique 711 *remplace* la voie logique 723 et que la voie logique 711 n'assurera jamais un trafic simultanément avec la voie logique 723;
- 3) tout en continuant à décoder les données G.723.1 de la voie logique 723, le récepteur se prépare en vue d'un passage imperceptible au décodage G.711.
Une telle préparation peut comporter le chargement d'un logiciel de décodage approprié.
Lorsque le récepteur est prêt à accepter le flux de données audio G.711, il répond par le message `OpenLogicalChannelAck` pour la voie logique 711. Pour les systèmes selon la Rec. UIT-T H.323, les adresses de transport de médias et de commande de médias retournées sont les mêmes que celles qui sont déjà utilisées pour la voie logique 723;
- 4) l'émetteur cesse d'envoyer des données audio G.723 sur la voie logique 723 et commence, de manière imperceptible, à envoyer des données audio G.711 sur la voie logique 711;
- 5) l'émetteur envoie immédiatement un message `CloseLogicalChannel` pour la voie logique 723, cette voie n'assurant plus de trafic et n'étant donc plus nécessaire;
- 6) *uniquement dans le cas d'un système H.323 utilisant le protocole RSVP*, si la nouvelle voie exige une largeur de bande moins grande que la voie initiale, l'émetteur et le récepteur réservent une largeur de bande RSVP plus petite (ceci n'est pas applicable dans cet exemple).

Dans tous les cas, le fonctionnement des entités LCSE et B-LCSE est conforme aux procédures normales. Le paramètre `replacementFor` informe simplement le récepteur qu'un changement de mode va se produire, que les deux voies logiques ne seront pas utilisées simultanément et que, par conséquent, la seconde voie logique peut (dans certaines implémentations) être acceptée dans des cas où elle serait normalement rejetée (en raison de l'absence de la capacité de recevoir une autre voie logique indépendante).

Il convient de noter que le récepteur peut, dans certains cas, rejeter la tentative d'ouvrir la voie logique au moyen du mécanisme `replacementFor` (par exemple, si un récepteur peut accepter le mécanisme `replacementFor` pour des voies audio, mais pas pour des voies vidéo). Dans ce cas, les émetteurs devraient tenter à nouveau de changer de mode sans recourir au mécanisme `replacementFor`, notamment en fermant la voie existante, puis en ouvrant une nouvelle, acceptant ainsi toute disparition temporaire de médias.

A noter également que dans les systèmes selon la Rec. UIT-T H.323, les récepteurs doivent réutiliser les adresses de transport de médias et de commande de médias existantes. Le point de passage à la nouvelle voie logique est marqué par l'en-tête de protocole RTP.

Appendice VI

Exemples de paramétrage des capacités H.263

Le présent appendice présente des exemples qui donnent des précisions sur l'utilisation des capacités H.263.

VI.1 Exemples de paramétrage des couches d'amélioration dans le mode H.245

Le Tableau VI.1 indique plusieurs exemples de paramétrage des couches d'amélioration.

Exemple n° 1: paramétrage d'une simple capacité vidéo de base H.263 à 10 trames par seconde, avec un débit maximal de 20 kbit/s et sans options.

Exemple n° 2: paramétrage de la capacité relative à une voie logique avec une couche d'amélioration spatiale et une résolution QCIF, à 10 trames par seconde, avec un débit binaire maximal de 5 kbit/s et sans autres options.

Exemple n° 3: paramétrage de la capacité relative à une voie logique avec une couche d'amélioration SNR et une résolution SQCIF, à 10 trames par seconde, avec un débit maximal de 5 kbit/s et sans autres options.

Exemple n° 4: paramétrage de la capacité relative à une voie logique avec trois couches d'amélioration, à savoir: deux couches d'amélioration SNR, l'une avec la résolution SQCIF et l'autre avec la résolution QCIF, à 10 trames par seconde et sans autres options, et une couche d'amélioration spatiale avec la résolution CIF, à 10 trames par seconde et sans autres options; les trois couches sont combinées à un débit maximal de 15 kbit/s.

Exemple n° 5: paramétrage de la capacité relative à une voie logique avec trois couches d'amélioration et une couche de base à un débit maximal de 25 kbit/s. La couche de base a une résolution QCIF sans options. En outre, le terminal accepte une couche d'amélioration SNR avec une résolution QCIF, à 10 trames par seconde et sans autres options, une couche d'amélioration SNR avec une résolution CIF, à 10 trames par seconde et sans autres options et une couche d'amélioration spatiale avec une résolution CIF, à 10 trames par seconde et sans autres options.

Tableau VI.1/H.245 – Exemples de paramétrage des couches d'amélioration dans le mode H.245

Paramètre de capacité H263	Exemples									
	1	2	3	4		5				
sqcifMPI	3	NP	NP	NP			NP			
qcifMPI	NP	NP	NP	NP			3			
cifMPI	NP	NP	NP	NP			NP			
cif4MPI	NP	NP	NP	NP			NP			
cif16MPI	NP	NP	NP	NP			NP			
maxBitRate	200	50	50	150			250			
unrestrictedVector	F	F	F	F			F			
arithmeticCoding	F	F	F	F			F			
advancedPrediction	F	F	F	F			F			
pbFrames	F	F	F	F			F			
temporalSpatialTradeOffCap	F	F	F	F			F			
hrd-B	NP	NP	NP	NP			NP			
bppMaxKb	NP	NP	NP	NP			NP			

Tableau VI.1/H.245 – Exemples de paramétrage des couches d'amélioration dans le mode H.245

		Exemples								
slowSqcifMPI		NP	NP	NP	NP			NP		
slowQcifMPI		NP	NP	NP	NP			NP		
slowCifMPI		NP	NP	NP	NP			NP		
slowCif4MPI		NP	NP	NP	NP			NP		
slowCif16MPI		NP	NP	NP	NP			NP		
errorCompensation		NP	NP	NP	NP			NP		
SET OF (EnhancementOptions ^{a)}) =		NP	NP	1	1	2		1	2	
snrEnhancement	sqcifMPI			3	3	NP		NP	NP	
	qcifMPI			NP	NP	3		3	NP	
	cifMPI			NP	NP	NP		NP	3	
	cif4MPI			NP	NP	NP		NP	NP	
	cif16MPI			NP	NP	NP		NP	NP	
	maxbitrate			50	50	50		50	50	
SET OF (EnhancementOptions ^{a)}) =		NP	1	NP	NP	NP	1	NP	NP	1
spatialEnhancement	sqcifMPI		NP				NP			NP
	qcifMPI		3				NP			NP
	cifMPI		NP				3			3
	cif4MPI		NP				NP			NP
	cif16MPI		NP				NP			NP
	maxbitrate		50				50			50
SET OF (EnhancementOptions ^{a)}) =		NP	NP	NP	NP	NP	NP			
bframeEnhancement	sqcifMPI									
	qcifMPI									
	cifMPI									
	cif4MPI									
	cif16MPI									
	maxbitrate									
NP	Non présent									
T	True (Vrai)									
F	False (Faux)									
a)	Les options figurant en dessous de l'option "maxbitrate" de la structure EnhancementOptions ne sont pas représentées.									

VI.2 Exemples de paramétrage de la voie vidéo inverse en mode H.245

Le présent paragraphe donne des exemples de paramétrage des capacités H.263 et des options H.263 (H263Options) pour le fonctionnement de la voie vidéo inverse.

Exemple 1: mode voie logique séparée

Dans ce mode, une voie logique bidirectionnelle supplémentaire est ouverte pour les messages vidéo dans le sens inverse. La dépendance entre une voie vidéo directe et une voie vidéo inverse est décrite par les paramètres **forwardLogicalChannelDependency** et **reverseLogicalChannelDependency** du message **OpenLogicalChannel**.

La voie logique pour les messages vidéo dans le sens inverse ne doit être établie qu'après l'établissement de la voie directe. Si un message OpenLogicalChannel est reçu avec la mention d'une dépendance à l'égard d'une voie non existante, le terminal répond par un message OpenLogicalChannelReject avec le code de motif invalidDependentChannel. Un exemple est donné ci-après:

- 1) une voie logique bidirectionnelle destinée aux données vidéo est ouverte entre le terminal A et le terminal B comme l'indique la Figure VI.1. Le message OpenLogicalChannel relatif à la voie logique bidirectionnelle comprend le paramètre RefPictureSelectionCapability de la séquence H263VideoCapability;

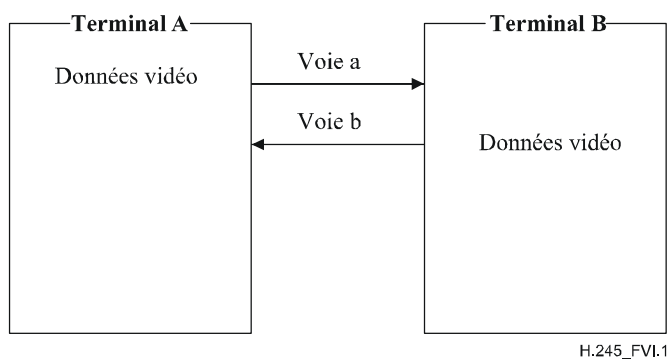


Figure VI.1/H.245 – Voies logiques pour les données vidéo

- 2) une voie logique bidirectionnelle destinée aux messages vidéo dans le sens inverse est ensuite ouverte, comme le montre la Figure VI.2. Dans cet exemple, nous supposons que le terminal A demande l'ouverture d'une voie logique bidirectionnelle. (Si le terminal B demande l'ouverture de la voie, le paramètre forwardLogicalChannelDependency est remplacé par le paramètre reverseLogicalChannelDependency et vice versa.) Le message OpenLogicalChannel correspondant à cette voie logique comprend l'élément forwardLogicalChannelDependency du paramètre forwardLogicalChannelParameters indiquant un numéro de voie logique "a" (Figure VI.2) et l'élément reverseLogicalChannelDependency du paramètre reverseLogicalChannelParameters indiquant un numéro de voie logique "b", ainsi que l'élément separateVideoBackChannel;

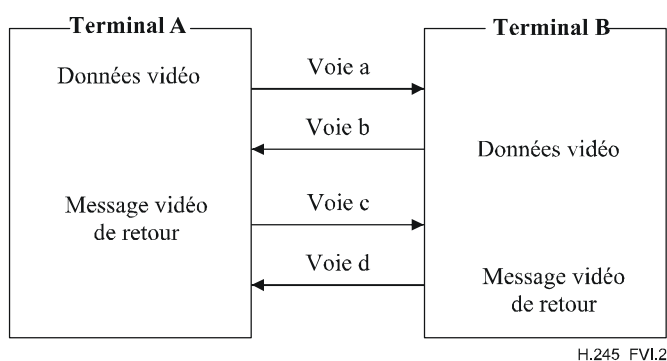


Figure VI.2/H.245 – Voies logiques pour le mode voie logique séparée

- 3) lorsque la voie logique destinée aux messages vidéo dans le sens inverse est établie, le terminal A envoie les données vidéo à la voie logique "a" et reçoit de la voie "d" les messages vidéo dans le sens inverse qui correspondent aux données vidéo envoyées à la voie "a". De même, le terminal A reçoit les données vidéo de la voie "b" et envoie à la voie "c" les messages vidéo dans le sens inverse qui correspondent aux données vidéo du terminal B.

Un exemple de définition des paramètres des capacités H.263 qui figurent dans chaque message OpenLogicalChannel est succinctement indiqué dans le Tableau VI.2. Par souci de simplicité, une partie seulement des capacités H.263 est représentée.

Tableau VI.2/H.245 – Exemples de paramétrage des capacités H263 pour les messages OpenLogicalChannel

Paramètres des capacités H263	Paramétrage des capacités H263 dans le message OpenLogicalChannel		
	Voies a, b	Voies c, d	Voies e, f
sqcifMPI	NP	NP	NP
qcifMPI	3	NP	3
cifMPI	NP	NP	NP
cif4MPI	NP	NP	NP
cif16MPI	NP	NP	NP
maxBitRate	240	10	240
refPictureSelection		NP	
additionalPictureMemory	Non spécifié	–	Non spécifié
videoMuxCapability	"Faux"	–	(doit être) "Vrai"
videoBackChannelSendCapability	ackAndNackMessage	–	AckAndNackMessage
separateVideoBackChannel	"Faux"	"Vrai"	"Faux"
NP non présent			

Exemple 2: mode VideoMux

Lorsqu'un terminal indique le paramètre videoMuxCapability dans la structure RefPictureSelectionCapability pendant un échange de capacités, un autre terminal peut utiliser ce mode pour envoyer des messages sur la voie vidéo inverse. Ces messages étant multiplexés dans le flux de bits vidéo codés, il n'est pas nécessaire pour les terminaux d'établir une voie logique supplémentaire pour les messages vidéo dans le sens inverse. Un exemple est donné ci-après.

- 1) une voie logique bidirectionnelle destinée aux données vidéo est ouverte par le message OpenLogicalChannel, qui comprend le paramètre refPictureSelectionCapability avec la valeur "Vrai" du mode VideoMux dans la structure H263VideoCapability (voir Figure VI.3);
- 2) lorsque la voie logique relative aux données vidéo est établie, le terminal A envoie les données vidéo à la voie logique "e" et reçoit de la voie "f" les messages vidéo dans le sens inverse qui correspondent aux données vidéo envoyées à la voie "e", multiplexées dans les données vidéo provenant du terminal B.

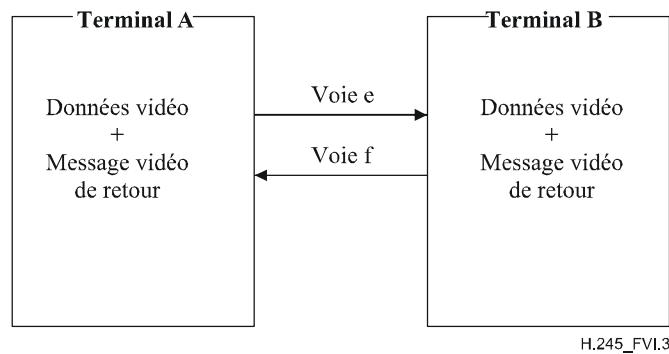


Figure VI.3/H.245 – Voies logiques pour le mode VideoMux

Le Tableau VI.2 résume un exemple de réglage des paramètres de capacité H.263 dans chaque message OpenLogicalChannel.

Exemple 3: mode voie logique séparée en communication vidéo unidirectionnelle

Cet exemple illustre le cas où seul le terminal A envoie des données vidéo, le terminal B envoyant uniquement des messages sur la voie vidéo inverse (voir Figure VI.4). Le paramétrage des voies logiques "g" et "h" est indiqué dans le Tableau VI.3.

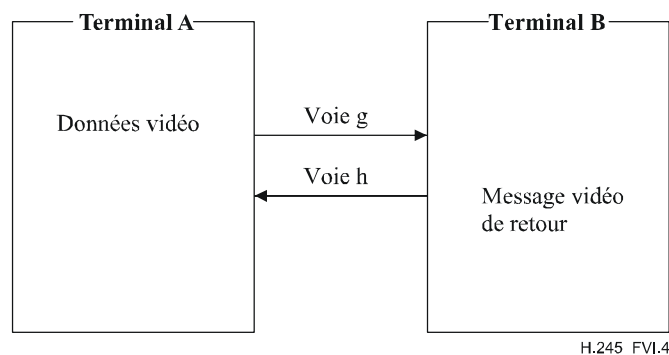


Figure VI.4/H.245 – Mode voie logique séparée en communication vidéo unidirectionnelle

Exemple 4: coexistence du mode voie logique séparée avec le mode VideoMux

L'exemple représenté à la Figure VI.5 illustre le cas où seul le terminal A utilise le mode voie logique séparée pour recevoir les messages vidéo dans le sens inverse du terminal B via la voie logique "l" et où le terminal B utilise le mode VideoMux pour recevoir les messages vidéo dans le sens inverse via la voie logique "i". Il est possible que cet exemple ne soit pas réaliste, mais c'est une configuration qui peut exister. Le paramétrage de chaque voie logique est indiqué au Tableau VI.3.

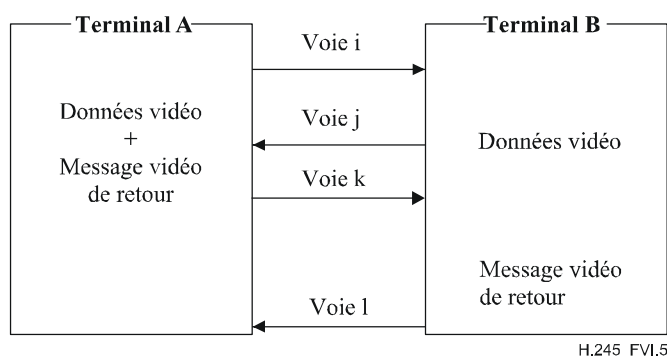


Figure VI.5/H.245 – Coexistence du mode voie logique séparée avec le mode VideoMux

Tableau VI.3/H.245 – Exemple de paramétrage des capacités H263 pour les messages OpenLogicalChannel

Paramétrage des capacités H263 dans le message OpenLogicalChannel						
Paramètre des capacités H263	Voie g	Voie h	Voie i	Voie j	Voie k	Voie l
sqcifMPI	NP	NP	NP	NP	NP	NP
qcifMPI	3	NP	3	3	NP	NP
cifMPI	NP	NP	NP	NP	NP	NP
cif4MPI	NP	NP	NP	NP	NP	NP
cif16MPI	NP	NP	NP	NP	NP	NP
maxBitRate	240	10	240	240	10	10
refPictureSelection		NP			NP	NP
additionalPictureMemory	Non spécifié	–	Non spécifié	Non spécifié	–	–
videoMuxCapability	F	–	F	(devra être) V	–	–
videoBackChannelSendCapability	ackAndNack Message	–	AckAndNack Message	ackAndNack Message	–	–
separateVideoBackChannel	F	V	F	F	F	V
NP non présent V "Vrai" F "Faux"						

Appendice VII

Procédure et gabarit pour la définition de nouvelles capacités à l'aide des capacités génériques H.245

Le présent appendice définit la procédure et un gabarit pour la définition de nouvelles capacités exprimées sous la forme de capacités génériques H.245. Il donne aussi un exemple de la manière dont ce gabarit aurait pu être utilisé pour décrire le codec H.261, qui a été décrit au moyen de la syntaxe ASN.1 dans la présente Recommandation. Ce nouveau mécanisme de définition des capacités dans la présente Recommandation est destiné à être utilisé pour toutes les nouvelles

capacités ajoutées dans la présente Recommandation; il n'est pas prévu de l'utiliser pour redéfinir les capacités existantes.

Les descriptions de capacité relatives aux Recommandations de l'UIT-T seront définies dans des annexes jointes soit à la présente Recommandation ou à la Recommandation elle-même (par exemple la Rec. UIT-T H.283).

D'autres descriptions de capacité pourront être définies dans des annexes à la présente Recommandation ou dans d'autres publications.

Le paramètre *GenericCapabilities*, qui inclut aussi bien des séquences de type *collapsing* que des séquences de type *nonCollapsing*, ne devrait pas inclure de structures de paramètre *GenericParameter* de différents types (*collapsing*, *nonCollapsing*) qui utilisent le même identificateur *parameterIdentifier*.

NOTE 1 – Une telle réutilisation du même identificateur *parameterIdentifier* pourrait provoquer une collision de valeurs d'identificateur *parameterIdentifier* si le paramètre était transféré automatiquement à un système, par exemple à un système H.320, qui ne possède pas le moyen de distinguer entre paramètres de type *collapsing* et paramètres de type *nonCollapsing*.

Le champ normal *parameterIdentifier* d'un paramètre *GenericParameter* ne devrait pas être assigné à la valeur 0.

NOTE 2 – Une telle assignation à la valeur 0 interférerait avec le transfert automatique à la signalisation H.320, par exemple comme cela est effectué dans l'Annexe A/H.239 et dans la Rec. UIT-T H.241.

VII.1 Procédure

VII.1.1 Définition de capacités génériques dans la présente Recommandation

Il convient d'appliquer la procédure ci-dessous si une définition doit être ajoutée aux annexes de la présente Recommandation:

- 1) définir un OBJECT IDENTIFIER (identificateur d'objet) pour cette capacité et le répertorier dans l'Annexe D;
- 2) définir la capacité à l'aide des capacités génériques dans une nouvelle annexe à la présente Recommandation.

Les identificateurs d'objet ont la forme: `{itu-t (0) recommendation (0) h (8) 245 generic-capabilities (1) capability-class capability-name}`.

La *capability-class* (*classe-de-capacité*) est l'une des suivantes: *video(0)*, *audio(1)*, *data(2)*, *control(3)* ou *multiplex(4)*. La valeur du *capability-name* (*nom-de-capacité*) est attribuée dans l'ordre numérique pour chaque *capability-class* (*classe-de-capacité*).

VII.1.2 Définition de capacités génériques dans d'autres Recommandations de l'UIT

Il convient d'appliquer la procédure ci-dessous si la définition doit être ajoutée à d'autres Recommandations de l'UIT:

- 1) définir un identificateur d'objet pour cette capacité dans la Recommandation elle-même et le répertorier dans l'Appendice VIII;
- 2) définir la capacité à l'aide des capacités génériques dans une nouvelle annexe à la Recommandation appropriée.

VII.1.3 Définition de capacités génériques dans des normes non UIT

Il convient d'appliquer la procédure ci-dessous si la définition doit être ajoutée à des normes non UIT:

- 1) définir un identificateur d'objet pour cette capacité dans une norme appropriée et le répertorier dans l'Appendice VIII;

- 2) définir la capacité à l'aide des Capacités génériques dans une norme appropriée.

VII.2 Gabarit

VII.2.1 Identificateur de la capacité

Une seule instance du Tableau VII.1 sera définie pour chaque description de capacité générique.

Tableau VII.1/H.245 – Gabarit de l'identificateur de la capacité

Nom de la capacité	Nom du codec, par exemple H.261.
Classe de la capacité	Classe de la capacité, par exemple vidéo, audio, etc.
Type de l'identificateur de capacité	Type de l'identificateur qui définit le codec: standard, h221NonStandard ou uuid.
Valeur de l'identificateur de capacité	Valeur de l'étiquette du codec, par exemple {itu-t (0) recommendation (0) h (8) 261 generic-capabilities (1) 0}. La valeur du paramètre generic-capabilities identifie un type ou un ensemble de paramètres associés à la capacité. Noter que le format effectif de cet identificateur d'objet relève de la responsabilité des autorités définissant les capacités mais qu'il devrait être défini en envisageant la possibilité d'extensions.
maxBitRate	Indique si le champ maxBitRate sera inclus, ne sera pas inclus ou si l'inclusion de ce champ est facultative.
nonCollapsingRaw	Spécifie le format de la chaîne d'octets et indique si elle sera incluse.
transport	Indique si le champ transport sera inclus, ne sera pas inclus ou si l'inclusion de ce champ est facultative.

VII.2.2 Paramètres de la capacité

Le présent paragraphe est applicable aux paramètres génériques de type collapsing et nonCollapsing. Une instance du Tableau VII.2 sera définie pour chaque paramètre générique. Le gabarit doit être subdivisé en sections afin de faire la distinction entre les paramètres destinés à la négociation des capacités et les paramètres qui sont propres à la signalisation de voie logique.

Tableau VII.2/H.245 – Gabarit d'un paramètre de la capacité

Nom du paramètre	Nom du paramètre, par exemple cifMPI.
Description du paramètre	Nom descriptif du paramètre, par exemple: "spécifie l'intervalle minimal entre images pour la résolution CIF".
Valeur de l'identificateur du paramètre	Entier identifiant ce paramètre "standard".
Statut du paramètre	Indique si le paramètre est obligatoire, obligatoire sous certaines conditions (par exemple en fonction d'un autre paramètre) ou facultatif.
Type du paramètre	Type du paramètre: logical, booleanArray, unsignedMin, unsignedMax, unsigned32Min, unsigned32Max, octetString [ou genericParameter].
Remplace	Les paramètres que ce paramètre remplace. Cet élément du tableau spécifiera zéro, un ou plusieurs paramètres que ce paramètre remplace. Le format sera: "nom-du-paramètre"("valeur-de-l'identificateur-du-paramètre"), par exemple qcifMPI (0).
NOTE – Ce tableau ne permet pas de spécifier le type ParameterTag (standard, h221NonStandard ou uuid) étant donné qu'il n'est à utiliser que pour les descriptions de capacités standards.	

VII.3 Exemple de gabarit – H.261

VII.3.1 Identificateur de la capacité H.261

Tableau VII.3/H.245 – Exemple d'identificateur de la capacité H.261

Nom de la capacité	UIT-T H.261
Classe de la capacité	Codec vidéo
Type de l'identificateur de capacité	Standard
Valeur de l'identificateur de capacité	{itu-t (0) recommendation (0) h (8) 261 generic-capabilities (1) 0}. C'est le premier (et le seul) ensemble de paramètres défini pour la Rec. UIT-T H.261.
maxBitRate	Le champ maxBitRate sera toujours inclus
nonCollapsingRaw	Ce champ ne sera pas inclus
transport	Ce champ ne sera pas inclus

VII.3.2 Paramètres de la capacité H.261

Il est à noter qu'il n'existe pas de tableau pour le champ de débit maximal qui figure dans la syntaxe ASN.1 de la Rec. UIT-T H.245 pour la Rec. UIT-T H.261. La raison en est que le débit maximal est donné au niveau le plus élevé de la structure GenericCapability. Il est aussi à noter que les paramètres temporalSpatialTradeOffCapability et stillImageTransmission auraient pu être combinés pour ne former qu'un seul paramètre générique de type booleanArray.

Tableau VII.4/H.245 – Exemple de paramètre de la capacité H.261 – qcifMPI

Nom du paramètre	qcifMPI
Description du paramètre	Si ce paramètre est présent, il indique l'intervalle minimal entre images (unité: 1/29,97) pour le codage et/ou le décodage d'images QCIF; s'il est absent, aucune capacité n'est indiquée pour les images QCIF.
Valeur de l'identificateur du paramètre	0
Statut du paramètre	Facultatif
Type du paramètre	unsignedMax
Remplace	–

Tableau VII.5/H.245 – Exemple de paramètre de la capacité H.261 – cifMPI

Nom du paramètre	cifMPI
Description du paramètre	Si ce paramètre est présent, il indique l'intervalle minimal entre images (unité: 1/29,97) pour le codage et/ou le décodage d'images CIF; s'il est absent, aucune capacité n'est indiquée pour les images CIF.
Valeur de l'identificateur du paramètre	1
Statut du paramètre	Facultatif
Type du paramètre	unsignedMax
Remplace	qcifMPI (0)

**Tableau VII.6/H.245 – Exemple de paramètre de la capacité H.261 –
temporalSpatialTradeOffCapability**

Nom du paramètre	temporalSpatialTradeOffCapability
Description du paramètre	La présence de ce paramètre indique que le codeur est en mesure de modifier son compromis entre résolution temporelle et résolution spatiale, comme commandé par le terminal distant. Il n'a pas de signification lorsqu'il fait partie d'une capacité de réception.
Valeur de l'identificateur du paramètre	2
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

**Tableau VII.7 /H.245 – Exemple de paramètre de la capacité H.261 –
stillImageTransmission**

Nom du paramètre	stillImageTransmission
Description du paramètre	La présence de ce paramètre indique la capacité de transmission d'images fixes, telle que spécifiée dans l'Annexe D/H.261.
Valeur de l'identificateur du paramètre	3
Statut du paramètre	Facultatif
Type du paramètre	Logique
Remplace	–

Appendice VIII

Liste des capacités et messages génériques définis dans d'autres Recommandations/Normes que la présente Recommandation

Le Tableau VIII.1 énumère les capacités génériques définies dans d'autres Recommandations ou Normes que la présente Recommandation.

**Tableau VIII.1/H.245 – Liste des capacités génériques définies dans d'autres
Recommandations/Normes**

Nom de la capacité	Classe de capacité	Identificateur de capacité	Nom de la Recommandation ou de la Norme qui définit cette capacité
H.283	Protocole de données	{itu-t(0) recommendation(0) h(8) 283 generic-capabilities(1) 0}	Rec. UIT-T H.283
G.722.1	Protocole audio	{itu-t(0) recommendation(0) g(7) 7221 generic-capabilities(1) 0}	Rec. UIT-T G.722.1
H.324	Protocole de données	{itu-t(0) recommendation(0) h(8) 324 generic-capabilities(1) http(0)}	Rec. UIT-T H.324
H.324 – Réinitialisation de session	protocole controle	{itu-t (0) recommendation (0) h (8) 324 generic-capabilities (1) SessionResetCapability (1)}	ITU-T Rec. H.324
H.263	Protocole vidéo	{itu-t(0) recommendation(0) h(8) 263 generic-capabilities(1) 0}	Rec. UIT-T H.263 NOTE – L'emploi de cette capacité pour signaler les "Profils et niveaux" H.263 conformes à l'Annexe X/H.263 devrait toujours être accompagné parallèlement de la signalisation des mêmes modes dans VideoCapability H263. Cela est nécessaire pour assurer que les systèmes qui ne reconnaissent pas les capacités génériques H.263 puissent continuer à fonctionner avec des systèmes plus récents.
H.224	Protocole de données	{itu-t(0) recommendation(0) h(8) 224 generic-capabilities(1) 0}	Rec. UIT-T H.224
G.722.2	Protocole audio	{itu-t (0) recommendation (0) g (7) 7222 generic-capabilities (1) 0}	Rec. UIT-T G.722.2
G.726	Protocole audio	{itu-t (0) recommendation (0) g (7) 726 generic-capabilities (1) version2003 (0)}	Rec. UIT-T G.726
H.241/H.264	Protocole vidéo	{itu-t (0) recommendation (0) h (8) 241 specificVideoCodecCapacités (0) h264 (0) generic-capabilities (1)}	Rec. UIT-T H.241
h239Control Capability	Protocole de commande	{itu-t (0) recommendation (0) h (8) 239 generic-capabilities (1) h239ControlCapability (1)}	Rec. UIT-T H.239
h239ExtendedVideo Capability	Protocole vidéo	{itu-t (0) recommendation (0) h (8) 239 generic-capabilities (1) h239ExtendedVideoCapability (2)}	Rec. UIT-T H.239

**Tableau VIII.1/H.245 – Liste des capacités génériques définies dans d'autres
Recommandations/Normes**

Nom de la capacité	Classe de capacité	Identificateur de capacité	Nom de la Recommandation ou de la Norme qui définit cette capacité
GenericH235security Capability	Protocole sécurité	{itu-t (0) recommendation (0) h (8) 235 version (0) 372}	Rec. UIT-T H.235 Annexe G – Pour MIKEY-PS – (Notes 1 et 2)
GenericH235security Capability	Protocole sécurité	{itu-t (0) recommendation (0) h (8) 235 version (0) 373}	Rec. UIT-T H.235 Annexe G – Pour MIKEY-DHHMAC – (Notes 1 et 2)
GenericH235security Capability	Protocole sécurité	{itu-t (0) recommendation (0) h (8) 235 version (0) 374}	Rec. UIT-T H.235 Annexe G – Pour MIKEY-PK-SIGN – (Notes 1 et 2)
GenericH235security Capability	Protocole sécurité	{itu-t (0) recommendation (0) h (8) 235 version (0) 375}	Rec. UIT-T H.235 Annexe G – Pour MIKEY-DH-SIGN – (Notes 1 et 2)
NOTE 1 – Utilisé à l'intérieur de H235SecurityCapability et dans H235Mode.			
NOTE 2 – Utilisé à l'intérieur de OLC, OLCAck pour les protocoles MIKEY.			

Le Tableau VIII.2 énumère les messages génériques définis dans d'autres Recommandations ou Normes que la présente Recommandation.

**Tableau VIII.2/H.245 – Liste de messages génériques définis dans des
Recommandations/Normes autres que la présente Recommandation**

Nom de message	Classe de message	Identificateur de message	Nom de la Recommandation ou Norme qui définit ce message
H.239	Message générique	{itu-t (0) recommendation (0) h (8) 239 generic-message (2)}	Rec. UIT-T H.239
GenericCommand, GenericIndication	Message générique	{itu-t (0) recommendation (0) h (8) 235 version (0) 372}	Rec. UIT-T H.235 Annexe G (Note)
GenericCommand, GenericIndication	Message générique	{itu-t (0) recommendation (0) h (8) 235 version (0) 373}	Rec. UIT-T H.235 Annexe G (Note)
GenericCommand, GenericIndication	Message générique	{itu-t (0) recommendation (0) h (8) 235 version (0) 374}	Rec. UIT-T H.235 Annexe G (Note)
GenericCommand, GenericIndication	Message générique	{itu-t (0) recommendation (0) h (8) 235 version (0) 375}	Rec. UIT-T H.235 Annexe G (Note)
NOTE – Utilisé pour le recalcul/renouvellement de clés TGK dans les protocoles MIKEY.			

Appendice IX

Utilisation de la syntaxe ASN.1 dans la présente Recommandation

Le présent appendice énumère les concepts ASN.1 qui ont été utilisés dans la présente Recommandation. La Commission d'études 16 se propose de limiter les extensions de la présente Recommandation à la seule utilisation de ces concepts. Les nouveaux concepts ASN.1 ne seront pris en considération que dans des cas exceptionnels.

IX.1 Etiquetage

Toutes les étiquettes mentionnées dans la présente Recommandation sont du type AUTOMATIC TAGS.

IX.2 Types

Les types suivants apparaissent dans les définitions ASN.1 de la présente Recommandation.

BIT STRING	IA5String	OCTET STRING
BMPString	INTEGER	SEQUENCE
BOOLEAN	NULL	SEQUENCE OF
CHOICE	NumericString	SET
GeneralString	OBJECT IDENTIFIER	SET OF

IX.3 Contraintes et gammes

La présente Recommandation utilise des contraintes de taille ("SIZE": chaînes, ensemble de et séquence de), des contraintes de gammes de valeurs (entiers) et des alphabets permis ("FROM").

IX.4 Extensibilité

La présente Recommandation utilise le marqueur d'extension (ellipse "...").

Appendice X

Résolution des scénarios conflictuels dans les voies unidirectionnelles et bidirectionnelles

Le présent appendice énumère les scénarios typiques des conflits qui se produisent en raison de l'utilisation de voies unidirectionnelles et bidirectionnelles. Il décrit également la procédure de résolution de conflit pour chacune de ces voies.

X.1 Les deux terminaux utilisent des messages bidirectionnels d'ouverture OLC

Dans ce scénario, les deux terminaux proposent une voie vidéo bidirectionnelle, comme représenté dans la Figure X.1.

Étant donné que les deux terminaux ne sont capables de traiter qu'un seul flux vidéo dans chaque sens, cela provoque un conflit.

Le terminal maître émet dans ce cas un message de rejet dont le champ de cause a une valeur égale à terminal masterSlaveConflict sur proposition de voie du terminal esclave.

Le terminal esclave devrait accepter la proposition du terminal maître et ne devrait pas essayer d'ouvrir une voie non conflictuelle en compensation.

Le terminal esclave pourrait également détecter les paramètres inverses inappropriés dans la proposition du terminal maître, auquel cas il devrait rejeter la proposition du terminal maître dont le champ de cause a une valeur égale à unsuitableReverseParameters, et renvoyer une proposition avec des paramètres directs et inverses appropriés, comme décrit au § C.5.1.

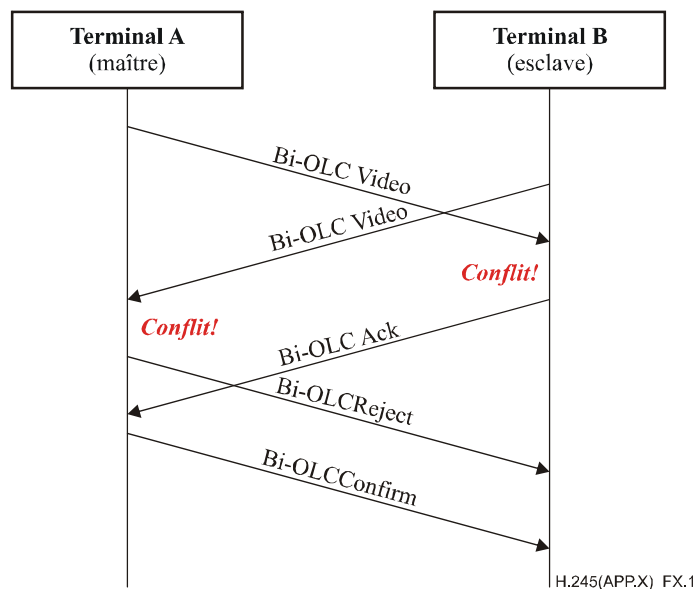


Figure X.1/H.245 – Les deux terminaux utilisent des messages bidirectionnels d'ouverture OLC

X.2 Le terminal maître propose des messages bidirectionnels d'ouverture OLC et le terminal esclave propose des messages unidirectionnels d'ouverture OLC

Dans ce scénario, le terminal maître propose une voie vidéo bidirectionnelle et le terminal esclave propose une voie vidéo unidirectionnelle, comme représenté dans la Figure X.2.

Le terminal maître émet dans ce cas un message de rejet dont le champ de cause a une valeur égale à terminal masterSlaveConflict sur proposition de voie du terminal esclave.

Le terminal esclave devrait accepter la proposition du terminal maître et ne devrait pas essayer d'ouvrir une voie non conflictuelle en compensation.

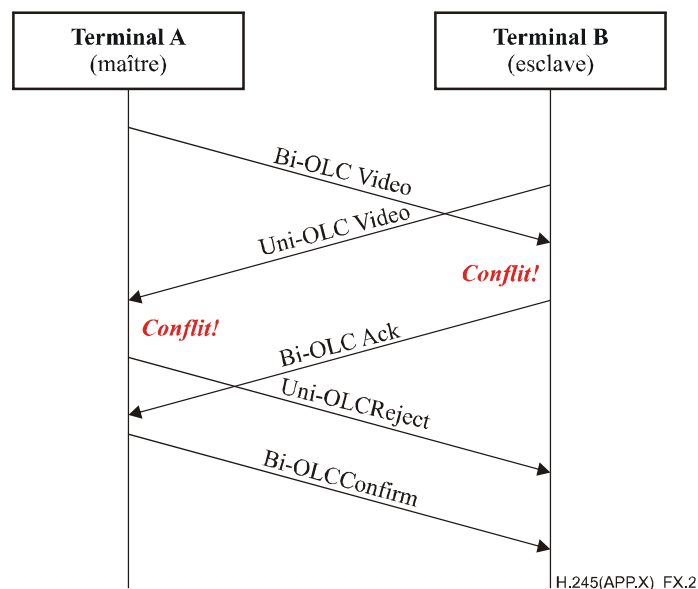


Figure X.2/H.245 – Le terminal maître propose des messages bidirectionnels d'ouverture OLC et le terminal esclave propose des messages unidirectionnels d'ouverture OLC

X.3 Le terminal maître propose des messages unidirectionnels d'ouverture OLC et le terminal esclave propose des messages bidirectionnels d'ouverture OLC

Dans ce scénario, le terminal maître propose une voie vidéo unidirectionnelle et le terminal esclave propose une voie vidéo bidirectionnelle, comme représenté dans la Figure X.3.

Le terminal maître émet dans ce cas un message de rejet dont le champ de cause a une valeur égale à terminal masterSlaveConflict sur proposition de voie du terminal esclave.

Le terminal esclave devrait accepter la proposition du terminal maître et devrait ouvrir une voie vidéo unidirectionnelle non conflictuelle en compensation.

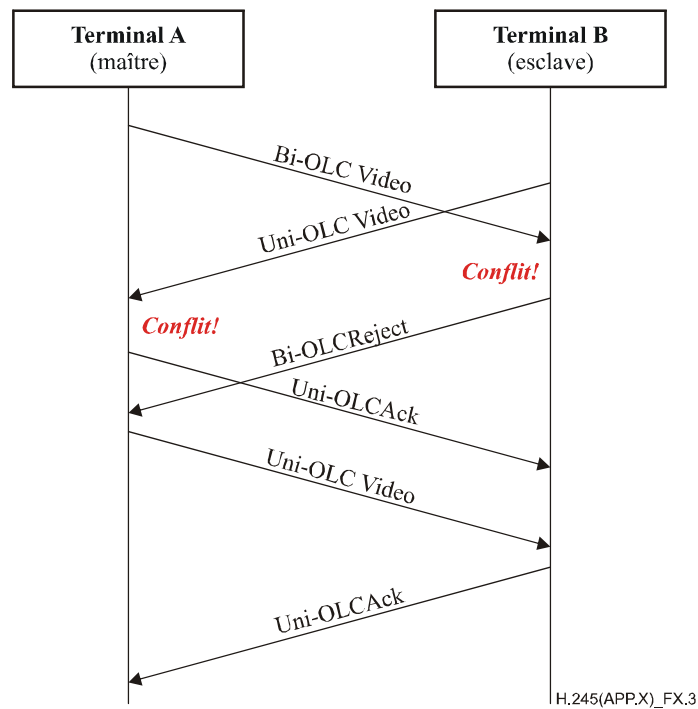


Figure X.3/H.245 – Le terminal maître propose des messages unidirectionnels d'ouverture OLC et le terminal esclave propose des messages bidirectionnels d'ouverture OLC

X.4 Le terminal maître propose des messages bidirectionnels d'ouverture OLC avec type nullData et le terminal esclave propose des messages unidirectionnels d'ouverture OLC

Dans ce scénario, le terminal maître propose une voie vidéo bidirectionnelle avec le type nullData dans les paramètres inverses, et le terminal esclave propose une voie vidéo unidirectionnelle, comme représenté dans la Figure X.4.

Etant donné que le résultat de ces propositions serait une seule voie vidéo dans un seul sens, aucun conflit ne devrait être détecté et chaque terminal devrait accepter la proposition offerte par l'autre terminal.

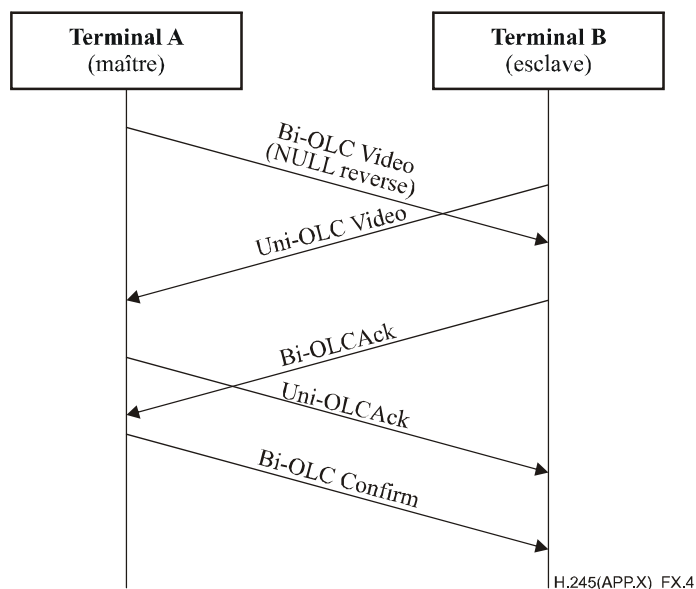


Figure X.4/H.245 – Le terminal maître propose des messages bidirectionnels d'ouverture OLC avec type nullData et le terminal esclave propose des messages unidirectionnels d'ouverture OLC

X.5 Les deux terminaux proposent des messages bidirectionnels d'ouverture OLC avec type nullData

Dans ce scénario, les deux terminaux proposent une voie vidéo bidirectionnelle avec type nullData dans les paramètres inverses, comme représenté dans la Figure X.5.

Étant donné que le résultat de ces propositions serait une seule voie vidéo dans un seul sens, aucun conflit ne devrait être détecté et chaque terminal devrait accepter la proposition offerte par l'autre terminal.

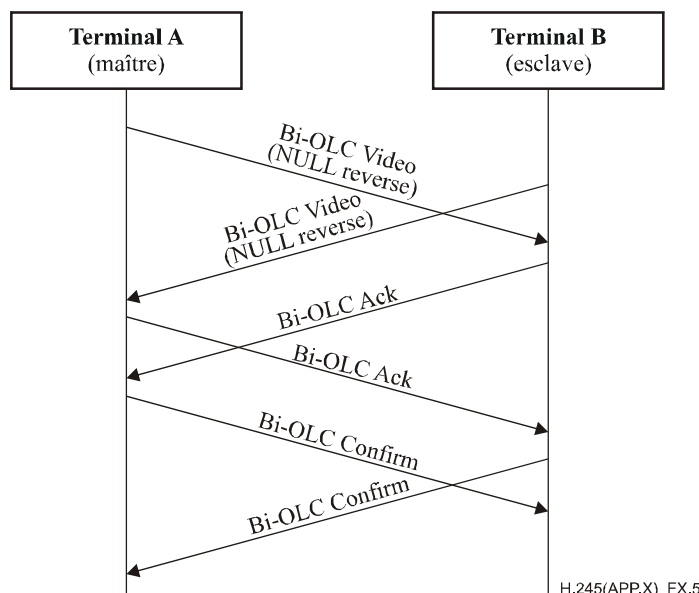


Figure X.5/H.245 – Les deux terminaux proposent des messages bidirectionnels d'ouverture OLC avec type nullData

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
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	Réseaux câblés et 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	Gestion des télécommunications y compris le RGT et maintenance des réseaux
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 de données, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication