



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.225.0

(11/2000)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET
MULTIMÉDIAS

Infrastructure des services audiovisuels – Multiplexage et
synchronisation en transmission

**Protocoles de signalisation d'appel et
paquetisation des flux monomédias pour les
systèmes de communication multimédias
en mode paquet**

Recommandation UIT-T H.225.0

(Antérieurement Recommandation du CCITT)

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.399
SERVICES COMPLÉMENTAIRES EN MULTIMÉDIA	H.450–H.499

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

Recommandation UIT-T H.225.0

Protocoles de signalisation d'appel et paquets des flux monomédias pour les systèmes de communication multimédias en mode paquet

Résumé

La présente Recommandation traite des spécifications techniques relatives aux services visiophoniques à bande étroite définis dans les Recommandations des séries H.200/AV.120, lorsque sur le trajet de transmission se trouvent un ou plusieurs réseaux à commutation par paquets configurés et gérés de manière à offrir une qualité de service (QS) non garantie et non équivalente à celle qui est offerte par le RNIS-BE, de sorte que les mécanismes additionnels de protection et de rétablissement autres que ceux exigés par l'UIT-T H.320 doivent être assurés par les terminaux. On remarque que l'UIT-T H.322 porte sur l'utilisation de certains autres types de réseaux locaux dont la qualité de service sous-jacente qu'ils peuvent offrir n'est pas prise en compte dans les UIT-T H.323 et H.225.0.

La présente Recommandation explique comment gérer les informations audio, vidéo, de données et de commande sur un réseau à commutation par paquets afin d'assurer des services conversationnels pour des équipements de type H.323.

L'Annexe G décrit des méthodes permettant de traiter la résolution d'adresse entre des domaines administratifs de systèmes H.323 en vue de l'établissement d'appels entre domaines. Un domaine administratif se présente vis-à-vis d'autres domaines administratifs sous la forme d'un élément logique appelé "élément frontière".

Les produits revendiquant la conformité à la version 4 de la présente Recommandation doivent en satisfaire toutes les exigences obligatoires. Les produits de la version 4 peuvent être identifiés par des messages H.225.0 contenant une valeur de champ **protocollidentifiant** égale à {itu-t (0) recommandation (0) h (8)2250 version (0) 4}.

Source

La Recommandation H.225.0 de l'UIT-T, révisée par la Commission d'études 16 (2001-2004) de l'UIT-T, a été approuvée le 17 novembre 2000 selon la procédure définie dans la Résolution 1 de l'AMNT.

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.

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 2002

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

TABLE DES MATIÈRES

	Page
1	Domaine d'application 1
2	Références normatives 4
3	Définitions 5
4	Conventions 5
5	Abréviations..... 6
5.1	Abréviations générales..... 6
5.2	Abréviations concernant les messages d'enregistrement, d'admission et d'état..... 7
6	Mécanisme de mise en paquets et de synchronisation..... 8
6.1	Méthode générale..... 8
6.2	Utilisation des protocoles RTP/RTCP 12
6.2.1	Signaux audio 14
6.2.2	Messages vidéo..... 15
6.2.3	Messages de données..... 16
7	Définition des messages H.225.0..... 16
7.1	Utilisation des messages Q.931 16
7.2	Éléments d'information Q.931 communs..... 19
7.2.1	Éléments d'information d'en-tête 19
7.2.2	Éléments d'information propres au message 20
7.3	Informations complémentaires concernant les messages Q.931 29
7.3.1	Alerte (ALERTING)..... 30
7.3.2	Appel en cours (CALL PROCEEDING)..... 31
7.3.3	Connexion (CONNECT)..... 32
7.3.4	Acquittement de connexion (CONNECT ACKNOWLEDGE) 34
7.3.5	Déconnexion (DISCONNECT)..... 34
7.3.6	Information (INFORMATION) 35
7.3.7	Progression (PROGRESS) 35
7.3.8	Libération (RELEASE) 37
7.3.9	Fin de libération (RELEASE COMPLETE) 37
7.3.10	Etablissement (SETUP)..... 38
7.3.11	Acquittement d'établissement (SETUP ACKNOWLEDGE)..... 42
7.3.12	Etat (STATUS)..... 43
7.3.13	Demande d'état (STATUS INQUIRY)..... 44
7.4	Détails des messages Q.932..... 45
7.4.1	Fonctionnalité (FACILITY) 45
7.4.2	Notification (NOTIFY)..... 48
7.4.3	Autres messages 48

	Page	
7.5	Temporisations Q.931.....	49
7.6	Eléments communs des messages H.225.0.....	49
7.7	Prise en charge requise des messages RAS.....	61
7.8	Messages de recherche de terminal et de passerelle.....	63
	7.8.1 Message GRQ (demande de portier).....	63
	7.8.2 Message GCF (confirmation de portier).....	64
	7.8.3 Message GRJ (refus de portier).....	64
7.9	Messages d'enregistrement de terminal et de portier.....	65
	7.9.1 Message de demande d'enregistrement RRQ.....	65
	7.9.2 Message RCF (confirmation d'enregistrement).....	67
	7.9.3 Message RRJ (refus d'enregistrement).....	70
7.10	Messages d'annulation d'enregistrement de terminal/portier.....	70
	7.10.1 Message de demande d'annulation d'enregistrement (URQ).....	70
	7.10.2 Message UCF (confirmation d'annulation d'enregistrement).....	72
	7.10.3 Message URJ (refus d'annulation d'enregistrement).....	72
7.11	Messages d'admission du terminal au portier.....	73
	7.11.1 Message de demande d'admission (ARQ, <i>admission request</i>).....	73
	7.11.2 Message ACF (confirmation d'admission).....	75
	7.11.3 Message ARJ (refus d'admission).....	77
7.12	Demandes de modification de largeur de bande émises par le terminal à l'intention du portier.....	78
	7.12.1 Message BRQ (demande de largeur de bande).....	78
	7.12.2 Message BCF (confirmation de largeur de bande).....	79
	7.12.3 Message BRJ (refus de largeur de bande).....	79
7.13	Messages de demande de localisation.....	80
	7.13.1 Message LRQ (demande de localisation).....	80
	7.13.2 Message LCF (confirmation de localisation).....	81
	7.13.3 Message LRJ (refus de localisation).....	82
7.14	Messages de désengagement.....	83
	7.14.1 Message DRQ (demande de désengagement).....	83
	7.14.2 Message DCF (confirmation de désengagement).....	84
	7.14.3 Message DRJ (refus de désengagement).....	85
7.15	Messages de demande d'état.....	85
	7.15.1 Message IRQ (demande d'information).....	86
	7.15.2 Message IRR (réponse à une demande d'information).....	87
	7.15.3 Message IACK (acquiescement de demande d'information).....	89
	7.15.4 Message INAK (acquiescement négatif de demande d'information).....	89
7.16	Message non normalisé.....	89
7.17	Message incompris.....	90

	Page
7.18	Messages de disponibilité de ressources de la passerelle 90
7.18.1	Message RAI (indication de disponibilité de ressources) 91
7.18.2	Message RAC (confirmation de disponibilité de ressources) 91
7.19	Temporisations RAS et message demande en cours (RIP, <i>request in progress</i>)..... 92
7.20	Messages de commande de service 94
7.20.1	ServiceControlIndication (SCI)..... 94
7.20.2	ServiceControlResponse (SCR) 94
8	Mécanismes permettant de conserver la qualité de service (QS) 95
8.1	Méthode générale et hypothèses 95
8.2	Utilisation du protocole RTCP pour la mesure de la qualité de service 96
8.2.1	Rapports d'expéditeur 96
8.2.2	Rapports du récepteur 96
8.3	Procédures relatives à la gigue audio/vidéo..... 97
8.4	Procédures relatives au décalage audio/vidéo 97
8.5	Procédures permettant de maintenir la qualité de service..... 97
8.6	Limitation de l'écho 98
	Annexe A – Protocoles RTP/RTCP 98
A.1	Introduction..... 99
A.2	Scénarios d'utilisation du protocole RTP..... 100
A.2.1	Audioconférence simple en mode multidiffusion 101
A.2.2	Conférence audio et vidéo 101
A.2.3	Mélangeurs et traducteurs..... 101
A.3	Définitions 102
A.4	Ordre des octets, alignement et format temporel..... 104
A.5	Protocole de transfert de données RTP..... 104
A.5.1	Champs de l'en-tête fixe RTP 104
A.5.2	Sessions RTP avec multiplexage des données 106
A.5.3	Modifications de l'en-tête RTP propres au profil 107
A.6	Protocole de commande RTP (RTCP)..... 108
A.6.1	Format de paquet RTCP 109
A.6.2	Intervalle de transmission RTCP..... 111
A.6.3	Rapports d'émetteur et de récepteur 113
A.6.4	SDES: paquet RTCP de description de source..... 120
A.6.5	BYE: paquet RTCP au revoir 122
A.6.6	APP: paquet RTCP défini par l'application..... 122

	Page	
A.7	Traducteurs et mélangeurs RTP.....	123
	A.7.1 Description générale.....	123
	A.7.2 Traitement RTCP dans les traducteurs.....	125
	A.7.3 Traitement RTCP dans les mélangeurs.....	126
	A.7.4 Mélangeurs en cascade.....	127
A.8	Attribution et utilisation des identificateurs SSRC.....	127
	A.8.1 Probabilité de collision.....	127
	A.8.2 Résolution des collisions et détection des boucles.....	128
A.9	Sécurité.....	131
A.10	Protocole RTP au-dessus des protocoles de réseau et de transport.....	131
A.11	Récapitulatif des constantes protocolaires.....	131
	A.11.1 Types de paquet RTCP.....	132
	A.11.2 Types d'éléments SDES.....	132
A.12	Spécifications de profil et de format de charge utile RTP.....	132
A.13	Algorithmes.....	134
A.14	Bibliographie.....	134
Annexe B – Profil RTP.....		135
B.1	Introduction.....	135
B.2	Formes de paquets RTP et RTCP et comportement des protocoles.....	136
B.3	Types de charge utile.....	136
B.4	Audio.....	137
	B.4.1 Recommandations indépendantes du codage.....	137
	B.4.2 Directives pour les codages audio à échantillonnage.....	138
	B.4.3 Directives pour les codages audio à trame.....	138
	B.4.4 Codages audio.....	139
B.5	Vidéo.....	140
B.6	Définitions des types de charge utile.....	140
B.7	Assignation des accès.....	141
Annexe C – Format de charge utile RTP pour les flux vidéo H.261.....		142
C.1	Introduction.....	142
C.2	Structure du flux de paquets.....	142
	C.2.1 Description générale de l'UIT-T H.261.....	142
	C.2.2 Mise en paquets.....	143
C.3	Spécification du système de mise en paquets.....	143
	C.3.1 Utilisation du protocole RTP.....	143
	C.3.2 Recommandations relatives au fonctionnement des codecs matériels.....	145

	Page
C.3.3	Perte des paquets 146
C.3.4	Utilisation des paquets de commande H.261 spécifiques facultatifs..... 146
C.3.5	Définition des paquets de commande..... 147
C.4	Bibliographie 148
Annexe D – Format de charge utile RTP pour les flux vidéo H.261A..... 148	
D.1	Introduction..... 148
D.2	Mise en paquets RTP H.261A 148
Annexe E – Mise en paquets de données vidéo..... 149	
E.1	H.263..... 150
Annexe F – Mise en paquets audio et en paquets multiplexés 150	
F.1	G.723.1..... 150
F.2	G.728..... 151
F.3	G.729..... 152
F.4	Suppression de silence 155
F.5	Codecs GSM..... 156
F.5.1	Groupage des trames par paquets 156
F.5.2	Références informatives 157
F.6	G.722.1..... 157
F.7	Vocodeur TIA/EIA-136 à codage ACELP 158
F.7.1	Format de trame TIA/EIA-136 à codage ACELP 158
F.7.2	Trames de suppression de silence des vocodeurs TIA/EIA-136 à codage ACELP..... 159
F.7.3	Mise en paquets dans les vocodeurs TIA/EIA-136 à codage ACELP 160
F.7.4	Références normatives des vocodeurs TIA/EIA-136 à codage ACELP 160
F.8	Vocodeur TIA/EIA-136 à codage US1..... 160
F.8.1	Format de trame TIA/EIA-136 à codage US1..... 160
F.8.2	Trames de suppression de silence des vocodeurs TIA/EIA-136 à codage US1 (TX-DTX)..... 160
F.8.3	Mise en paquets dans les vocodeurs TIA/EIA-136 à codage US1 161
F.8.4	Références normatives des vocodeurs TIA/EIA-136 à codage US1 161
F.9	Codec à débit variable amélioré (EVRC) selon la norme IS-127..... 161
F.9.1	Description du codec EVRC IS-127..... 161
F.9.2	Mise en paquets dans les vocodeurs EVRC à codage IS-127 162
F.9.3	Références normatives des vocodeurs EVRC à codage IS-127 163
F.10	Mise en paquets d'unités MUX-PDU à codage H.223 163
F.10.1	Introduction 163
F.10.2	Format de mise en paquets des unités MUX-PDU..... 164

	Page
Annexe G – Communication entre domaines administratifs	164
G.1 Domaine d'application	164
G.2 Définitions	166
G.3 Abréviations.....	166
G.4 Références.....	167
G.5 Modèles système.....	167
G.5.1 Hiérarchique	167
G.5.2 Répartition ou maillage total	168
G.5.3 Résolveur d'adressage.....	168
G.5.4 Point d'agrégation	169
G.5.5 Chevauchement de domaines administratifs	169
G.6 Conventions d'adressage.....	169
G.7 Fonctionnement	170
G.7.1 Canevas et descripteurs d'adresse.....	170
G.7.2 Découverte d'un élément frontière ou d'un ensemble d'éléments frontière ...	172
G.7.3 Procédures de résolution.....	173
G.7.4 Echange d'information d'utilisation.....	174
G.8 Protocole	174
G.8.1 Considérations relatives à la sécurité.....	175
G.8.2 Définitions des messages.....	175
G.9 Exemples de signalisation.....	193
G.9.1 Répartition ou maillage total	193
G.9.2 Résolveur d'adressage.....	197
Annexe H – Syntaxe des messages H.225.0 (ASN.1)	214
Annexe I – Groupage par paquets vidéo H.263+.....	249
Appendice I – Algorithmes RTP/RTCP	249
Appendice II – Profil RTP	250
Appendice III – Mise en paquets H.261	250
Appendice IV – Fonctionnement du mode H.225.0 sur différentes piles protocolaires de réseau en mode paquet.....	250
IV.1 TCP/IP/UDP	250
IV.1.1 Recherche du portier.....	251
IV.1.2 Communications extrémité à extrémité.....	254
IV.2 SPX/IPX.....	254
IV.2.1 Découverte du portier	254
IV.2.2 Communication d'extrémité à extrémité.....	254

	Page
Appendice V – Utilisation de la notation ASN.1 dans la présente Recommandation.....	255
V.1 Balisage.....	255
V.2 Types.....	255
V.3 Contraintes et étendues	255
V.4 Extensibilité	255
Appendice VI – Identificateurs H.225.0 des protocoles de signalisation canalisés en tunnel.	255

Recommandation UIT-T H.225.0

Protocoles de signalisation d'appel et paquets des flux monomédias pour les systèmes de communication multimédias en mode paquet

L'UIT-T,

considérant

que l'UIT-T H.320 est largement adoptée et de plus en plus utilisée dans le monde pour les services de visiophonie et de visioconférence sur des réseaux conformes aux caractéristiques du RNIS spécifiées dans les Recommandations de la série I,

reconnaissant

qu'il doit être souhaitable et avantageux de faire en sorte que les services précités soient acheminés, en totalité ou en partie, sur des réseaux locaux tout en pouvant interfonctionner avec des terminaux de type H.320,

et notant

les caractéristiques et la qualité offerte par les nombreux types de réseaux locaux susceptibles de présenter un intérêt,

recommande

d'utiliser pour ces services, des systèmes et des équipements conformes aux spécifications de l'UIT-T H.322 ou de l'UIT-T H.323.

1 Domaine d'application

La présente Recommandation décrit la façon dont les signaux audio, vidéo, de données et de commande sont associés, codés puis mis en paquets pour être acheminés entre des équipements H.323 sur un réseau à commutation par paquets. Pour cela, une passerelle H.323 est utilisée, cette passerelle pouvant être connectée à des terminaux H.320, H.324 ou H.310/H.321 sur le RNIS-BE, le RTGC ou le RNIS-LB respectivement. L'UIT-T H.323 contient une description des équipements et procédures tandis que la présente Recommandation traite des protocoles et des formats de message. La communication par l'intermédiaire d'une passerelle H.323 vers une passerelle H.322 pour les réseaux locaux avec qualité de service garantie et, par conséquent, vers des extrémités H.322 est également possible.

La présente Recommandation est applicable à des réseaux à commutation par paquets de types différents: IEEE 802.3, à jeton circulant, etc. La présente Recommandation se positionne au-dessus de la couche Transport TCP/IP/UDP, le SPX/IPX, etc. Des profils particuliers pour les suites protocolaires de transport particulières sont présentés dans l'Appendice IV. ***Ainsi, le domaine d'application de la communication H.225.0 se situe entre entités H.323 sur le même réseau à commutation par paquets, utilisant le même protocole de transport.*** Ce réseau à commutation par paquets peut être constitué par un seul segment ou un anneau ou bien logiquement être un réseau de données d'entreprise comprenant plusieurs réseaux à commutation par paquets interconnectés ou reliés pour former un seul réseau interconnecté. Il convient de souligner que le fonctionnement des terminaux H.323 sur l'ensemble du réseau Internet ou même sur plusieurs réseaux à commutation par paquets interconnectés, peut se traduire par des performances médiocres. Les moyens qui permettent d'offrir une certaine qualité de service sur le réseau à commutation par paquets considéré ou sur Internet n'entrent pas dans le domaine d'application de la présente Recommandation. Cependant, la présente Recommandation permet à l'utilisateur d'un équipement H.323 de savoir que les problèmes

de qualité qu'il rencontre tiennent à un encombrement sur le réseau à commutation par paquets et de disposer de procédures permettant d'exécuter des actions correctives. Il convient également de noter que l'utilisation de plusieurs passerelles H.323 connectées sur le RNIS public est une méthode directe qui permet d'améliorer la qualité de service.

L'UIT-T H.323 et la présente Recommandation étendent les conférences de l'UIT-T H.320 et les connexions de l'UIT-T H.221 aux réseaux à commutation par paquets à qualité de service non garantie. En tant que tel, le modèle de conférence principal¹ est applicable à un nombre de participants allant de quelques personnes à plusieurs milliers, comparé à la diffusion à grande échelle avec commande d'admission poussée et gestion étroite de conférence.

La présente Recommandation utilise le protocole de transport en temps réel/protocole de commande de transport en temps réel (RTP/RTCP, *real-time transport protocol/real-time transport control protocol*) pour la mise en paquets et la synchronisation du flux multimédia pour tous les réseaux à commutation par paquets sous-jacents (voir les Annexes A, B et C). Il convient de noter que l'utilisation du protocole RTP/RTCP telle que spécifiée dans la présente Recommandation n'est liée en aucune façon à l'utilisation du protocole TCP/IP/UDP. La présente Recommandation prend pour hypothèse un modèle d'appel dans lequel on utilise la signalisation initiale sur une adresse de transport non RTP pour l'établissement de l'appel et la négociation de capacité (voir l'UIT-T H.323 et l'UIT-T H.245) suivis par l'établissement d'une ou plusieurs connexions RTP/RTCP. La présente Recommandation contient les détails sur l'utilisation des protocoles RTP/RTCP.

Dans l'UIT-T H.221, des signaux audio, vidéo, de données et de commande sont multiplexés en une ou plusieurs connexions RCC physiques synchronisées. Du côté réseau à commutation par paquet d'un appel H.323 ces concepts ne sont pas applicables. Il n'est pas nécessaire d'appliquer à partir du côté RCC le concept H.221 d'un appel à $P \times 64$ kbit/s, par exemple 2×64 kbit/s, 3×64 kbit/s, etc. Ainsi, du côté du réseau à commutation par paquets, par exemple, il n'y a que des appels à connexion unique avec un débit maximal de 128 kbit/s et non pas des appels à débit fixe 2×64 kbit/s. Dans un autre exemple, les appels de réseau à commutation par paquets à connexion unique avec un débit maximal limité à 384 kbit/s sont en interfonctionnement avec un appel de 6×64 kbit/s du réseau à commutation de circuits (RCC)². La raison essentielle de cette méthode est de concentrer la complexité dans la passerelle et non pas dans le terminal et ainsi d'éviter l'intégration dans le réseau à commutation par paquets de fonctions H.320 qui sont étroitement liées au RNIS sauf si cela est nécessaire.

En général, les terminaux H.323 en interfonctionnement via une passerelle H.323 ne connaissent pas directement le débit de transfert H.320; en effet, la passerelle utilise les messages **FlowControlCommand** H.245 pour limiter le débit du média sur chaque canal logique utilisé à celui autorisé par le multiplex H.221. La passerelle peut permettre l'utilisation de débits vidéo côté réseau à commutation par paquets très inférieurs à ceux utilisés côté RCC (ou inversement), pour cela on fait appel à une fonction de réduction du débit et des trames de remplissage H.261. Les détails de ce mode de fonctionnement sortent du domaine d'application de l'UIT-T H.323 et de la

¹ Un modèle de conférence pour diffusion facultative seulement est à l'étude; de par sa nature le modèle de diffusion ne permet pas des admissions strictes ou la gestion de conférence.

² Il convient de noter que les débits vidéo et de données du côté LAN doivent correspondre aux débits vidéo et de données du côté RCC du multiplex H.320. Il n'est pas exigé de correspondance des débits audio et de commande. Autrement dit, on doit normalement s'attendre à ce que du fait de l'utilisation du contrôle de flux H.245, la passerelle LAN/RCC oblige les débits vidéo et de données à être compatibles avec le multiplex RCC H.221. Cependant, comme les signaux audio peuvent être souvent transcodés dans la passerelle, on constatera souvent que les débits audio sur le LAN et sur le RCC ne correspondent pas. On ne devrait pas également s'attendre à ce que le débit H.221 utilisé pour la commande (800 bit/s) corresponde au débit H.245 du côté LAN. Il convient aussi de noter que le débit du LAN peut diminuer le débit vidéo ou de donnée, mais ce débit ne pourra être supérieur au débit qui est appliqué au multiplex côté RCC.

présente Recommandation. Il convient de noter que le terminal H.323 connaît indirectement les débits de transfert H.320 grâce aux champs de débit maximal vidéo H.245 et aux champs de débit maximal H.245, et qu'il ne doit pas émettre à des débits supérieurs à ces débits.

La présente Recommandation est conçue de manière à rendre possible avec une passerelle H.323, l'interopérabilité avec les terminaux H.320 (1990), H.320 (1993) et H.320 (1996). Cependant, certains éléments de la présente Recommandation pourraient faciliter la compatibilité avec les futures versions de l'UIT-T H.320. Il se peut également que la qualité de service côté H.320 dépende des caractéristiques et des capacités de la passerelle H.323 (voir Figure 1).

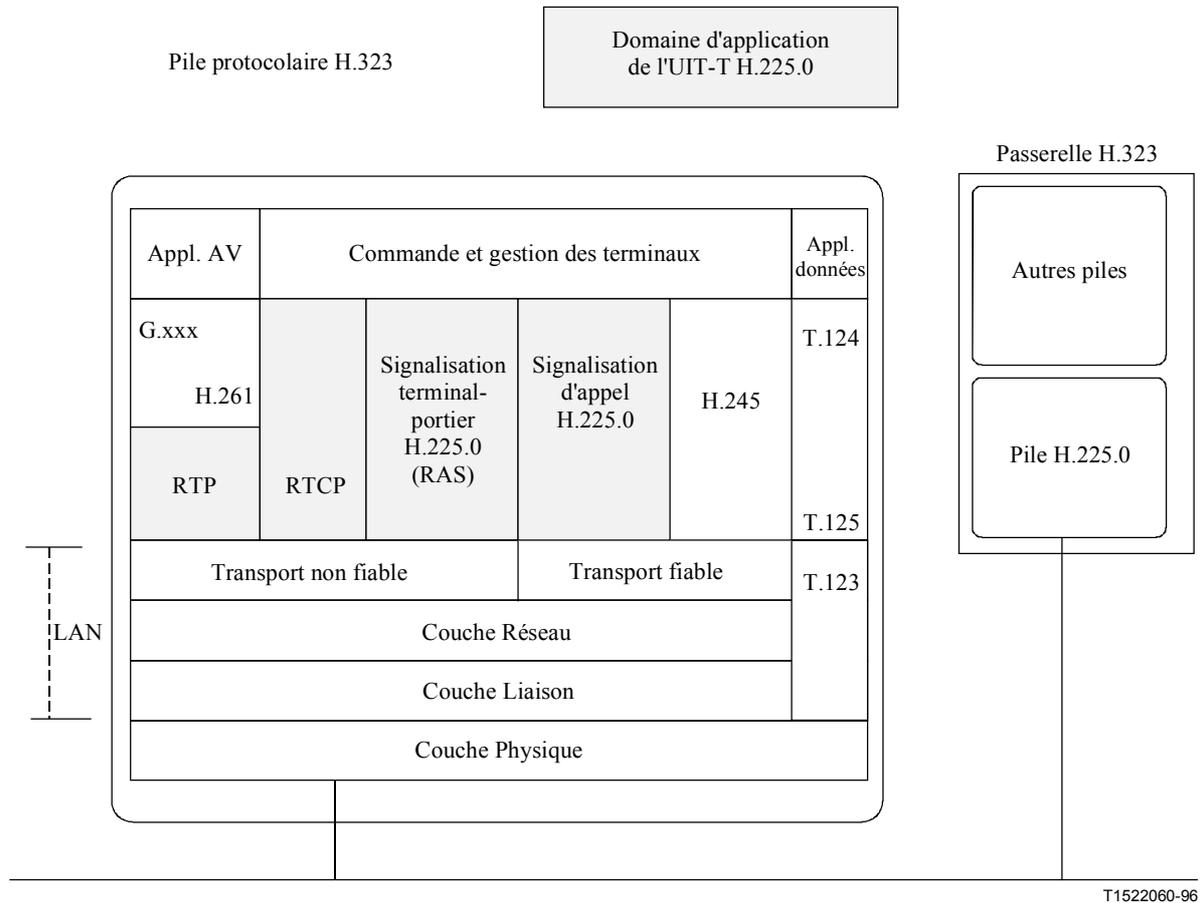


Figure 1/H.225.0 – Domaine d'application H.225.0

D'une manière générale, la présente Recommandation vise à décrire un moyen permettant de synchroniser les paquets qui utilisent les facilités sous-jacentes réseau à commutation par paquets/transport. Elle n'exige pas de combiner tous les médias et toutes les commandes en un seul flux, qui est alors mis en paquets. Les mécanismes de tramage décrits dans l'UIT-T H.221 ne sont pas utilisés pour les raisons suivantes:

- la non-utilisation de ces mécanismes permet l'utilisation de types de traitement des erreurs adaptés à chaque média;
- ces mécanismes sont relativement sensibles à une perte de groupes aléatoires de bits; la mise en paquets offre une plus grande fiabilité dans un environnement de réseau à commutation par paquets;
- les messages H.245 et Q.931 peuvent être envoyés sur les liaisons fiables offertes par le réseau à commutation par paquets;
- la souplesse et la puissance offertes par le protocole H.245 comparativement à celles offertes par le protocole H.242.

2 Références normatives

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

- [1] UIT-T G.711 (1988), *Modulation par impulsions et codage (MIC) des fréquences vocales.*
- [2] UIT-T G.722 (1988), *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s.*
- [3] 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.*
- [4] UIT-T G.723.1 (1996), *Codeur vocal à double débit pour communications multimédias acheminées à 5,3 kbit/s et à 6,3 kbit/s.*
- [5] 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.*
- [6] UIT-T H.221 (1999), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels.*
- [7] UIT-T H.230 (1999), *Signaux de commande et d'indication synchrones de la trame pour les systèmes audiovisuels.*
- [8] UIT-T H.233 (1995), *Système de confidentialité pour les services audiovisuels.*
- [9] UIT-T H.242 (1999), *Procédures pour l'établissement de communications entre terminaux audiovisuels sur des canaux numériques d'un débit allant jusqu'à 2 Mbit/s.*
- [10] 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.*
- [11] UIT-T H.245 (2000), *Protocole de commande pour communications multimédias.*
- [12] UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s.*
- [13] UIT-T H.263 (1998), *Codage vidéo pour communications à faible débit.*
- [14] UIT-T H.320 (1999), *Systèmes et équipements terminaux visiophoniques à bande étroite.*
- [15] UIT-T T.122 (1998), *Service de communication multipoint – Définition du service.*
- [16] UIT-T T.123 (1999), *Piles de protocoles de données propres au réseau pour conférences multimédias.*
- [17] UIT-T T.125 (1998), *Spécification du protocole du service de communication multipoint.*
- [18] UIT-T H.321 (1998), *Adaptation des terminaux visiophoniques H.320 aux environnements RNIS à large bande.*
- [19] UIT-T H.322 (1996), *Systèmes et équipements terminaux visiophoniques pour réseaux locaux offrant une qualité de service garantie.*
- [20] UIT-T H.324 (1998), *Terminal pour communications multimédias à faible débit.*
- [21] UIT-T H.310 (1998), *Systèmes et terminaux de communication audiovisuels à large bande.*
- [22] UIT-T Q.931 (1998), *Spécification de la couche 3 de l'interface utilisateur-réseau RNIS pour la commande de l'appel de base.*

- [23] UIT-T Q.932 (1998), *Système de signalisation d'abonné numérique n° 1 – Procédures génériques pour la commande des services complémentaires RNIS.*
- [24] UIT-T X.680 (1997) | ISO/CEI 88241:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- [25] UIT-T X.681/Amd.1 (1997), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- [26] UIT-T X.691 (1997) | ISO/CEI 8852-2:1998, *Technologies de l'information – Règles de codage ASN.1 – Spécification des règles de codage compact.*
- [27] UIT-T E.164 (1997), *Plan de numérotage des télécommunications publiques internationales.*
- [28] ISO/CEI 10646-1:2000, *Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC) – Partie 1: Architecture et plan multilingue de base.*
- [29] UIT-T Q.850 (1998), *Utilisation des indications de cause et de localisation dans le système de signalisation d'abonné numérique n° 1 et le sous-système utilisateur du RNIS du système de signalisation n° 7.*
- [30] UIT-T Q.950 (2000), *Protocoles pour services complémentaires, structure et principes généraux.*
- [31] UIT-T H.235 (2000), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245).*
- [32] CEI 11571 (1998), *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Réseaux privés à intégration de services – Adressage.*
- [33] IETF RFC 1738 (1994), *Uniform Resource Locators (URL) (Localisateurs de ressource normalisé).*
- [34] IETF RFC 2068 (1997), *Hypertext Transfer Protocol (Protocole de transfert hypertexte) – HTTP/1.1.*
- [35] IETF RFC 1766 (1995), *Tags for the Identification of Languages (Balises pour l'identification des langues).*
- [36] UIT-T H.248 (2000), *Protocole de commande de passerelle.*

3 Définitions

Voir les définitions de l'UIT-T H.323. Selon celle-ci une "extrémité" est un terminal, une passerelle ou un pont de conférence qui a la propriété de pouvoir recevoir et lancer des appels. Dans la présente Recommandation, le terme "terminal" est souvent utilisé dans un sens général dans les descriptions d'établissement d'appel et doit être pris comme désignant un élément qui peut intervenir dans l'établissement d'appel, y compris une passerelle ou un pont de conférence.

4 Conventions

Dans la présente Recommandation le présent du verbe "devoir" correspond à des prescriptions obligatoires tandis que le mode conditionnel de ce verbe correspond à des procédures facultatives. L'auxiliaire "peut" correspond à un processus facultatif sans qu'il y ait de préférences exprimées à ce sujet.

Lorsqu'un terme tel que "pont MCU" est utilisé, il s'agit d'un pont MCU H.323. S'il s'agit d'un pont MCU H.231, cela doit être explicitement mentionné.

Dans la présente Recommandation, le terme "kbit" désigne 1000 éléments binaires. Ainsi le terme "64 kbit/s" désigne exactement 64 000 bit/s.

Sauf indication contraire, la variante "aligned" des règles de codage compact (PER) de l'ASN.1 doit être utilisée pour toutes les déclarations ASN.1 dans la présente Recommandation.

Les noms des messages Q.931 sont en Majuscules; l'ASN.1 est en **gras**.

5 Abréviations

La présente Recommandation utilise les abréviations suivantes:

5.1 Abréviations générales

BAS	signal d'attribution de débit (<i>bit rate allocation signal</i>)
CIF	format intermédiaire commun (<i>common intermediate format</i>)
CRV	valeur de référence d'appel (<i>call reference value</i>)
ECS	signal de commande de chiffrement (<i>encryption control signal</i>)
GOB	groupe de blocs (<i>group of blocks</i>)
H-MLP	protocole multicouche à grande vitesse (<i>high speed multi-layer protocol</i>)
HSD	données à grande vitesse (<i>high speed data</i>)
IA5	alphabet international n° 5 (<i>international alphabet n° 5</i>)
IE	élément d'information (<i>information element</i>)
IETF	groupe de travail d'ingénierie Internet (<i>Internet engineering task force</i>)
IP	protocole Internet (<i>Internet protocol</i>)
LAN	réseau local (<i>local area network</i>)
LD-CELP	prédiction linéaire à faible délai à excitation par code (<i>low delay – Code excited linear prediction</i>)
LSB	bit de plus faible poids (<i>least significant bit</i>)
LSD	données à faible vitesse (<i>low speed data</i>)
MB	macrobloc (voir l'UIT-T H.261)
MBE	extension multioctet (<i>multi-byte extension</i>)
MCC	conférence à commande multipoint (<i>multipoint command conference</i>)
MCN	négation à commande multipoint (<i>multipoint command negating</i>)
MCS	service de communication multipoint (<i>multipoint communication service</i>)
MCS	transmission de données symétriques de commande multipoint (<i>multipoint command symmetrical data transmission</i>)
MCU	pont de conférence; unité de commande multipoint (<i>multipoint control unit</i>)
MF	multitrames (<i>multiframe</i>)
MIC	modulation par impulsions et codage
MLP	protocole multicouche (<i>multi-layer protocol</i>)
MPI	intervalle d'image minimal (<i>minimum picture interval</i>)
MSB	bit de plus fort poids (<i>most significant bit</i>)

NA	non applicable
NS	non normalisé (<i>non-standard</i>)
NSAP	point d'accès au service de réseau (<i>network service access point</i>)
PDU	unité de données protocolaire (<i>protocol data unit</i>)
QCIF	quart de format intermédiaire commun (<i>quarter common intermediate format</i>)
QS	qualité de service
RAS	enregistrement, admission et statut (<i>registration, admission and status</i>)
RCC	réseau à commutation de circuits
RTCP	protocole de commande de transport en temps réel (<i>real-time transport control protocol</i>)
RTP	protocole de transport en temps réel (<i>real-time transport protocol</i>)
SBE	extension à un octet (<i>single byte extension</i>)
SC	canal de service (<i>service channel</i>)
SCM	mode de communication sélectionné (<i>selected communications mode</i>)
TCP	protocole de commande de transport (<i>transport control protocol</i>)
TSAP	point d'accès au service de transport (<i>transport service access point</i>)
UDP	protocole datagramme d'utilisateur (<i>user datagram protocol</i>)
URL	localisateur de ressource normalisé (<i>uniform resource locator</i>)
VCF	commande vidéo "demande d'arrêt sur image" (<i>video command "freeze picture request"</i>)
VCU	commande vidéo "demande d'actualisation rapide" (<i>video command "fast update request"</i>)

5.2 Abréviations concernant les messages d'enregistrement, d'admission et d'état

ACF	confirmation d'admission (<i>admissions confirm</i>)
ARJ	refus d'admission (<i>admissions reject</i>)
ARQ	demande d'admission (<i>admissions request</i>)
BCF	confirmation de largeur de bande (<i>bandwidth confirm</i>)
BRJ	refus de largeur de bande (<i>bandwidth reject</i>)
BRQ	demande de largeur de bande (<i>bandwidth request</i>)
DCF	confirmation de désengagement (<i>disengage confirm</i>)
DRJ	refus de désengagement (<i>disengage reject</i>)
DRQ	demande de désengagement (<i>disengage request</i>)
GCF	confirmation de portier (<i>gatekeeper confirm</i>)
GRJ	refus de portier (<i>gatekeeper reject</i>)
GRQ	demande de portier (<i>gatekeeper request</i>)
IACK	accusé de réception de demande d'information (<i>information request acknowledgement</i>)
INAK	accusé de réception négatif de demande d'information (<i>information request negative acknowledgement</i>)
IRQ	demande d'information (<i>information request</i>)

IRR	réponse à une demande d'information (<i>information request response</i>)
LCF	confirmation de localisation (<i>location confirm</i>)
LRJ	refus de localisation (<i>location reject</i>)
LRQ	demande de localisation (<i>location request</i>)
RAC	confirmation de disponibilité de ressources (<i>resource availability confirmation</i>)
RAI	indication de disponibilité de ressources (<i>resource availability indication</i>)
RCF	confirmation d'enregistrement (<i>registration confirm</i>)
RIP	demande en cours (<i>request in progress</i>)
RRJ	refus d'enregistrement (<i>registration reject</i>)
RRQ	demande d'enregistrement (<i>registration request</i>)
SCI	indication de commande de service (<i>service control indication</i>)
SCR	réponse de commande de service (<i>service control response</i>)
UCF	confirmation de non-enregistrement (<i>unregistration confirm</i>)
URJ	refus d'annulation d'enregistrement (<i>unregistration reject</i>)
URQ	demande d'annulation d'enregistrement (<i>unregistration request</i>)

6 Mécanisme de mise en paquets et de synchronisation

6.1 Méthode générale

Avant le lancement d'un appel, une extrémité peut repérer un portier et s'enregistrer auprès de celui-ci. Si tel est le cas, il est souhaitable pour l'extrémité de connaître le "millésime" du portier auprès duquel il s'enregistre et inversement pour le portier de connaître le "millésime" de l'extrémité qu'il enregistre. Pour ces raisons, les séquences de *repérage* et d'enregistrement contiennent un identificateur d'objet H.245 qui permet de déterminer le "millésime" de la version de l'UIT-T H.323 implémentée. Cette séquence peut aussi contenir des parties de message non standards facultatives pour permettre aux extrémités d'établir des relations non standards. A la fin de cette séquence, les portiers et les extrémités connaissent les numéros des versions et la situation non standard réciproque.

Le numéro de version est obligatoire et l'information non standard est facultative dans la séquence établissement/connexion décrite ci-après pour permettre aux deux points d'extrémité de connaître réciproquement leur millésime et leur situation non standard. Il convient de noter cependant, que tous les messages Q.931 ont un champ pour un message non standard facultatif dans l'élément d'information Utilisateur-utilisateur et que tous les messages de canaux RAS ont un champ facultatif pour l'information non standard. En outre, un message RAS non standard (enregistrement, admission, état) a été défini et peut être envoyé à tout moment.

Le canal non fiable destiné à la messagerie enregistrement, admission et état est appelé le canal RAS. La méthode générale pour lancer un appel consiste à émettre d'abord une demande d'admission obligatoire sur le canal RAS³, puis un message d'établissement initial sur une adresse de transport par canal fiable (cette adresse peut avoir été renvoyée dans le message de confirmation d'admission, ou peut avoir été communiquée au terminal appelant). L'émission de ce message initial est suivie par une séquence d'établissement d'appel Q.931 avec les améliorations décrites ci-après. La séquence se termine lorsque le terminal reçoit dans un message de connexion, une adresse de transport fiable sur laquelle il envoie le message de commande H.245⁴.

Lorsque les messages sont envoyés sur la voie de signalisation d'appel H.225.0 fiable, seul un message complet doit être envoyé dans les frontières définies par le transport fiable; il ne peut y avoir de fragmentation des messages H.225.0 dans plusieurs unités PDU de transport. (Dans les mises en œuvre du protocole IP, dont il est question dans l'Appendice IV, cette unité PDU est définie par TPKT.)

Lorsque le canal de commande H.245 fiable a été établi, des canaux additionnels pour les signaux audio, vidéo et de données peuvent être établis sur la base du résultat de l'échange de capacités en utilisant les procédures de canal logique H.245. La nature de la conférence multimédia côté réseau à commutation par paquets (centralisée ou distribuée/multidiffusion) est négociée pour chaque connexion⁵. Cette négociation est effectuée pour chaque média dans le sens où, par exemple, les signaux audio/vidéo peuvent être décentralisés, alors que les données et les commandes sont centralisées.

Lorsque les messages sont envoyés sur le canal de commande H.245 fiable, plusieurs messages peuvent être envoyés dans les frontières définies par l'unité PDU de transport fiable aussi longtemps que des messages complets sont envoyés; il ne peut y avoir de fragmentation des messages dans plusieurs unités PDU de transport. (Dans les implémentations du protocole IP, dont il est question dans l'Appendice IV, cette unité PDU est définie par TPKT.)

Les terminaux H.225.0 doivent pouvoir envoyer des signaux audio et vidéo en utilisant le protocole RTP sur des canaux non fiables afin de minimiser les délais. La correction des erreurs ou toute action de rétablissement peut être appliquée pour pallier la perte des paquets; en général les paquets audio/vidéo ne sont pas rémis pour ne pas allonger les délais de manière excessive dans un environnement de réseau à commutation par paquets⁶. On suppose que les erreurs sur les bits sont détectées dans les couches inférieures et que les paquets contenant des erreurs ne sont pas envoyés jusqu'au terminal H.225.0. Il convient de noter que les informations audio/vidéo et la signalisation/commande d'appel H.245 ne sont jamais envoyées sur le même canal, et ne partagent pas une structure de message commune. Les terminaux H.225.0 doivent pouvoir envoyer et recevoir des informations audio et vidéo sur des adresses de transport distinctes en utilisant des instances de protocole RTP distinctes pour permettre l'utilisation de numéros de séquence de trame propres aux médias et le traitement distinct de la qualité de service pour chaque média. Cependant, le mode facultatif dans lequel les paquets audio et vidéo sont mélangés en une seule trame, envoyée vers une adresse de transport unique, appelle un complément d'étude.

³ Un terminal qui n'est pas enregistré auprès d'un portier n'est pas tenu d'envoyer une demande d'admission.

⁴ Noter que l'adresse H.245 peut être envoyée dans le message ALERTING ou CALL PROCEEDING pour diminuer le temps d'établissement de l'appel. On notera que le canal H.245 peut être ouvert immédiatement après la réception de l'adresse H.245 dans le message d'établissement.

⁵ Une conférence côté LAN peut être partiellement centralisée et décentralisée selon le choix de la commande multipoint gérant la conférence, ce qu'ignore le terminal. En général, tous les terminaux verront bien évidemment le même mode de communication sélectionné (SCM, *selected communications mode*) (voir l'UIT-T H.243 pour la définition).

⁶ La mise à jour rapide de toutes les trames, de tous les macroblocs ou de tous les groupes de blocs peut être demandée par la signalisation H.245.

Les capacités T.120 sont négociées en utilisant les procédures H.245, et dès réception des messages appropriés, les conférences T.120 sont établies en utilisant les piles transport/réseau à commutation par paquets de l'UIT-T T.123 selon le cas. Les capacités T.120 doivent être acheminées sur le réseau à commutation par paquets entre les extrémités sur une autre adresse de transport. Le Tableau 1 montre le nombre d'identificateurs TSAP utilisés pour chaque média dans un appel point à point. Il est également vrai qu'un terminal H.323 donné peut être en mesure de participer simultanément à plusieurs conférences, ce qui se traduit par l'utilisation d'identificateurs TSAP supplémentaires. Tous les canaux H.245 utilisés sont unidirectionnels sauf ceux qui sont associés au protocole T.120 qui sont bidirectionnels.

Tableau 1/H.225.0 – Identificateurs TSAP utilisés dans le cadre de la présente Recommandation pour chaque appel de unidiffusion point à point

Utilisation d'identificateurs TSAP	Fiable ou non fiable	Connu ou dynamique
Audio/RTP	Non fiable	Dynamique
Audio/RTCP	Non fiable	Dynamique
Vidéo/RTP	Non fiable	Dynamique
Vidéo/RTCP	Non fiable	Dynamique
Signalisation d'appel	Fiable	Connu ou dynamique
H.245	Fiable	Dynamique
Données (T.120)	Fiable	Connu ou dynamique
RAS	Non fiable	Connu ou dynamique
NOTE – Si l'on utilise des identificateurs TSAP connus, il ne peut y avoir seulement qu'une seule extrémité par adresse réseau. Aussi, dans le modèle d'appel direct, l'appelant doit disposer d'un identificateur TSAP connu pour la voie de signalisation d'appel pour pouvoir lancer l'appel.		

Bien que l'adresse de transport pour, par exemple, les informations audio et vidéo puisse partager la même adresse de réseau à commutation par paquets et différer uniquement par l'identificateur TSAP, certains fabricants peuvent choisir d'utiliser différentes adresses de réseau à commutation par paquets pour les données audio et pour les données vidéo. La seule condition à satisfaire est de respecter la convention des Annexes A et B pour la numérotation des identificateurs TSAP dans la session RTP⁷.

Le Tableau 1 décrit le cas élémentaire d'un mode d'exploitation point à point entre deux terminaux. Afin de faciliter la fabrication des passerelles, des unités MCU et des portiers, on peut utiliser des identificateurs TSAP dynamiques au lieu d'identificateurs TSAP connus. Les Tableaux 2 et 3 décrivent un exemple d'utilisation des identificateurs TSAP accés dans le cas d'une passerelle/MCU, et dans le cas d'un portier.

⁷ Il convient de noter que l'on peut utiliser un identificateur TSAP quelconque pour la session RTP, la raison principale de respecter la convention RTP est de permettre l'interopérabilité éventuelle IETF RTP.

**Tableau 2/H.225.0 – Identificateurs TSAP utilisés sur un accès MCU/passerelle
(exemple unidiffusion)**

Utilisation d'identificateurs TSAP	Fiable ou non fiable	Connu ou dynamique
Audio/RTP	Non fiable	Dynamique
Audio/RTCP	Non fiable	Dynamique
Vidéo/RTP	Non fiable	Dynamique
Vidéo/RTCP	Non fiable	Dynamique
Signalisation d'appel	Fiable	Dynamique (Note)
H.245	Fiable	Dynamique
Données (T.120)	Fiable	Dynamique
RAS	Non fiable	Dynamique (Note)
NOTE – Voir la Note 1 du Tableau 3.		

**Tableau 3/H.225.0 – Utilisation d'identificateurs TSAP par le portier H.225.0 pour
chacune des extrémités prenant en charge le modèle d'appel avec intervention
du portier décrit à la Figure 28/H.323 pour un appel point à point**

Utilisation d'identificateurs TSAP	Fiable ou non fiable	Connu ou dynamique	Nombre de canaux
Signalisation d'appel	Fiable	Dynamique ou connu (Note 1)	2 par appel (Note 2)
H.245	Fiable	Dynamique	2 par appel (Note 2)
RAS	Non fiable	Connu	1
NOTE 1 – Si l'on utilise l'identificateur TSAP connu, le portier peut être limité à une seule extrémité par dispositif; par conséquent, il convient d'utiliser des identificateurs TSAP dynamiques.			
NOTE 2 – 0 pour le modèle d'appel direct; 2 pour le modèle d'appel avec intervention d'un portier.			

Il convient de noter qu'une adresse de transport fiable connue est utilisée pour l'établissement d'appel dans le cas d'une communication de terminal à terminal, et également pour le cas d'une communication avec intervention du portier. La connexion sémaphore d'appel fiable doit être maintenue active en respectant les règles suivantes:

- 1) pour une signalisation d'appel de terminal à terminal (voir Figure 26/H.323), chaque terminal peut décider de fermer la voie de signalisation d'appel fiable ou de la laisser ouverte;
- 2) pour le cas de signalisation d'appel avec intervention du portier (voir Figure 25/H.323), les terminaux doivent laisser actif l'accès fiable pendant l'appel. Cependant, le portier peut choisir de fermer ce canal de signalisation, mais il doit maintenir le canal ouvert pour tous les appels qui font intervenir des passerelles. Cela permet une transmission de bout en bout des éléments d'information Q.931 (l'information d'affichage, par exemple);
- 3) si pour certaines raisons, la liaison fiable devient inactive suite à une anomalie au niveau transport ou à un autre problème, la liaison doit être réouverte et l'appel ne doit pas être interrompu. L'état de l'appel et l'utilisation de la valeur CRV (valeur de référence d'appel Q.931) ne sont pas affectés par la fermeture de la liaison fiable sauf si le canal H.245 est fermé, indiquant ainsi la fin de l'appel.

Il convient de noter qu'il est possible d'ouvrir simultanément plusieurs canaux H.245 c'est-à-dire qu'une extrémité peut correspondre à plusieurs appels/conférences simultanément. Il convient de noter également que dans le cadre d'un appel spécifique, un terminal peut avoir plusieurs canaux du même type ouvert, par exemple deux canaux audio pour la stéréophonie. La seule limitation est qu'il doit y avoir un et un seul canal de commande H.245 dans chaque sens par appel point à point.

La signalisation de canal logique H.245 est utilisée pour lancer et arrêter l'utilisation des protocoles vidéo, audio et de données. Ce processus demande la fermeture du canal ouvert et sa réouverture avec un nouveau mode de fonctionnement. Dans le cadre du processus d'ouverture du canal, avant l'envoi de l'acquittement d'ouverture de canal logique, l'extrémité utilise la séquence ARQ/ACF ou BRQ/BCF pour faire en sorte qu'une largeur de bande suffisante soit disponible pour le nouveau canal (à moins qu'une largeur de bande suffisante soit disponible d'une précédente séquence ARQ/ACF ou BRQ/BCF). Dans certains cas, la passerelle peut s'apercevoir que la modification de mode RCC est plus rapide que la modification de mode côté réseau à commutation par paquets, ce qui peut induire une perte de l'information audio. La passerelle peut adopter plusieurs approches au choix du constructeur:

- a) la passerelle peut transcoder l'information audio, dissimulant ainsi les modifications de mode RCC;
- b) la passerelle peut simplement rejeter l'information audio;
- c) la passerelle peut fonctionner comme un pont MCU H.231, prenant le contrôle par rapport à toutes les modifications de mode côté RCC.

Il n'existe pas de règle générale concernant la priorité entre les procédures H.245 et RTP (voir les Annexes A, B et C); chaque conflit et sa résolution est mentionné de manière spécifique dans la présente Recommandation.

Il convient de noter également qu'il n'y a pas d'association fixe entre les sources de synchronisation (SSRC) et les canaux logiques; l'UIT-T H.245 fournit cette association qui peut être utilisée pour la synchronisation audio/vidéo.

En général, deux modes de fonctionnement conférence sont possibles du côté réseau à commutation par paquets: le mode décentralisé et le mode centralisé. Il est également possible de choisir différents modes pour les différents médias, par exemple décentralisé pour l'audio/vidéo et centralisé pour les données. Les procédures permettant de déterminer le type de conférence à mettre en place sont décrites dans l'UIT-T H.323; les messages de la présente Recommandation permettent de prendre en charge toutes les combinaisons autorisées. Il convient de noter aussi que le cas commande et données décentralisées appelle un complément d'étude bien que pris en charge par la signalisation de capacité H.245.

6.2 Utilisation des protocoles RTP/RTCP

L'extrémité H.225.0 doit utiliser des identificateurs TSAP distincts pour l'audio et la vidéo et pour le canal RTCP associé (voir les Annexes A et B). Les extrémités peuvent choisir facultativement d'utiliser des adresses de réseau à commutation par paquets différentes pour l'audio et pour la vidéo, mais pour chaque adresse la convention décrite dans les Annexes A et B doit être respectée pour l'utilisation des identificateurs TSAP. L'utilisation de la signalisation H.245 permettra d'établir d'autres canaux audio et vidéo à condition que le terminal dispose des capacités nécessaires.

La possibilité d'utiliser une adresse de transport unique pour l'audio et pour la vidéo appelle un complément d'étude.

Sauf mention explicite, les réalisations devront correspondre aux réalisations du protocole RTP comme indiqué dans l'Annexe A sauf si le texte de la présente Recommandation apportait des modifications. Les implémentations devront se conformer au profil RTP (voir l'Annexe B) seulement dans les cas explicitement prévus dans la présente Recommandation.

Les traducteurs et les mélangeurs RTP ne sont pas des éléments du système H.323 et tout renseignement les concernant, figurant dans les Annexes A et B, doit être considéré comme étant donné pour information. Il convient de noter que les passerelles et les unités MCU disposent de certaines fonctions de traduction et de mélange et que les informations données dans les Annexes A et B peuvent être utiles pour l'implémentation de passerelles et d'unités MCU. Toutefois, ces unités ne sont pas des mélangeurs et inversement. Il convient aussi de noter que par exemple, dans un appel entre réseaux à commutation par paquets via une passerelle, celle-ci peut se comporter comme un traducteur.

Version (V): la version 2 du protocole RTP doit être utilisée.

Décompte CSRC (CC): l'utilisation du décompte CSRC (source contributive) dans la présente Recommandation est facultative. Lorsqu'elle n'est pas utilisée, la valeur CC doit être zéro (0). Le CSRC peut être utilisé par des unités MCU pour fournir des informations sur les contributeurs à la somme audio en cas de traitement audio décentralisé. Il convient de noter qu'il n'existe pas de fonction permettant de comprendre le décompte CSRC et qu'en conséquence le MCU/MC n'a pas de possibilité de savoir si le terminal de conférence utilise cette information et comment il l'utilise.

CNAME (nom canonique): dans le cas le plus simple d'une connexion point à point sur le réseau à commutation par paquets, le champ SSRC (source de synchronisation) est utilisé pour identifier une source audio/vidéo à partir d'un terminal, et les flux sont associés par un CNAME fourni par la même extrémité comme spécifié dans l'Annexe A.

Lorsque le protocole RTCP est utilisé, les paquets RR ou les paquets SR doivent être envoyés périodiquement comme cela est décrit dans l'Annexe A. Il faut utiliser le message CNAME SDES. D'autres messages SDES (voir l'Annexe A) sont facultatifs, mais ne doivent pas être utilisés pour la direction de la conférence ou pour l'information conférence lorsque les fonctions de commande H.245 ou T.120 sont utilisées. Les informations fournies par l'UIT-T H.245 ou T.120 devront être considérées comme des informations correctes.

Il ne faut pas compter sur le message RTCP BYE pour la fin de session RTP. Le terminal H.323 détecte la déconnexion d'un appel au moyen des procédures UIT-T H.323. L'utilisation du paquet RTCP BYE n'est obligatoire que pour la résolution des collisions SSRC.

Le terminal H.323 utilisé dans une conférence, point à point ou multipoint, doit ramener le débit du canal logique intégré sur une période de temps telle qu'elle est définie dans l'UIT-T H.245, sur celui signalé dans le message **FlowControlCommands** H.245, les commandes de canal logique H.245 et le mécanisme de commande de flux T.120.

Lorsque le terminal H.323 est connecté à une passerelle H.323, celle-ci doit utiliser les moyens offerts par l'UIT-T H.245 et l'UIT-T T.120 pour obliger le terminal H.323 à transmettre à des débits inférieurs ou égaux aux débits média côté RCC et recevoir un débit égal ou supérieur au débit RCC, avec les exceptions suivantes:

- la largeur de bande de commande sur le réseau à commutation par paquets ne doit pas correspondre à celle de l'UIT-T H.221;
- la largeur de bande audio sur le réseau à commutation par paquets peut correspondre à celle de l'UIT-T H.221 sur le RCC, mais avec transcodage de passerelle, une correspondance n'est pas requise;
- dans le cas où la passerelle utilise un réducteur de débit: le terminal H.323 côté réseau à commutation par paquets ne doit pas dépasser le débit H.245 signalé, qui est probablement inférieur au débit émis sur le RCC.

Le chiffrement pour les extrémités H.323 appelle un complément d'étude.

6.2.1 Signaux audio

Avant d'examiner comment la mise en paquets audio est effectuée au moyen du protocole RTP, il faut étudier la façon dont cette opération est signalée via le protocole H.245, et la relation de cette signalisation avec protocole RTP. En général, lorsqu'un canal audio est ouvert, un canal logique H.245 est aussi ouvert. La signalisation H.245 dans la structure **AudioCapability** est donnée en termes de nombre maximal de trames par paquets. Dans la présente Recommandation, la taille de trame varie avec le codage audio utilisé.

Tous les terminaux H.323 assurant la communication audio devront prendre en charge la modulation G.711. Pour tous les codecs audio fonctionnant en mode trame, les récepteurs doivent signaler le nombre maximal de trames audio qu'ils sont capables d'accepter dans un seul paquet audio. Les émetteurs peuvent envoyer un nombre entier quelconque de trames audio dans chaque paquet, jusqu'au maximum indiqué par le récepteur. Les émetteurs ne doivent pas fractionner les trames audio à travers les paquets et doivent envoyer un nombre entier d'octets dans chaque paquet audio.

Les codecs à échantillonnage, par exemple de type G.711 ou G.722, devront être considérés comme étant de type trame avec une taille de trame égale à huit échantillons. (Se reporter à l'Annexe B pour plus de détails concernant les directives pour les codages audio à échantillonnage.) Pour les algorithmes audio, tel celui décrit dans l'UIT-T G.723.1, qui utilisent plusieurs tailles de trame audio, les limites de taille audio dans chaque paquet doivent être signalées au canal audio par signalisation dans la bande.

Pour les algorithmes audio à taille de trame fixe (voir l'UIT-T G.728 et l'UIT-T G.729 pour la taille de trame utilisée par chacun), les limites de trame audio doivent être déduites du rapport taille des paquets/taille de trame audio, en d'autres termes seules les trames audio entières doivent être insérées dans le paquet RTP.

Type de charge utile (PT, *payload type*): seuls les types de charge utile définis par l'UIT-T tels (0)[PCMU], (8)[PCMA], (9)[G722] et (15)[G728] devront être utilisés dans le cas des codecs définis par l'UIT signalés dans l'UIT-T H.245. Les types de charge utile transmis en utilisant la signalisation H.245 devront être utilisés pour tout type de charge utile défini par l'UIT-T et qui n'est pas cité dans l'Annexe B.

En cas d'interruption d'un numéro de séquence, il faudra que le récepteur puisse répéter les derniers sons reçus de sorte que l'amplitude du son répété décroisse jusqu'au silence; d'autres procédures analogues qui peuvent être utilisées sont laissées à la discrétion du fabricant.

Chaque octet G.711 doit être aligné sur un octet de paquet RTP. Le bit de signe de chaque octet G.711 doit correspondre au bit de plus fort poids de l'octet considéré du paquet RTP (c'est-à-dire que dans l'hypothèse où les échantillons G.711 sont manipulés sous forme d'octets dans le serveur, le bit de signe doit être le bit de plus fort poids de l'octet tel que défini par le format du serveur).

Lorsqu'elle envoie un signal MIC à 48/56 kbit/s en direction du réseau à commutation par paquets, la passerelle H.323 doit effectuer un remplissage avec un ou deux bits supplémentaires dans chaque octet conformément à la Note 2 du Tableau 1b/G.711, et utiliser des valeurs RTP pour la MIC-A ou la MIC-U (8 ou 0). Pour la loi μ le remplissage consiste à placer des 1 dans le 7^e et le 8^e bit. Pour la loi A, le 7^e bit doit être à 0 et le 8^e à 1. Dans le sens opposé, la passerelle H.323 tronquera le signal G.711 à 64 kbit/s du côté réseau à commutation par paquets pour adapter le débit G.711 utilisé en H.320. Ainsi, du côté réseau à commutation par paquets on ne doit utiliser que des signaux G.711 à 64 kbit/s.

Lorsqu'elle envoie un signal G.722 à 48/56 kbit/s en direction du réseau à commutation par paquets, la passerelle H.323 doit remplir d'un ou de deux bits supplémentaires chaque octet, et utiliser les types de charge utile RTP dynamique signalés dans l'UIT-T H.245 pour distinguer les signaux à 64 kbit/s (qui utilisent $PT = 9$) des signaux à débit réduit. Dans le sens inverse, la passerelle H.323 tronquera le signal G.722 à 64 kbit/s du côté réseau à commutation par paquets pour que le débit corresponde au débit G.711 utilisé dans H.320. Ainsi, du côté réseau à commutation par paquets seuls des signaux G.722 à 64 kbit/s doivent être utilisés.

Si possible, le terminal H.323 devrait utiliser la fonction de suppression de silence offerte par le protocole RTP, et particulièrement lorsque la conférence est de type multidiffusion. Le terminal H.323 doit être en mesure de recevoir des flux RTP avec compression des silences. Les codeurs peuvent ne pas envoyer de signal audio pendant les périodes de silence après l'envoi d'une unique trame de silence ou peuvent envoyer des trames remplies d'un silence de fond si ces techniques sont spécifiées par la Recommandation en vigueur sur les codecs audio.

6.2.2 Messages vidéo

Type de charge utile (PT): seuls les types de charge utile définis par l'UIT-T tels ceux de l'UIT-T H.261 ou H.263 doivent être utilisés dans le cas des codecs définis par l'UIT signalés dans l'UIT-T H.245. Des types de charge utile dynamiques pourront être utilisés dans le cas de codecs qui peuvent être signalés par l'intermédiaire de l'UIT-T H.245 et pour lesquels les formats de mise en paquets n'ont pas été définis.

Marqueur (M): le bit de marqueur doit être positionné conformément aux procédures décrites dans l'Annexe A sauf dans les cas où il augmenterait le temps de transmission de bout en bout.

Afin de pouvoir se rétablir après la perte de paquets vidéo, les messages H.245 **VideoFastUpdatePicture**, **VideoFastUpdateMB** et **VideoFastUpdateGOB** doivent être pris en charge. L'utilisation des paquets de commande RTCP demande interne (FIR, *full intra request*) [envoyez-moi une trame complète] et acquittement négatif (NACK, *negative acknowledgment*) [envoyez-moi certains paquets] est facultative; elle est signalée dans les capacités H.245.

Il est possible que la méthode 3) de reprise sur erreur décrite au C.3.3 soit inutile si le paquet NACK n'arrive pas en un seul instant de trame.

Un flux H.261 est mis en paquets du côté réseau à commutation par paquets comme indiqué dans l'Annexe C. Aussi longtemps que des paquets RTP suffisamment longs sont disponibles, la fragmentation sur les limites de macroblocs MB par l'émetteur n'est pas nécessaire. Cependant, si le terminal H.323 fragmente les paquets H.261 au niveau RTP, cette fragmentation doit se produire sur les limites des macroblocs. Tous les terminaux H.323 doivent être en mesure de recevoir des paquets de macroblocs fragmentés ainsi que des paquets fragmentés de groupes de blocs ou des paquets comportant un mélange de macroblocs et de groupes de blocs. Il convient de noter que la non-prise en charge de la fragmentation de macroblocs dans l'émetteur peut se traduire par la perte d'un groupe de blocs entier, et peut aussi abaisser le débit de paquets. La taille des paquets utilisés ne doit pas dépasser la taille de l'unité maximale de transfert (MTU, *maximum transfer unit*) sur un réseau à commutation par paquets donné pour maximiser la fiabilité de fonctionnement. Cependant, si le plus petit élément du schéma de codage codé séparément (un macrobloc, par exemple) dépasse la taille de l'unité MTU, il n'est pas tenu de répartir le paquet en plusieurs unités MTU. Les macroblocs ne doivent pas être ventilés à travers les paquets; tous les paquets doivent se terminer sur une limite de groupes de blocs ou de macroblocs. L'émetteur H.323 peut facultativement choisir de compléter un paquet contenant un petit groupe de blocs avec des macroblocs.

Pour éviter que plusieurs images soient corrompues en raison de la perte d'un paquet RTP, le dispositif de mise en paquets RTP situé au niveau d'une extrémité H.323 doit inclure les signaux vidéo d'au plus une image dans chaque paquet RTP.

SBIT est le nombre de bits de plus fort poids qu'il faut ignorer dans le premier octet de données. EBIT est le nombre de bits de plus faible poids qu'il faut ignorer dans le dernier octet de données.

Le dispositif de mise en paquets RTP ne doit pas obliger à un alignement des signaux vidéo en début d'octet dans chaque nouveau paquet RTP. Autrement dit, si $EBIT = n$ dans un paquet RTP, SBIT dans le paquet RTP suivant vaudra $8 - n$, $0 < n < 8$, et si $EBIT = 0$ dans un paquet RTP, SBIT dans le paquet RTP suivant vaudra 0. Cette prescription permet d'éviter un éventuel temps de transmission de bout en bout supplémentaire dû à un décalage de bits. Cette prescription s'appliquera aux frontières d'image.

L'Annexe D spécifie une extension H.323 pour les en-têtes de paquets vidéo qui contiennent un décompte d'octets. L'utilisation de cette extension facultative est décrite dans l'Annexe D.

On trouvera dans l'Appendice IV des conseils propres au réseau à commutation par paquets pour la mise en paquets de signaux vidéo.

6.2.3 Messages de données

Il n'existe pas de messages de données ou de formats de données spéciaux; les protocoles T.120 sont utilisés sur le réseau à commutation par paquets, conformément aux indications de l'UIT-T T.123. Une comparaison entre les conférences de données centralisées ou décentralisées sur le réseau à commutation par paquets est faite dans l'UIT-T H.323, et est négociée via le protocole H.245.

La commande de flux T.120 sur le réseau à commutation par paquets est gérée au moyen des protocoles de réseau à commutation par paquets lorsqu'elle est demandée par les messages H.245 **FlowControlCommand** et les limites **maxBitRate**.

On se reportera à l'UIT-T H.323 pour les procédures utilisées pour connecter une conférence T.120 en cours à une conférence H.323, ou ajouter un appel H.323 à une conférence T.120.

Le protocole H.224 à utiliser sur le réseau à commutation par paquets appelle un complément d'étude.

7 Définition des messages H.225.0

Le présent paragraphe concerne la définition des messages pour l'établissement d'appel, la commande d'appel et les communications entre terminaux, passerelles, portiers et unités MCU.

La définition en ASN.1 de tous les messages H.225.0 est donnée dans l'Annexe H.

7.1 Utilisation des messages Q.931

Les implémentations doivent être conformes à l'UIT-T Q.931, comme cela est spécifié dans la présente Recommandation. Les terminaux peuvent également prendre en charge les messages facultatifs Q.931 et H.450. Les messages contiendront tous les éléments d'information obligatoires et pourront contenir tout élément d'information facultatif défini dans l'UIT-T Q.931, comme cela est décrit dans la présente Recommandation. Il convient de noter que l'extrémité H.225.0 peut, d'après l'UIT-T Q.931, ignorer tous les messages facultatifs qu'il ne prend pas en charge sans gêner l'interopérabilité, mais doit répondre à un message inconnu par un message d'état.

Chaque extrémité H.225.0 doit être en mesure de recevoir et d'identifier un message entrant Q.931 ou H.450. Il doit avoir la capacité de traiter les messages Q.931 obligatoires; il peut avoir la capacité de traiter les messages Q.931 facultatifs. Dans tous les cas, chaque extrémité H.225.0 devra pouvoir ignorer tout message inconnu sans perturber le fonctionnement.

Chaque extrémité H.225.0 doit être en mesure d'interpréter et de produire des éléments d'information rendus obligatoires dans ce qui suit pour les messages respectifs Q.931 et H.450. Il peut aussi interpréter et produire les éléments d'information facultatifs définis ci-dessous. Il peut aussi interpréter tout autre élément d'information du protocole H.450 ou Q.931 ou d'autres protocoles de la série Q. Les extrémités doivent être en mesure d'ignorer les éléments d'information inconnus contenus dans un message Q.931 ou H.450 sans perturber le fonctionnement. Les procédures applicables à la réception d'éléments d'information "nécessaires à la compréhension" non reconnus doivent s'appliquer conformément au 5.8.7.1/Q.931. Les extrémités H.225.0 ne doivent pas envoyer de multiples éléments d'information du même type dans le même message; par exemple, elles ne doivent pas envoyer de multiples éléments d'information Numéro de l'appelant comme décrit dans l'Annexe A/Q.951.

Les éléments d'information doivent être codés conformément à l'UIT-T Q.931, sauf indication contraire dans la présente Recommandation. L'UIT-T Q.931 doit cependant régir l'ordre approprié de tous les éléments d'information contenus dans un message, quel que soit l'ordre des éléments énumérés dans la présente Recommandation.

Les systèmes intermédiaires (passerelles et portiers) doivent se conformer aux règles suivantes en ce qui concerne les messages et les éléments d'information facultatifs Q.931:

- 1) la passerelle devrait et le portier devra retransmettre, après modification convenable, tous les éléments d'information (facultatifs ou obligatoires) associés aux messages Q.931 obligatoires, soit du terminal vers la passerelle/le portier et en sens inverse. Cela inclut des éléments d'information telles les informations d'utilisateur-utilisateur et les informations d'affichage;
- 2) une passerelle devrait retransmettre tous les éléments d'information et messages facultatifs Q.931 ou H.450 dans les deux sens. Si la voie de signalisation d'appel n'est pas conservée par le portier, cette retransmission n'est pas possible;
- 3) aussi longtemps que la voie de signalisation d'appel Q.931 est en fonctionnement, un portier retransmettra tous les éléments d'information et messages facultatifs Q.931 ou H.450 dans les deux sens après modification convenable. Si la voie de signalisation d'appel n'est pas conservée par le portier, cette retransmission n'est pas possible. A noter que le portier peut agir comme un élément de signalisation pouvant offrir des fonctions (fonctions de services complémentaires par exemple) et qu'il peut donc modifier, terminer ou envoyer des messages Q.931.

Les passerelles H.323 peuvent avoir la capacité de convertir des services complémentaires de la série H.450 et des messages H.225.0 en services complémentaires et messages selon l'ISO/CEI 11582, l'ISUP et les autres normes de signalisation du RCC. Les détails font l'objet de l'UIT-T H.246 et de ses annexes.

Les passerelles H.323 peuvent aussi avoir la capacité de transmettre sans modification des messages de signalisation selon l'ISO/CEI 11582, l'ISUP et les autres normes de signalisation du RCC au moyen de la canalisation en tunnel de la signalisation non H.323 dans les signaux H.225.0. Les détails sont dans l'Annexe M/H.323 (voir M.1/H.323, M.2/H.323, etc.).

Dans la présente version de la Recommandation, toutes les références concernent la version de 1998 de l'UIT-T Q.931. Les procédures décrites au 3.1/Q.931 pour l'établissement d'une connexion en mode circuit sont respectées. Cependant, il est rappelé à la personne chargée de l'implémentation que si la signalisation indique un support, il n'existe pas de canaux B réels du RNIS du côté réseau à commutation par paquets. L'aboutissement de l'appel se traduit par la mise en place d'un canal fiable de bout en bout prenant en charge les messages H.245. L'établissement d'un "support" réel est effectué au moyen des procédures H.245. Cependant, l'utilisation du mode Q.931 du côté réseau à commutation par paquets permet l'interfonctionnement avec le mode Q.931 du côté RCC, tout en disposant d'un ensemble éprouvé pour les caractéristiques d'appel générales orientées connexion.

En général, les procédures symétriques décrites dans l'Annexe D/Q.931 sont utilisées. Cela implique que la machine à états Q.931 fonctionne conformément à l'Annexe D/Q.931 sauf que la procédure du D.3/Q.931 (collisions d'appels) ne doit pas être suivie; la reprise à partir de cette situation est laissée à la couche Application.

Les extrémités qui ne prennent pas en charge les jeux de code avec verrouillage Q.931 ignoreront tous les messages Q.931 utilisant ces méthodes.

Le Tableau 4 ci-dessous montre les messages obligatoires et facultatifs pour l'établissement d'appels H.323 et H.225.0 au moyen de la procédure Q.931 sur le réseau à commutation par paquets.

Tableau 4/H.225.0 – Utilisation de messages Q.931/Q.932 par la Recommandation H.225.0

	Emission (M, F, O, CM)^{a)}	Réception et action (M, F, O^{b)}, CM)
Messages d'établissement d'appel		
Alerting	M	M
Call Proceeding	O	CM ^{c)f)}
Connect	M	M
Connect Acknowledge	F	F
Progress	O	CM ^{f)}
Setup	M	M
Setup Acknowledge	O	O
Messages de libération d'appel		
Disconnect	F	F
Release	F	F
Release Complete	M ^{d)}	M
Messages de la phase information de l'appel		
Resume	F	F
Resume Acknowledge	F	F
Resume Reject	F	F
Suspend	F	F
Suspend Acknowledge	F	F
Suspend Reject	F	F
User Information	O	O
Messages divers		
Congestion Control	F	F
Information	O	CM ^{f)}
Notify	O	O

Tableau 4/H.225.0 – Utilisation de messages Q.931/Q.932 par la Recommandation H.225.0

	Emission (M, F, O, CM)^{a)}	Réception et action (M, F, O^{b)}, CM)
Status	M ^{e)}	M
Status Inquiry	O	M
Messages Q.932/H.450		
Facility	M	M
Hold	F	F
Hold Acknowledge	F	F
Hold Reject	F	F
Retrieve	F	F
Retrieve Acknowledge	F	F
Retrieve Reject	F	F
<p>a) M: obligatoire (<i>mandatory</i>), F: interdit (<i>forbidden</i>), O: facultatif (<i>optional</i>), CM: obligatoire conditionnel (<i>conditionally mandatory</i>). Un message est obligatoire conditionnel s'il est obligatoire lorsqu'une option est prise en charge.</p> <p>b) Il convient de noter qu'il ne faut pas envoyer de message Status en réponse à un message classé ici "O"; le récepteur devra simplement ignorer le message s'il ne le prend pas en charge.</p> <p>c) Les terminaux utilisant des passerelles pourront recevoir le message CALL PROCEEDING et y réagir.</p> <p>d) Le message RELEASE COMPLETE est exigé pour toute situation dans laquelle un canal de signalisation d'appel fiable H.225.0 est ouvert. Si ce canal n'est pas ouvert, la fin de la session H.245 peut être utilisée pour terminer la conférence.</p> <p>e) L'extrémité réagira à un message inconnu avec un message STATUS; la réaction à un message STATUS INQUIRY est également obligatoire. Cependant, une extrémité n'est pas tenue d'envoyer un message STATUS INQUIRY. Dans la pratique, l'extrémité doit avoir la capacité de comprendre un message d'état reçu en réaction à un message envoyé qui n'a pas été connu du récepteur.</p> <p>f) Les extrémités qui prennent en charge des éléments de service facultatifs utilisant ces messages (tels que la canalisation en tunnel, les services complémentaires H.450, la canalisation en tunnel des protocoles, ou les éléments de service qui utilisent la structure genericData) doivent traiter ces messages.</p>		

7.2 Eléments d'information Q.931 communs

7.2.1 Eléments d'information d'en-tête

Pour tous les messages Q.931, il y a trois champs communs qui sont obligatoires outre le type de message; ce type est décrit dans le présent sous-paragraphe.

7.2.1.1 Discriminateur de protocole

Tel que défini au 4.2/Q.931.

Sa valeur est 08H – Cette valeur identifie le message comme un message USER-NETWORK Q.931/I.451 (codé conformément à la Figure 4-2/Q.931). Lorsqu'un portier agit comme un réseau pour fournir les services complémentaires, il peut être utile d'utiliser une autre valeur. Ce point appelle un complément d'étude.

7.2.1.2 Référence d'appel

Tel que défini au 4.3/Q.931.

Une longueur de valeur de référence d'appel de deux octets doit être prise en charge par toute extrémité H.323.

La valeur référence d'appel est choisie du côté où l'appel a été déclenché et doit être localement univoque. Pour la communication subséquente, le côté appelant et le côté appelé doivent utiliser cette valeur de référence d'appel dans tous les messages associés de l'appel considéré.

La valeur est codée conformément à la Figure 4-5/Q.931 pour une valeur de référence d'appel à deux octets. L'octet de plus fort poids de la valeur de référence est toujours codé dans l'octet n° 2.

Il convient de noter que la valeur de référence d'appel est seulement univoque sur un tronçon particulier d'un appel, par exemple entre deux terminaux, ou entre un terminal et un portier. Si un terminal donné a deux appels dans une même conférence, chacun de ces appels devra avoir le même identificateur de conférence mais des valeurs de référence d'appel différentes.

Le fanion de référence d'appel doit être choisi conformément aux procédures décrites dans l'UIT-T Q.931.

A noter que les valeurs de référence d'appel transmises dans les messages RAS doivent être conformes à la structure spécifiée dans l'UIT-T Q.931. En particulier, le fanion de référence d'appel doit correspondre au bit de plus fort poids de la valeur de référence d'appel. La valeur de référence d'appel effective ne peut alors être comprise qu'entre 0 et 32 767 inclus.

La référence d'appel globale, qui est représentée sur la Figure 4-5/Q.931 et qui a la valeur numérique 0, est utilisée pour faire référence à tous les appels se trouvant dans la voie de signalisation d'appel ou dans la voie de signalisation RAS.

7.2.1.3 Type de message

Le type de message est codé conformément à la Figure 4-6/Q.931 en utilisant les valeurs spécifiées au Tableau 4-2/Q.931. Des extensions propres à la présente Recommandation appellent un complément d'étude.

7.2.2 Eléments d'information propres au message

Les règles de codage générales pour les éléments d'information suivants sont définies au 4.5.1/Q.931 et au Tableau 4-3/Q.931. Ces règles doivent être respectées. Le mécanisme d'échappement (Figure 4-8/Q.931) est facultatif.

7.2.2.1 Capacité support

Cet élément d'information est codé conformément à la Figure 4-11/Q.931 et au Tableau 4-6/Q.931. Lorsque cet élément d'information est reçu dans un appel entre réseaux à commutation par paquets, il peut être ignoré par le récepteur. Lorsque cet élément d'information figure dans un message SETUP pour une connexion sémaphore indépendante de l'appel telle que définie dans l'UIT-T H.450.1, le codage doit être conforme au 7.2.2.1.2. Dans tous les autres cas, le codage doit être conforme au 7.2.2.1.1. Les références de numéro d'octet renvoient à la Figure 4-11/Q.931).

7.2.2.1.1 Codage par défaut de la capacité support

Les entités H.323 doivent coder comme suit l'élément d'information Capacité support, sauf indication contraire dans des paragraphes subséquents.

Bit d'extension pour l'octet n° 3 (bit 8)

- Doit être mis à "1".

Norme de codage (octet n° 3, bits 6 et 7)

- Doit être mis à "00", indiquant "UIT-T".

Capacité de transfert d'informations (octet n° 3, bits 1 à 5)

- Pour les appels issus d'une extrémité RNIS, les informations indiquées à la passerelle doivent être réexpédiées.

NOTE – Cela vise à permettre de réexpédier vers l'extrémité H.323 certaines informations anticipées sur la nature de la connexion, par exemple voix seulement ou données ou vidéo; cela aura une incidence sur la largeur de bande requise ainsi que sur la capacité/disposition à accepter l'appel ou non.

- Les appels issus d'une extrémité H.323 doivent utiliser ce champ pour indiquer leur intention d'établir une communication audiovisuelle. Ce champ doit donc être mis à la valeur "informations numériques sans restriction", c'est-à-dire "01000" ou "informations numériques avec restriction", c'est-à-dire "01001". Si un appel uniquement vocal doit être établi, le terminal H.323 doit mettre le champ de capacité de transfert d'informations à la valeur "parole" (soit "00000") ou "audio à 3,1 kHz" (soit "10000").

Bit d'extension pour l'octet n° 4 (bit 8)

- Doit être mis à "0" si le débit de transfert d'informations est mis à "multidébit"; sinon, doit être mis à "1".

Mode de transfert (octet n° 4, bits 6 et 7)

- Doit spécifier "mode circuit", valeur "00".

Débit de transfert d'informations (octet n° 4, bits 1 à 5)

- Doit être codé conformément au Tableau 4-6/Q.931, sauf que la valeur "00000" (pour le mode paquet) n'est pas autorisée si la passerelle n'est pas connectée à un réseau en mode paquet.

Multiplicateur de débit (octet n° 4.1)

- Doit être présent si le débit de transfert d'informations est mis à "multidébit".
- Le bit d'extension (bit 8) doit être mis à "1".
- Les bits 1 à 7 doivent indiquer la largeur de bande nécessaire pour l'appel, comme défini ci-dessous (noter que, contrairement à l'UIT-T Q.931, une valeur de "00000001" est autorisée ici).
- Dans le cas d'un appel issu d'une extrémité RNIS, la passerelle doit simplement faire suivre les informations qu'elle reçoit du RNIS.
- Dans le cas d'un appel entrant en provenance d'une extrémité H.324, la passerelle doit mettre le multiplicateur de débit à la valeur "01H".
- Dans le cas d'un appel entrant en provenance du RNIS-LB, il est nécessaire d'effectuer une certaine conversion de l'UIT-T Q.2391 à l'UIT-T Q.931. Ce point fera l'objet d'un complément d'étude.
- Dans le cas d'un appel issu d'une extrémité H.323, ce champ doit être utilisé pour indiquer la largeur de bande à utiliser pour cet appel. Si le système appelé est une autre extrémité H.323, cette valeur peut refléter la largeur de bande à utiliser dans le réseau en mode paquet mais le terminal récepteur n'est pas tenu de suivre ces informations. Si une passerelle est impliquée, cette valeur doit refléter le nombre de connexions externes à établir. La largeur de bande nécessaire pour l'appel est celle qui est nécessaire du côté RCC: elle peut concorder ou ne pas concorder avec la largeur de bande autorisée par les messages ACF/BCF dans le réseau en mode paquet.

Protocole de couche 1 (octet n° 5)

- Le bit d'extension (bit 8) doit être mis à "1".
- Les bits 6 et 7 doivent indiquer l'identificateur de couche 1, c'est-à-dire "01".
- Les bits 1 à 5 doivent indiquer le protocole de couche 1.
- Les valeurs autorisées sont G.711 (loi A "00011" et loi μ "00010") pour indiquer un appel uniquement vocal et H.221 ou H.242 ("00101") pour indiquer un appel vidéophonique H.323.

Les octets n° 5a, 5b, 5c, 5d, 6 et 7 ne doivent pas être présents.

7.2.2.1.2 Codage de la capacité support pour connexions sémaphores H.450.1 indépendantes de l'appel

Les entités H.323 doivent coder comme suit l'élément d'information Capacité support dans les connexions sémaphores indépendantes de l'appel comme défini dans l'UIT-T H.450.1.

Bit d'extension pour l'octet n° 3 (bit 8)

- Doit être mis à "1".

Norme de codage (octet n° 3, bits 6 et 7)

- Doit être mis à "01" pour indiquer "Autre norme internationale". Noter que, lorsque cette norme de codage est indiquée, le codage défini dans l'UIT-T Q.931 doit s'appliquer aux octets 1 et 2 et au bits 8 des octets 3 et 4. La capacité de transfert d'informations, le mode de transfert et le débit de transfert d'informations doivent être codés comme indiqué et aucun autre octet ne doit être inclus.

Capacité de transfert d'informations (octet n° 3, bits 1 à 5)

- Doit être mis à "01000" pour indiquer "Informations numériques sans restriction".

Bit d'extension pour l'octet n° 4 (bit 8)

- Doit être mis à "1".

Mode de transfert (octet n° 4, bits 6 et 7)

- Doit être mis à "00" pour indiquer "Connexion sémaphore indépendante de l'appel".

Débit de transfert d'informations (octet n° 4, bits 1 à 5)

- Doit être mis à "00000" pour indiquer "connexion sémaphore indépendante de l'appel".

Les octets 4.1 et au-delà ne doivent pas être inclus.

7.2.2.2 Identité de l'appel

L'utilisation éventuelle de l'élément d'information "identité de l'appel" nécessite un complément d'étude. Cette étude tiendra compte de la numérotation en plusieurs étapes de type terminal-à-portier-à-terminal et terminal-à-passerelle-à-terminal et du routage à source indéterminée.

7.2.2.3 Etat d'appel

Cet élément d'information est codé conformément à la Figure 4-13/Q.931.

Octet n° 3 norme de codage (bits 8-7)

- Mis à "00" pour le codage normalisé de l'UIT-T.

Valeur d'état d'appel (octet n° 3, bits 1-6)

- Codée conformément au Tableau 4-8/Q.931, mais n'utilise pas la valeur d'état de l'interface globale. Ces valeurs sont interprétées comme état d'utilisateur lorsqu'on utilise l'Annexe D/Q.931. Il convient de noter que la plupart des codes indiqués ne doivent pas être produits par un terminal H.323.

7.2.2.4 Numéro de l'appelé

Cet élément d'information est codé conformément à la Figure 4-14/Q.931 et au Tableau 4-9/Q.931.

Octet n° 3 extension (bit 8)

- Mis à "1".

Type de numéro (octet n° 3, bits 5-7)

- Codé selon les valeurs et les règles spécifiées dans le Tableau 4-9/Q.931.

Identification du plan de numérotage (octet n° 3, bits 1-4)

- Codé selon les valeurs et les règles spécifiées dans le Tableau 4-9/Q.931. Un numéro sous forme de chaîne de chiffres composés manuellement devrait être codé par "0000" (inconnu). Lorsque le code est "1001" (plan de numérotage privé) dans un appel au départ d'un réseau à commutation par paquets, cela indique que:
 - 1) la chaîne de chiffres composés manuellement n'est pas présente dans le message SETUP;
 - 2) l'appel doit être routé via une adresse pseudonyme dans l'information utilisateur-utilisateur.

Type de numéro (octet n° 3, bits 5-7)

- Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-9/Q.931. Un numéro dont l'identification dans le plan de numérotage est codée "0000" (inconnu) doit être codé "000" (inconnu). Un numéro dont l'identification dans le plan de numérotage est codée "0001" (plan de numérotage téléphonique/RNIS selon l'UIT-T E.164) et dont le type est codé "000" (inconnu) peut être utilisé pour la compatibilité amont.

Chiffres du numéro

- Numéro composé de caractères IA5 selon les formats spécifiés dans le plan de numérotage/de numérotation.

NOTE – Un numéro E.164 ne doit comporter que des caractères IA5: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" et "0".

7.2.2.5 Sous-adresse de l'appelé

A utiliser comme indiqué dans l'UIT-T Q.931.

7.2.2.6 Numéro de l'appelant

Cet élément d'information est codé conformément à la Figure 4-16/Q.931 et au Tableau 4-11/Q.931.

Type de numéro (octet n° 3, bits 5-7)

- Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-11/Q.931. Un numéro dont l'identification dans le plan de numérotage est codée "0000" (inconnu) doit être codé "000" (inconnu). Un numéro dont l'identification dans le plan de numérotage est codée "0001" (plan de numérotage téléphonique/RNIS selon l'UIT-T E.164) et dont le type est codé "000" (inconnu) peut être utilisé pour la compatibilité amont.

Identificateur du plan de numérotage (octet n° 3, bits 1-4)

- Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-11/Q.931. Un numéro se présentant sous la forme d'une chaîne de chiffres composés manuellement devrait être codé "0000" (inconnu). Si sa valeur est "1001" (plan de numérotage privé) dans un appel au départ d'un réseau à commutation par paquets, cela indique que:
 - 1) la chaîne de chiffres composés manuellement n'est pas présente dans le message SETUP;
 - 2) que l'appel doit être acheminé via une adresse pseudonyme dans les informations d'utilisateur-utilisateur.

Octet n° 3a

- Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-11/Q.931.

Chiffres du "numéro"

- Numéro composé de caractères IA5, selon les formats spécifiés dans le plan de numérotage/de numérotation.

NOTE – Un numéro E.164 ne doit comporter que des caractères IA5: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" et "0".

Les extrémités H.323 ne doivent pas envoyer de multiples éléments d'information "Numéro de l'appelant" dans le même message. Les passerelles peuvent fournir un appui pour l'interfonctionnement avec des messages SETUP Q.931 contenant de multiples éléments d'information "Numéro de l'appelant". Les passerelles qui fournissent cet appui doivent insérer le premier élément d'information "Numéro de l'appelant" Q.931 dans l'élément d'information "Numéro de l'appelant" du message SETUP H.225.0 puis insérer les éléments d'information "Numéro de l'appelant" Q.931 subséquents dans le champ **additionalSourceAddresses** du message SETUP H.225.0.

7.2.2.7 Sous-adresse de l'appelant

Conformément à l'UIT-T Q.931.

7.2.2.8 Cause

Lorsque cet élément d'information est reçu, les règles définies dans l'UIT-T Q.850 sont applicables. Il convient de noter que l'un ou l'autre de l'élément d'information cause et de l'élément **ReleaseCompleteReason** est obligatoire pour le message RELEASE COMPLETE; l'élément d'information "Cause" est facultatif partout ailleurs. L'élément d'information "Cause" et l'élément **ReleaseCompleteReason** (partie du message Release Complete) s'excluent mutuellement. Les passerelles doivent mapper un élément **ReleaseCompleteReason** vers l'élément d'information cause lorsqu'elles envoient un message RELEASE COMPLETE au côté à commutation de circuit depuis le côté réseau à commutation par paquets (voir Tableau 5). (Le mappage inverse n'est pas nécessaire car les entités du réseau à commutation par paquets sont tenues de décoder l'élément d'information cause.)

Tableau 5/H.225.0 – Mappage entre l'élément ReleaseCompleteReason et l'élément d'information "cause"

Code de l'élément ReleaseCompleteReason	Valeur de cause Q.931/Q.850 correspondante
noBandwidth	34 – Pas de circuit/canal disponible
gatekeeperResources	47 – Ressources non disponibles, non spécifiées
unreachableDestination	3 – Pas d'acheminement vers la destination
destinationRejection	16 – Libération normale de l'appel
invalidRevision	88 – Destination incompatible
noPermission	111 – Erreur de protocole, non spécifiée
unreachableGatekeeper	38 – Réseau en dérangement
gatewayResources	42 – Encombrement de l'équipement de commutation
badFormatAddress	28 – Format de numéro non valide
adaptiveBusy	41 – Dérangement temporaire
inConf	17 – Usager occupé
undefinedReason	31 – Normal, non spécifié
facilityCallDeflection	16 – Libération normale de l'appel
securityDenied	31 – Normal, non spécifié
calledPartyNotRegistered	20 – Abonné absent
callerNotRegistered	31 – Normal, non spécifié
newConnectionNeeded	47 – Ressources non disponibles, non spécifiées
nonStandardReason	127 – Interfonctionnement, non spécifié
replaceWithConferenceInvite	31 – Normal, non spécifié
genericDataReason	31 – Normal, non spécifié
neededFeatureNotSupported	31 – Normal, non spécifié
tunnelledSignallingRejected	127 – Interfonctionnement, non spécifié

7.2.2.9 Identification de canal

Appelle un complément d'étude; peut être utilisé pour avoir des rétroactions sur des tentatives d'appel multiples.

7.2.2.10 Numéro connecté

Codé conformément au 5.4.1/Q.951.

7.2.2.11 Sous-adresse du numéro connecté

Codé conformément au 5.4.2/Q.951.

7.2.2.12 Niveau d'encombrement

Ne doit pas être utilisé.

7.2.2.13 Date/heure

Codé conformément à la Figure 4-21/Q.931.

7.2.2.14 Affichage

Codé conformément à la Figure 4-22/Q.931. La longueur maximale de cet élément d'information est de 82 octets.

7.2.2.15 Élément d'information "Fonctionnalité étendue"

Lorsque cet élément d'information est utilisé pour indiquer une sémantique non modifiée telle que définie dans les Recommandations de la série Q.95x, il doit être codé conformément au 8.2.4/Q.932. Dans ce cas, les unités PDU de service doivent être constituées en fonction de l'élément ROSE [qui utilise l'UIT-T X.208 (Spécification de la syntaxe abstraite numéro un (ASN.1)) et l'UIT-T X.209 (Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1))] comme cela est défini dans l'UIT-T X.229.

7.2.2.16 Fonctionnalité

Pour signaler un réacheminement d'appel propre aux procédures H.323 (renvoyer un appel, réacheminer un appel vers le contrôleur multipoint ou imposer qu'un appel soit routé au portier) ou en cas de signalisation d'un service complémentaire conformément à l'UIT-T H.450, l'élément d'information Facility-UUIE est utilisé. Ce cas particulier doit être indiqué par le codage d'un élément d'information Fonctionnalité de longueur zéro, c'est-à-dire que l'élément d'information Fonctionnalité doit comporter exactement 2 octets définis comme suit:

- l'octet n° 1 (identificateur de l'élément d'information) doit être mis à "00011100" ("1C"H) pour indiquer qu'il s'agit de l'élément d'information Fonctionnalité;
- l'octet n° 2 (longueur de l'élément d'information) doit être mis à "0" pour indiquer que cet élément d'information ne comprend pas d'autre octet.

Pour indiquer un renvoi d'appel, l'élément d'information Fonctionnalité doit être vide et l'élément d'information **Facility-UUIE** doit indiquer dans le champ **alternativeAddress** ou **alternativeAliasAddress** le terminal auquel l'appel doit être réacheminé. Dans ce cas, le champ **facilityReason** doit être mis à **callForwarded**.

Dans le cas où une extrémité est amenée à appeler une autre extrémité car l'extrémité appelante souhaite participer à une conférence et l'extrémité appelée n'incorpore pas le contrôleur multipoint, l'élément d'information Fonctionnalité doit aussi être vide. L'élément **conferenceID** devra indiquer la conférence à laquelle l'extrémité souhaite participer et la cause figurant dans l'élément **Facility-UUIE** doit être **routeCallToMC**.

De même, dans le cas où l'extrémité appelante est amenée à signaler l'extrémité appelée au portier de ce dernier, l'élément d'information Fonctionnalité est vide. Le champ **conferenceID** de l'élément d'information **Facility-UUIE** devra indiquer la conférence à laquelle l'extrémité souhaite participer et la cause figurant dans l'élément **Facility-UUIE** doit être **routeCallToGatekeeper**.

Lorsque l'élément d'information Fonctionnalité est utilisé pour indiquer une sémantique non modifiée telle que définie dans les Recommandations de la série Q.95x, il doit être codé conformément au 8.2.3/Q.932. Dans ce cas, les unités PDU de service doivent être constituées en fonction de l'élément ROSE [qui utilise l'UIT-T X.208 (Spécification de la syntaxe abstraite numéro un (ASN.1)) et l'UIT-T X.209 (Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1))] comme cela est défini dans l'UIT-T X.229.

7.2.2.17 Compatibilité de couche supérieure

Pour étude complémentaire.

7.2.2.18 Fonctionnalité Clavier

Codé conformément à la Figure 4-24/Q.931. L'utilisation du caractère de point d'exclamation "!" doit représenter une indication de double appel. Les extrémités ne prenant pas en charge la réception de l'indication de double appel ne doivent pas tenir compte du caractère "!" si elles le reçoivent.

7.2.2.19 Compatibilité de couche inférieure

Pour étude complémentaire.

7.2.2.20 Données à suivre

Ne doit pas être utilisé.

7.2.2.21 Fonctionnalités propres au réseau

Ne doit pas être utilisé.

7.2.2.22 Indicateur de notification

Codé conformément au 4.5.22/Q.931.

7.2.2.23 Indicateur de progression

Codé conformément à la Figure 4-29/Q.931 et au Tableau 4-20/Q.931.

Cet élément d'information n'est requis que pour l'interfaçage d'un terminal H.323 avec un terminal RNIS-ATM lorsque les informations de traitement d'appel détaillées sont disponibles. Dans ce cas, la passerelle transmettra ces informations au terminal H.323. Le système final H.323 n'a pas besoin d'interpréter cet élément d'information.

Si cet élément d'information est produit par un terminal H.323, les restrictions suivantes sont applicables:

Norme de codage (octet n° 3, bits 6, 7)

- Doit indiquer "UIT-T" ("00").

Emplacement

- Conformément au Tableau 4-20/Q.931.
- Les valeurs "utilisateur" ("0000"), "réseau privé desservant l'utilisateur local" ("0001"), et "réseau privé desservant l'utilisateur distant" ("0101") sont autorisées.

Description de la progression

- Conformément au Tableau 4-20/Q.931.

7.2.2.24 Numéro renvoyant

Cet élément doit être codé conformément au 4.6.7/Q.931. Noter que cet élément d'information n'est fourni que pour faciliter l'interfonctionnement avec le RCC et non pour offrir un mécanisme pour les services de déviation d'appel de type H.323 qui sont définis par l'UIT-T H.450.3.

7.2.2.25 Indicateur de répétition

Ne doit pas être utilisé.

7.2.2.26 Indicateur de reprise

Ne doit pas être utilisé.

7.2.2.27 Message fractionné

Ne doit pas être utilisé. Il convient de noter qu'il n'y a pas de limite supérieure critique pour la taille d'un message dans l'UIT-T H.323 et la présente Recommandation.

7.2.2.28 Numérotation complète

Codé conformément à la Figure 4-33/Q.931.

Il n'y a pas de restriction.

7.2.2.29 Signal

Codé conformément à la Figure 4-34/Q.931 et au Tableau 4-24/Q.931.

Il n'y a pas de restriction.

7.2.2.30 Sélection du réseau de transit

Ne doit pas être utilisé.

7.2.2.31 Utilisateur-utilisateur

Codé conformément à la Figure 4-36/Q.931 et au Tableau 4-26/Q.931, tels que modifiés ici.

L'élément d'information "Utilisateur-utilisateur" doit être utilisé par toutes les entités H.323 pour acheminer les informations H.323 associées. L'information réelle utilisateur-utilisateur à échanger entre les terminaux intervenant est emboîtée dans le champ **user-data** de l'unité PDU **H.323-UserInformation** (qui ne fait l'objet d'aucune restriction).

Les restrictions suivantes sont applicables:

Longueur des contenus utilisateur-utilisateur

- Doit être de 2 octets et non de 1 (comme cela est indiqué sur la Figure 4-36/Q.931).

Discriminateur de protocole

- Doit indiquer une information d'utilisateur codée à UIT-T X.208 et UIT-T X.209 (ASN.1) ("00000101").

NOTE – Ce codage est conforme à la révision de 1998 de l'UIT-T Q.931 qui renvoie aux révisions précédentes de l'ASN.1. Les références correctes à l'ASN.1 sont l'UIT-T X.680 (syntaxe) et l'UIT-T X.691 (règles de codage compactes PER).

Information d'utilisateur

- Doit contenir une structure ASN.1 (**H323-UserInformation**) qui, parallèlement aux informations pertinentes H.323, inclut des données d'utilisateur réelles, par exemple comme suit. L'ASN.1 est codé en utilisant la variation "aligned" des règles de codage compactes spécifiées dans l'UIT-T X.691.

La structure **H323-UserInformation** contient les champs **h323-uu-pdu** et **user-data**.

Le champ **h323-uu-pdu** de la structure **H323-UserInformation** contient les sous-champs suivants. Noter que tous les sous-champs du champ **h323-uu-pdu** ne sont pas autorisés dans tous les messages. Voir les restrictions indiquées dans la description individuelle de chaque message.

- **h323-message-body** – Ce champ contient des informations propres à un message Q.931 particulier, comme décrit aux 7.3 et 7.4. Un expéditeur peut choisir l'option **empty** s'il n'est pas nécessaire d'envoyer le champ de l'élément d'information "Utilisateur-utilisateur" (**Facility-UUIE**, etc.) dans un message particulier, par exemple lorsqu'un message FACILITY est utilisé pour transporter des informations non associées à un appel. Noter qu'à partir de la version 4 de la présente Recommandation, si un message est associé à un

appel particulier, l'expéditeur doit inclure le champ de l'élément d'information "Utilisateur-utilisateur". Cela est nécessaire afin de fournir le champ **callIdentifiant**.

- **nonStandardData** – Ce champ contient des informations non définies dans la présente Recommandation (par exemple, données non normalisées).
- **H4501SupplementaryService** – Ce champ contient une séquence d'unités APDU de type H4501SupplementaryService comme défini dans le Tableau 3/H.450.1.
- **h245Tunneling** – Cet élément est mis à la valeur Vrai si la canalisation en tunnel des messages H.245 est activée. Les systèmes conformes au moins à la version 4 de la présente Recommandation doivent mettre cet élément à la valeur Vrai si la procédure de connexion rapide est utilisée pour établir l'appel.
- **h245Control** – Ce champ contient une séquence d'unités PDU H.245 canalisées en tunnel. Chaque chaîne d'octets doit contenir exactement une seule unité PDU H.245.
- **nonStandardControl** – Ce champ contient des informations de commande non définies dans la présente Recommandation (p. ex. des informations de commande non normalisées).
- **callLinkage** – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir l'UIT-T H.323.
- **tunnelledSignallingMessage** – Ce champ désigne un message de signalisation entièrement canalisé en tunnel dans son format d'origine afin de prendre en charge une signalisation additionnelle de commande d'appel de bout en bout. Le champ **tunnelledProtocolID** désigne le protocole à mettre en tunnel. Le champ **messageContent** est une séquence de messages réels entièrement canalisés en tunnel, dans leur format binaire d'origine; cela permet l'agrégation de messages canalisés en un seul message H.225.0. Si le champ **tunnelingRequired** est présent, l'appel ne peut être établi que si la canalisation en tunnel est prise en charge.
- **provisionalRespToH245Tunneling** – Ce fanion est utilisé pour signaler que l'entité appelée n'a pas encore déterminé si la canalisation en tunnel H.245 est applicable pour l'appel considéré. S'il est présent, le fanion **h245Tunneling** doit être négligé par l'entité réceptrice.
- **stimulusControl** – Ce champ est réservé pour utilisation future par l'UIT-T pour un protocole fondé sur un stimulus.
- **genericData** – Ce champ est une liste d'éléments génériques associés à des éléments de service définis en dehors de la spécification H.225.0 de base. Ces paramètres peuvent être utilisés, par exemple, pour la canalisation en tunnel transparente d'informations conformes au protocole H.225.0.

Le champ **user-data** de la structure **H323-UserInformation** contient les champs suivants:

- **protocol-discriminator** – Ce champ est codé conformément au Tableau 4-26/Q.931.
- **user-information** – Ce champ est codé conformément au 4.5.30/Q.931.

7.3 Informations complémentaires concernant les messages Q.931

Il convient de noter que les longueurs des éléments d'information spécifiées dans les tableaux ci-après concernent les messages qui sont produits par des terminaux H.323 uniquement. La taille indiquée de l'élément d'information Utilisateur-utilisateur correspond à la taille de la structure **user-data** de **H323-UserInformation** et ne comprend pas l'élément **h323-UU-PDU**. La taille totale de **H323-UserInformation** est limitée à 65 536 octets. Indépendamment des tailles spécifiées, les messages transmis du côté RCC peuvent avoir des tailles différentes (très grandes).

Il convient également de noter que pour les éléments d'information, les termes obligatoires, facultatifs et proscrits se rapportent à la possibilité pour les terminaux H.323 de produire ou non ces éléments d'information.

7.3.1 Alerte (ALERTING)

Ce message peut être envoyé par l'utilisateur appelé pour indiquer que l'alerte de l'utilisateur appelé a été déclenchée. En termes courants cela veut dire que "le téléphone sonne".

Se conformer au Tableau 3-2/Q.931 (version 1998) tel que modifié ci-après dans le Tableau 6.

Tableau 6/H.225.0 – Contenu du message ALERTING (Alerte)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Capacité support	O	5-6
Fonctionnalité étendue	O	8-*
Identification du canal	A étudier	NA
Fonctionnalité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Signal	O	2-3
Compatibilité de couche supérieure	A étudier	NA
Utilisateur-utilisateur	M	2-131

L'élément d'information Utilisateur-utilisateur contient l'élément d'information Alerting-UUIE défini dans la syntaxe des messages H.225.0. L'élément **Alerting-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type d'extrémité) pour permettre à l'appelant de déterminer si l'appel fait intervenir une passerelle ou non.

h245Address – Adresse de transport spécifique sur laquelle l'extrémité appelée ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse peut également être envoyée dans le message CALL PROCEEDING, PROGRESS ou CONNECT.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message SETUP avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CALLPROCEEDING, ALERTING, PROGRESS ou CONNECT.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion d'appel rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel** définie dans l'UIT-T H.245, mais l'expéditeur de **fastStart** indique les modes qu'il préfère recevoir et envoyer ainsi que les adresses de transport auxquelles il devrait recevoir les flux multimédias.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

alertingAddress – Contient les adresses pseudonymes pour le correspondant à l'origine de l'alerte.

presentationIndicator – Indique si la présentation de l'adresse **alertingAddress** doit être autorisée ou limitée.

screeningIndicator – Indique si l'adresse **alertingAddress** a été communiquée par l'extrémité ou le réseau (portier) et si elle a été filtrée par un portier.

fastConnectRefused – Une extrémité appelée devrait renvoyer cet élément dans tout message jusqu'au message CONNECT inclus lors de l'établissement d'un appel afin d'indiquer qu'elle refuse la procédure de connexion rapide.

serviceControl – Ce champ contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées dans le cadre de la procédure d'établissement par l'extrémité appelante (p. ex. pour afficher un menu d'options de déviation d'appel) comme cela est décrit dans l'Annexe K/H.323.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un moment donné, à condition que le message ALERTING considéré concerne une communication active. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

7.3.2 Appel en cours (CALL PROCEEDING)

Ce message peut être envoyé par l'utilisateur appelé pour indiquer que l'établissement d'appel demandé a été déclenché et pour indiquer qu'aucune nouvelle information d'établissement d'appel n'est plus acceptée. Voir Tableau 7.

Tableau 7/H.225.0 – Contenu du message CALL PROCEEDING (Appel en cours)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Capacité support	O	5-6
Fonctionnalité étendue	O	8-*
Identification du canal	A étudier	NA
Fonctionnalité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Compatibilité de couche supérieure	A étudier	NA
Utilisateur-utilisateur	M	2-131

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **CallProceeding-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **CallProceeding-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type d'extrémité) pour permettre à l'appelant de déterminer si l'appel fait intervenir une passerelle ou non.

h245Address – Adresse de transport spécifique sur laquelle l'extrémité appelée ou le portier qui traite l'appel aimerait établir la signalisation H.245.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message SETUP avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CALL PROCEEDING, ALERTING, PROGRESS ou CONNECT.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel** définie dans l'UIT-T H.245, mais l'expéditeur de **fastStart** indique les modes qu'il préfère recevoir et envoyer ainsi que les adresses de transport auxquelles il devrait recevoir les flux multimédias.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

fastConnectRefused – Une extrémité appelée devrait renvoyer cet élément dans tout message jusqu'au message CONNECT inclus lors de l'établissement d'un appel afin d'indiquer qu'elle refuse la procédure de connexion rapide.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

7.3.3 Connexion (CONNECT)

Ce message est envoyé par le demandé au demandeur (portier, passerelle ou terminal appelant) pour signaler que l'appelé accepte l'appel. Se conformer au Tableau 3-4/Q.931, tel que modifié dans le Tableau 8 ci-dessous.

Tableau 8/H.225.0 – Contenu du message CONNECT (Connexion)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Capacité support	O (Note)	5-6
Fonctionnalité étendue	O	8-*
Identification du canal	A étudier	NA
Fonctionnalité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Date/heure	O	8
Numéro connecté	O	2-*
Sous-adresse connectée	O	2-23
Compatibilité de couche inférieure	A étudier	NA
Compatibilité de couche supérieure	A étudier	NA
Utilisateur-utilisateur	M	2-131
NOTE – L'élément d'information Capacité support est obligatoire si le message est échangé entre un terminal et une passerelle.		

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **Connect-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **Connect-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

h245Address – Adresse de transport spécifique sur laquelle l'extrémité appelée ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse doit être envoyée si elle a été envoyée antérieurement dans le message ALERTING, PROGRESS ou CALL PROCEEDING.

destinationInfo – Contient un **EndpointType** (type d'extrémité) pour permettre à l'appelant de déterminer si l'appel fait intervenir une passerelle ou non.

conferenceID – Contient un numéro propre permettant d'identifier de manière univoque la conférence; il s'agit du numéro reçu dans le message SETUP.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message SETUP avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CALL PROCEEDING, ALERTING, PROGRESS ou CONNECT.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel** définie dans l'UIT-T H.245, mais l'expéditeur de **fastStart** indique les modes qu'il préfère recevoir et envoyer ainsi que les adresses de transport auxquelles il devrait recevoir les flux multimédias.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

language – Indique le ou les langages dans lesquels l'utilisateur souhaiterait de préférence recevoir les annonces et les invites. Ce champ contient une ou plusieurs étiquettes de langage conformes au document RFC 1766.

connectedAddress – Contient les adresses pseudonymes pour le correspondant connecté (qui répond); la chaîne de chiffres composés manuellement du correspondant connecté figure dans l'élément d'information Numéro connecté.

presentationIndicator – Indique si la présentation de l'adresse du numéro connecté **connectedAddress** doit être autorisée ou limitée. Si l'indicateur **presentationIndicator** et l'indicateur de présentation de l'élément d'information Numéro connecté sont tous les deux présents mais incompatibles, l'indicateur de présentation de l'élément d'information Numéro connecté doit être utilisé.

screeningIndicator – Indique si l'adresse du numéro connecté **connectedAddress** a été communiquée par l'extrémité ou le réseau (portier) et si elle a été filtrée par un portier. Si l'indicateur **screeningIndicator** et l'indicateur de filtrage de l'élément d'information Numéro connecté sont tous les deux présents mais incompatibles, l'indicateur de filtrage de l'élément d'information Numéro connecté doit être utilisé.

fastConnectRefused – Une extrémité appelée devrait renvoyer cet élément dans tout message jusqu'au message CONNECT inclus lors de l'établissement d'un appel afin d'indiquer qu'elle refuse la procédure de connexion rapide.

serviceControl – Contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées dans le cadre de la procédure d'établissement par l'extrémité appelante (par exemple pour afficher un menu d'options de déviation d'appel) comme cela est décrit dans l'Annexe K/H.323.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un moment donné, à condition que le message ALERTING considéré concerne une communication active. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques qui se rapportent à l'appel considéré.

7.3.4 Acquittement de connexion (CONNECT ACKNOWLEDGE)

Ce message ne devra pas être envoyé.

7.3.5 Déconnexion (DISCONNECT)

Ce message ne doit pas être envoyé par une entité H.323.

Le contenu et la sémantique d'un message DISCONNECT reçu à partir d'un réseau sont définis dans le Tableau 3-6/Q.931 et au 10.5 de l'ISO/CEI 11582.

7.3.6 Information (INFORMATION)

Ce message peut être envoyé afin de fournir des compléments d'information. Il peut être utilisé pour transmettre des informations relatives à l'établissement des communications (par exemple la signalisation avec chevauchement) ou pour transmettre des informations diverses concernant les appels.

Ce message peut être envoyé par une entité H.323.

Ce message se conforme au Tableau 3-7/Q.931 moyennant les modifications suivantes (voir Tableau 9).

Tableau 9/H.225.0 – Contenu du message INFORMATION (Information)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Fin de numérotation	O	1
Affichage	O	2-82
Fonctionnalité clavier	O	2-34
Signal	O	2-3
Numéro appelé	O	2-35
Utilisateur-utilisateur	M	2-131

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **Information-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **Information-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

callIdentifier – Identificateur d'appel unique à l'échelle mondiale, qui est activé par l'extrémité d'origine et qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est décrite dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

fastStart – Ce champ ne doit pas être inclus. Il doit être ignoré dès réception.

fastConnectRefused – Ce champ ne doit pas être inclus. Il doit être ignoré dès réception.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

7.3.7 Progression (PROGRESS)

Ce message peut être envoyé par une passerelle H.323 pour indiquer la progression d'un appel en cas d'interfonctionnement avec un RCC. Ce message peut aussi être envoyé par une extrémité H.323 avant le message CONNECT, en fonction de l'interaction avec des services complémentaires.

Se conformer au Tableau 3-9/Q.931 et au 10.10 de l'ISO/CEI 11582, tels que modifiés au Tableau 10 ci-dessous.

Tableau 10/H.225.0 – Contenu du message PROGRESS (Progression)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Capacité support	O (Note)	5-6
Cause	O	2-32
Fonctionnalité étendue	O	8-*
Identification du canal	A étudier	NA
Fonctionnalité	O	8-*
Indicateur de progression	M	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Compatibilité de couche supérieure	A étudier	NA
Utilisateur-utilisateur	M	2-131
NOTE – L'élément d'information Capacité support est obligatoire si le message est échangé entre un terminal et une passerelle.		

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **Progress-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **Progress-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de la Recommandation H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type d'extrémité) pour permettre à l'appelant de déterminer si l'appel fait intervenir une passerelle ou non.

h245Address – Adresse de transport spécifique sur laquelle l'extrémité appelée ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse doit être envoyée si elle a été envoyée antérieurement dans le message CALL PROCEEDING, ALERTING ou CONNECT.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message SETUP avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CALL PROCEEDING, ALERTING, PROGRESS ou CONNECT.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**tokens**) chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion d'appel rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel** définie dans l'UIT-T H.245, mais l'expéditeur de **fastStart** indique les modes qu'il préfère recevoir et envoyer ainsi que les adresses de transport auxquelles il devrait recevoir les flux multimédias.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

fastConnectRefused – Une extrémité appelée devrait renvoyer cet élément dans tout message jusqu'au message CONNECT inclus lors de l'établissement d'un appel afin d'indiquer qu'elle refuse la procédure de connexion rapide.

7.3.8 Libération (RELEASE)

Ce message ne doit pas être envoyé par une entité H.323.

Le contenu et la sémantique d'un message RELEASE reçu en provenance du réseau sont définis dans le Tableau 3-10/Q.931 et au 10.5 de l'ISO/CEI 11582.

7.3.9 Fin de libération (RELEASE COMPLETE)

Ce message doit être envoyé par un terminal pour indiquer la libération de l'appel si la voie de signalisation d'appel fiable est ouverte. La valeur de la référence d'appel (CRV, *call reference value*) devient ensuite disponible pour réemploi éventuel.

La séquence déconnexion/libération/fin de libération n'est pas utilisée étant donné que la seule valeur ajoutée est qu'un élément d'information réseau-utilisateur peut être adjoint au message de libération. Comme elle ne s'applique pas à un environnement de réseau à commutation par paquets, la méthode à une seule étape pour l'envoi uniquement du message fin de libération est utilisée.

Ce message est conforme au Tableau 3-11/Q.931. Les modifications du Tableau 11 s'y appliquent.

Tableau 11/H.225.0 – Contenu du message RELEASE COMPLETE (Fin de libération)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Cause	CM (Note)	2-32
Fonctionnalité	O	8-*
Indicateur de notification	O	2-*
Affichage	O	2-82
Signal	O	2-3
Utilisateur-utilisateur	M	2-131
NOTE – L'un ou l'autre de l'élément d'information Cause et de l'élément ReleaseCompleteReason doit être présent.		

Si ce message est envoyé en réponse à un message FACILITY où l'élément d'information Fonctionnalité est vide, l'élément ReleaseCompleteReason doit être mis à **facilityCallDeflection**.

Si ce message est retransmis à partir d'un RCC par une passerelle, la valeur de cause doit être fixée comme spécifié dans l'UIT-T Q.931.

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **ReleaseComplete-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **ReleaseComplete-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

reason – Plus de renseignements sur la cause de la libération de l'appel. Une valeur de cause **genericDataReason** indique que l'appel a été libéré à cause d'un élément générique de service ou de réseau. Dans ce cas, des informations additionnelles peuvent être spécifiées dans le champ **genericData** de l'unité **h323-uu-pdu** du message considéré. Une valeur de cause **neededFeatureNotSupported** indique qu'un élément de service requis par une entité n'est pas pris en charge par une autre entité. Une valeur de cause **tunnelledSignallingRejected** est envoyée si l'appel est libéré parce que l'expéditeur n'autorise pas la signalisation non H.323 en tunnel et parce que la canalisation en tunnel est requise afin que l'appel soit établi.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

busyAddress – Contient les adresses pseudonymes pour le correspondant occupé.

presentationIndicator – Indique si la présentation de l'adresse **busyAddress** doit être autorisée ou limitée.

screeningIndicator – Indique si l'adresse **busyAddress** a été communiquée par l'extrémité ou le réseau (portier) et si elle a été filtrée par un portier.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice une fois que l'appel indiqué dans le message Release Complete considéré a été libéré. Lorsqu'elle envoie ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

serviceControl – Contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées dans le cadre de la procédure d'établissement par l'extrémité appelante (p. ex. pour afficher un menu d'options de déviation d'appel) comme cela est décrit dans l'Annexe K/H.323.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

7.3.10 Etablissement (SETUP)

Ce message doit être envoyé par une entité H.323 appelante pouvant indiquer qu'elle souhaite établir une connexion avec l'entité appelée.

Se conformer au Tableau 13-15/Q.931 modifié comme au Tableau 12.

Tableau 12/H.225.0 – Contenu du message SETUP (Etablissement)

Élément d'information	Statut H.225.0 (M/F/O/CM)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M (Note 2)	3
Type de message	M	1
Fin de numérotation	O	1
Indication de répétition	F	NA
Capacité support	M	5-6
Fonctionnalité étendue	O	8-*
Identification du canal	A étudier	NA

Tableau 12/H.225.0 – Contenu du message SETUP (Etablissement)

Élément d'information	Statut H.225.0 (M/F/O/CM)	Longueur H.225.0
Fonctionnalité	O	8-*
Indicateur de progression	F	NA
Fonctionnalités propres au réseau	F	NA
Indicateur de notification	O	2-*
Affichage	O	2-82
Fonctionnalité clavier	O	2-34
Signal	O	2-3
Numéro de l'appelant	O	2-131
Sous-adresse de l'appelant	CM (Note 1)	NA
Numéro de l'appelé	O	2-131
Sous-adresse de l'appelé	CM (Note 1)	NA
Numéro renvoyant	O	2-*
Sélection du réseau de transit	F	NA
Indicateur de répétition	F	NA
Compatibilité de couche inférieure	A étudier	NA
Compatibilité de couche supérieure	A étudier	NA
Utilisateur-utilisateur	M	2-131
NOTE 1 – Les sous-adresses sont nécessaires pour certains scénarios d'appel RCC; elles ne doivent pas être utilisées pour des appels côté réseau à commutation par paquets seulement. NOTE 2 – Si un message ARQ a été précédemment envoyé, la valeur de référence d'appel utilisée ici doit être la même.		

L'élément d'information Utilisateur-utilisateur contient l'élément d'information **Setup-UUIE** défini dans la syntaxe des messages H.225.0. L'élément **Setup-UUIE** comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

h245Address – Adresse de transport spécifique sur laquelle l'extrémité appelée ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse ne doit être fournie par l'expéditeur que s'il est en mesure de traiter les procédures H.245 avant de recevoir un message CONNECT sur la voie de signalisation d'appel.

sourceAddress – Contient les adresses pseudonymes de la source. L'adresse primaire doit se trouver en premier. Noter que le numéro E.164 de la source, s'il existe, doit être contenu dans l'élément d'information "Numéro de l'appelant".

sourceInfo – Contient un élément **EndpointType** pour permettre à l'appelé de déterminer si l'appel fait intervenir une passerelle ou non.

destinationAddress – Adresse à laquelle l'extrémité souhaite être connectée. L'adresse primaire devra se trouver en premier. Pour appeler une extrémité en utilisant uniquement la chaîne de chiffres composés manuellement, cette adresse doit être placée dans l'élément d'information Numéro de l'appelé Q.931. Si le champ **destinationAddress** est disponible, il doit être inclus dans le message SETUP par les terminaux conformes à la version 2 ou à une version supérieure de la présente Recommandation.

destCallSignalAddress – Nécessaire pour informer le portier de l'adresse de transport utilisée par le terminal de destination pour la signalisation d'appel; redondant dans le cas direct terminal à terminal. Dans les cas où l'expéditeur du message SETUP dispose de l'information, ce champ doit être rempli.

destExtraCallInfo – Nécessaire pour rendre possibles des appels sur canaux additionnels, c'est-à-dire pour un appel 2×64 kbit/s du côté RCC. Ne doit contenir que les chaînes de chiffres composés manuellement, les numéros E.164 ou les numéros de plan privé et ne doit pas contenir le numéro du canal initial. (Voir Note.)

destExtraCRV – Valeurs CRV pour les autres appels RCC spécifiés par **destExtraCallInfo**. Leur utilisation appelle un complément d'étude. Elles peuvent être utilisées pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

activeMC – Indique que l'extrémité appelante est sous l'influence d'un contrôleur multipoint activé.

conferenceID – Identificateur univoque de la conférence.

conferenceGoal:

- **create** – Lancer une nouvelle conférence;
- **invite** – Inviter un correspondant à une conférence existante;
- **join** – Rejoindre une conférence en cours;
- **capability-negotiation** – Négocier les capacités d'une conférence ultérieure souple;
- **callIndependentSupplementaryService** – Transporter des unités APDU de services complémentaires hors de tout appel.

callServices – Fournit des informations sur la prise en charge des protocoles facultatifs de la série Q à l'intention du portier et du terminal appelé.

callType – Lorsqu'il utilise cette valeur, le portier du demandé peut essayer de déterminer la largeur de bande réellement utilisée. La valeur par défaut est **pointToPoint** pour tous les appels; il convient de noter que le type d'appel peut changer dynamiquement tout au long de l'appel et que le type d'appel définitif peut ne pas être connu au moment de l'envoi du message SETUP.

sourceCallSignalAddress – Contient l'adresse de transport utilisée par la source; cette valeur doit être utilisée dans le message ARQ par le destinataire du message SETUP. Dans tous les cas où l'expéditeur du message SETUP dispose de l'information, ce champ doit être rempli. La valeur du champ **sourceCallSignalAddress** doit être identique à la valeur qui a été utilisée dans le message ARQ par l'expéditeur du message SETUP et doit être utilisée par l'extrémité recevant le message SETUP dans son message ARQ.

remoteExtensionAddress – Contient l'adresse pseudonyme d'une extrémité appelée dans les cas où cette information est nécessaire pour traverser plusieurs passerelles. Dans tous les cas où l'expéditeur du message SETUP dispose de l'information, ce champ doit être rempli.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

h245SecurityCapability – Ensemble des capacités que l'expéditeur peut utiliser pour fiabiliser le canal H.245.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel**

définie dans l'UIT-T H.245, mais l'expéditeur de **fastStart** indique les modes qu'il préfère recevoir et envoyer ainsi que les adresses de transport auxquelles il devrait recevoir les flux multimédias.

mediaWaitForConnect – Si ce paramètre a la valeur Vrai, cela indique que le destinataire du message d'établissement ne doit pas émettre de données multimédias avant d'avoir envoyé le message de connexion.

canOverlapSend – Si la valeur de ce paramètre est Vrai cela indique que l'expéditeur du message SETUP doit pouvoir effectuer une numérotation avec chevauchement.

endpointIdentif – Il s'agit d'un identificateur d'extrémité qui a été attribué au terminal dans le message RCF. Ce champ doit être présent lorsque le message d'établissement SETUP est envoyé au portier auprès duquel l'extrémité est enregistré; il ne doit pas être présent quand le message d'établissement est envoyé à une autre entité.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

ConnectionParameters – Permet de spécifier les paramètres utiles pour les passerelles qui assurent plusieurs types de connexion et/ou l'agrégation (ou regroupement) de canaux (par exemple les passerelles H.323/H.320):

- **scnConnectionType** – Fournit à une passerelle des informations sur le type de connexion individuelle utilisée pour établir d'un bout à l'autre la communication RCC. Les extrémités ou les portiers doivent remplir ce champ si ces informations leur sont communiquées. Si l'option "multidébit" est indiquée, l'octet de débit de transfert d'information de la capacité support doit aussi indiquer "multidébit" et l'octet multiplicateur de débit doit indiquer le nombre de connexions. Dans tous les autres cas, si le champ **scnConnectionType** est présent, il prévaut sur toute indication concernant le type de connexion individuelle figurant dans le débit de transfert (octet n° 4) et le multiplicateur de débit (octet n° 4.1) de l'élément d'information Capacité support.
- **numberOfSCNConnections** – Indique le nombre de connexions de type **scnConnectionType** qui sont regroupées ensemble pour former la communication RCC. Ce champ, lorsqu'il est multiplié par la largeur de bande de la connexion individuelle spécifiée dans **scnConnectionType**, indique la largeur de bande de la communication établie d'un bout à l'autre sur le réseau RCC. Les extrémités ou les portiers doivent remplir ce champ si ces informations leur sont communiquées. Il convient de noter que si le champ **scnConnectionType** est mis sur "valeur non connue", on prend pour hypothèse une unité de largeur de bande de 64 kbit/s. Si ce champ et le champ **scnConnectionType** sont tous les deux présents, la largeur de bande totale indiquée doit correspondre à la largeur de bande RCC totale indiquée par le débit de transfert (octet n° 4) et le multiplicateur de débit (octet n° 4.1) de l'élément d'information Capacité support.
- **scnConnectionAggregation** – Indique comment les connexions individuelles sont regroupées ensemble pour former la totalité de la communication RCC. Les extrémités ou les portiers doivent remplir ce champ si les informations leur sont communiquées. L'option par défaut, à utiliser lorsque le mécanisme de regroupement effectif est inconnu, est "automatique". Lorsque l'on sait que le réseau est mis à la masse, mais que l'on ignore le mode précis de mise à la masse, l'option "mode de mise à la masse 1" doit être utilisée.

language – Indique le ou les langages dans lesquels l'utilisateur souhaiterait de préférence recevoir les annonces et les invites. Ce champ contient une ou plusieurs étiquettes de langage conformes au document RFC 1766.

presentationIndicator – Indique si la présentation de l'adresse de la source **sourceAddress** doit être autorisée ou limitée. Si l'indicateur **presentationIndicator** et l'indicateur de présentation de l'élément d'information Numéro de l'appelant sont tous les deux présents mais incompatibles, l'indicateur de présentation de l'élément d'information Numéro de l'appelant doit être utilisé.

screeningIndicator – Indique si l'adresse de la source **sourceAddress** a été communiquée par l'extrémité ou le réseau (portier) et si elle a été filtrée par un portier. Si l'indicateur **screeningIndicator** et l'indicateur de filtrage de l'élément d'information Numéro de l'appelant sont tous les deux présents mais incompatibles, l'indicateur de filtrage de l'élément d'information Numéro de l'appelant doit être utilisé.

serviceControl – Contient des données propres au service ou des références à celui-ci qui peuvent être utilisées dans le cadre de la procédure d'établissement par l'extrémité appelée (p. ex. une image ou une icône à afficher pendant l'alerte) comme cela est décrit dans l'Annexe K/H.323.

symmetricOperationRequired – S'il est à Vrai, ce champ indique que l'extrémité appelée doit sélectionner des capacités identiques d'émission et de réception audio. Cet élément ne doit pas être inclus si l'élément **fastStart** n'est pas également inclus.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un moment donné, à condition que le message SETUP considéré concerne une communication active. Lorsqu'elle envoie ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

desiredProtocols – Ce champ identifie, par ordre de préférence, les types de protocole recherchés par l'extrémité pour son appel (p. ex. voix ou télécopie). Une entité de résolution peut utiliser ce champ pour localiser une extrémité qui prend également en charge le protocole, compte tenu de l'ordre de préférence.

neededFeatures – Ce champ spécifie une liste d'éléments de service génériques qui sont nécessaires à l'établissement de l'appel.

desiredFeatures – Ce champ spécifie une liste d'éléments de service génériques qui sont préférés pour l'appel mais qui ne sont pas nécessaires à son établissement.

supportedFeatures – Ce champ spécifie une liste d'éléments de service génériques que l'expéditeur prend en charge et a choisi de déclarer.

parallelH245Control – Ce champ contient une séquence d'unités PDU H.245 d'ensemble de capacités de terminal mises en tunnel ainsi que, facultativement, d'unités PDU de détermination de relation maître-esclave. Chaque chaîne d'octets doit contenir exactement une seule unité PDU H.245.

additionalSourceAddresses – Ce champ contient une séquence d'adresses pseudonymes qui correspond au deuxième élément d'information "Numéro de l'appelant" et aux suivants dans les réseaux non H.323. Par exemple, dans le RNIS, plusieurs numéros d'appelant peuvent être présents afin de prendre en charge "l'option d'acheminement de deux éléments d'information Numéro de l'appelant" définie dans l'Annexe A/Q.951.

NOTE – Si le champ **destExtraCallInfo** est présent, une valeur de référence d'appel (CRV) pour chaque appel à effectuer peut être fournie dans le champ **destExtraCRV**. Ces valeurs CRV doivent être utilisées pour identifier toute réponse à chaque appel lancé. Ces procédures appellent un complément d'étude. Si le champ **destExtraCRV** n'est pas présent, une passerelle regroupera toutes les informations d'appel en une seule réponse et de ce fait, si un des appels échoue du côté RCC, l'appel entier doit être traité comme ayant échoué.

7.3.11 Acquiescement d'établissement (SETUP ACKNOWLEDGE)

Ce message peut être envoyé par une entité H.323. Cependant, il peut être retransmis à partir du réseau par l'intermédiaire d'une passerelle. Son traitement dès réception est facultatif mais une entité

qui indique **canOverlapSend** dans le message SETUP devra prendre en charge le message SETUP ACKNOWLEDGE.

Le contenu et la sémantique d'un message SETUP ACKNOWLEDGE reçu en provenance du réseau sont définis dans le Tableau 3-16/Q.931, tel que modifié par le Tableau 13.

Tableau 13/H.225.0 – SETUP ACKNOWLEDGE (acquittement d'établissement)

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Identification de canal	A étudier	NA
Affichage	O	2-82
Utilisateur-utilisateur	M	2-131

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4/H.225.0, l'expéditeur de ce message ne doit pas inclure le champs **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information "Utilisateur-utilisateur".

L'élément d'information Utilisateur-utilisateur contient l'élément **SetupAcknowledge-UUIE** qui est défini dans la syntaxe de messagerie H.225.0 et qui contient les champs suivants:

protocolIdentifier – Ce champ est mis au numéro de version de la présente Recommandation qui est pris en charge.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

7.3.12 Etat (STATUS)

Le message STATUS doit être envoyé en réponse à un message de signalisation d'appel inconnu ou à un message de demande d'état Status Inquiry.

Se conformer au Tableau 3-17/Q.931 tel que modifié par le Tableau 14.

Tableau 14/H.225.0 – STATUS (état)

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel (Note)	M	3
Type de message	M	1
Cause	M	4-32
Etat d'appel	M	3
Affichage	O	2-82
Utilisateur-utilisateur	M	2-131
NOTE – Ce message peut contenir la référence d'appel globale si le message s'applique à tous les appels d'une connexion transportant des appels multiples.		

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4/H.225.0, l'expéditeur de ce message ne doit pas inclure le champs **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information "Utilisateur-utilisateur".

L'élément d'information Utilisateur-utilisateur contient l'élément **Status-UUIE** qui est défini dans la syntaxe de messagerie H.225.0 et qui contient les champs suivants:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

7.3.13 Demande d'état (STATUS INQUIRY)

Le message STATUS INQUIRY peut être utilisé pour demander l'état d'un appel tel que décrit au 8.4.2/H.323.

Se conformer au Tableau 3-18/Q.931 tel que modifié par le Tableau 15.

Tableau 15/H.225.0 – STATUS INQUIRY (demande d'état)

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel (Note)	M	3
Type de message	M	1
Affichage	O	2-82
Utilisateur-utilisateur	M	2-131
NOTE – Ce message peut contenir la référence d'appel globale si le message s'applique à tous les appels d'une connexion transportant des appels multiples.		

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4/H.225.0, l'expéditeur de ce message ne doit pas inclure le champs **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information "Utilisateur-utilisateur".

L'élément d'information Utilisateur-utilisateur contient l'élément **StatusInquiry-UUIE** qui est défini dans la syntaxe de messagerie H.225.0 et qui contient les champs suivants:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée:

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

7.4 Détails des messages Q.932

Les messages définis ci-dessous dérivent des UIT-T Q.932 et H.450. On se référera à celles-ci pour de plus amples détails.

7.4.1 Fonctionnalité (FACILITY)

Le message FACILITY doit être utilisé pour fournir des informations sur l'endroit où un appel doit être dirigé (**FacilityReason = routeCallToMC**) ou doit être utilisé par une extrémité pour indiquer que l'appel entrant doit passer par un portier (**FacilityReason = routeCallToGatekeeper**).

Pour signaler un réacheminement d'appel propre aux procédures H.323, l'élément d'information **Facility-UUIE** est utilisé. Ce cas particulier doit être indiqué par le codage d'un élément d'information Fonctionnalité de longueur zéro. Dans ce cas, l'élément d'information Fonctionnalité doit comporter exactement 2 octets. Une entité H.323 doit traiter correctement l'élément d'information Fonctionnalité (propre aux procédures H.323) vide et doit avoir la capacité de sauter les autres éléments d'information de fonctionnalité qu'elle ne comprend pas.

Le message FACILITY peut être utilisé pour demander un service complémentaire ou en accuser réception conformément aux Recommandations de la série H.450. C'est pourquoi une ou plusieurs unités APDU de services complémentaires H.450 doivent être transportées à l'intérieur de l'élément d'information Facility-UUIE. Les unités APDU de services complémentaires H.450 doivent être codées conformément au paragraphe 8/H.450.1. L'élément d'information Fonctionnalité contenu doit être de longueur nulle. A noter qu'un message FACILITY de version 2 ou 3 de la présente Recommandation, qui ne transporte que des unités APDU de services complémentaires H.450, pourrait choisir de ne pas utiliser l'élément d'information Facility-UUIE, mais d'utiliser à la place la valeur "**empty**" de l'élément **h323-message-body**. Dans ce cas, un message FACILITY ne contiendra pas de champ **callIdentifier**. Dans les versions 4 et au-delà de la présente Recommandation, un expéditeur doit inclure un élément d'information "Utilisateur-utilisateur" de message FACILITY contenant un champ **callIdentifier** dans chaque message FACILITY associé à l'appel, et doit mettre la valeur du champ **reason** à "**transportedInformation**".

Si un élément d'information Fonctionnalité transportant la sémantique de l'UIT-T Q.932 et codé conformément aux dispositions des UIT-T Q.932 et Q.95x est présent, il doit être composé d'au moins 8 octets, comme cela est exigé dans le Tableau 7-2/Q.932. L'utilisation d'éléments d'information de fonctionnalité de ce type appelle un complément d'étude.

Le message FACILITY peut être utilisé par une extrémité ou par un portier pour demander au destinataire d'établir une voie H.245 entre les deux entités (**FacilityReason = startH245**).

Le message FACILITY peut être utilisé par une extrémité ou par un portier afin d'envoyer un nouvel ensemble de jetons dans le champ **tokens** et/ou **cryptoTokens** du message FACILITY (**FacilityReason = newTokens**). Cela peut être utile, par exemple, pour des applications dans lesquelles des jetons sont utilisés pour permettre à une certaine action de n'avoir lieu que pendant une durée limitée.

Se conformer au 7.1.1/Q.932 et au 10.8 de l'ISO/CEI 11582, tels que modifiés au Tableau 16.

Tableau 16/H.225.0 – Contenu du message FACILITY (Fonctionnalité)

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel (Note 1)	M	3
Type de message	M	1
Fonctionnalité étendue	O (Note 2)	8-*
Fonctionnalité	O (Note 2)	2 ou 8-*
Indicateur de notification	O	2-*
Affichage	O	2-82
Numéro de l'appelant	F	NA
Numéro de l'appelé	F	NA
Utilisateur-utilisateur	M	2-131
<p>NOTE 1 – Ce message peut contenir la référence d'appel globale si le message s'applique à tous les appels d'une connexion transportant des appels multiples.</p> <p>NOTE 2 – Si le message FACILITY est utilisé pour transporter la signalisation de services complémentaires Q.95x, l'un des deux éléments d'information fonctionnalité et fonctionnalité étendue est nécessaire. Si le message FACILITY est utilisé pour le contrôle des services complémentaires conformément aux Recommandations de la série H.450.x ou s'il est utilisé pour le reroutage vers les fonctions du contrôleur multipoint/portier, alors l'élément d'information Fonctionnalité de longueur zéro est nécessaire.</p>		

Codage de l'élément d'information de type de message

L'élément d'information "Type de message" du message FACILITY doit être codé "0110 0010".

L'élément d'information Utilisateur-utilisateur contient l'élément d'information Facility-UUIE défini dans la syntaxe des messages H.225.0. L'élément Facility-UUIE comprend ce qui suit:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

alternativeAddress – Adresse de transport spécifique vers laquelle l'appelant doit diriger l'appel; si ce champ est présent, le champ **alternativeAliasAddress** n'est pas nécessaire.

alternativeAliasAddress – Contient des pseudonymes qui peuvent être utilisés pour réacheminer l'appel; si un alias est fourni, le champ **alternativeAddress** n'est pas nécessaire.

conferenceID – Identificateur univoque de la conférence; pas nécessaire si le champ **conferences** est utilisé.

reason – Plus de renseignements sur le message FACILITY. Une valeur **featureSetUpdate** de ce champ indique que l'objet du message est de mettre à jour des informations du champ **featureSet** qui ont été envoyées antérieurement. Une valeur **forwardedElements** de ce champ indique que l'objet du message est de renvoyer des éléments d'un autre message si le message considéré ne peut pas être envoyé, comme ce serait le cas si un portier de routage recevait un message CALL PROCEEDING après avoir déjà envoyé ce même message. Un champ **reason** contenant la valeur **transportedInformation** indique que le but du message est de transporter des informations de couche supérieure, par exemple dans le champ **h4501SupplementaryService**; dans ce cas, l'élément **Facility-UUIE** n'est inclus qu'afin de fournir l'identificateur d'appel.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

destExtraCallInfo – Nécessaire pour rendre possibles des appels sur canaux additionnels, c'est-à-dire pour un appel 2×64 kbit/s du côté RCC. Ne doit contenir que des chaînes de chiffres composés manuellement, des numéros E.164 ou des numéros de plan privé. Ne doit pas contenir le numéro du canal initial.

remoteExtensionAddress – Contient l'adresse pseudonyme d'une extrémité appelé dans les cas où cette information est nécessaire pour traverser plusieurs passerelles.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

conferences – Une ou plusieurs conférences qu'il est possible de rejoindre.

h245Address – Adresse spécifique de transport vers laquelle l'extrémité où le portier envoyant le message FACILITY souhaite que le destinataire établisse la voie de signalisation H.245. Noter que ce champ peut être présent lorsqu'une entité de signalisation intermédiaire achemine le champ **h245Address** à partir d'un message CALL PROCEEDING. L'entité réceptrice n'est appelée à lancer les procédures H.245 que lorsque le champ **reason** a la valeur **startH245**.

fastStart – Utilisé seulement dans la procédure de connexion rapide, ce champ **fastStart** prend en charge la signalisation nécessaire pour ouvrir une voie logique. A cette fin, il utilise la structure **OpenLogicalChannel** qui est définie dans l'UIT-T H.245 mais l'expéditeur de cette structure indique les modes qu'il préfère recevoir et émettre, ainsi que les adresses de transport auxquelles il prévoit de recevoir les flux médias. Ce champ est présent dans un message FACILITY lorsqu'un portier de routage l'a reçu dans un message CALL PROCEEDING issu de l'appelé et que cette information est renvoyée à l'appelant. Ce champ ne doit pas être inclus par une extrémité.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

maintainConnection – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

fastConnectRefused – Une extrémité appelée devrait renvoyer cet élément dans tout message jusqu'au message CONNECT inclus lors de l'établissement d'un appel afin d'indiquer qu'elle refuse la procédure de connexion rapide. Ce champ est présent dans un message FACILITY lorsqu'un portier de routage l'a reçu dans un message CALL PROCEEDING issu de l'appelé et que cette information est renvoyée à l'appelant.

serviceControl – Contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées par une extrémité ou par une passerelle (p. ex. pour afficher un menu d'options à l'intention d'un participant à une communication) comme cela est décrit dans l'Annexe K/H.323.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

destinationInfo – Contient un **EndpointType** (type d'extrémité) pour permettre à l'appelant de déterminer si l'appel fait intervenir une passerelle ou non. Ce champ est présent dans un message FACILITY lorsqu'un portier de routage l'a reçu dans un message CALL PROCEEDING issu de

l'appelé et que cette information est renvoyée à l'appelant. Ce champ n'existait pas dans le message FACILITY avant la version 4 de la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message SETUP avec activation du champ **h245SecurityCapability** doit répondre avec le mode de sécurité acceptable correspondant, **h245SecurityMode**, dans le message CALL PROCEEDING, ALERTING, PROGRESS ou CONNECT. Ce champ est présent dans un message FACILITY lorsqu'un portier de routage l'a reçu dans un message CALL PROCEEDING issu de l'appelé et que cette information est renvoyée à l'appelant. Ce champ n'existait pas dans le message FACILITY avant la version 4 de l'UIT-T H.225.0.

7.4.2 Notification (NOTIFY)

Ce message peut être envoyé par une entité H.323. Le traitement à la réception est facultatif.

Se conformer au Tableau 3-8/Q.931, tel que modifié par le Tableau 17.

Tableau 17/H.225.0 – NOTIFY (notification)

Élément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Discriminateur de protocole	M	1
Référence d'appel	M	3
Type de message	M	1
Capacité support	O (Note)	5-6
Indicateur de notification	M	3
Affichage	O	2-82
Utilisateur-utilisateur	M	2-131
NOTE – Inclus pour indiquer un changement de la capacité support.		

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4/H.225.0, l'expéditeur de ce message ne doit pas inclure le champs **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information "Utilisateur-utilisateur".

L'élément Utilisateur-utilisateur contient l'élément **Notify-UUIE** qui est défini dans la syntaxe de messagerie H.225.0. L'élément **Notify-UUIE** contient les champs suivants:

protocolIdentifier – Positionné sur la version de l'UIT-T H.225.0 acceptée.

callIdentifier – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité d'origine, qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

7.4.3 Autres messages

Les messages de commande d'appel qui peuvent comporter les éléments d'information facultatifs fonctionnalité, fonctionnalité étendue et indicateur de notification sont spécifiés au 8.3.

7.5 Temporisations Q.931

Deux temporisations Q.931 doivent être prises en charge:

- le temporisateur "Setup" T303 (voir Tableau 9-1/Q.931 et Tableau 9-2/Q.931) qui définit la durée pendant laquelle l'extrémité appelante doit attendre un message ALERTING, CALL PROCEEDING, CONNECT, RELEASE COMPLETE ou bien un autre message niveau 1 envoyé par l'extrémité appelée après qu'elle a envoyé un message SETUP. Cette temporisation doit être d'au moins 4 s. A noter que certaines applications pourront être utilisées dans des réseaux dont les temps de transmission sont intrinsèquement plus longs (on peut par exemple comparer Internet et un réseau local d'entreprise ou intranet);
- le temporisateur "establishment" T301 (voir Tableau 9-1/Q.931 et Tableau 9-2/Q.931) qui définit le temps après lequel l'extrémité appelante doit cesser d'attendre la réponse de l'extrémité appelée. Cette temporisation est déclenchée après la réception d'un message ALERTING et s'achève normalement sur un message CONNECT ou lorsque l'appelant met fin à la tentative d'appel et envoie un message RELEASE COMPLETE. Cette temporisation doit être d'au moins 180 secondes (3 minutes).

Il convient de noter que les valeurs de ces temporisations côté réseau à commutation par paquets sont les mêmes que celles utilisées dans le RCC.

D'autres temporisateurs peuvent être pris en charge dans le cadre des Recommandations de la série H.450 sur les services complémentaires facultatifs.

7.6 Eléments communs des messages H.225.0

Le présent paragraphe contient une description des structures ASN.1 qui sont utilisées dans plusieurs messages (enregistrement, admission et état). Certains de ces messages peuvent être utilisés dans la partie utilisateur-utilisateur des messages Q.931.

La structure **requestSeqNum** dans les messages est utilisée pour conserver trace des demandes multiples exceptionnelles. Tous les messages de réponse associés (aboutissement ou échec) contiendront la structure **requestSeqNum** correspondante qui est renvoyée avec chaque message. Les messages retransmis devront avoir le même numéro **requestSeqNum**. Le numéro **RequestSeqNum** incrémente de 1 modulo 65536.

La structure **protocolIdentifier** fait partie des séquences recherche, enregistrement et établissement/connexion pour permettre aux parties concernées de déterminer les millésimes des implémentations utilisées.

nonStandardParameter – Ce paramètre est facultatif dans les séquences recherche, enregistrement, établissement/connexion pour permettre aux correspondants concernés de déterminer le statut non standard des extrémités. Un portier ou une passerelle n'est pas tenu de transmettre cette structure **nonStandardData** qu'il ne prend pas en charge ou comprend lorsque cette structure peut gêner son fonctionnement.

La structure **TransportAddress** est destinée à saisir les différents formats de transport et inclut tous les modes spécifiques de transport outre la référence locale possible à un identificateur TSAP.

L'octet le plus significatif des adresses IPv4 et IPv6 doit être le premier octet de la chaîne d'octets, par exemple le "130" de l'adresse IPv4 de classe B 130.1.2.97 doit être codé dans le premier octet de la chaîne d'octets, suivi du "1" et ainsi de suite.

Le "a1" de l'adresse IPv6 a148:2:3:4:a:b:c:d doit être codé dans le premier octet, le "48" dans le deuxième, le "00" dans le troisième, le "02" dans le quatrième et ainsi de suite.

Une adresse **TransportAddress** de type **ipSourceRoute** dans laquelle la SEQUENCE **route** ne comporte aucune indication doit être assimilée à une adresse du type **ipAddress** qui contient les mêmes valeurs pour **ip** et **port**.

L'octet le plus significatif de chaque champ des adresses IPX **node**, **netnum** et **port** doit être le premier octet de la chaîne d'octets.

Il convient de noter que ces structures n'utilisent pas d'adresse de transport = langage "adresse de réseau à commutation par paquets plus identificateur TSAP" de l'UIT-T H.323. On utilise seulement les termes communs à chaque domaine de transport.

La structure **EndpointType** achemine les informations concernant l'entité H.323 à l'extrémité du canal sémaphore. L'entité H.323 remplirait un ou plusieurs des éléments de message **gatekeeper**, **gateway**, **mcu** ou **terminal**. Si l'entité H.323 comporte un contrôleur multipoint, le booléen **mc** a la valeur Vrai. Le paragraphe 6.3/H.323 décrit la représentation d'un pont MCU lorsque celui-ci est copositionné avec une passerelle; dans ce cas, le dispositif H.323 peut contenir à la fois les éléments **gateway** et **mcu** dans sa définition de type d'extrémité **EndpointType**. La présence de l'élément **set** indique que l'entité est un dispositif (SET, *simple endpoint type*) (type d'extrémité simple) tel que défini dans l'Annexe F/H.323. Les positions binaires à l'intérieur de cet élément indiquent le type de dispositif SET; leur signification est définie dans l'Annexe F/H.323 et dans d'autres Recommandations qui spécifient les types de dispositifs SET. Le champ **supportedTunnelledProtocols** fournit une liste par priorités (décroissantes) des protocoles mis en tunnel qui sont pris en charge.

La structure **TunnelledProtocol** désigne un protocole de signalisation mis en tunnel comme décrit par exemple aux M.1/H.323 et M.2/H.323. Le champ **tunnelledProtocolObjectID** est un identificateur d'objet qui désigne le protocole mis en tunnel. La structure **TunnelledProtocolAlternateID** offre un autre format d'identificateur. Le champ **subIdentifieur** permet la spécification d'une version particulière d'un protocole normalisé.

La structure **TunnelledProtocolAlternateIdentifieur** offre un format d'identificateur de type chaîne pour un protocole mis en tunnel. Le champ **protocolType** indique le type général de protocole, comme ISUP. Le champ **protocolVariant** indique une variante spécifique de cette norme, comme ANSI.

Les protocoles en tunnel définis dans la présente Recommandation sont indiqués dans les Tableaux VI.1 et VI.2. Noter que la mise en tunnel n'est pas limitée aux protocoles énumérés dans ces tableaux.

La structure **GatewayInfo** contient un élément **protocol** qui permet à la passerelle d'indiquer les protocoles qu'elle prend en charge.

La structure **SupportedProtocols** indique un choix de protocoles avec lesquels une entité H.323 a la capacité d'interfonctionner. Par exemple, la sélection de l'option **h310** indique que l'entité assure l'interfonctionnement avec le format H.310.

Dans chaque structure de capacité de protocole prise en charge (**H310Caps**, **H320Caps**, etc.), l'élément **dataRatesSupported** indique les débits binaires assurés pour chaque protocole pris en charge par le dispositif. L'élément **supportedPrefixes** indique les préfixes associés à un protocole pris en charge et, dans certains cas, également associés aux débits.

La structure **McuInfo** contient un élément **protocol** qui permet au pont MCU d'indiquer les protocoles qu'il prend en charge.

La structure **CapacityReportingCapability** indique l'aptitude d'une extrémité à communiquer des informations de capacité d'appel.

La structure **CapacityReportingSpecification** indique les informations de capacité d'appel qu'une extrémité est appelée à signaler. La structure **callStart** indique une demande d'information de capacité au début de l'appel (c'est-à-dire dans le message ARQ ou SETUP). La structure **callEnd** indique une demande d'informations de capacité à la fin de l'appel (c'est-à-dire dans le message DRQ ou RELEASE COMPLETE). Une séquence **when** vide indique une demande de non-communication des informations de capacité d'appel par l'extrémité.

La structure **CallCapacityInfo** permet à une extrémité d'indiquer sa capacité d'acceptation d'appel pour chaque type d'appel pris en charge par cette extrémité. Elle représente donc l'état de repos actuel de l'extrémité. Par exemple, dans une passerelle vocale, la structure **CallCapacityInfo** représentera le nombre de circuits au repos.

La structure **CallCapacity** permet à une extrémité d'indiquer sa capacité d'acceptation d'appels pour chaque type d'appel ainsi que sa capacité actuellement disponible pour chaque type d'appel qu'elle prend en charge.

La structure **CallsAvailable** représente un sous-ensemble de la capacité totale d'appel de l'extrémité. Le champ **group** permet d'identifier le sous-ensemble par une étiquette de groupe. Le champ **group** peut être celui qui a été signalé dans la structure **CircuitIdentifier**.

La structure **DataRate** donne des informations sur les débits associés aux protocoles de la passerelle. **channelRate** est le débit de base d'un canal en centaines de bits. **channelMultiplier** indique le nombre de canaux dont le débit vaut **channelRate**. Par exemple, si une passerelle prend en charge un appel 3B, **channelMultiplier** = 3 et **channelRate** = 640 pour un canal à 64 kbit/s.

La structure **VendorIdentifier** permet à un vendeur d'identifier un produit. L'élément **vendor** permet une identification en termes d'indicatif de pays, d'extension et de code de fabricant. **productId** et **versionId** sont des chaînes de texte qui peuvent renseigner sur les produits.

La structure **h221NonStandard** permet de définir un champ non normalisé. L'élément **t35CountryCode** doit désigner le pays, comme décrit dans l'Annexe A/T.35. L'élément **t35Extension** doit contenir une extension d'indicatif de pays qui est attribuée au niveau national, à moins que l'élément **t35CountryCode** n'ait la valeur binaire "1111 1111", auquel cas ce champ doit contenir l'indicatif de pays indiqué dans l'Annexe B/T.35. L'élément **manufacturerCode** doit être attribué au niveau national afin de désigner un équipementier.

La structure **AliasAddress** est destinée à saisir les différents formats d'adresse extérieurs qui référencent un emplacement de transport particulier sur le réseau à commutation par paquets. Lorsqu'une extrémité enregistre chez un portier une adresse composée de chiffres composés manuellement E.164, elle doit utiliser le champ **dialedDigits** et ne doit utiliser que les chiffres 0 à 9. Lorsqu'une extrémité enregistre chez un portier une adresse E.164, elle doit utiliser le champ **e164Number** et ne doit utiliser que les chiffres 0 à 9. Lorsqu'elle enregistre ou représente autrement un préfixe, une extrémité doit utiliser le champ **dialedDigits** et ne doit utiliser que les chiffres 0 à 9 et les caractères '#' et '*'. Le champ **mobileUIM** est un module d'identification pour les systèmes compatibles avec les réseaux sans fil des 2^e et 3^e générations. Il permet l'interfonctionnement avec les réseaux mobiles de télécommunication publics qui sont décrits, par exemple, dans l'Annexe E/H.246.

La structure **AddressPattern** permet de spécifier une adresse pseudonyme **AliasAddress** sous forme d'une structure générique ou une gamme de numéros de correspondant **PartyNumber**. Le champ **wildcard** représente l'extension possible de la structure **AliasAddress** sous une forme générique. Pour les chiffres composés manuellement ou les numéros E.164, cette extension est possible à la fin du numéro. Pour les adresses de courrier électroniques, l'extension est possible à leur début. Par exemple, si la structure générique est "+1 303", la structure pourra représenter un numéro quelconque dans l'indicatif régional de Denver. Le champ **range** de la structure **AddressPattern** représente une étendue d'adresses, y compris l'indication de début et de fin d'étendue.

Les mécanismes qu'une extrémité utilise pour déterminer le type d'adresse doivent être choisis au moment de l'implémentation. La représentation des divers types de numéro dans les messages est donnée dans le Tableau 18. A noter que si une extrémité ne connaît pas le type ou la portée d'une adresse, elle doit normalement la représenter comme un "numéro inconnu de plan privé" lorsqu'elle le code dans les messages Q.931 et comme une adresse pseudonyme contenant des chiffres composés manuellement **dialedDigits AliasAddress** lorsqu'elle le code dans des messages de signalisation RAS.

Tableau 18/H.225.0 – Mappage des représentations des types de numéros

Type de numéro	Représentation Q.931	Représentation par élément d'information H.225.0	H.225.0 UIE representation
Inconnu (par défaut et mode d'interopérabilité avec la version 1)	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	Plan de numérotage privé, Type de numéro = inconnu ("000")	dialedDigits AliasAddress (Note 2)
Numéro inconnu de plan privé	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	dialedDigits AliasAddress (Note 2)
Numéro régional privé, niveau 2	Plan de numérotage privé, Type de numéro = Level 2 Regional Number ("001")	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	privateNumber of PartyNumber AliasAddress, TypeOfNumber = level2RegionalNumber
Numéro régional privé, niveau 1	Plan de numérotage privé, Type de numéro = Level 1 Regional Number ("010")	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	privateNumber of PartyNumber AliasAddress, TypeOfNumber = level1RegionalNumber
Numéro privé propre à un RPIS	Plan de numérotage privé, Type de numéro = PISN specific Number ("011")	Plan de numérotage privé, Type de numéro = inconnu ("000") (Note 1)	privateNumber of PartyNumber AliasAddress, TypeOfNumber = pISNSpecificNumber
Numéro régional privé, niveau 0 (Local)	Plan de numérotage privé, Type de numéro = Level 0 Regional Number ("100")	Plan de numérotage privé, Type de numéro = inconnu ("000")	privateNumber of PartyNumber AliasAddress, TypeOfNumber = localNumber
Numéro E.164 public, inconnu	Plan de numérotage RNIS/téléphonie, Type de numéro = inconnu ("000")	Plan de numérotage RNIS/téléphonie, Type de numéro = inconnu ("000")	e164Number of PartyNumber AliasAddress, TypeOfNumber = Unknown
Numéro E.164 public, numéro international	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro international ("001")	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro international ("001")	e164Number of PartyNumber AliasAddress, TypeOfNumber = internationalNumber
Numéro E.164 public, numéro national	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro national ("010")	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro national ("010")	e164Number of PartyNumber AliasAddress, TypeOfNumber = nationalNumber

Tableau 18/H.225.0 – Mappage des représentations des types de numéros

Type de numéro	Représentation Q.931	Représentation par élément d'information H.225.0	H.225.0 UIIE representation
Numéro E.164 public, numéro propre au réseau	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro propre au réseau ("011")	Plan de numérotage RNIS/téléphonie, Type de numéro = NetworkSpecific Number ("011")	e164Number of PartyNumber AliasAddress, TypeOfNumber = networkSpecificNumber
Numéro E.164 public, numéro d'abonné	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro d'abonné ("100")	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro d'abonné ("100")	e164Number of PartyNumber AliasAddress, TypeOfNumber = subscriberNumber
Numéro E.164 public, numéro abrégé	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro abrégé ("110")	Plan de numérotage RNIS/téléphonie, Type de numéro = numéro abrégé ("110")	e164Number of PartyNumber AliasAddress, TypeOfNumber = abbreviatedNumber
<p>NOTE 1 – Si l'identification du plan de numérotage indique un plan privé, les chiffres du numéro privé sont codés dans le champ privateNumber de la structure PartyNumber, qui inclut le type de numéro. Le champ de type de numéro dans l'élément d'information doit être ignoré à la réception et codé selon ce tableau à l'émission.</p> <p>NOTE 2 – Une structure privateTypeOfNumber = Unknown PartyNumber AliasAddress doit être traitée comme une structure AliasAddress = dialedDigits.</p>			

La structure **MobileUIM** représente un module d'identification pour les systèmes compatibles avec les réseaux sans fil de 2^e et de 3^e générations. Les options offertes sont les suivantes:

- **ansi-41-uim** – Cette option concerne les réseaux sans fil définis par des normes américaines.
- **gsm-uim** – Cette option concerne les réseaux sans fil définis par des normes européennes.

La structure **ANSI-41-UIM** désigne un module d'identification pour les systèmes conformes aux normes américaines pour les réseaux sans fil. Les options offertes sont les suivantes:

- **imsi** – Cette option concerne les numéros internationaux d'identification de station mobile.
- **min** – Cette option concerne les numéros d'identification de station mobile.
- **mdn** – Cette option concerne les numéros d'annuaire du service mobile.
- **msisdn** – Cette option concerne les numéros de station mobile RNIS.
- **esn** – Cette option concerne les numéros de série électronique.
- **mscid** – Cette option concerne les numéros de commutateur du service mobile plus les numéros d'identification de marché ou de système.
- **sid** – Cette option concerne les numéros d'identification de système.
- **mid** – Cette option concerne les numéros d'identification de marché.
- **systemMyTypeCode** – Cette option concerne les numéros d'identification de vendeur.
- **systemAccessType** – Cette option concerne le type d'accès du système.
- **qualificationInformationCode** – Cette option concerne le code d'information de qualification.

- **sesn** – Cette option concerne les numéros de série électronique de module SIM.
- **soc** – Cette option concerne les codes d'opérateur de système.

La structure **GSM-UIM** désigne un module d'identification pour les systèmes conformes aux normes européennes pour les réseaux sans fil. Les options offertes sont les suivantes:

- **imsi** – Cette option concerne l'identification internationale de station mobile.
- **tmsi** – Cette option concerne l'identification temporaire de station mobile.
- **msisdn** – Cette option concerne les numéros de station mobile RNIS.
- **imei** – Cette option concerne les numéros d'identification internationale d'équipements mobiles.
- **hplmn** – Cette option concerne les numéros publics d'abonné d'un réseau mobile terrestre.
- **vplmn** – Cette option concerne les numéros publics d'étranger d'un réseau mobile terrestre.

La structure **ExtendedAliasAddress** permet d'associer des informations communes à des adresses pseudonymes. La structure **presentationIndicator** indique si la présentation de l'**adresse** doit être autorisée ou interdite. La structure **screeningIndicator** indique si l'adresse a été fournie par l'extrémité ou par le réseau et si celui-ci l'a filtrée.

La structure **Endpoint** sert à indiquer des informations de sauvegarde, des informations redondantes ou d'autres informations concernant une extrémité:

- **nonStandardData** – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).
- **aliasAddress** – Liste des adresses pseudonymes, au moyen desquelles d'autres points d'extrémité peuvent identifier l'extrémité considérée.
- **callSignalAddress** – Adresse de transport utilisée par l'extrémité considérée pour la signalisation d'appel.
- **rasAddress** – Adresse de transport utilisée par l'extrémité considérée pour les messages d'enregistrement et d'état.
- **endpointType** – Spécifie le type d'extrémité.
- **tokens** – Jetons associés à cette extrémité (c'est-à-dire au point décrit dans la structure **Endpoint**).
- **cryptoTokens** – Cryptojetons associés à cette extrémité (c'est-à-dire au point décrit dans la structure **Endpoint**).
- **priority** – Utilisé lorsqu'une SEQUENCE de points d'extrémité est présentée. Les extrémités ayant un petit numéro de priorité sont prioritaires par rapport à celles qui ont un grand numéro de priorité. Les extrémités sans numéro de priorité sont équivalentes à celles qui ont la priorité 0 (premier rang de priorité).
- **remoteExtensionAddress** – Élément contenant l'adresse pseudonyme d'une extrémité lorsque cette information est nécessaire pour traverser des passerelles multiples.
- **destExtraCallInfo** – Contient des adresses externes pour les appels multiples.
- **alternateTransportAddresses** – Indique la prise en charge de modes de transport autres que TCP.

La structure **AlternateTransportAddresses** achemine des adresses de signalisation d'appel pour des modes de transport autres que TCP.

La structure **UseSpecifiedTransport** définit un choix de protocoles de transport de signalisation. La valeur **tcp** désigne le protocole TCP. La valeur **annexE** désigne le protocole défini par l'Annexe E/H.323.

La structure **AlternateGK** sert à indiquer une liste de portiers de remplacement ou de secours:

- **rasAddress** – Adresse de transport utilisée pour la signalisation des messages RAS.
- **gatekeeperIdentifier** – Inclus facultativement pour identifier le portier de secours ou de remplacement. Si ce champ est fourni, il devra figurer dans les futurs messages RAS envoyés au portier de secours.
- **needToRegister** – Mis à Vrai pour indiquer que l'extrémité doit s'enregistrer auprès du portier de remplacement avant d'envoyer d'autres demandes RAS.
- **priority** – Indique le rang de priorité du portier de secours ou de remplacement. Plus le numéro est petit, meilleur est le rang de priorité.

La structure **AltGKInfo** sert à donner des renseignements sur les portiers de remplacement:

- **alternateGatekeeper** – Séquence de portiers de remplacement classés par ordre de priorité.
- **altGKisPermanent** – Ce paramètre a la valeur Vrai pour indiquer que tous les futurs signaux RAS doivent normalement être renvoyés vers un portier figurant dans la liste du champ **alternateGatekeeper**; il a la valeur Faux si seul le message qui a causé le rejet doit être renvoyé. Ce fanion doit être mis à la valeur Vrai si un fanion **needToRegister** est mis à Vrai dans le champ **alternateGatekeeper**.

La structure **QseriesOptions** fournit des informations au portier ou aux autres points d'extrémité sur la prise en charge assurée par un terminal des protocoles facultatifs de la série Q. Elle est utilisée dans les messages ARQ, SETUP et RRQ.

Les identificateurs **GloballyUniqueID** et **ConferenceIdentifier** sont supposés être uniques sur le plan mondial (**GloballyUniqueID**); leur utilisation est décrite dans l'UIT-T H.323. L'identificateur **GloballyUniqueID** est codé avec l'octet zéro codé en premier. Il est constitué conformément au Tableau 19.

Tableau 19/H.225.0 – Formation de l'identificateur mondial unique

Champ	Type de données	Octet n°	Note
time_low	Entier non signé sur 32 bits	0-3	Champ de plus faible poids de l'horodate
time_mid	Entier non signé sur 16 bits	4-5	Champ de poids moyen de l'horodate
time_hi_and_version	Entier non signé sur 16 bits	6-7	Champ de plus fort poids de l'horodate multiplexé avec le numéro de version
clock_seq_hi_and_reserved	Entier non signé sur 8 bits	8	Champ de plus fort poids de la séquence d'horloge multiplexé avec la variante
clock_seq_low	Entier non signé sur 8 bits	9	Champ de plus faible poids de la séquence d'horloge
node	Entier non signé sur 48 bits	10-15	Identificateur de nœud unique spatialement

L'identificateur **GloballyUniqueID** est composé d'un enregistrement de 16 octets et ne doit pas contenir de bits de remplissage entre les champs. La taille totale est de 128 bits.

Pour éviter au maximum les confusions concernant les affectations de bits à l'intérieur des octets, l'enregistrement **GloballyUniqueID** n'est défini qu'en termes de champs composés d'un nombre entier d'octets. Le numéro de version est multiplexé avec l'horodate (*time_high*) et le champ de variante est multiplexé avec la séquence d'horloge (*clock_seq_high*).

L'horodate est une valeur de 60 bits représentée en temps universel coordonné (UTC, *coordinated universal time*) sous forme du nombre d'intervalles de 100 ns depuis le 15 octobre 1582 à 00:00:00.00 (date de la réforme grégorienne du calendrier chrétien).

Le numéro de version est multiplexé dans les 4 éléments binaires de plus fort poids du champ *time_hi_and_version* et sa valeur est 1 ("0001" en binaire).

Le champ de variante détermine la présentation de l'identificateur **GloballyUniqueID**. La structure d'un identificateur **GloballyUniqueID** d'ETCD est fixe d'une version à l'autre. Il est possible que d'autres variantes d'identificateur **GloballyUniqueID** ne soient pas compatibles avec un identificateur **GloballyUniqueID** d'ETCD donné. La compatibilité des identificateurs **GloballyUniqueID** est définie comme la possibilité d'appliquer certaines opérations comme la conversion, la comparaison et l'ordonnancement lexicologique de chaînes entre des systèmes différents. Le champ *variant* est composé d'un nombre variable de bits de plus fort poids du champ *clock_seq_hi_and_reserved* (voir le Tableau 20).

Tableau 20/H.225.0 – Contenu du champ de variante d'ETCD

Bit de plus fort poids 1	Bit de plus fort poids 2	Bit de plus fort poids 3	Description
0	–	–	Réservé, rétrocompatibilité NCS
1	0	–	Variante d'ETCD
1	1	0	Réservé, Microsoft Corporation GUID
1	1	1	Réservé pour une définition future

La séquence d'horloge est nécessaire pour détecter les éventuelles pertes de monotonie de l'horloge. Elle est codée dans les 6 bits de plus faible poids du champ *clock_seq_hi_and_reserved* et dans le champ *clock_seq_low*.

Le champ *node* est composé de l'adresse IEEE, généralement l'adresse du serveur. Pour les systèmes comportant plusieurs nœuds IEEE 802, on peut utiliser l'adresse de n'importe quel nœud disponible. L'octet de plus faible poids de l'adresse (octet numéro 10) contient le bit global/local et le bit unidiffusion/multidiffusion et c'est l'octet de l'adresse qui est transmis en premier dans un réseau à commutation par paquets 802.3.

Il convient de modifier la valeur de la séquence d'horloge chaque fois que:

- le générateur d'identificateur **GloballyUniqueID** détecte que la valeur locale du temps UTC a fait marche arrière; cela peut être dû à un fonctionnement normal du service de temps d'ETCD;
- le générateur d'identificateur **GloballyUniqueID** a perdu l'état de sa dernière valeur de temps UTC utilisée, indiquant que le temps a pu faire marche arrière; c'est généralement le cas lors des réinitialisations.

Tant qu'un nœud est opérationnel, le générateur d'identificateur **GloballyUniqueID** sauvegarde toujours le dernier temps UTC utilisé pour créer un identificateur **GloballyUniqueID**. Chaque fois qu'un nouvel identificateur **GloballyUniqueID** est créé, le temps *UTC* courant est comparé à la valeur sauvegardée et si soit la valeur courante est inférieure (cas d'une horloge non monotone) soit la valeur sauvegardée a été perdue, alors la *séquence d'horloge* est incrémentée modulo 16 384, ce qui permet d'éviter la production d'identificateurs **GloballyUniqueID** en double.

La *séquence d'horloge* doit être initialisée à une valeur aléatoire afin de minimiser la corrélation entre systèmes.

Chaque identificateur **GloballyUniqueID** est produit conformément à l'algorithme suivant:

- 1) déterminer les valeurs de l'horodate fondée sur le temps UTC et de la séquence d'horloge à utiliser dans l'identificateur **GloballyUniqueID**;
- 2) positionner le champ *time_low* sur les 32 bits de plus faible poids (bits numérotés de 0 à 31 inclus) de l'horodate en conservant l'ordre de poids;
- 3) positionner le champ *time_mid* sur les bits numérotés de 32 à 47 inclus de l'horodate en conservant l'ordre de poids;
- 4) positionner les 12 bits de plus faible poids (bits numérotés de 0 à 11 inclus) du champ *time_hi_and_version* sur les bits numérotés de 48 à 59 inclus de l'horodate en conservant l'ordre de poids;
- 5) positionner les 4 bits de plus fort poids (bits numérotés de 12 à 15 inclus) du champ *time_hi_and_version* sur les 4 bits de numéro de version correspondant à la version de l'identificateur **GloballyUniqueID** en cours de création, comme représenté au Tableau 20;
- 6) positionner le champ *clock_seq_low* sur les 8 bits de plus faible poids (bits numérotés de 0 à 7 inclus) de la *séquence d'horloge* en conservant l'ordre de poids;
- 7) positionner les 6 bits de plus faible poids (bits numérotés de 0 à 5 inclus) du champ *clock_seq_hi_and_reserved* sur les 6 bits de plus fort poids (bits numérotés de 8 à 13 inclus) de la *séquence d'horloge* en conservant l'ordre de poids;
- 8) positionner les 2 bits de plus fort poids (bits numérotés 6 et 7) du champ *clock_seq_hi_and_reserved* à 0 et 1, respectivement;
- 9) positionner le champ *node* sur les 48 bits de l'adresse IEEE en conservant l'ordre de poids des bits de l'adresse.

Si un système souhaite produire un identificateur **GloballyUniqueID** mais ne dispose pas de carte réseau conforme à la norme IEEE 802 ni d'autre source d'adresses IEEE 802, il convient d'utiliser une autre méthode pour produire une valeur de remplacement pour l'adresse. La solution idéale consiste à obtenir un nombre aléatoire de qualité cryptographique de 47 bits et à l'utiliser dans les 47 bits de plus fort poids de l'identificateur de nœud, le bit de plus faible poids du premier octet de l'identificateur de nœud étant mis à 1. Ce bit est le bit de unidiffusion/multidiffusion, qui ne doit être jamais positionné dans les adresses IEEE 802 obtenues à partir de cartes réseau; par conséquent, il ne pourra jamais y avoir de conflit entre des identificateurs **GloballyUniqueID** produits par des machines avec et sans carte de réseau.

Si un système ne dispose pas de primitive pour produire des nombres aléatoires de qualité cryptographique, alors, dans la plupart des systèmes, il existe généralement un nombre relativement grand de sources génératrices de nombres aléatoires à partir desquelles un nombre aléatoire de qualité cryptographique peut être produit. Ces sources sont propres au système, mais elles comprennent souvent le pourcentage de mémoire utilisé, la taille de la mémoire principale en octets, le volume disponible de la mémoire principale en octets, la taille de l'unité de mémoire à accès direct ou du fichier de permutation en octets, le volume disponible de l'unité de mémoire à accès direct ou du fichier de permutation en octets, la taille totale de l'espace d'adresse virtuelle de l'utilisateur en octets, l'espace total disponible de l'adresse utilisateur en octets, la taille de l'espace disque de l'unité d'initialisation en octets, l'espace disque disponible de l'unité d'initialisation en octets, le temps courant, le temps écoulé depuis la réinitialisation du système, la taille de chacun des fichiers situés dans les divers répertoires du système, etc.

Pour une utilisation dans un texte lisible par l'utilisateur, la représentation sous forme de chaîne d'un identificateur **GloballyUniqueID** est spécifiée sous la forme d'une séquence de champs, certains d'entre eux étant séparés par des tirets simples.

Chaque champ est traité comme un entier et sa valeur est imprimée sous forme de chaîne de chiffres hexadécimaux remplie de zéros, le chiffre de plus fort poids étant placé en premier. Les valeurs

hexadécimales a à f inclus sont représentées en minuscules en sortie et aucune distinction n'est faite entre les minuscules et les majuscules en entrée. La séquence est la même que le type construit **GloballyUniqueID**.

La définition formelle de la représentation sous forme de chaîne d'un identificateur **GloballyUniqueID** est donnée ci-après dans le formalisme BNF étendu:

```

UUID          = <time_low> <hyphen> <time_mid> <hyphen>
               <time_high_and_version> <hyphen>
               <clock_seq_and_reserved>
               <clock_seq_low> <hyphen> <node>
time_low      = <hexOctet> <hexOctet> <hexOctet> <hexOctet>
time_mid      = <hexOctet> <hexOctet>
time_high_and_version = <hexOctet> <hexOctet>
clock_seq_and_reserved = <hexOctet>
clock_seq_low = <hexOctet>
node          = <hexOctet><hexOctet><hexOctet>
               <hexOctet><hexOctet><hexOctet>
hexOctet      = <hexDigit> <hexDigit>p
hexDigit      = <digit> | <a> | <b> | <c> | <d> | <e> | <f>
digit         = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
               "8" | "9"
hyphen        = "-"
a             = "a" | "A"
b             = "b" | "B"
c             = "c" | "C"
d             = "d" | "D"
e             = "e" | "E"
f             = "f" | "F"

```

La chaîne suivante est un exemple de la représentation sous forme de chaîne d'un identificateur **GloballyUniqueID**:

f81d4fae-7dec-11d0-a765-00a0c91e6bf6

timeToLive est un nombre de secondes pendant lesquelles un enregistrement doit être considéré comme valide.

La structure **H248PackagesDescriptor** est un élément **PackagesDescriptor** comme défini dans l'UIT-T H.248, en format binaire.

La structure **H248SignalsDescriptor** est un élément **SignalsDescriptor** comme défini dans l'UIT-T H.248, en format binaire.

La structure **FeatureDescriptor** est un élément **GenericData** qui est utilisé pour identifier de façon générique un élément de service.

La structure **CircuitInfo** – Donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré. Le champ **sourceCircuitID** donne des renseignements sur le circuit source lorsque l'appel provient du RCC. Il peut être utilisé par une passerelle d'entrée pour signaler au portier l'identificateur du circuit source. Le champ **destinationCircuitID** donne des renseignements sur le circuit de destination lorsque l'appel aboutit au RCC. Il peut être utilisé par un portier afin de sélectionner un circuit de destination dans une passerelle d'égression.

La structure **CircuitIdentif** désigne un dispositif permettant un compte rendu par une passerelle ou une sélection par un portier. La structure **CircuitIdentif** prend en charge diverses interfaces.

La structure **CicInfo** désigne des voies supports SS7. Le champ **cic** est le code identificateur de circuit comme défini dans l'UIT-T Q.763. Il est codé avec les bits de plus faible poids dans le premier octet et les bits de plus fort poids dans le dernier octet. Le champ **pointCode** contient le code de point sémaphore comme défini dans l'UIT-T Q.763. Le premier octet de l'élément **pointCode** désigne le réseau (code indicatif de réseau) et les autres octets désignent la valeur du

code de point sémaphore SS7. Les champs **cic** et **pointCode** ont une longueur variable afin d'autoriser des variantes nationales.

La structure **GroupID** désigne un groupe (**group**) physique ou logique ainsi qu'un membre (**member**) (ou ensemble de membres) appartenant à ce groupe. Par exemple, le champ **group** peut désigner une interface physique, alors que le champ **member** peut désigner un certain signal DS0 à cette interface. Si le champ **member** est omis, la passerelle est censée choisir un dispositif disponible dans le groupe spécifié dans l'élément **group**.

La structure **ServiceControlDescriptor** contient des données propres au service ou des références à ces données, destinées à être présentées à l'utilisateur; il peut également contenir d'autres communications de commande de service comme décrit dans l'Annexe K/H.323. Les options suivantes sont offertes:

- **url** – Cette option contient un protocole ou une ressource dont la référence est donnée par une localisation URL.
- **signal** – Cette option contient un élément **SignalsDescriptor** tel que défini dans l'UIT-T H.248, en format binaire. Les éléments facultatifs **streamID** et **notifyCompletion** doivent être omis de la séquence **Signal** dans l'élément **SignalsDescriptor**.
- **nonStandard** – Cette option contient des informations non définies dans la présente Recommandation (par exemple des données privées).
- **callCreditServiceControl** – Cette option contient des informations relatives à la commande de la durée d'une communication et donnant à l'utilisateur des renseignements sur le solde de son compte.

La structure **ServiceControlSession** contient une description d'une session de commande de service comme indiqué dans l'Annexe K/H.323. Elle contient les champs suivants:

- **sessionId** – Nombre entier désignant la session considérée. Ce nombre est unique pour le client. Noter que les identificateurs reçus par différents conduits de signalisation (comme la signalisation RAS et la signalisation d'appel) sont orthogonaux et peuvent se superposer.
- **contents** – Structure de commande de service **ServiceControl** possédant le contenu ou le mécanisme de communication approprié.
- **reason** – Indique s'il s'agit d'une nouvelle session (**open**) ou d'une modification à une session existante (**refresh**) ou si la session va être close par le fournisseur (**close**) et s'il y a lieu de fermer des ressources existantes telles qu'une interface GUI, etc.

La structure **RasUsageInfoTypes** énumère les types d'information sur le taux d'utilisation qui peuvent faire l'objet d'un compte rendu à un portier de la part d'une extrémité. Celle-ci utilise cette structure pour demander des types particuliers d'informations sur le taux d'utilisation. Le champ **nonStandardUsageTypes** permet à un vendeur de faire référence à des types non normalisés d'informations sur le taux d'utilisation. Les champs **startTime** et **endTime** se rapportent aux instants auxquels, respectivement, une communication a commencé et a fini. Le paramètre **terminationCause** se rapporte à la raison pour laquelle la communication a fini.

La structure **RasUsageSpecification** est un gabarit qui permet à un portier de demander des types particuliers d'informations sur le taux d'utilisation à des points spécifiques de la communication. Le champ **when** indique l'instant ou les instants de la communication auxquels l'extrémité est appelée à signaler les informations. Le champ **start** désigne le début de la communication; le champ **end** désigne la fin de la communication et le champ **inIrr** désigne des messages IRR spontanés. Le champ **callStartingPoint** définit l'instant ou les instants de la communication qui doivent être considérés comme le début de la communication aux fins de la signalisation des informations sur le taux d'utilisation; la valeur **connect** se rapporte à l'émission ou à la réception du message Connect et la valeur **alerting** se rapporte à l'émission ou à la réception du message Alerting. Le champ **required** indique les types d'informations sur le taux d'utilisation que l'extrémité est appelée à signaler. Une

structure **RasUsageSpecification** dans le champ **when** ou **required** de laquelle aucune valeur n'est sélectionnée indique une demande de désactivation de la signalisation des informations sur le taux d'utilisation.

La structure **RasUsageInformation** est un ensemble de données de taux d'utilisation se rapportant à une communication particulière. Le champ **nonStandardUsageFields** permet à un vendeur d'énumérer des types non normalisés d'informations sur le taux d'utilisation. Le champ **alertingTime** indique l'instant auquel le message ALERTING a été émis ou reçu. Le champ **connectTime** indique l'instant auquel le message CONNECT a été émis ou reçu. Le champ **endTime** indique l'instant auquel le message RELEASE COMPLETE a été émis ou reçu.

La structure **CallTerminationCause** indique la raison de la fin d'une communication. Le champ **releaseCompleteReason** indique le champ **reason** qui était spécifié dans le message RELEASE COMPLETE. Le champ **releaseCompleteCauseIE** extrait l'élément d'information Cause du message RELEASE COMPLETE.

La structure **BandwidthDetails** définit des informations additionnelles sur le taux d'utilisation de la largeur de bande, qui ne sont pas disponibles dans la structure **BandWidth**. Le champ **sender** est mis à la valeur Vrai si le message est expédié par l'émetteur du flux, ou à la valeur Faux si le message est expédié par le récepteur. Le champ **bandwidth** indique la largeur de bande utilisée pour le flux, en centaines de bits par seconde. Le champ **rtcpAddresses** indique les adresses RTCP utilisées pour le flux média.

La structure **CallCreditCapability** indique certaines capacités d'une extrémité concernant la facturation d'une communication. Par défaut, une extrémité est censée ne pas avoir ces capacités facultatives. Si un champ n'est pas inclus dans cette structure, cela signifie que l'état de la capacité représentée par ce champ n'a pas changé depuis la dernière fois qu'il a été signalé. Le champ **canDisplayAmountString** indique si l'extrémité peut afficher une chaîne alphanumérique contenant le montant en monnaie d'un compte d'utilisateur. Le champ **canEnforceDurationLimit** indique si une extrémité possède la capacité de libérer une communication lorsque la limite de durée de communication indiquée par le portier s'est écoulée.

La structure **CallCreditServiceControl** permet à un portier de fournir à une extrémité certaines commandes et informations relatives à la facturation. Cette structure contient les champs suivants:

- **amountString** – Ce champ indique la quantité d'argent inscrite dans un compte d'utilisateur, p. ex. "10.00\$". La chaîne doit inclure le symbole de monnaie approprié. Noter que les abréviations normalisées des types de monnaie, telles que "USD" pour les dollars des Etats-Unis d'Amérique, sont définies dans l'ISO 4217. Le champ **amountString** doit être codé selon l'ISO/CEI 10646-1 de base (Unicode).
- **billingMode** – Ce champ indique le mode de facturation pour la communication considérée. Le mode **debit** indique que la communication produira des taxes débitées de la quantité d'argent inscrite au compte d'un usager. Le mode **credit** indique que la communication produira des taxes à payer ultérieurement. Une extrémité peut utiliser ces informations pour, par exemple, déterminer le type d'annonce à restituer ou à afficher.
- **callDurationLimit** – Ce champ indique la durée restant disponible pour une communication particulière.
- **enforceCallDurationLimit** – Ce champ indique si l'extrémité est appelée à libérer la communication à l'expiration de la durée indiquée par le champ **callDurationLimit**. L'extrémité doit interpréter l'absence éventuelle de ce champ comme indiquant l'absence de modification de la directive depuis son état antérieur.
- **callStartingPoint** – Ce champ indique l'instant de la communication auquel le chronométrage est appelé à commencer si le mesurage de la durée de communication est assuré par l'extrémité.

La structure **GenericData** se compose d'un champ **id** désignant les données et du champ **parameters** afin d'acheminer les paramètres proprement dits.

La structure **GenericIdentifieur** offre divers moyens pour identifier un objet.

La structure **EnumeratedParameter** fournit un paramètre générique. Elle se compose d'un champ **id** désignant le paramètre et d'un champ **content** acheminant d'éventuelles données associées.

La structure **Content** prend en charge un certain nombre de types de données différentes comme **raw**, **text**, **unicode**, **bool**, **number8**, **number16**, **number32**, **id**, **alias**, **transport**, **compound** et **nested**. Cela permet une définition souple d'un paramètre générique. L'option **raw** permet de choisir un paramètre ou un jeu de paramètres dont la structure de données réelle est définie ailleurs; elle peut par exemple se composer de notation ASN.1 en codage PER ou de données sous forme type-longueur-valeur, ou d'un message encapsulé d'un autre protocole de signalisation.

La structure **FeatureSet** permet à une entité de spécifier des informations génériques sur les éléments de service. L'entité spécifie l'ensemble des éléments de service dont elle a besoin pour l'aboutissement normal de l'appel au moyen du champ **neededFeatures**; l'ensemble des éléments de service qu'elle préfère mais qu'elle n'exige pas au moyen du champ **desiredFeatures**; et l'ensemble des éléments de service qu'elle prend en charge au moyen du champ **supportedFeatures**. L'opérateur BOOLÉEN **replacementFeatureSet** est mis à la valeur Vrai afin d'indiquer que l'ensemble d'éléments de service désigné remplace tout autre ensemble d'éléments de service déjà envoyé; sinon, il prend la valeur Faux.

La structure **TransportChannelInfo** donne des renseignements sur une voie de transport multimédia. Le champ **sendAddress** est l'adresse de transport de l'expéditeur et le champ **recvAddress** est l'adresse de transport du destinataire.

La structure **RTPSession** décrit une session RTP. Elle comporte les champs suivants :

- **rtpAddress** – Ce champ fournit les adresses d'émission et de réception du flux RTP.
- **rtcpAddress** – Ce champ fournit les adresses d'émission et de réception du flux RTCP.
- **cname** – Ce champ fournit le nom CNAME comme spécifié au paragraphe 6 et dans l'Annexe A.
- **ssrc** – Ce champ sert à identifier la source d'un flux RTP, comme décrit dans le paragraphe 6 et dans l'Annexe A.
- **sessionID** – Ce champ fournit l'identificateur de la session RTP en cours, comme décrit dans l'UIT-T H.245.
- **associatedSessionIds** – Ce champ fournit les identificateurs des sessions RTP associées, comme décrit dans l'UIT-T H.245.
- **multicast** – Ce champ indique s'il s'agit d'une session multidiffusée.
- **bandwidth** – Ce champ indique la largeur de bande utilisée pour le flux, en centaines de bits par seconde.

7.7 Prise en charge requise des messages RAS

Le Tableau 21 montre les différents messages RAS qui sont pris en charge par différents types d'extrémité.

Tableau 21/H.225.0 – Statut des messages RAS

Message RAS	Extrémité (émission)	Extrémité (réception)	Portier (émission)	Portier (réception)
GRQ	O			M
GCF		O	M	
GRJ		O	M	
RRQ	M			M
RCF		M	M	
RRJ		M	M	
URQ	O	M	O	M
UCF	M	O	M	O
URJ	O	O	M	O
ARQ	M			M
ACF		M	M	
ARJ		M	M	
BRQ	M	M	O	M
BCF	M (Note 1)	M	M	O
BRJ	M	M	M	O
IRQ		M	M	
IRR	M			M
IACK		O	CM	
INAK		O	CM	
DRQ	M	M	O	M
DCF	M	M	M	M
DRJ	M (Note 2)	M	M	M
LRQ	O		O	M
LCF		O	M	O
LRJ		O	M	O
NSM	O	O	O	O
XRS	M	M	M	M
RIP	CM	M	CM	M
RAI	O			M
RAC		O	M	
SCI	O	O	O	O
SCR	O	O	O	O

M: obligatoire (*mandatory*), O: facultatif (*optional*), F: interdit (*forbidden*), CM: obligatoire conditionnel (*conditionally mandatory*), espace blanc "non applicable".

NOTE 1 – Si un portier envoie un message BRQ demandant un débit plus faible, l'extrémité répondra avec un message BCF si le débit plus faible est pris en charge et avec un message BRJ dans le cas contraire. Si le portier envoie un message BRQ demandant un débit élevé, l'extrémité peut répondre par un message BCF ou BRJ.

NOTE 2 – Lorsqu'un terminal est en communication, il ne doit pas envoyer de message DRJ en réponse à un message DRQ envoyé par son portier.

7.8 Messages de recherche de terminal et de passerelle

Un portier qui reçoit un message GRQ est tenu de répondre avec un message GCF l'autorisant à s'enregistrer. Le message GRJ est un rejet de cette demande indiquant que l'extrémité demandant doit chercher un autre portier.

7.8.1 Message GRQ (demande de portier)

Il convient de noter qu'un message GRQ est envoyé par chaque extrémité logique; ainsi un pont MCU ou une passerelle peut en envoyer plusieurs.

Le message GRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

protocolIdentifier – Identifie le millésime H.225.0 de l'extrémité expéditeur.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

rasAddress – Adresse de transport utilisée par l'extrémité considérée pour les messages d'enregistrement et d'état.

endpointType – Spécifie le ou les types de l'extrémité qui s'enregistre (le bit MC ne doit pas être fixé par lui-même).

gatekeeperIdentifier – Chaîne permettant d'identifier le portier dont le terminal aimerait recevoir l'autorisation d'enregistrement. Une chaîne **gatekeeperIdentifier** manquante ou nulle indique que le terminal recherche tout portier disponible.

callServices – Fournit des informations sur la prise en charge des protocoles facultatifs de la série Q au portier et au terminal appelé.

endpointAlias – Liste d'adresses pseudonymes au moyen desquelles d'autres terminaux peuvent identifier le terminal considéré.

alternateEndpoints – Séquence d'autres points d'extrémité possibles classés par ordre de priorité pour les éléments **rasAddress**, **endpointType** ou **endpointAlias**.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

authenticationCapability – Indique les mécanismes d'authentification pris en charge par l'extrémité.

algorithmOIDs – Indique l'ensemble complet d'algorithmes de chiffrement pris en charge par l'extrémité.

integrity – Indique au destinataire le mécanisme d'intégrité qui doit être appliqué aux messages RAS.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

supportsAltGK – Indique si l'extrémité prend en charge le mécanisme de portier de remplacement.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.8.2 Message GCF (confirmation de portier)

Le message GCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message GRQ.

protocolIdentifier – Identifie le millésime du portier qui accepte la demande.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

gatekeeperIdentifier – Chaîne permettant d'identifier le portier qui envoie le message GCF.

rasAddress – Adresse de transport utilisée par le portier pour les messages d'enregistrement et d'état.

alternateGatekeeper – Séquence d'autres portiers possibles classés par ordre de priorité pour les éléments **gatekeeperIdentifier** et **rasAddress**.

authenticationMode – Indique le mécanisme d'authentification à utiliser. Le portier doit choisir ce mode parmi les mécanismes que l'extrémité a indiqués dans l'élément **authenticationCapability** du message GRQ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

algorithmOID – Indique l'algorithme de chiffrement dont a besoin le portier.

integrity – Indique au destinataire le mécanisme d'intégrité qui doit être appliqué aux messages RAS.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.8.3 Message GRJ (refus de portier)

Le message GRJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message GRQ.

protocolIdentifier – Identifie le millésime du portier qui refuse la demande.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

gatekeeperIdentifier – Chaîne permettant d'identifier le portier qui envoie le message GRJ.

rejectReason – Codes indiquant la cause du rejet du message GRQ par ce portier. Une cause avec la valeur **genericDataReason** indique que la demande a été rejetée à cause d'un élément de service

générique; dans ce cas, des informations complémentaires peuvent être spécifiées dans le champ **genericData**.

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.9 Messages d'enregistrement de terminal et de portier

Le message RRQ est une demande d'enregistrement soumise par un terminal à un portier. Si le portier répond par un message RCF, le terminal doit utiliser le portier qui a répondu pour les appels futurs. Si le portier répond par un message RRJ, le terminal doit chercher un autre portier auprès duquel il pourra s'enregistrer.

7.9.1 Message de demande d'enregistrement RRQ

Le message RRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans toute réponse associée à ce message spécifique.

protocolIdentifier – Identifie le millésime H.225.0 de l'extrémité expéditeur.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

discoveryComplete – Mis à Vrai si l'extrémité demandeur a fait précéder ce message de la procédure de recherche de portier; mis à Faux s'il s'agit d'un enregistrement seulement. Il convient de noter que l'enregistrement peut "devenir caduque avec le temps" et l'extrémité ne réussira pas à envoyer de message RRQ ou ARQ, le code de motif étant respectivement **discoveryRequired** ou **notRegistered**. Cela indique que l'extrémité doit exécuter la procédure de recherche (dynamique ou statique) avant d'envoyer un message RRQ où **discoveryComplete** est mis à Vrai.

callSignalAddress – Adresse de transport utilisée par l'extrémité considérée pour la signalisation d'appel. Si plusieurs transports sont pris en charge, ils doivent être tous enregistrés une fois.

rasAddress – Adresse de transport utilisée par l'extrémité considérée pour les messages d'enregistrement et d'état.

terminalType – Spécifie le ou les types de l'extrémité qui sont enregistrés; il convient de noter que le bit **mc** ne doit pas être fixé par lui-même; le bit **terminal**, **mcu**, **gateway** ou **gatekeeper** doit être également fixé. Si des informations relatives au vendeur (**vendor**) sont fournies, elles doivent être identiques à celles qui sont contenues dans la structure **endpointVendor**. Si le type de terminal est **gateway** ou **mcu**, la valeur facultative **supportedPrefixes** est une liste d'adresses de préfixe au moyen desquelles d'autres extrémités peuvent désigner les protocoles et débits binaires RCC pris en

charge par cette entité. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAlias** et **terminalAliasPattern**. Tous les préfixes pris en charge par l'extrémité doivent être inclus dans chaque message RRQ à moins que l'option **additiveRegistration** ne soit spécifiée, auquel cas les préfixes pris en charge dans un message RRQ doivent être ajoutés à la liste des préfixes en cours d'enregistrement pour l'extrémité. Avec le message RRQ additif, les préfixes pris en charge et déjà enregistrés auprès de cette extrémité doivent être considérés comme restant enregistrés. Noter que les préfixes ne font pas partie d'un numéro d'abonné (**PartyNumber**) (E.164 ou autre format). Afin d'enregistrer un numéro d'abonné (ou une série ou un type de tels numéros), l'extrémité doit utiliser les champs **terminalAlias** et **terminalAliasPattern** qui sont décrits ci-dessous.

terminalAlias – Cette valeur facultative est une liste d'adresses pseudonymes, au moyen desquelles d'autres terminaux peuvent identifier le terminal considéré. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAliasPattern** et **supportedPrefixes**. Si la séquence **terminalAlias** est nulle, une adresse de type **dialedDigits** peut être assignée par le portier et incluse dans le message RCF. Si un identificateur de courrier électronique (**email-ID**) est disponible pour l'extrémité, il doit être enregistré. Il convient de noter que plusieurs adresses pseudonymes peuvent renvoyer aux mêmes adresses de transport. Tous les pseudonymes de l'extrémité que celle-ci souhaite enregistrer doivent figurer dans cette liste à moins que l'option **additiveRegistration** ne soit spécifiée, auquel cas les pseudonymes d'extrémité contenus dans un message RRQ doivent être ajoutés à la liste des pseudonymes en cours d'enregistrement pour l'extrémité.

gatekeeperIdentifier – Chaîne permettant d'identifier le portier auprès duquel le terminal souhaite s'enregistrer.

endpointVendor – Information sur le vendeur de l'extrémité.

alternateEndpoints – Séquence d'autres points d'extrémité possibles classés par ordre de priorité pour les éléments **callSignalAddress**, **rasAddress**, **terminalType** ou **terminalAlias**.

timeToLive – Durée de validité de l'enregistrement, en secondes. Une fois ce temps écoulé, le portier peut considérer l'enregistrement comme périmé.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

keepAlive – Si ce champ est mis à Vrai, il indique que l'extrémité a envoyé ce message RRQ comme un "maintien d'enregistrement". Une extrémité peut envoyer un message RRQ "allégé" composé uniquement des champs **rasAddress**, **keepAlive**, **endpointIdentifier**, **gatekeeperIdentifier**, **tokens** et **timeToLive**. Un portier qui reçoit un message RRQ avec le champ **keepAlive** mis à Vrai doit ignorer les champs autres que **endpointIdentifier**, **gatekeeperIdentifier**, **tokens** et **timeToLive**. Le champ **rasAddress** dans un message RRQ "allégé" ne doit être utilisé par un portier que comme adresse de destination d'un message RRJ lorsque l'extrémité n'est pas enregistrée.

endpointIdentifier – Identificateur d'extrémité (**endpointIdentifier**) fourni par le portier dans le message RCF d'origine.

willSupplyUIEs – Si ce champ est mis à Vrai, il indique que l'extrémité fournira des informations de message Q.931 dans les messages IRR si le portier le demande.

maintainConnection – Si ce champ est mis à Vrai, il indique que l'expéditeur du message est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

alternateTransportAddresses – Ce champ achemine des adresses de signalisation d'appel pour des modes de transport autres que TCP. L'inclusion d'une adresse indique la prise en charge du transport correspondant.

additiveRegistration – S'il est présent, ce champ indique que le message considéré est une demande RRQ "additive", c'est-à-dire que l'extrémité a expédié cette demande RRQ en complément des informations relatives à un enregistrement existant. Une extrémité peut envoyer une demande RRQ additive composée seulement des champs **callSignalAddress**, **rasAddress**, **terminalType**, **terminalAlias**, **terminalAliasPattern**, **alternateEndpoints**, **endpointIdentifier**, **gatekeeperIdentifier** et **tokens**. Un portier recevant un message RRQ contenant le champ **additiveRegistration** doit ignorer les autres champs. Le champ **rasAddress** dans une demande RRQ additive doit être utilisé par un portier comme destination pour le message RRJ subséquent si l'extrémité n'est pas enregistrée ou si les champs **terminalAlias** et/ou **terminalAliasPattern** contreviennent à la politique d'enregistrement du portier.

terminalAliasPattern – Cette valeur facultative est une liste de structures d'adresse spécifiant les pseudonymes et les adresses permettant à d'autres extrémités d'identifier l'extrémité considérée. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAlias** et **supportedPrefixes**. Tous les pseudonymes et toutes toutes adresses de l'extrémité doivent figurer dans chaque demande RRQ à moins que l'option **additiveRegistration** n'ait la valeur Vrai, auquel cas les pseudonymes et adresses d'extrémité se trouvant dans la demande RRQ doivent être ajoutés à la liste des pseudonymes actuellement enregistrés pour l'extrémité considérée.

supportsAltGK – Ce champ indique si l'extrémité prend en charge le mécanisme de portier de remplacement.

usageReportingCapability – Ce champ peut être inclus par l'extrémité afin d'annoncer sa capacité de collecter et de signaler divers types d'informations sur le taux d'utilisation.

multipleCalls – S'il est à Vrai, ce champ indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion sémaphore d'appel.

supportedH248Packages – Ce champ indique une liste de paquetages H.248 pris en charge par l'extrémité considérée.

callCreditCapability – Ce champ indique certaines capacités d'une extrémité concernant la facturation d'une communication.

capacityReportingCapability – Ce champ indique l'aptitude d'une extrémité à communiquer des informations de capacité d'appel.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un moment donné. Lors de l'envoi de ce champ, l'extrémité doit inclure les éléments **maximumCallCapacity** et **currentCallCapacity**.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.9.2 Message RCF (confirmation d'enregistrement)

Le message RCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message RRQ.

protocolIdentifier – Identifie le millésime du portier qui a accepté la demande d'enregistrement.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

callSignalAddress – Matrice d'adresses de transport pour les messages de signalisation d'appel H.225.0; une adresse pour chaque transport auquel le portier répond. Cette adresse inclut l'identificateur TSAP.

terminalAlias – Cette valeur facultative est une liste d'adresses pseudonymes, au moyen desquelles d'autres terminaux peuvent identifier le terminal en question. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAliasPattern** et **supportedPrefixes**. Il spécifie les adresses pseudonymes qui ont été acceptées parmi celles qui ont été proposées dans le message RRQ associé. Si aucune adresse n'a été proposée dans le message RRQ, cette liste indique les pseudonymes attribués par le portier.

gatekeeperIdentifier – Chaîne permettant d'identifier le portier qui a accepté d'enregistrer les terminaux.

endpointIdentifier – Chaîne d'identité du terminal assignée par le portier; on doit la retrouver dans les messages RAS ultérieurs.

alternateGatekeeper – Séquence d'autres éléments possibles classés par ordre de priorité pour les éléments **gatekeeperIdentifier** et **rasAddress**.

timeToLive – Durée de validité de l'enregistrement, en secondes. Une fois ce temps écoulé, le portier peut considérer l'enregistrement comme périmé.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

willRespondToIRR – Vrai si le portier envoie un message IACK ou INAK en réponse à un message IRR non sollicité dont le champ **needsResponse** est mis à Vrai.

preGrantedARQ – Indique les événements pour lesquels le portier a préaccordé l'admission. Cela permet d'établir plus rapidement les appels dans les environnements où l'admission est garantie par des moyens autres que l'échange des messages ARQ/ACF. A noter que même si ces champs sont mis à Vrai, une extrémité peut toujours envoyer un message ARQ au portier pour certains motifs (traduction d'une adresse, non-prise en charge par l'extrémité du mode de signalisation modifié, etc.). Si la séquence **preGrantedARQ** est absente, il faut alors utiliser la signalisation ARQ dans tous les cas. Les fanions sont les suivants:

- **makeCall** – Si le fanion **makeCall** est à Vrai, le portier a préaccordé à l'extrémité l'autorisation de lancer des appels sans avoir à envoyer d'abord un message ARQ. Si le fanion **makeCall** est à Faux, l'extrémité doit toujours envoyer un message ARQ pour obtenir l'autorisation de lancer un appel.
- **useGKCallSignalAddressToMakeCall** – Si les fanions **makeCall** et **useGKCallSignalAddressToMakeCall** sont tous les deux à Vrai, alors si l'extrémité n'envoie pas de message ARQ au portier pour lancer un appel, il doit envoyer toute la signalisation d'appel H.225 au canal de signalisation d'appel du portier.

- **answerCall** – Si le fanion **answerCall** est à Vrai, le portier a préaccordé à l'extrémité l'autorisation de répondre aux appels sans avoir à envoyer d'abord un message ARQ. Si le fanion **answerCall** est à Faux, l'extrémité doit toujours envoyer un message ARQ pour obtenir l'autorisation de répondre à un appel.
- **useGKCallSignalAddressToAnswer** – Si les fanions **answerCall** et **useGKCallSignalAddressToAnswer** sont tous les deux mis à Vrai, alors lorsqu'une extrémité n'envoie pas de message ARQ au portier pour répondre à un appel, il doit s'assurer que toute la signalisation d'appel H.225.0 provient du portier. Si une extrémité est chargée d'utiliser le portier pour répondre, mais qu'il ne sait pas si un appel entrant est arrivé du portier (pour cela, il faut peut-être regarder l'adresse de transport), il doit envoyer un message ARQ indépendamment de l'état du fanion **useGKCallSignalAddressToAnswer**.
- **irrFrequencyInCall** – Indique la fréquence, en secondes, des messages IRR envoyés au portier lorsque l'extrémité participe à une ou plusieurs communications. S'il n'est pas présent, le portier ne veut pas de messages IRR non sollicités. Lorsque l'extrémité envoie ces messages IRR, la valeur de référence d'appel doit être unique pour le terminal, du fait qu'elle aura été émise dans une demande d'admission. Toutefois, il ne s'agit pas d'une valeur CRV "normale" car elle ne peut pas être réutilisée pour une autre communication (DRQ, IRQ ou BRQ). L'identificateur d'appel doit être le même que celui qui est utilisé dans les messages de canal de signalisation d'appel pour l'appel en question.
- **totalBandwidthRestriction** – Ce champ limite l'utilisation de la largeur de bande à la seule extrémité lorsque celui-ci participe à des communications. S'il n'est pas présent, il n'y a pas de limitation constante de la largeur de bande.
- **alternateTransportAddresses** – Ce champ achemine des adresses de signalisation d'appel pour des modes de transport autres que TCP. L'inclusion d'une adresse indique la prise en charge du transport correspondant.
- **useSpecifiedTransport** – Ce champ permet au portier d'indiquer à l'extrémité le protocole de transport de signalisation à utiliser pour établir des communications. Si ce champ est inclus et que le transport spécifié ne soit pas **tcp**, le champ **alternateTransportAddresses** doit également être inclus dans le message considéré.

maintainCorrection – S'il est à Vrai, ce champ indique que le portier (en cas de routage par portier) est en mesure de prendre en charge une connexion sémaphore lorsque aucun appel n'est en cours de signalisation sur la connexion.

serviceControl – Contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées par l'extrémité pour une communication avec le réseau de commande de service non associée à l'appel comme cela est décrit dans l'Annexe K/H.323.

supportsAdditiveRegistration – S'il est à Vrai, ce champ indique que le portier prend en charge les capacités d'enregistrement additif. S'il est absent, le portier ne prend pas en charge l'enregistrement additif.

terminalAliasPattern – Cette valeur facultative est une liste de structures d'adresse spécifiant les pseudonymes et les adresses permettant à d'autres extrémités d'identifier l'extrémité considérée. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAlias** et **supportedPrefixes** – Il spécifie les pseudonymes et les adresses qui ont été acceptés parmi ceux qui ont été dans le message RRQ associé. Si aucun pseudonyme n'a été proposé dans le message RRQ, cette liste indique les pseudonymes et adresses attribués par le portier.

supportedPrefixes – Cette valeur facultative indique une liste de préfixes permettant à d'autres extrémités d'identifier l'extrémité considérée. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAlias** et **terminalAliasPattern**. Il spécifie les préfixes d'adresse qui ont été acceptés parmi ceux qui ont été proposés dans le message RRQ associé. Si aucun préfixe n'a été proposé dans le message RRQ, cette liste indique les préfixes attribués par le portier.

usageSpec – Ce champ peut être inclus par le portier afin de demander à l'extrémité de collecter et de signaler à des instants spécifiés les informations indiquées de taux d'utilisation.

featureServerAlias – Ce champ est réservé pour utilisation future par l'UIT-T pour un protocole fondé sur un stimulus.

capacityReportingSpec – Ce champ indique le type d'informations de capacité d'appel qu'une extrémité est appelée à signaler.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.9.3 Message RRJ (refus d'enregistrement)

Le message RRJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message RRQ.

protocolIdentifiant – Identifie le millésime du portier qui refuse la demande d'enregistrement.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

rejectReason – Motif du refus d'enregistrement. Ce champ peut contenir une valeur **invalidTerminalAliases**, auquel cas il contient également une liste de pseudonymes, d'adresses et de préfixes pris en charge qui ont été déterminés comme étant invalides dans le message RRQ associé. Une cause avec la valeur **genericDataReason** indique que la demande a été rejetée à cause d'un élément de service générique; dans ce cas, des informations complémentaires peuvent être spécifiées dans le champ **genericData**.

gatekeeperIdentifiant – Chaîne permettant d'identifier le portier qui a refusé d'enregistrer le terminal.

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.10 Messages d'annulation d'enregistrement de terminal/portier

7.10.1 Message de demande d'annulation d'enregistrement (URQ)

Le message URQ demande la rupture de l'association entre un terminal et un portier. Il convient de noter que l'annulation d'enregistrement est bidirectionnelle, c'est-à-dire qu'un portier peut demander à un terminal de considérer que son enregistrement est annulé et un terminal peut informer un portier qu'il renonce à un enregistrement antérieur.

Le message URQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans toute réponse associée à ce message spécifique.

callSignalAddress – Une ou plusieurs des adresses de transport utilisées par l'extrémité considérée pour la signalisation de l'appel, dont l'enregistrement doit être annulé.

endpointAlias – Cette valeur facultative est une liste d'adresses pseudonymes au moyen desquelles d'autres terminaux peuvent identifier le terminal considéré. Ce champ peut être utilisé en complément ou en remplacement des champs **endpointAliasPattern** et **supportedPrefixes**. Si ce champ, le champ **endpointAliasPattern** et le champ **supportedPrefixes** ne sont pas présents, tous les pseudonymes font l'objet d'une annulation d'enregistrement dans un seul message. La chaîne de chiffres composés manuellement **dialedDigits**, si elle est assignée, est requise. Seules les valeurs énumérées ici font l'objet d'une annulation d'enregistrement; cela permet, par exemple, d'annuler l'enregistrement d'un identificateur **h323-ID** tout en conservant l'enregistrement de la chaîne de chiffres composés manuellement **dialedDigits**.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

endpointIdentifier – Confirmation d'identité; n'est pas envoyée par le portier.

alternateEndpoints – Séquence d'autres points d'extrémité possibles classés par ordre de priorité pour les éléments **callSignalAddress** ou **endpointAlias**.

gatekeeperIdentifier – Identificateur de portier **gatekeeperIdentifier** que l'extrémité a reçu dans la liste **alternateGatekeeper** d'un message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message URJ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

reason – Utilisé lorsque le portier envoie le message URQ pour indiquer les raisons pour lesquelles il considère que l'enregistrement de l'extrémité est annulé. Une valeur **maintenance** du champ **reason** indique que le portier ou l'extrémité va être mis hors service pour maintenance.

endpointAliasPattern – Cette valeur facultative est une liste de structures d'adresse spécifiant les pseudonymes et les adresses permettant à d'autres extrémités d'identifier l'extrémité considérée. Ce champ peut être utilisé en complément ou en remplacement des champs **endpointAlias** et **supportedPrefixes** – Si ce champ, le champ **endpointAlias** et le champ **supportedPrefixes** ne sont pas présents, tous les pseudonymes et toutes les adresses font l'objet d'une annulation d'enregistrement dans un seul message. Sinon, seules les valeurs énumérées ici font l'objet d'une annulation d'enregistrement.

supportedPrefixes – Cette valeur facultative indique une liste de préfixes permettant à d'autres extrémités d'identifier l'extrémité considérée. Ce champ peut être utilisé en complément ou en remplacement des champs **terminalAlias** et **terminalAliasPattern**. Si ce champ, le champ **endpointAlias** et le champ **endpointAliasPattern** ne sont pas présents, tous les pseudonymes et toutes les adresses font l'objet d'une annulation d'enregistrement dans un seul message. Sinon, seules les valeurs énumérées ici font l'objet d'une annulation d'enregistrement.

alternateGatekeeper – Séquence d'autres portiers possibles classés par ordre de priorité pour les éléments **gatekeeperIdentifier** et **rasAddress**.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.10.2 Message UCF (confirmation d'annulation d'enregistrement)

Le message UCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message URQ.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.10.3 Message URJ (refus d'annulation d'enregistrement)

Le message URJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message URQ.

rejectReason – Motif du refus d'annulation de l'enregistrement.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.11 Messages d'admission du terminal au portier

Le message ARQ demande l'attribution à une extrémité d'un accès au réseau à commutation par paquets par le portier, qui accepte la demande avec un message ACF ou la refuse avec un message ARJ.

7.11.1 Message de demande d'admission (ARQ, *admission request*)

Le message ARQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

callType – Lorsqu'il utilise cette valeur, le portier peut essayer de déterminer la largeur de bande réellement utilisée. La valeur par défaut est **pointToPoint** pour tous les appels. Il convient d'admettre qu'un type d'appel peut être modifié dynamiquement au cours de l'appel et que le type d'appel définitif peut être inconnu lorsque le message ARQ est envoyé.

callModel – Si sa valeur est **direct**, l'extrémité demande le modèle d'appel direct terminal à terminal. Si sa valeur est **gatekeeperRouted**, l'extrémité demande le modèle avec intervention du portier. Le portier n'est pas tenu de se conformer à cette demande.

endpointIdentifiant – Identificateur d'extrémité qui a été assigné au terminal par le message RCF.

destinationInfo – Séquence d'adresses pseudonymes pour la destination, telles que des champs **dialedDigits**, **PartyNumber** (**e164Number** ou **privateNumber**) ou des identificateurs **h323-ID**. Lors de l'envoi du message ARQ en réponse à un appel, **destinationInfo** indique la destination de l'appel (extrémité qui répond). Si au moins une adresse pseudonyme est enregistrée auprès d'un portier et si le message ARQ ne comporte pas deux adresses pseudonymes enregistrées auprès de personnes différentes, le portier doit reconnaître le message ARQ comme se rapportant à l'identité enregistrée. Dans le cas d'adresses pseudonymes incompatibles, il convient de refuser la demande d'admission en indiquant comme motif **AliasesInconsistent**.

destCallSignalAddress – Adresse de transport utilisée à la destination pour la signalisation d'appel.

destExtraCallInfo – Contient les adresses externes pour les appels multiples.

srcInfo – Séquence d'adresses pseudonymes pour l'extrémité source, telles que des champs **dialedDigits**, **PartyNumber** (**e164Number** ou **privateNumber**) ou des identificateurs **h323-ID**. Lors de l'envoi du message ARQ en réponse à un appel, **srcInfo** indique qui est à l'origine de l'appel.

srcCallSignalAddress – Adresse de transport utilisée à la source pour la signalisation d'appel.

bandWidth – Largeur de bande bidirectionnelle demandée pour la communication, exprimée en multiples de 100 bit/s. Par exemple, un appel à 128 kbit/s doit être signalé dans une demande de 256 kbit/s. Cette valeur ne concerne que le débit audio et vidéo à l'exclusion des en-têtes et des surdébits.

callReferenceValue – Valeur CRV/Q.931 pour cet appel; n'a qu'une validité locale. Elle est utilisée par le portier pour associer le message ARQ à un appel particulier.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

callServices – Fournit des informations sur la prise en charge des protocoles facultatifs de la série Q au portier et au terminal appelé.

conferenceID – Identificateur univoque de la conférence.

activeMC – Si sa valeur est à Vrai, l'appelant incorpore un contrôleur multipoint activé; dans le cas contraire, sa valeur est Faux.

answerCall – Sert à indiquer l'arrivée d'un appel à un portier.

canMapAlias – Si sa valeur est à Vrai, indique que si le message ACF résultant contient les champs **destinationInfo**, **destExtraCallInfo** et/ou **remoteExtensionAddress**, l'extrémité doit copier ces informations respectivement dans le champ **destinationAddress**, **destExtraCallInfo** ou **remoteExtensionAddress** du message SETUP, ou dans l'élément d'information Numéro de l'appelé, s'il y a lieu. Si l'extrémité est une passerelle utilisée pour sortir du réseau H.323, la passerelle convertira les informations de destination dans le format de signalisation approprié qui est utilisé à l'extérieur du réseau H.323 (par exemple en tonalités DTMF). Si le portier remplaçait les informations d'adressage provenant du message ARQ et que **canMapAlias** soit à Faux, le portier devrait refuser le message ARQ. Les systèmes conformes à la présente version 4 de l'UIT-T H.225.0 et aux versions ultérieures doivent mettre ce champ à Vrai.

callIdentifieur – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

srcAlternatives – Séquence d'autres points d'extrémité sources possibles classés par ordre de priorité pour les éléments **srcInfo**, **srcCallSignalAddress** ou **rasAddress**.

destAlternatives – Séquence d'autres points d'extrémité de destination possibles classés par ordre de priorité pour les éléments **destinationInfo** ou **destCallSignalAddress**.

gatekeeperIdentifieur – Identificateur de portier (**gatekeeperIdentifieur**) que l'extrémité a reçu dans la liste **alternateGatekeeper** d'un message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message ARJ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

transportQOS – Une extrémité peut utiliser cet élément pour indiquer sa capacité à réserver des ressources de transport. La structure TransportQOS comprend ce qui suit:

- **endpointControlled** – L'extrémité appliquera son propre mécanisme de réservation.
- **gatekeeperControlled** – Le portier effectuera la réservation de ressources au nom de l'extrémité.
- **noControl** – Aucune réservation de ressources n'est nécessaire.

willSupplyUIEs – Si sa valeur est à Vrai, indique que l'extrémité fournira des informations de message Q.931 dans les messages IRR si le portier le demande.

callLinkage – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir le paragraphe 10/H.323.

gatewayDataRate – Débit binaire demandé du côté RCC d'un appel passant par une passerelle. Ce débit binaire doit, le cas échéant, être égal au débit binaire spécifié dans l'élément d'information Capacité support du message SETUP. Un portier peut utiliser ce champ lors de la sélection d'une passerelle afin de gérer l'appel.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un moment donné, à condition que le portier confirme la demande ARQ en envoyant un message ACF. Lors de l'envoi de ce champ, l'extrémité doit inclure un élément **currentCallCapacity**.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

desiredProtocols – Ce champ identifie, par ordre de préférence, les types de protocole recherchés par l'extrémité d'origine pour son appel (p. ex. voix ou télécopie). Une entité de résolution peut utiliser ce champ pour localiser une extrémité qui prend également en charge le protocole, compte tenu de l'ordre de préférence.

desiredTunnelledProtocol – Ce champ désigne un protocole dont la canalisation en tunnel est demandée.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

NOTE – La présence simultanée des deux séquences **destinationInfo** et **destCallSignalAddress** est facultative, mais une de ces séquences au moins doit être présente sauf si l'extrémité répond à un appel. Il n'existe pas de règle absolue indiquant quelle séquence est préférée étant donné que cela peut dépendre du site; néanmoins, l'adresse doit être fournie si elle est disponible. Les meilleurs résultats doivent être obtenus en considérant la nature des protocoles de transport utilisés.

7.11.2 Message ACF (confirmation d'admission)

Le message ACF comprend ce qui suit:

requestSeqNum – Aura la même valeur que celle qui a été transmise dans le message ARQ.

bandWidth – Largeur de bande maximale attribuée pour l'appel; peut être inférieure à celle qui a été demandée.

callModel – Indique au terminal si la signalisation d'appel envoyée à l'adresse **destCallSignalAddress** est destinée à un portier ou à un terminal. La valeur **gatekeeperRouted** indique que la signalisation d'appel transite par le portier alors que la valeur **direct** indique que le mode d'appel extrémité à extrémité est utilisé.

destCallSignalAddress – Adresse de transport utilisée à laquelle doit être envoyée la signalisation d'appel Q.931, mais peut être une adresse d'extrémité ou de portier selon le modèle d'appel utilisé.

irrFrequency – Fréquence, en secondes, à laquelle l'extrémité doit envoyer des messages IRR au portier pendant qu'il est en phase d'appel ou en phase de maintien. Si la fréquence est absente, l'extrémité n'envoie pas de message IRR pendant la phase active d'un appel et on s'attend à ce que le portier interroge l'extrémité.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

destinationInfo – Adresse du canal initial, utilisé lorsque l'appel traverse une passerelle.

destExtraCallInfo – Nécessaire pour rendre possibles des appels sur canaux additionnels, c'est-à-dire pour un appel 2×64 kbit/s du côté RCC. Ne doit contenir que les adresses de type **dialedDigits** ou **PartyNumber** et ne doit pas contenir le numéro du canal initial.

destinationType – Spécifie le type de l'extrémité de destination.

remoteExtensionAddress – Contient l'adresse pseudonyme d'une extrémité appelée dans les cas où cette information est nécessaire pour traverser plusieurs passerelles.

alternateEndpoints – Séquence d'autres points d'extrémité possibles classés par ordre de priorité pour les éléments **destCallSignalAddress** ou **destinationInfo**.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

transportQOS – Le portier peut indiquer à l'extrémité qui a la responsabilité de la réservation des ressources. Si le portier reçoit un élément **TransportQOS** dans un message ARQ, il doit alors inclure l'élément **transportQOS** (éventuellement modifié conformément à l'implémentation du portier) dans le message ACF.

willRespondToIRR – Vrai si le portier envoie un message IACK ou INAK en réponse à un message IRR non sollicité lorsque le champ **needsResponse** du message IRR est à Vrai.

uuiesRequested – Sur demande du portier, l'extrémité peut devoir informer le portier des messages de signalisation d'appel H.225.0 qu'il envoie ou reçoit s'il a indiqué cette capacité dans le message ARQ en positionnant **willSupplyUIEs** sur Vrai. **uuiesRequested** indique l'ensemble des messages de signalisation d'appel H.225.0 que l'extrémité doit signaler au portier.

language – Indique le ou les langages dans lesquels l'utilisateur souhaiterait de préférence recevoir les annonces et les invites. Ce champ contient une ou plusieurs étiquettes de langage conformes au document RFC 1766.

alternateTransportAddresses – Ce champ achemine des adresses de signalisation d'appel pour des modes de transport autres que TCP. L'inclusion d'une adresse indique la prise en charge du transport correspondant.

useSpecifiedTransport – Ce champ permet au portier d'indiquer à l'extrémité le protocole de transport de signalisation à utiliser pour établir des communications. Si ce champ est inclus et que le transport spécifié ne soit pas **tcp**, le champ **alternateTransportAddresses** doit également être inclus dans le message considéré.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré. Par exemple, il permet à une passerelle d'entrée de signaler les ressources RCC particulières à utiliser pour l'appel.

usageSpec – Ce champ peut être inclus par le portier afin de demander à l'extrémité de collecter et de signaler à des instants spécifiés les informations indiquées de taux d'utilisation.

supportedProtocols – Ce champ indique les protocoles pris en charge par l'extrémité de destination.

serviceControl – Ce champ contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées par une extrémité (p. ex. un message à restituer à l'appelant) comme cela est décrit dans l'Annexe K/H.323.

multipleCalls – S'il est mis à Vrai, ce champ indique que l'extrémité de destination possède la capacité de signaler plusieurs appels dans une même connexion sémaphore d'appel. S'il a la valeur Faux, l'extrémité de destination ne possède pas cette capacité. Si ce champ n'est pas présent, le portier ne sait pas si l'extrémité distante possède cette capacité.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.11.3 Message ARJ (refus d'admission)

Le message ARJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message ARQ.

rejectReason – Il s'agit de la cause du refus de la demande d'admission. Il convient de noter que le choix de **routeCallToSCN** comme motif du refus **rejectReason** n'est approprié que lorsque le message ARJ est acheminé vers une passerelle d'entrée (le message ARQ a été envoyé par une passerelle et la valeur booléenne **answerCall** du message ARQ est Faux). Si la cause du refus **rejectReason** est **routeCallToSCN**, la cause de refus **rejectReason** pour ce choix concerne également un numéro de téléphone ou une liste de numéros de téléphone vers lesquels la passerelle peut réacheminer l'appel dans le réseau RCC, si la passerelle autorise une telle procédure. Si le champ **rejectReason** a la valeur **exceedsCallCapacity**, le portier a déterminé que la destination ne possède pas la capacité d'accepter cet appel à cet instant. Si le champ **rejectReason** a la valeur **collectDestination**, cela indique que le portier demande que la passerelle collecte l'adresse de destination finale, et que le champ **serviceControl** du message ARJ indique l'invitation à présenter à l'utilisateur. Si le champ **rejectReason** a la valeur **collectPIN**, cela indique que le portier demande que la passerelle collecte un numéro d'identification personnel ou un code d'autorisation, et que le champ **serviceControl** du message ARJ indique l'invitation à présenter à l'utilisateur. Une cause avec la valeur **genericDataReason** indique que la demande a été rejetée à cause d'un élément de service générique; dans ce cas, des informations complémentaires peuvent être spécifiées dans le champ **genericData**.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

callSignalAddress – Adresse de signalisation d'appel du portier renvoyée lorsque la cause du rejet est **routeCallToGatekeeper**.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

serviceControl – Ce champ contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées par une extrémité (p. ex. pour afficher la raison de l'échec d'un appel) comme cela est décrit dans l'Annexe K/H.323.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.12 Demandes de modification de largeur de bande émises par le terminal à l'intention du portier

Le message BRQ est une demande adressée au portier de modification de la largeur de bande du réseau à commutation par paquets, le portier accède à la demande par un message BCF ou la refuse par un message BRJ.

Le portier peut, au moyen d'un message BRQ, demander à ce qu'une extrémité augmente ou diminue la largeur de bande utilisée. S'il s'agit d'une augmentation, l'extrémité peut répondre au moyen d'un message BRJ ou BCF, s'il s'agit d'abaisser le débit, l'extrémité doit répondre par un message BCF si le débit plus bas est pris en charge, sinon avec BRJ.

7.12.1 Message BRQ (demande de largeur de bande)

Le message BRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentif – Identificateur d'extrémité qui a été attribué au terminal par un message RCF.

conferenceID – Identificateur de l'appel pour lequel la largeur de bande doit être modifiée.

callReferenceValue – Valeur CRV Q.931 pour cet appel; sa validité est uniquement locale. Elle est utilisée par un portier pour associer le message BRQ à un appel donné.

callType – Lorsqu'il utilise cette valeur, le portier peut essayer de déterminer la largeur de bande "réellement" utilisée.

bandWidth – Nouvelle largeur de bande bidirectionnelle demandée pour l'appel, en unités de 100 bit/s. Il s'agit d'une valeur absolue qui inclut seulement les flux de données audio et vidéo et qui ne tient pas compte des en-têtes et suffixes. Les flux uniques de diffusion s'ajouteront une seule fois à la largeur de bande totale utilisée, même si ces flux ont de multiples destinataires.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

callIdentif – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

gatekeeperIdentif – Identificateur de portier (**gatekeeperIdentif**) que l'extrémité a reçu dans la liste **alternateGatekeeper** d'un message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message BRJ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

answeredCall – Mis à Vrai pour indiquer que ce participant était la destination d'origine (ce participant a répondu à l'appel).

callLinkage – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir le paragraphe 10/H.323.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un instant donné, à condition que le portier confirme le message BRQ par un message BCF. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

usageInformation – Ce champ permet à l'extrémité de signaler des informations relatives au taux d'utilisation pour la communication considérée. Un portier ne doit pas inclure ce champ lors de l'envoi d'un message BRQ.

bandwidthDetails – Ce champ donne des informations sur la largeur de bande pour chaque flux média que l'extrémité émet ou reçoit actuellement, avec les mêmes unités que le champ **bandWidth**. Chaque flux multidiffusé ne doit être signalé qu'une seule fois, même si le flux média a des destinataires multiples.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.12.2 Message BCF (confirmation de largeur de bande)

Le message BCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message BRQ.

bandWidth – Valeur maximale autorisée à l'instant considéré par pas de 100 bit/s.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un instant donné. Ce champ n'est pas inclus lorsque le message BCF est envoyé par un portier. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.12.3 Message BRJ (refus de largeur de bande)

Le message BRJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message BRQ.

rejectReason – Cause pour laquelle la modification a été refusée par le portier.

allowedBandWidth – Maximum autorisé à l'instant considéré par pas de 100 bit/s y compris l'attribution courante.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.13 Messages de demande de localisation

Un message LRQ est une demande soumise à un portier pour fournir une traduction d'adresse. Le portier répond par un message LCF contenant l'adresse de transport de la destination ou rejette simplement la demande avec un message LRJ.

7.13.1 Message LRQ (demande de localisation)

Le message LRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentifrier – Identificateur d'extrémité qui a été attribué par le terminal au moyen d'un message RCF.

destinationInfo – Séquence d'adresses pseudonymes pour la destination, telles que des champs **dialedDigits** ou **partyNumber** (**E164Number** ou **PrivateNumber**), ou des identificateurs **h323-ID**. Si au moins une adresse pseudonyme est enregistrée auprès d'un portier et si le message LRQ ne compte pas deux adresses pseudonymes enregistrées auprès de personnes différentes, le portier reconnaîtra le message LRQ comme se rapportant à l'identité enregistrée. Dans le cas d'adresses pseudonymes incompatibles, il convient de refuser la demande d'admission en indiquant comme cause **AliasesInconsistent**. Si le portier n'assure pas la validation, il considèrera la première adresse enregistrée comme étant celle de la destination.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

replyAddress – Adresse de transport à laquelle doivent être envoyés les messages LCF/LRJ.

sourceInfo – Indique l'expéditeur du message LRQ. Le portier peut utiliser cette information pour décider de la manière dont il va répondre au message LRQ.

canMapAlias – Si sa valeur est à Vrai, ce champ indique que si le message LCF résultant contient les champs **destinationInfo**, **destExtraCallInfo** et/ou **remoteExtensionAddress**, l'extrémité peut copier ces informations respectivement dans les champs **destinationAddress**, **destExtraCallInfo** et **remoteExtensionAddress** du message SETUP. Si le portier remplaçait les informations d'adressage provenant du message LRQ et que **canMapAlias** soit à Faux, le portier devrait refuser le message LRQ. Les systèmes conformes à la présente version 4 de l'UIT-T H.225.0 et aux versions supérieures doivent mettre ce champ à Vrai.

gatekeeperIdentifier – Identificateur de portier (**gatekeeperIdentifier**) que l'extrémité a reçu dans la liste **alternateGatekeeper** d'un message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message LRJ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

desiredProtocols – Ce champ identifie, par ordre de préférence, les types de protocole recherchés par l'extrémité d'origine pour son appel (p. ex. voix ou télécopie). Une entité de résolution peut utiliser ce champ pour localiser une extrémité qui prend également en charge le protocole, compte tenu de l'ordre de préférence.

desiredTunnelledProtocol – Ce champ désigne un protocole dont la canalisation en tunnel est demandée.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

hopCount – Ce champ définit le nombre de portiers au travers desquels le message considéré peut se propager. Lorsqu'un portier reçoit une demande LRQ et détermine que le message doit être réexpédié vers un autre portier, il commence par décrémenter le champ **hopCount**. Si la valeur de ce champ est supérieure à 0, le portier insère la nouvelle valeur du décompte de bonds dans le message à réexpédier. Si la valeur **hopCount** est arrivée à 0, le portier ne doit pas réexpédier le message.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

7.13.2 Message LCF (confirmation de localisation)

Le message LCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message LRQ.

callSignalAddress – Adresse de transport à laquelle doit être envoyée la signalisation d'appel Q.931; utilise l'accès connu fiable ou l'accès dynamique, mais peut être une adresse d'extrémité ou de portier selon le modèle d'appel utilisé.

rasAddress – Adresse utilisée par l'extrémité localisé pour les messages d'enregistrement, d'admissions et d'état.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

destinationInfo – Séquence d'adresses pseudonymes pour la destination, telles que des champs **dialedDigits** ou **partyNumber** (**e164Number** ou **privateNumber**), ou des identificateurs **h323-ID**.

destExtraCallInfo – Contient des adresses externes pour les appels multiples.

destinationType – Spécifie le type de l'extrémité de destination.

remoteExtensionAddress – Contient l'adresse pseudonyme d'une extrémité appelé dans les cas où cette information est nécessaire pour traverser plusieurs passerelles.

alternateEndpoints – Séquence d'autres points d'extrémité possibles classés par ordre de priorité pour les éléments **callSignalAddress**, **rasAddress**, ou **destinationInfo**.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

alternateTransportAddresses – Ce champ achemine des adresses de signalisation d'appel pour des modes de transport autres que TCP. L'inclusion d'une adresse indique la prise en charge du transport correspondant.

supportedProtocols – Ce champ indique les protocoles pris en charge par l'extrémité de destination.

multipleCalls – S'il est mis à Vrai, ce champ indique que l'extrémité de destination possède la capacité de signaler plusieurs appels dans une même connexion sémaphore d'appel. S'il a la valeur Faux, l'extrémité de destination ne possède pas cette capacité. Si ce champ n'est pas présent, le portier ne sait pas si l'extrémité distante possède cette capacité.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

serviceControl – Ce champ contient des informations d'adressage que l'extrémité peut utiliser pour des communications de service associées à l'appel avec un réseau tel que décrit, par exemple, dans l'Annexe K/H.323.

7.13.3 Message LRJ (refus de localisation)

Le message LRJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message LRQ.

rejectReason – Il s'agit de la cause pour laquelle la demande de localisation a été refusée. Si la cause du refus **rejectReason** est **routeCallToSCN**, la cause de refus **rejectReason** pour ce choix concerne également un numéro de téléphone ou une liste de numéros de téléphone vers lesquels la passerelle peut réacheminer l'appel dans le réseau RCC, si la passerelle autorise une telle procédure. Si la valeur de ce champ est **resourceUnavailable**, le taux d'utilisation de la largeur de bande est trop grand ou aucune entité enregistrée auprès du portier ne possède actuellement la capacité de traiter un appel jusqu'à l'emplacement demandé. Une cause avec la valeur **genericDataReason** indique que la demande a été rejetée à cause d'un élément de service générique; dans ce cas, des informations complémentaires peuvent être spécifiées dans le champ **genericData**.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques se rapportant à l'appel considéré.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

serviceControl – Ce champ contient des informations d'adressage que l'extrémité peut utiliser pour des communications de service associées à l'appel avec un réseau tel que décrit, par exemple, dans l'Annexe K/H.323.

7.14 Messages de désengagement

7.14.1 Message DRQ (demande de désengagement)

Lorsqu'il est envoyé à partir d'une extrémité vers un portier, le message DRQ informe le portier qu'une extrémité est abandonnée. S'il est envoyé par un portier à une extrémité, le message DRQ oblige d'abandonner un appel, une telle demande ne doit pas être refusée. Le message DRQ n'est pas envoyé entre extrémité directement.

Il convient de noter que le message DRQ n'est pas le même que le message **ReleaseComplete** étant donné que son objet est d'informer le portier de la terminaison d'un appel; le portier peut ne pas recevoir de message de fin de libération s'il ne ferme pas la voie de signalisation d'appel.

Le message DRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentif – Identificateur d'extrémité qui a été assigné au terminal par un message RCF.

conferenceID – Identificateur de l'appel pour lequel la largeur de bande doit être libérée.

callReferenceValue – Valeur CRV Q.931 pour cet appel; sa validité est uniquement locale. Elle est utilisée par un portier pour associer le message à un appel particulier.

disengageReason – Cause pour laquelle la modification a été demandée par le portier ou le terminal.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

callIdentif – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

gatekeeperIdentif – Identificateur de portier (**gatekeeperIdentif**) que l'extrémité a reçu dans la liste **alternateGatekeeper** d'un message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message DRJ.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

answeredCall – Mis à Vrai pour indiquer que ce participant était la destination d'origine (ce participant a répondu à l'appel).

callLinkage – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir le paragraphe 10/H.323.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice à un instant donné, à condition que le portier confirme le message DRQ par un message DCF. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**. Ce champ n'est pas inclus lorsque le message DRQ est envoyé par un portier.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

usageInformation – Ce champ permet à l'extrémité de signaler des informations relatives au taux d'utilisation pour la communication considérée. Un portier ne doit pas inclure ce champ lors de l'envoi d'un message DRQ.

terminationCause – Ce champ décrit la raison pour laquelle la communication s'est terminée. Cette information est plus spécifique que la raison contenue dans le champ **disengageReason**. Un portier ne doit pas inclure ce champ lors de l'envoi d'un message DRQ.

serviceControl – Ce champ contient des données propres au service, ou des références à celui-ci qui peuvent être utilisées par une extrémité comme cela est décrit dans l'Annexe K/H.323. Le portier peut utiliser ce champ pour indiquer que la communication va se terminer parce qu'un compte a été épuisé ou que le montant prépayé pour la communication a été consommé.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.14.2 Message DCF (confirmation de désengagement)

Le message DCF comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message DRQ.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul,

l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

capacity – Ce champ indique la capacité d'appel disponible à l'extrémité émettrice une fois que l'appel indiqué dans le message DCF a été désengagé. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity**. Ce champ n'est pas inclus lorsque le message DCF est envoyé par un portier.

circuitInfo – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

usageInformation – Ce champ permet à l'extrémité de signaler des informations relatives au taux d'utilisation pour la communication considérée. Un portier ne doit pas inclure ce champ lors de l'envoi d'un message DCF.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.14.3 Message DRJ (refus de désengagement)

Le message **DRJ** est envoyé par le portier si l'enregistrement de l'extrémité est annulé.

Le message DRJ comprend ce qui suit:

requestSeqNum – Doit avoir la même valeur que celle qui a été transmise dans le message DRQ.

rejectReason – Cause du refus de la demande.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.15 Messages de demande d'état

Le message IRQ est envoyé par un portier à un terminal pour demander les informations d'état sous la forme d'un message IRR. Le message IRR peut également être envoyé par le terminal à un intervalle spécifié dans le message ACF sans réception d'un message IRQ émis par le portier. Il ne faut pas confondre ce message avec le message STATUS Q.931.

Lorsqu'une extrémité envoie un message IRR non sollicité à un portier de version 2 ou supérieure, il peut indiquer dans le champ **needResponse** qu'il souhaite que le portier accuse réception du message IRR. Dans ce cas, il remplit le champ **requestSeqNum** avec un numéro autre que 1. Le portier renvoie un message IACK (acquiescement positif) ou un message INAK (acquiescement négatif) et doit renvoyer le même numéro dans le champ **requestSeqNum**.

7.15.1 Message IRQ (demande d'information)

Le message IRQ comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

callReferenceValue – Valeur CRV de l'appel pour lequel on demande des informations. Si elle est égale à zéro, ce message est interprété comme une demande de message IRR pour chaque appel pour lequel le terminal est actif. Si le terminal n'est actif pour aucun appel, un message IRR comportant tous les champs appropriés doit être envoyé en réponse à une valeur nulle de **CallReferenceValue**. Si la valeur **callReferenceValue** est égale à 0, l'extrémité doit ignorer l'identificateur **callIdentifiant** – en pareil cas, le portier doit attribuer à l'identificateur **callIdentifiant** la valeur 0.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

replyAddress – Adresse de transport à laquelle doit être envoyé le message IRR, peut-être une adresse autre que celle du portier.

callIdentifiant – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

uuiesRequested – Sur demande du portier, l'extrémité peut devoir informer le portier des messages de signalisation d'appel H.225.0 qu'il envoie ou reçoit s'il a indiqué cette capacité dans le message ARQ en positionnant **willSupplyUIEs** sur Vrai. **uuiesRequested** indique l'ensemble des messages de signalisation d'appel H.225.0 que l'extrémité doit signaler au portier.

callLinkage – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir le paragraphe 10/H.323.

usageInfoRequested – Ce champ peut être inclus par un portier afin de demander que l'extrémité signale dans le message IRR les informations de taux d'utilisation indiquées pour l'appel.

segmentedResponseSupported – Ce champ indique si le portier autorisera l'extrémité à renvoyer des informations d'appel pour tous les appels dans plusieurs messages IRR ou à les renvoyer par segments. Si ce champ est présent, la segmentation est autorisée. Sinon, elle ne l'est pas. Ce champ n'est valide que si le portier envoie un message IRQ avec une valeur 0 du champ **callReferenceValue**. Sinon, il ne doit pas être présent.

nextSegmentRequested – Si le portier envoie un message IRQ avec une valeur 0 du champ **callReferenceValue** et s'il inclut le champ **segmentedResponseSupported**, l'extrémité peut renvoyer un message IRR avec une indication partielle seulement des informations d'appel, par inclusion du champ de segment dans le message IRR. Le portier peut demander le segment suivant en renvoyant le message IRQ précédent avec le champ **nextSegmentRequested** mis à la valeur du prochain segment que le portier s'attend à recevoir.

capacityInfoRequested – S'il est présent, ce champ indique que le portier demande que l'extrémité inclue les informations de capacité d'appel dans le message IRR.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.15.2 Message IRR (réponse à une demande d'information)

Le message IRR comprend ce qui suit:

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

requestSeqNum – Dans le cas d'une réponse IRR sollicitée, ce champ doit contenir le numéro de séquence extrait de la demande IRQ. Dans le cas d'un rapport non sollicité soumis à un portier de version 1, ce champ doit contenir la valeur un (1). Dans toutes les autres réponses IRR non sollicitées, ce champ doit contenir un nombre croissant de façon monotone (que le portier doit renvoyer dans sa réponse si **needResponse** est à Vrai).

endpointType – Fournit des informations sur l'extrémité.

endpointIdentifiant – Valeur assignée par le portier dans le message RCF.

rasAddress – Adresse pour l'enregistrement, l'admission, etc.

callSignalAddress – Adresse pour la signalisation d'appel H.225.0.

endpointAlias – Le ou les pseudonymes associés à l'extrémité.

perCallInfo – Informations associées à un appel donné:

- **nonStandardData** – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées);
- **callReferenceValue** – Valeur CRV Q.931 de l'appel sur lequel porte la réponse;
- **conferenceID** – Identificateur univoque de la conférence;
- **originator** – Lorsqu'il est égal à Vrai, l'extrémité qui fait l'objet de la demande est l'appelant, s'il est égal à Faux, l'extrémité est l'appelé;
- **audio** – Informations concernant le ou les canaux audio. L'élément **multicast** doit être inclus si la session est multidiffusée;
- **video** – Informations concernant le ou les canaux vidéo. L'élément **multicast** doit être inclus si la session est multidiffusée;
- **data** – Informations concernant le ou les canaux de données;
- **h245** – Adresse de transport du canal de commande H.245;
- **callSignaling** – Adresse de transport du canal de signalisation d'appel H.225.0;
- **callType** – renseigne sur la topologie de l'appel;
- **bandwidth** – Largeur de bande utilisée par pas de 100 bit/s; n'inclut que les signaux audio et vidéo, à l'exclusion de tout en-tête ou préfixe;
- **callModel** – Indique le modèle d'appel utilisé, selon l'extrémité;
- **callIdentifier** – Identificateur d'appel unique sur le plan mondial, fixé par l'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée qui est utilisée dans la présente Recommandation;
- **tokens** – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message;
- **cryptoTokens** – Jetons (**token**) chiffrés;

- **substituteConfIDs** – Liste de tous les identificateurs ConferenceID reçus dans les messages SubstituteCID H.245 se rapportant à l'élément **conferenceID** de l'élément **perCallInfo** du message RAS d'origine;
- **pdu**:
 - **h323pdu** – Copie d'une unité PDU H.225.0 et Q.931, comme demandé par le portier dans l'élément **uuiesRequested** du message ACF ou IRQ;
 - **sent** – Mis à Vrai pour indiquer que l'extrémité a envoyé l'élément **h323pdu**; mis à Faux pour indiquer que l'extrémité a reçu l'élément **h323pdu**.
- **callLinkage** – Le contenu de ce champ est normalement commandé par un service d'assemblage d'appels. Pour les procédures et la sémantique de ce champ, voir le paragraphe 10/H.323.
- **usageInformation** – Ce champ permet à l'extrémité de signaler des informations relatives au taux d'utilisation pour la communication considérée.
- **circuitInfo** – Ce champ donne des renseignements sur le ou les circuits RCC utilisés pour l'appel considéré.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

needResponse – Si sa valeur est à Vrai et que le portier ait indiqué que dans le message RCF ou ACF qu'il répondrait aux messages IRR non sollicités (en positionnant **willRespondToIRR** sur Vrai), alors le portier devra répondre avec un message IACK ou INAK. Si le portier n'a indiqué ni dans le message RCF ni dans le message ACF qu'il répondrait aux messages IRR non sollicités (en positionnant **willRespondToIRR** sur Faux), alors il peut ignorer le booléen **needResponse**.

capacity – Ce champ indique la capacité d'appel de l'extrémité émettrice à l'instant considéré. Lors de l'envoi de ce champ, l'extrémité doit inclure l'élément **currentCallCapacity** et ne devrait inclure l'élément **maximumCallCapacity** que dans une réponse à une demande IRC contenant l'élément **capacityInfoRequested**.

irrStatus – Cet élément doit normalement être renvoyé dans les messages IRR en réponse à un message IRQ envoyé par le portier. L'absence de cet élément indique que le message IRR contient des informations complètes sur les détails de la communication. Les valeurs suivantes sont possibles:

- **complete** – Indique que le message IRR considéré contient le dernier segment des informations d'appel pour un message IRQ demandant tous les détails de la communication. Lorsque la segmentation n'est pas utilisée, ce champ indique que le message IRR contient tous les détails de communication contenus dans un même message IRR;
- **incomplete** – Indique que l'extrémité n'est pas en mesure d'insérer toutes les informations d'appel demandées dans un même message IRR lors d'une réponse à un message IRQ contenant une valeur 0 du champ **callReferenceValue**;
- **segment** – Ce champ indique le numéro de segment, qui augmente de façon monotone modulo 65536, du message IRR considéré lorsque des messages IRR segmentés sont envoyés en réponse à un message IRQ contenant un champ **callReferenceValue** de valeur 0;
- **invalidCall** – Ce champ indique que l'appel indiqué dans le message IRQ n'existe pas.

unsolicited – Les extrémités de version 4/H.323 et ultérieures doivent mettre ce champ à la valeur Vrai dans les messages de demande IRR non sollicitée, comme décrit dans le 8.4.2/H.323 et doivent le mettre à la valeur Faux dans les demandes IRR sollicitées.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.15.3 Message IACK (acquiescement de demande d'information)

Le message IACK comprend ce qui suit:

requestSeqNum – Ce champ doit contenir le numéro **requestSeqNum** qui figurait dans le message IRR.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

7.15.4 Message INAK (acquiescement négatif de demande d'information)

Le message INAK comprend ce qui suit:

requestSeqNum – Ce champ doit contenir le numéro **requestSeqNum** qui figurait dans le message IRR.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

nakReason – Cause pour laquelle le message IRR a fait l'objet d'un acquiescement négatif.

altGKInfo – Informations facultatives sur des portiers de remplacement.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

7.16 Message non normalisé

La structure de **NonStandardMessage** est la suivante:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.17 Message incompris

Ce message est envoyé à chaque fois qu'une extrémité H.323 reçoit un message RAS qu'elle ne comprend pas ou qu'elle ne peut pas décoder. Si l'adresse de transport jusqu'à destination du message XRS n'est pas disponible (c'est-à-dire que le message RAS reçu ne peut pas être décodé), le message XRS peut être envoyé à l'adresse de transport à partir de laquelle le message RAS incompris a été reçu. Cette adresse de transport peut être obtenue de la couche de transport sous-jacente. Un message XRS ne doit pas être envoyé en réponse à un message XRS entrant. Les extrémités H.323 ne devraient pas envoyer plus d'un seul message XRS par seconde à la même adresse de transport afin d'éviter l'encombrement du réseau en cas de réception de messages corrompus.

requestSeqNum – Ce champ doit indiquer le numéro **requestSeqNum** du message inconnu, si celui-ci peut être décodé. Si le message inconnu ne peut pas être décodé, ce champ est un numéro croissant de façon monotone qui est attribué à titre unique à l'expéditeur. Il y a lieu d'utiliser ce champ pour assurer la compatibilité amont avec les extrémités conformes à la version 3 de l'UIT-T H.323 et aux versions antérieures. Les extrémités conformes à la version 4 et aux versions postérieures doivent normalement examiner le paramètre **messageNotUnderstood** afin d'associer le message XRS à un message déjà émis.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

messageNotUnderstood – Copie du message qui a été reçu et qui n'a pas été compris.

7.18 Messages de disponibilité de ressources de la passerelle

L'indication de disponibilité de ressources (RAI, *resource availability indication*) sert à une passerelle à indiquer à un portier sa capacité d'appel courante pour chaque protocole de la série H et

le débit associé à chaque protocole. Le portier répond avec une confirmation de disponibilité de ressources (RAC, *resource availability confirmation*) à la réception d'un message RAI pour en accuser réception.

7.18.1 Message RAI (indication de disponibilité de ressources)

Le message RAI comprend ce qui suit:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

protocolIdentifier – Identifie le millésime de l'extrémité expéditeur.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

endpointIdentifier – Chaîne d'identité de l'extrémité assignée par un portier.

protocols – Indique les débits courants pour chaque protocole qui peut être pris en charge compte tenu de l'état courant du dispositif.

almostOutOfResources – Lorsque ce champ est à Vrai, le dispositif utilise toute sa capacité ou presque. Toute action fondée sur ce champ est à la discrétion du fabricant. Si le dispositif est loin d'utiliser toute sa capacité, ce champ doit être mis à Faux.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

capacity – Ce champ indique la capacité d'appel à l'extrémité émettrice à un moment donné. Noter que si ce champ est fourni, le destinataire ne devrait pas tenir compte de l'opérateur booléen du champ **almostOutOfResources** car le champ **capacity** fournit des informations plus détaillées; cependant, l'opérateur booléen du champ **almostOutOfResources** doit tout de même être réglé correctement afin de conserver la rétrocompatibilité. Lors de l'envoi du champ **capacity**, l'extrémité doit inclure les éléments **currentCallCapacity**.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.18.2 Message RAC (confirmation de disponibilité de ressources)

Le message RAC comprend ce qui suit:

requestSeqNum – Valeur qui a été transmise dans le message RAI.

protocolIdentifier – Identifie le millésime du portier qui accuse réception de l'indication de disponibilité de ressources.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.19 Temporisations RAS et message demande en cours (**RIP, request in progress**)

On trouvera au Tableau 22 les valeurs par défaut de temporisations recommandées pour la réponse aux messages RAS et les nombres de nouvelles tentatives si aucune réponse n'est reçue. (Ces valeurs sont susceptibles de changer lorsqu'on aura acquis plus d'expérience d'implémentation.)

Tableau 22/H.225.0 – Valeurs de temporisation recommandées par défaut

Message RAS	Valeur de temporisation (s)	Nombre de nouvelles tentatives
GRQ	5	2
RRQ (Note 1)	3	2
URQ	3	1
ARQ	5	2
BRQ	3	2
IRQ	3	1
IRR (Note 2)	5	2
DRQ	3	2
LRQ	5	2
RAI	3	2
SCI	3	2

NOTE 1 – La valeur de temporisation doit être recalculée d'après la durée de vie (qui peut être indiquée par le portier dans le message RCF) et d'après le nombre souhaité de nouvelles tentatives.

NOTE 2 – Dans les cas où le portier est censé répondre à un message IRR non sollicité avec un message IACK ou INAK, la temporisation s'écoule tant qu'aucune réponse au message IRR n'est reçue.

Si une entité reçoit une demande de la part d'une entité de version 2 (ou postérieure) à laquelle une réponse ne peut pas être produite dans le cadre d'une temporisation typique de nouvelle tentative, elle peut envoyer un message RIP spécifiant la période (dans le champ **delay**) au bout de laquelle une réponse doit avoir été produite. Dès qu'une réponse est disponible, l'entité qui répond doit l'envoyer et ne doit pas attendre l'expiration du délai indiqué dans le message RIP. Si une entité demandeuse n'a pas reçu de réponse au moment où le délai indiqué dans le message RIP expire, elle doit envoyer à nouveau la demande. L'entité qui répond peut alors envoyer une copie de la réponse ou un autre message RIP. La Figure 2 donne un exemple d'échange de messages qui montre un certain nombre d'aspects de la stratégie de nouvelle tentative.

Les vendeurs doivent savoir que toute nouvelle tentative aura une incidence sur le temps d'établissement de l'appel, qui doit être minimisé. Des temps courts de nouvelle tentative sont donc souhaitables. Pour que les entités distantes puissent anticiper les délais types de nouvelle tentative

afin de décider de l'instant où elles enverront un message RIP, les entités doivent éviter les périodes de nouvelle tentative inférieures à 100 ms. Il est souhaitable d'utiliser des temps d'attente exponentiels et de faire une adaptation sur les temps aller-retour mesurés. Pour cela, les entités peuvent utiliser la mesure du temps aller-retour du processus d'enregistrement RRQ/RCF pour modifier une évaluation qui était prudente au départ (quelques secondes). Les entités peuvent aussi utiliser le processus d'enregistrement pour échanger les numéros de version afin de garantir que le mécanisme de nouvelle tentative fondé sur le message RIP n'est pas utilisé lorsque des entités de version 1 interviennent dans la signalisation.

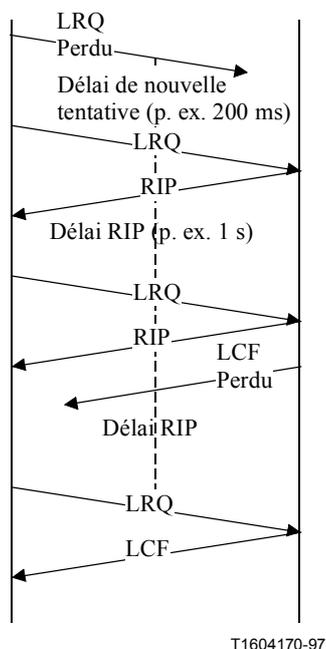


Figure 2/H.225.0 – Exemple d'utilisation du message RIP

Le message RIP comprend ce qui suit:

requestSeqNum – Numéro de la demande en cours de traitement.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

delay – Spécifie le temps en millisecondes au bout duquel une extrémité peut faire une nouvelle tentative. L'extrémité qui répond peut répondre avant l'expiration de cette période.

7.20 Messages de commande de service

7.20.1 ServiceControlIndication (SCI)

Le message SCI est envoyé par un fournisseur de services afin d'indiquer au client du service qu'une session distincte de commande de service peut être lancée vers l'adresse fournie. Ce message peut être envoyé par un portier à une extrémité (p. ex. pour la présentation d'éléments de service à l'utilisateur) ou par une extrémité à un portier (par exemple, pour exporter une logique de traitement d'appel). Noter que les entités H.323 conformes à la version 3 ou à une version antérieure ne possèdent pas la capacité de décoder ce message et n'y répondront donc pas.

Le message SCI contient les éléments suivants:

requestSeqNum – Numéro croissant de façon monotone propre à l'expéditeur. Il doit être renvoyé par le destinataire dans toute réponse associée à ce message spécifique.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

serviceControl – Transporte un ensemble d'informations relatives à la session de commande de service.

endpointIdentif – Mis à la valeur reçue du portier dans le message RCF si celui-ci est envoyé par une extrémité à son portier.

callSpecific – Ce champ est fourni si les sessions indiquées se rapportent à une même communication spécifique. Les champs **callIdentif**, **conferenceID** et **answeredCall** doivent être mis à la même valeur que dans le message ARQ auquel la session de service se rapporte.

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

7.20.2 ServiceControlResponse (SCR)

Le message SCR est envoyé pour accuser réception d'un message SCI mais n'implique pas nécessairement que le client du service lancera la session indiquée dans le message SCI.

Le message SCR contient les champs suivants:

requestSeqNum – Ce champ doit contenir la valeur qui a été transmise dans le message SCI.

result – Ce champ indique le résultat du traitement des informations contenues dans le message SCI. Les valeurs suivantes sont définies:

- **started** – La commande de service demandée a été lancée;
- **failed** – Une erreur s'est produite avec la demande, qui a donc échoué;
- **stopped** – La commande de service a été arrêtée;

- **notAvailable** – La commande de service demandée n'était pas disponible au moment de la demande.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple des données privées).

tokens – Données qui peuvent être nécessaires pour permettre l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons (**token**) chiffrés.

integrityCheckValue – Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur **integrityCheckValue**, ce champ doit être ignoré et doit être vide. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ **integrityCheckValue** et transmet le message.

featureSet – Ce champ spécifie un ensemble d'éléments de service génériques.

genericData – Ce champ énumère les éléments génériques associés à des éléments de service définis en dehors de la spécification de base H.225.0. Ces paramètres peuvent par exemple être utilisés pour canaliser en transparence des informations à travers le service RAS.

8 Mécanismes permettant de conserver la qualité de service (QS)

8.1 Méthode générale et hypothèses

La qualité de service du transport sur un réseau à commutation par paquets inclut certaines caractéristiques comme:

- le taux d'erreur;
- le taux de perte de paquets;
- les temps de propagation.

Toute signalisation associée à la qualité de service transport (par exemple une demande de réservation à l'intention d'un routeur) est faite par le terminal dès que possible, ou par le portier au nom du terminal. Le terminal peut souhaiter faire certaines réservations puisque le portier peut ne pas se trouver logiquement proche du terminal, ou être en mesure de formuler des demandes relatives à la qualité de service au nom du terminal. Les moyens par lesquels le terminal ou le portier fait des réservations de qualité de service ou de largeur de bande n'entrent pas dans le domaine d'application de la présente Recommandation.

Les rapports d'émetteur et de récepteur du protocole RTCP sont des moyens par lesquels la qualité de service doit être évaluée.

Il y a deux types d'encombrement associés au temps de propagation qui peuvent être mesurés:

- les augmentations à court terme des temps de propagation qui se traduiront par une diminution du débit de trame perceptible mais non gênante;
- une augmentation générale des temps de propagation due à un encombrement du réseau à commutation par paquets avec le temps de sorte qu'un mécanisme de rétroaction est utile.

Essentiellement, les rafales d'erreurs à court terme peuvent être compensées par une dissimulation des erreurs, et l'encombrement à plus long terme peut être compensé en réduisant la charge multimédia. L'hypothèse retenue est que tous les terminaux multimédias du réseau à commutation par paquets sont des terminaux H.323, et tous tenteront de diminuer l'utilisation du réseau en question en cas d'encombrement et ne tenteront pas de "voler" de la largeur de bande aux autres.

Les erreurs binaires sur un réseau à commutation par paquets sont en général corrigées dans les couches inférieures ou se traduisent par des pertes de paquets de sorte qu'elles ne doivent être pas examinées plus avant dans le présent paragraphe.

La perte de paquets exige de la part du récepteur qu'il soit capable de la compenser de manière à dissimuler les erreurs au maximum. Pour les informations de données et de commande, on utilise la retransmission au niveau de la couche Transport. Pour les données audio et vidéo la retransmission appelle un complément d'étude.

Un niveau donné de qualité de service de transport correspond à un niveau de qualité de service audio/vidéo perçue par l'utilisateur et qui dépend en partie de l'efficacité des méthodes utilisées pour résoudre les problèmes de qualité de service de transport.

8.2 Utilisation du protocole RTCP pour la mesure de la qualité de service

8.2.1 Rapports d'expéditeur

Le rapport d'expéditeur a trois objets principaux:

- 1) permettre la synchronisation de plusieurs flux RTP tels des flux audio et vidéo;
- 2) permettre au récepteur de connaître le débit de données attendu et le débit de paquets attendu;
- 3) permettre au récepteur de mesurer la distance en temps de l'expéditeur.

Sur ces trois objectifs, l'objectif 1) est celui qui concerne le plus la présente Recommandation. Les constructeurs peuvent utiliser les rapports d'émetteur comme bon leur semble.

Le champ utilisé pour la synchronisation de flux est l'horodate RTP et l'horodate NTP figurant dans le rapport de l'expéditeur du RTCP. L'horodate NTP (lorsqu'elle est disponible) donne la durée chronométrée et correspond à l'horodate RTP qui a les mêmes unités et le même décalage aléatoire étant donné que le protocole RTP prélève les horodates dans les paquets médias.

8.2.2 Rapports du récepteur

Quatre paramètres des rapports du récepteur sont utilisées dans la présente Recommandation pour la mesure de la qualité de service:

- 1) la perte fractionnaire;
- 2) la perte cumulative des paquets;
- 3) le numéro de séquence le plus élevé étendu reçu;
- 4) gigue entre arrivées.

Les paramètres 2) et 3) sont utilisés pour calculer le nombre de paquets perdus depuis le précédent rapport de récepteur. Cette valeur peut être considérée comme une mesure à long terme de l'encombrement du réseau à commutation par paquets. Le A.6.3.4 donne un exemple de calcul. Si le taux de perte dépasse la valeur fixée par le constructeur, le terminal H.225.0 devra réduire les débits de transmission des médias côté réseau à commutation par paquets conformément aux procédures décrites au 8.4 ci-après. Si le paramètre 1) dépasse la valeur fixée par le constructeur, il peut être aussi souhaitable de prendre des mesures correctives.

Si l'intervalle entre les rapports de récepteur dépasse une valeur fixée par le constructeur, les terminaux H.323 devront utiliser le paramètre 1) comme indicateur d'un encombrement grave nécessitant une réduction du débit des médias côté réseau à commutation par paquets.

Le paramètre 4) doit être utilisé comme une indication d'encombrement imminent. Si la gigue entre arrivées augmente au cours de trois rapports de récepteur consécutifs, le terminal H.323 émetteur doit prendre les mesures correctives.

8.3 Procédures relatives à la gigue audio/vidéo

L'UIT-T H.245 spécifie des commandes et des procédures permettant d'obtenir des indications aller-retour au moyen des structures **RoundTripDelayRequest** et **RoundTripDelayResponse**. Dans un appel multipoint, le contrôleur multipoint répond à une demande émanant de l'extrémité. Le protocole RTCP contient une méthode de calcul des temps aller-retour à partir des messages des rapports d'expéditeur et de récepteur. Il convient de noter que la quantité mesurée dans chaque cas n'est pas la même, de sorte qu'il n'existe pas de conflit lorsqu'on utilise les deux méthodes pour mesurer la gigue.

On se reportera au 6.2.5/H.323 montrant comment la signalisation de niveau H.245 peut être utilisée pour diminuer facultativement les délais associés à la gigue.

8.4 Procédures relatives au décalage audio/vidéo

On se reportera au 6.2.6/H.323 pour avoir de plus amples détails sur la façon dont la signalisation de niveau H.245 est utilisée pour limiter le décalage entre les différents canaux logiques.

8.5 Procédures permettant de maintenir la qualité de service

Il existe un certain nombre de méthodes permettant à la passerelle/terminal H.323 de réagir à une augmentation de la perte de paquets ou de la gigue entre arrivées dans le récepteur distant. Ces méthodes peuvent être groupées en méthodes de réaction rapide à un problème à court terme tels la perte d'un paquet ou le retard d'un paquet et celles qui conviennent à une réponse à un problème à plus long terme tel un encombrement croissant sur le réseau à commutation par paquets. Il convient de noter que ces méthodes ne cherchent pas à maintenir la qualité de service actuelle, mais plutôt à obtenir une dégradation ordonnée du service. Les priorités ci-dessous doivent être observées de sorte que la dégradation affectera les médias dans l'ordre suivant, par ordre décroissant d'importance: vidéo, données, audio, commande.

Réactions à court terme:

- réduire le débit de trame pendant une courte période de temps: cela peut se traduire dans la passerelle H.323 par l'envoi de trames de remplissage supplémentaires H.261 dans le sens réseau à commutation par paquets → RCC pour compenser le sous-débit de paquets;
- diminuer le débit de paquets en passant au mode facultatif dans lequel les flux audio/vidéo sont mélangés en un paquet (appelle un complément d'étude);
- le débit de paquets peut être réduit en utilisant la fragmentation de macroblocs du flux vidéo.

Réactions à plus long terme:

- la diminution du débit média (par exemple en passant de 384 kbit/s à 256 kbit/s): cette réaction peut être déclenchée par une simple instruction donnée au codeur dans un terminal ou faire intervenir une fonction de réduction de débit dans la passerelle H.323. Ces modifications sont signalées par des commandes **FlowControl** H.245 ou par une signalisation de canal logique selon le cas;
- arrêter le média d'importance moindre (par exemple stopper le flux vidéo pour permettre un fort volume de trafic T.120);
- renvoi d'un signal d'occupation (occupation adaptative) au récepteur pour indiquer l'encombrement du réseau à commutation par paquets. Cette action peut être associée avec l'arrêt d'un média, ou même de tous les médias autres que l'accès transport de commande. L'occupation adaptative est signalée par une valeur de cause Q.931 dans le message **RELEASE COMPLETE**.

Il convient de noter que la réaction à une gigue entre arrivées dans un trajet multirouteur, dans lequel un large pourcentage de paquets arrivent avec des défauts, est difficile. Il peut être impossible de

distinguer cette source de gigue des autres sources, ou de fonder une stratégie de récupération des erreurs sur la gigue mesurée. Cependant, la perte des paquets est quantifiable et non ambiguë.

8.6 Limitation de l'écho

La responsabilité de la limitation de l'écho acoustique relève du terminal selon la série H. En général, compte tenu du délai nécessaire à la compression vidéo/audio, on suppose que tous les terminaux H.320, H.323 et H.324 disposent de la même forme de limitation d'écho (annulation ou commutation).

Cependant, lorsque le terminal H.323 est en communication avec un poste téléphonique du RTGC, on se trouve dans le cas type où ce poste ne dispose pas de système de limitation d'écho. Ainsi, l'utilisateur du terminal H.323 peut entendre le retour d'écho acoustique provenant du côté RTGC. Cet écho acoustique peut être minimisé par l'utilisation d'un téléphone à haut-parleur avec limitation d'écho, ou par l'utilisation d'un combiné ou d'écouteurs. Les constructeurs peuvent aussi ajouter un affaiblisseur sur le trajet audio lorsqu'un terminal H.323 est connecté à un téléphone du service téléphonique de base du RTGC.

Limitation de l'écho dû au transformateur différentiel (2 fils/4 fils). Le circuit hybride offre une interface entre les systèmes de transmission à 4 fils et les terminaux à 2 fils. L'annulation d'écho n'est pas nécessaire pour les communications vocales du RNIS qui sont transportées par le RTGC à 64 kbit/s. L'annulation d'écho n'est pas autorisée pour les communications de données à 64 kbit/s.

Dans le cas d'une passerelle décomposée en interface avec un réseau SS7, les indications de fourniture d'annulation d'écho sont transportés dans le message de signalisation de l'ISUP, comme spécifié dans l'UIT-T Q.115. Le contrôleur de passerelle média H.323 (MGC, *media gateway controller*) peut interpréter ces informations de signalisation et activer ou désactiver l'annulation d'écho dans la passerelle média (MG, *media gateway*). Dans le cas des communications vocales, le contrôleur MGC peut activer l'annulation d'écho sans effets préjudiciables sur la qualité vocale même si le RTGC a fourni l'annulation d'écho dans son réseau.

Pour les communications de données en bande vocale (par modem) traversant un réseau H.323 ou y aboutissant, la commande d'annulation d'écho est assurée par les modems au moyen de tonalités dans la bande. Aucune signalisation hors bande n'est requise par les éléments de réseau RTGC ou par les contrôleurs MGC.

ANNEXE A

Protocoles RTP/RTCP

On notera que toutes les références de la présente annexe, qui sont des références à une bibliographie, ne sont données qu'à titre d'information, à l'exception de la référence [A-10] (norme ISO/CEI 10646-1), qui apparaît également dans le paragraphe relatif aux références de la présente Recommandation. Dans certains cas, il apparaîtra une référence à l'Appendice I; ces références ne sont données qu'à titre d'information. Tous les détails nécessaires à l'implémentation de l'UIT-T H.323 et de la présente Recommandation figurent dans la présente annexe et dans d'autres annexes et Recommandations ou Normes internationales publiées par l'UIT-T ou l'ISO.

On notera que la présente annexe ne contient pas la spécification primaire et complète des protocoles RTP/RTCP; on trouvera la référence à ces informations à l'Appendice I. La présente annexe ne se rapporte qu'à l'UIT-T H.323 et à la présente Recommandation.

On notera également que la terminologie utilisée dans la présente annexe et celle qui est utilisée dans l'UIT-T H.323 et la présente Recommandation présentent les différences indiquées dans le Tableau A.1.

Tableau A.1/H.225.0 – Correspondance terminologique

Terme utilisé dans H.323 et H.225.0	Terme utilisé dans l'Annexe A (Protocoles RTP/RTCP)
Flux multimédia	Données
Adresse de transport	Adresse de transport
Adresse de réseau à commutation par paquets	Adresse de réseau
Identificateur TSAP	Accès
Annexe A	Spécification ou document

On notera en outre que les "traducteurs" et les "mélangeurs" ne font pas partie du système H.323. Mais les extrémités de système H.323 – Passerelles et unités MCU par exemple – Ayant quelques-unes des caractéristiques des traducteurs et des mélangeurs, ce texte a été retenu pour servir de guide aux personnes chargées de l'implémentation. Toutefois, les traducteurs et les mélangeurs n'étant pas pris en charge dans le système H.323, ces sous-paragraphes ne doivent pas être considérés comme normatifs.

Enfin, il est rappelé aux personnes implémentant le protocole RTP qu'elles doivent impérativement se conformer aux dispositions de la présente Recommandation, y compris des Annexes A, B et C, qui donnent des précisions sur les systèmes H.323/H.225.0. Dans tous les cas, le texte de la présente Recommandation l'emportera sur le texte de la présente annexe et des Annexes B ou C.

A.1 Introduction

La présente annexe spécifie le protocole de transport en temps réel (RTP, *real-time transport protocol*), qui permet d'assurer des services de remise de bout en bout pour les données ayant des caractéristiques de temps réel, comme les données audio et vidéo interactives. Ces services comprennent l'identification du type de charge utile, la numérotation des séquences, l'horodatage et le contrôle de la remise. Les applications emploient généralement le protocole RTP au-dessus du protocole UDP pour utiliser ses services de multiplexage et de contrôle de somme; les deux protocoles contribuent à la fonctionnalité du protocole de transport. Le protocole RTP peut toutefois être utilisé avec d'autres protocoles de transport ou de réseau sous-jacents (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport). Le protocole RTP prend en charge le transfert de données vers plusieurs destinations au moyen de la distribution en mode multidiffusion lorsqu'elle est fournie par le réseau sous-jacent.

On notera que le protocole RTP ne prévoit aucun mécanisme fournissant une garantie de remise en temps utile ou d'autres garanties de qualité de service; ce sont les services de couche inférieure qui doivent fournir ce mécanisme. Le protocole RTP ne garantit pas la remise, n'empêche pas les remises désordonnées, il ne suppose pas non plus que le réseau sous-jacent est fiable et effectue une remise des paquets en séquence. Les numéros de séquence figurant dans l'en-tête des paquets RTP permettent au récepteur de reconstituer la séquence de paquets de l'émetteur; ces numéros peuvent également servir à déterminer l'emplacement exact d'un paquet, dans un décodage vidéo par exemple, sans avoir à décoder les paquets de la séquence.

Le protocole RTP est d'abord conçu pour répondre aux besoins des conférences multimédias multiparticipants, mais il peut servir pour d'autres applications: enregistrement de données continues, simulation répartie interactive, badge actif, applications de commande et de mesure.

La présente Recommandation définit le protocole RTP, qui est constitué de deux parties étroitement liées:

- le protocole de transport en temps réel (RTP), pour transporter des données ayant des caractéristiques de temps réel;

- le protocole de commande RTP (RTCP, *real-time transport control protocol*), pour contrôler la qualité de service et pour acheminer des renseignements sur les participants au cours d'une session. Cette deuxième finalité du protocole RTCP peut suffire pour les sessions "à commande souple", c'est-à-dire pour lesquelles il n'existe pas de procédures explicites de contrôle et de raccordement des membres, mais le but n'est pas nécessairement de prendre en charge toutes les prescriptions de communication de commande d'une application. Cette fonctionnalité peut être entièrement ou partiellement incluse dans un protocole de commande de session distinct, mais cela sort du cadre de la présente Recommandation.

Le protocole RTP représente un nouveau style de protocole qui suit les principes de tramage au niveau application et de traitement de couche intégré proposés par Clark et Tennenhouse [A-1]. Cela signifie que le protocole RTP doit être malléable pour pouvoir fournir les informations requises par une application donnée et qu'il doit être souvent intégré dans le traitement au niveau application et non pas implémenté sous forme de couche distincte. Le protocole RTP est un cadre de protocole qui est délibérément incomplet. La présente Recommandation spécifie les fonctions qui doivent être communes pour toutes les applications utilisant le protocole RTP. A la différence des protocoles classiques pour lesquels on peut accepter des fonctions supplémentaires qui rendent le protocole plus général ou qui ajoutent un mécanisme facultatif nécessitant une analyse, le protocole RTP peut être adapté par des modifications ou des ajouts aux en-têtes, lorsque c'est utile. Des exemples sont donnés au A.5.3, modifications de l'en-tête RTP propres au profil.

Pour spécifier entièrement le protocole RTP pour une application donnée, il faudra donc compléter la présente Recommandation par un ou plusieurs documents associés (voir les Annexes B et C):

- un document de spécification de profil, qui définit un ensemble de codes de type de charge utile et leur mappage avec les formats de charge utile (codages multimédias par exemple). Un profil peut également définir des extensions ou des modifications du protocole RTP qui sont propres à une classe d'applications. Une application ne fonctionnera généralement que conformément à un seul profil. On trouvera à l'Annexe B un profil pour les données audio et vidéo;
- des documents de spécification de format de charge utile, qui définissent la manière dont une charge utile donnée, comme un codage audio ou vidéo, doit être transportée dans le protocole RTP. Voir l'Annexe C.

Plusieurs applications RTP, à la fois expérimentales et commerciales, ont déjà été implémentées à partir des premières spécifications. Ces applications comprennent des outils audio et vidéo ainsi que des outils de diagnostic comme des contrôleurs de trafic. Les utilisateurs de ces outils se comptent par milliers. Toutefois, le réseau Internet actuel ne peut pas encore prendre en charge toute la demande potentielle pour des services en temps réel. Les services à large bande utilisant le protocole RTP, comme les services vidéo, peuvent dégrader gravement la qualité de service des autres services de réseau. Les personnes chargées de la réalisation doivent donc prendre les précautions nécessaires pour limiter l'utilisation accidentelle de large bande. La documentation sur les applications doit indiquer clairement les limites et l'influence possible de l'exploitation de services en temps réel à large bande sur l'exploitation des services Internet et des services d'autres réseaux.

A.2 Scénarios d'utilisation du protocole RTP

Les sous-paragraphes suivants décrivent certaines caractéristiques d'utilisation du protocole RTP. Les exemples ont été choisis de façon à illustrer l'exploitation de base des applications utilisant le protocole RTP, mais ils ne visent pas à limiter les possibilités d'utilisation du protocole RTP. Dans ces exemples, le protocole RTP, utilisé au-dessus des protocoles IP et UDP, suit les conventions établies dans le profil pour les données audio et vidéo spécifié dans l'Annexe B.

A.2.1 Audioconférence simple en mode multidiffusion

Un groupe de travail de l'IETF se réunit pour discuter du projet de protocole sous sa dernière forme, qui utilise les services de multidiffusion IP d'Internet pour les communications vocales. Par l'intermédiaire d'un certain mécanisme d'attribution, la présidence du groupe de travail obtient une adresse de groupe de multidiffusion et deux accès, l'un pour les données audio et l'autre pour les paquets (RTCP) de commande. Ces informations d'adresse et d'accès sont transmises aux participants voulus. Si l'on souhaite que les informations soient confidentielles, les paquets de données et de commande peuvent être chiffrés selon les spécifications de l'UIT-T H.323. Par l'intermédiaire de l'application d'audioconférence, chaque participant à l'audioconférence envoie des données audio par petites tranches de 20 ms de durée par exemple. Chaque tranche de données audio est précédée d'un en-tête RTP; cet en-tête et les données figurent à leur tour dans un paquet UDP. L'en-tête RTP indique le type de codage audio (MIC, MICDA ou LPC) de chaque paquet; les émetteurs peuvent ainsi modifier le codage au cours d'une conférence, par exemple pour pouvoir communiquer avec un nouveau participant qui est raccordé via une liaison à faible largeur de bande ou pour réagir à des indications d'encombrement de réseau.

Comme pour les autres réseaux par paquets, il arrive à Internet de perdre des paquets, de les redemander et de les retransmettre avec des retards variables. Pour faire face à ces dégradations, l'en-tête RTP contient des informations de rythme et un numéro de séquence qui permettent aux récepteurs de reconstituer le rythme produit par la source de sorte que, dans cet exemple, le locuteur produit de façon continue des tranches de données audio de 20 ms. Le rythme est reconstitué séparément pour chaque source de paquets RTP de la conférence. Le numéro de séquence peut également servir au récepteur à évaluer le nombre de paquets perdus.

Etant donné que les membres du groupe de travail peuvent arriver et partir au cours de la conférence, il est utile de connaître les participants à un moment donné et de savoir avec quelle qualité ils reçoivent les données audio. Pour cela, chaque instance de l'application audio dans la conférence diffuse périodiquement un rapport de réception ainsi que le nom de l'utilisateur se trouvant à l'accès (de commande) RTCP. Le rapport de réception, qui indique avec quelle qualité le locuteur du moment est reçu, peut servir à commander des codages adaptatifs. Le nom de l'utilisateur peut être accompagné d'autres informations d'identification sous réserve de satisfaire aux limites de largeur de bande de commande. Un site envoie le paquet RTCP BYE (voir A.6.5, BYE – Paquet RTCP au revoir) lorsqu'il quitte la conférence.

A.2.2 Conférence audio et vidéo

Si les médias audio et vidéo sont tous deux utilisés dans une conférence, ils sont transmis dans des sessions RTP séparées; pour chacun de ces supports, les paquets RTCP sont transmis au moyen de deux couples d'accès UDP différents ou de deux adresses de multidiffusion différentes. Il n'existe pas de couplage direct au niveau RTP entre les sessions audio et vidéo, mis à part le fait qu'un utilisateur participant aux deux sessions doit utiliser le même nom (canonique) distinctif dans les paquets RTCP afin de pouvoir associer ces deux sessions.

L'une des raisons de cette séparation est de permettre aux participants à la conférence qui le souhaitent de ne recevoir qu'un seul support d'information. On trouvera plus de détails au A.5.2, Sessions RTP avec multiplexage des données. Malgré la séparation, il est possible de synchroniser les données audio et vidéo d'une source grâce aux informations de rythme figurant dans les paquets RTCP pour les deux sessions.

A.2.3 Mélangeurs et traducteurs

Jusqu'ici, nous avons supposé que tous les sites souhaitent recevoir les données de média dans le même format. Toutefois, cette hypothèse ne convient pas dans tous les cas. Considérons le cas où les participants d'une zone sont raccordés via une liaison à faible vitesse à la majorité des participants à la conférence qui bénéficient d'un accès au réseau à vitesse élevée. Au lieu de forcer tout le monde à utiliser une largeur de bande moins élevée et un codage audio de qualité réduite, un relais au niveau

RTP, appelé mélangeur, peut être placé près de la zone à faible largeur de bande. Ce mélangeur resynchronise les paquets audio entrants pour reconstituer l'espacement constant de 20 ms produit par l'émetteur, mélange ces flux audio reconstitués pour ne former qu'un seul flux, traduit le codage audio en un codage à largeur de bande plus faible et transmet le flux de paquets à largeur de bande plus faible sur la liaison à faible vitesse. Ces paquets peuvent être monodiffusés vers un seul destinataire ou multidiffusés à une adresse différente vers plusieurs destinataires. L'en-tête RTP doit permettre aux mélangeurs d'identifier les sources qui ont contribué à un paquet mélangé de façon à pouvoir fournir une indication correcte de locuteur aux récepteurs.

Certains participants prévus à l'audioconférence peuvent être raccordés par l'intermédiaire de liaisons à large bande mais il est possible qu'on ne puisse pas les joindre directement via la multidiffusion IP. Ils peuvent par exemple se trouver derrière une barrière au niveau application qui ne laisse passer aucun paquet IP. Pour ces sites, on n'utilise pas de mélangeur, mais on a recours à un autre type de relais au niveau RTP, appelé traducteur. Deux traducteurs sont installés, un de chaque côté de la barrière; celui qui se trouve à l'extérieur envoie tous les paquets de multidiffusion reçus sur une connexion sûre en forme d'entonnoir vers le traducteur se trouvant à l'intérieur. Ce traducteur-ci renvoie ces paquets sous forme de paquets de multidiffusion à un groupe de multidiffusion restreint au réseau interne du site.

Les mélangeurs et les traducteurs peuvent être conçus pour diverses applications. Citons par exemple le cas d'un mélangeur vidéo qui met les images d'une personne donnée dans différents flux vidéo après leur avoir fait subir un changement d'échelle et qui compose ces flux dans un seul flux vidéo pour simuler une scène de groupe. Parmi les exemples de traduction, citons le raccordement d'un groupe de serveurs n'ayant que les protocoles IP/UDP à un groupe de serveurs ne comprenant que le protocole ST-II, ou encore la traduction de flux vidéo provenant de sources distinctes en codage paquet par paquet sans resynchronisation ni mélange. On trouvera des détails sur le fonctionnement des mélangeurs et des traducteurs au A.7, Traducteurs et mélangeurs RTP.

A.3 Définitions

La présente annexe définit les termes suivants:

A.3.1 charge utile RTP: données transportées par le protocole RTP dans un paquet, par exemple des échantillons audio ou des données vidéo compressées. Le format de charge utile et son interprétation sortent de la présente Recommandation.

A.3.2 paquet RTP: paquet de données comprenant l'en-tête RTP fixe, une liste de sources contributives éventuellement vide (voir plus loin) et des données de charge utile. Certains protocoles sous-jacents peuvent nécessiter une encapsulation du paquet RTP à définir. Un paquet du protocole sous-jacent contient généralement un seul paquet RTP, mais il peut en contenir plusieurs si la méthode d'encapsulation le permet (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport).

A.3.3 paquet RTCP: paquet de commande comportant une partie d'en-tête fixe semblable à celle des paquets de données RTP, suivie d'éléments structurés qui varient selon le type de paquet RTCP. Les formats sont définis au A.6, protocole de commande RTP – Protocole RTCP. Généralement, plusieurs paquets RTCP sont envoyés ensemble sous la forme d'un paquet RTCP composé dans un unique paquet du protocole sous-jacent; cela est permis par le champ de longueur se trouvant dans l'en-tête fixe de chaque paquet RTCP.

A.3.4 accès: "abstraction que les protocoles de transport utilisent pour faire la distinction entre plusieurs destinations à l'intérieur d'un serveur donné. Les protocoles TCP/IP identifient les accès au moyen de petits entiers" positifs [A-2]. Les sélecteurs de transport (TSEL, *transport selectors*) utilisés par la couche Transport OSI sont équivalents à des accès. Le protocole RTP s'appuie sur le protocole de couche inférieure pour fournir un mécanisme (des accès par exemple) permettant de multiplexer les paquets RTP et RTCP d'une session.

A.3.5 adresse de transport: combinaison d'une adresse de réseau et d'un accès qui identifie une extrémité au niveau transport, par exemple une adresse IP et un accès UDP. Les paquets sont transmis d'une adresse de transport de source vers une adresse de transport de destination.

A.3.6 session RTP: association de participants communiquant grâce au protocole RTP. Pour chaque participant, la session est définie par un couple d'adresses de transport de destination (une adresse de réseau plus un couple d'accès pour les protocoles RTP et RTCP). Le couple d'adresses de transport de destination peut être commun à tous les participants, comme dans le cas de la multidiffusion IP, ou chaque participant peut avoir un couple d'adresses différent, comme dans le cas des accès et des adresses de réseau d'unidiffusion individuels. Dans une session multimédia, chaque média est transporté dans une session RTP distincte avec ses propres paquets RTCP. On peut faire la distinction entre les diverses sessions RTP grâce à des couples de numéros d'accès différents ou à des adresses de multidiffusion différentes.

A.3.7 source de synchronisation (SSRC, *synchronization source*): source d'un flux de paquets RTP, caractérisée par un identificateur SSRC numérique à 32 bits figurant dans l'en-tête RTP de sorte qu'il ne dépende pas de l'adresse de réseau. Tous les paquets provenant d'une source de synchronisation appartiennent au même espace de rythme et de numéro de séquence, les récepteurs peuvent donc regrouper les paquets par source de synchronisation avant de les lire. Parmi les exemples de sources de synchronisation, citons un mélangeur RTP (voir plus loin) ou encore un émetteur qui envoie un flux de paquets découlant d'une source de signaux comme un microphone ou une caméra. Une source de synchronisation peut modifier son format de données, codage audio par exemple, au cours du temps. L'identificateur SSRC est une valeur choisie au hasard qui est globalement unique dans une session RTP donnée (voir A.8, Attribution et utilisation des identificateurs SSRC). Les participants n'ont pas besoin d'utiliser le même identificateur SSRC pour toutes les sessions RTP d'une session multimédia; le lien entre les différents identificateurs SSRC est assuré par le protocole RTCP (voir A.6.4.1, CNAME – Élément SDES identificateur d'extrémité canonique). Si un participant génère plusieurs flux dans une même session RTP, à partir de diverses caméras vidéo par exemple, chaque flux doit être identifié comme provenant d'une source SSRC différente.

A.3.8 source contributive (CSRC, *contributing source*): source d'un flux de paquets RTP qui contribue au flux combiné produit par un mélangeur RTP (voir plus loin). Le mélangeur insère la liste des identificateurs SSRC des sources qui contribuent à la génération d'un paquet donné, dans l'en-tête RTP de ce paquet. Cette liste s'appelle la liste CSRC. Comme exemple d'application, citons une audioconférence dans laquelle un mélangeur indique tous les locuteurs dont les paroles ont été combinées pour produire le paquet sortant, ce qui permet au récepteur d'indiquer le locuteur du moment, même si tous les paquets audio contiennent le même identificateur SSRC (celui du mélangeur).

A.3.9 système d'extrémité: application qui génère le contenu à envoyer dans des paquets RTP ou qui consomme le contenu des paquets RTP reçus. Un système d'extrémité joue généralement le rôle d'une seule source de synchronisation dans une session RTP donnée, mais il peut aussi jouer le rôle de plusieurs sources.

A.3.10 mélangeur: système intermédiaire qui reçoit des paquets RTP en provenance d'une ou plusieurs sources, qui modifie éventuellement le format des données, qui combine les paquets d'une certaine manière et qui transmet ensuite un nouveau paquet RTP. Etant donné que le rythme parmi les différentes sources d'entrée ne doit être généralement pas synchronisé, le mélangeur procédera à des ajustements de rythme entre les flux et produira son propre rythme pour le flux combiné. On considérera alors que la source de synchronisation de tous les paquets de données provenant d'un mélangeur est le mélangeur.

A.3.11 traducteur: système intermédiaire qui transmet des paquets RTP avec leur identificateur de source de synchronisation intact. Exemples de traducteurs: dispositifs convertissant les codages sans

effectuer de mélange, dispositifs de passage de la multidiffusion à l'unidiffusion, filtres au niveau application dans les barrières.

A.3.12 contrôleur: application qui reçoit les paquets RTCP envoyés par les participants à une session RTP, en particulier les rapports de perception, et qui évalue la qualité de service courante aux fins de contrôle de la distribution, de diagnostic des défauts et de statistiques à long terme. La fonction de contrôle devrait être intégrée dans la ou les applications participant à la session, mais elle pourra aussi constituer une application distincte qui ne participe pas à la session et qui n'envoie pas de paquets de données RTP et n'en reçoit pas. On parle alors de contrôleurs tiers.

A.3.13 moyens non RTP: protocoles et mécanismes qui peuvent être nécessaires en plus du protocole RTP pour fournir un service facile à utiliser. Pour les conférences multimédias en particulier, une application de commande de conférence peut distribuer des adresses de multidiffusion et des clés de chiffrement, négocier l'algorithme de chiffrement à utiliser et définir les mappages dynamiques entre les valeurs de type de charge utile RTP et les formats de charge utile pour lesquels aucune valeur de type de charge utile n'est prédéfinie. Pour les applications simples, on peut également utiliser le courrier électronique ou une base de données de conférence. La spécification de ces protocoles et mécanismes sort du cadre de la présente Recommandation.

A.4 Ordre des octets, alignement et format temporel

Tous les champs d'entiers sont ordonnés dans l'ordre réseau des octets, c'est-à-dire l'octet le plus significatif en premier. Cet ordre des octets est appelé en anglais "big-endian". L'ordre de transmission est décrit en détail dans la référence [A-3]. Sauf indication contraire, les constantes numériques sont décimales (base 10).

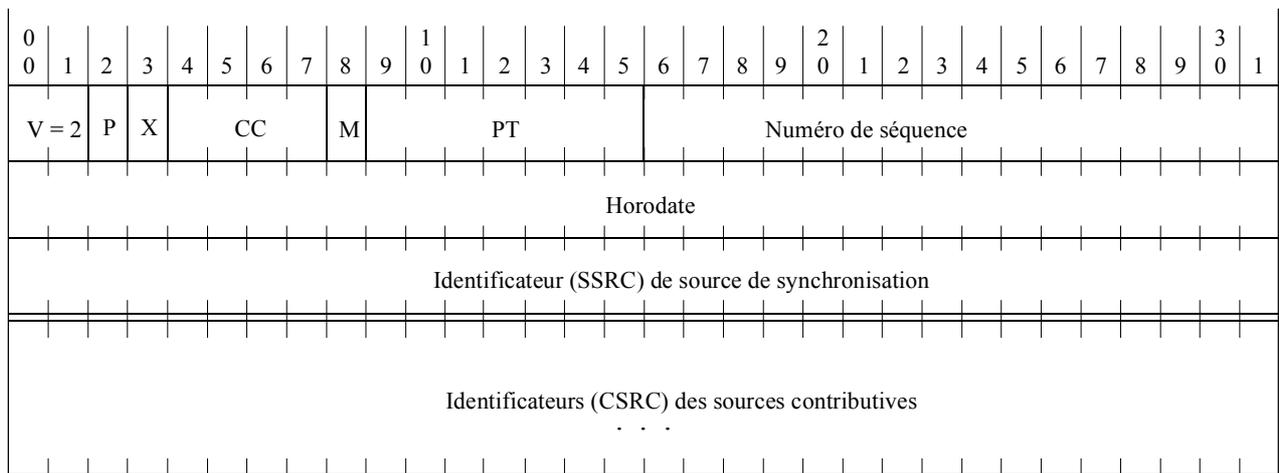
Toutes les données d'en-tête sont alignées sur leur longueur naturelle, c'est-à-dire que les champs à 16 bits sont alignés sur des décalages pairs, les champs à 32 bits sont alignés sur des décalages divisibles par quatre, etc. Les octets désignés comme étant des octets de bourrage ont la valeur zéro.

L'heure lue sur une horloge murale (heure absolue) est représentée à l'aide du format d'horodate du protocole temporel de réseau (NTP, *network time protocol*), qui est en secondes par rapport à 0h UTC du 1^{er} janvier 1900 [A-4]. L'horodate NTP à résolution complète est un nombre à virgule fixe non signé sur 64 bits, la partie entière se trouvant dans les 32 premiers bits et la partie décimale dans les 32 derniers bits. Dans certains champs où il convient d'utiliser une représentation plus compacte, seuls les 32 bits du milieu sont utilisés: à savoir les 16 bits inférieurs de la partie entière et les 16 bits supérieurs de la partie décimale. Les 16 bits supérieurs de la partie entière doivent être déterminés de façon indépendante.

A.5 Protocole de transfert de données RTP

A.5.1 Champs de l'en-tête fixe RTP

L'en-tête RTP a le format suivant:



T1527560-97

Les douze premiers octets figurent dans chaque paquet RTP, alors que la liste des identificateurs CSRC n'est présente que lorsqu'elle est insérée par un mélangeur. Les champs ont la signification suivante:

version (V): 2 bits. Ce champ identifie la version du protocole RTP. La version définie par la présente Recommandation correspond à 2. (La valeur 1 est utilisée par le projet de première version du protocole RTP et la valeur 0 est utilisée par le protocole qui a été implémenté au départ dans l'outil audio "vat".)

bouillage (P) (padding): 1 bit. Si le bit de bouillage est positionné, le paquet contient un ou plusieurs octets supplémentaires de bouillage à la fin qui ne font pas partie de la charge utile. Le dernier octet de bouillage indique le nombre d'octets de bouillage qu'il faut ignorer. Le bouillage peut être nécessaire pour certains algorithmes de chiffrement avec des tailles de bloc fixes ou pour le transport de plusieurs paquets RTP dans une unité de données protocolaires de couche inférieure.

extension (X): 1 bit. Si le bit d'extension est positionné, l'en-tête fixe est suivi d'une extension d'en-tête et d'une seule, avec le format défini au A.5.3, Modifications de l'en-tête RTP propres au profil.

compte CSRC (CC): 4 bits. Le compte CSRC contient le nombre d'identificateurs CSRC qui suivent l'en-tête fixe.

marqueur (M): 1 bit. L'interprétation du marqueur est définie dans les profils. Il sert à repérer des événements significatifs, comme les frontières de trame, dans le flux de paquets. Chaque profil peut définir des bits de marqueur supplémentaires ou spécifier qu'il n'y a pas de bit de marqueur en modifiant le nombre de bits dans le champ de type de charge utile (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

type de charge utile (PT): 7 bits. Ce champ identifie le format de la charge utile RTP et détermine la façon dont l'application doit l'interpréter. Chaque profil spécifie un mappage statique par défaut entre les codes de type de charge utile et les formats de charge utile. D'autres codes de type de charge utile peuvent être définis dynamiquement grâce aux moyens non RTP (voir A.3, Définitions). L'Annexe B spécifie un ensemble initial de mappages par défaut pour les données audio et vidéo. A un instant donné, un émetteur RTP transmet un type de charge utile RTP et un seul; ce champ n'est pas prévu pour le multiplexage de divers flux de média (voir A.5.2, Sessions RTP avec multiplexage des données).

numéro de séquence: 16 bits. Le numéro de séquence, incrémenté d'une unité à chaque paquet de données RTP envoyé, peut servir au récepteur à détecter les pertes de paquets et à restaurer la séquence de paquets. La valeur initiale du numéro de séquence est aléatoire (imprévisible) pour rendre plus difficiles les attaques en clair connues contre le chiffrement, même si la source n'effectue pas de chiffrement, car les paquets peuvent passer dans un traducteur qui effectue des chiffrements.

Les techniques permettant de choisir des nombres imprévisibles sont abordées dans la référence [A-5].

horodate: 32 bits. L'horodate correspond à l'instant d'échantillonnage du premier octet dans le paquet de données RTP. Cet instant doit être lu sur une horloge indiquant des dates croissantes et espacées linéairement dans le temps pour permettre la synchronisation et les calculs de gigue (voir A.6.3.1, SR: paquet RTCP rapport de l'émetteur). La résolution de l'horloge doit être suffisante pour la précision de synchronisation souhaitée et pour la mesure de la gigue d'arrivée des paquets (un top d'horloge par trame vidéo n'est généralement pas suffisant). La fréquence de l'horloge, fonction du format des données de la charge utile, est spécifiée statiquement dans le profil ou dans la spécification de format de charge utile qui définit le format, ou peut être spécifiée dynamiquement pour les formats de charge utile définis par des moyens non-RTP. Si les paquets RTP sont produits périodiquement, il faut utiliser l'instant d'échantillonnage nominal déterminé à partir de l'horloge d'échantillonnage, et non pas lire l'horloge du système. Pour des données audio à débit fixe par exemple, il devrait y avoir un nouveau top d'horloge à chaque période d'échantillonnage. Si une application audio lit des blocs couvrant 160 périodes d'échantillonnage et provenant du dispositif d'entrée, l'estampille horaire doit être incrémentée de 160 à chaque bloc, que le bloc soit transmis dans un paquet ou qu'il soit éliminé en tant que silence.

La valeur initiale de l'horodate est aléatoire, comme pour le numéro de séquence. Plusieurs paquets RTP consécutifs peuvent avoir des horodates égales s'ils sont produits (logiquement) en même temps, par exemple s'ils appartiennent à la même trame vidéo. L'horodate d'un paquet RTP peut être aussi bien postérieure qu'antérieure à l'horodate du paquet RTP consécutif si les données ne sont pas transmises dans leur ordre d'échantillonnage, comme c'est le cas pour les trames vidéo interpolées MPEG. (Mais le numéro de séquence des paquets transmis doit être toujours monotone croissante.)

SSRC: 32 bits. Le champ SSRC identifie la source de synchronisation. Cet identificateur est choisi au hasard, avec comme intention que deux sources de synchronisation quelconques à l'intérieur d'une même session RTP aient des identificateurs SSRC différents. Le paragraphe I.6 présente un exemple d'algorithme générant un identificateur aléatoire. La probabilité pour que plusieurs sources choisissent le même identificateur est faible, mais toutes les implémentations du protocole RTP doivent prévoir la détection et la résolution des collisions. Le paragraphe A.8, Attribution et utilisation des identificateurs SSRC, donne la probabilité de collision et décrit un mécanisme permettant de résoudre les collisions et de détecter les boucles au niveau RTP sur la base de l'unicité des identificateurs SSRC. Si une source change d'adresse de transport, elle doit aussi choisir un nouvel identificateur SSRC pour éviter d'être interprétée comme une source bouclée.

liste CSRC: 0 à 15 éléments, 32 bits chacun. La liste CSRC identifie les sources qui ont contribué à la charge utile du paquet. Le nombre d'identificateurs est donné par le champ CC. Si le nombre de sources contributives est supérieur à 15, seules 15 peuvent être identifiées. Les mélangeurs insèrent des identificateurs CSRC, en utilisant les identificateurs SSRC des sources contributives. Pour les paquets audio par exemple, on énumère les identificateurs SSRC de toutes les sources qui ont été mélangées ensemble pour créer un paquet, ce qui permet de fournir une indication correcte de locuteur au récepteur.

A.5.2 Sessions RTP avec multiplexage des données

Pour un fonctionnement efficace du protocole, le nombre de points de multiplexage doit être réduit au minimum, comme le décrit le principe de traitement de couche intégré [A-1]. Dans le protocole RTP, le multiplexage est permis par l'adresse de transport de destination (adresse de réseau et numéro d'accès) qui définit une session RTP. Par exemple, dans une téléconférence composée de média audio et vidéo codés séparément, chaque médium doit être transporté dans une session RTP distincte avec sa propre adresse de transport de destination. Il n'est pas prévu de transporter les données audio et vidéo dans une seule session RTP et de les démultiplexer en fonction du type de

charge utile ou des champs SSRC. L'entrelacement de paquets avec des types de charge utile différents mais utilisant le même identificateur SSRC serait à l'origine de plusieurs problèmes:

- 1) si l'un des types de charge utile changeait au cours d'une session, il n'y aurait pas de moyen général qui permettrait de déterminer l'ancienne valeur que la nouvelle valeur a remplacée;
- 2) par définition, un identificateur SSRC désigne un unique espace de rythme et de numéro de séquence. Si divers types de charge utile sont entrelacés, il faudrait alors différents espaces de rythme si les fréquences d'horloge diffèrent selon les médias et il faudrait différents espaces de numéro de séquence pour savoir quel type de charge utile a subi une perte de paquet;
- 3) les rapports d'émetteur et de récepteur RTCP (voir A.6.3, Rapports d'émetteur et de récepteur) ne peuvent décrire qu'un seul espace de rythme et de numéro de séquence par identificateur SSRC et ne comportent pas de champ de type de charge utile;
- 4) un mélangeur RTP n'est pas en mesure de combiner des flux entrelacés de média incompatibles en un seul flux;
- 5) le transport de plusieurs médias dans une même session RTP empêche:
 - l'utilisation de conduits de réseau différents ou d'attributions de ressources de réseau différentes lorsque c'est possible;
 - la réception d'une partie seulement des médias lorsqu'on le souhaite, par exemple la réception des données audio uniquement dans le cas où les données vidéo dépasseraient la largeur de bande disponible;
 - l'implémentation de récepteurs qui utilisent des processus distincts pour les différents médias,alors que le recours à des sessions RTP distinctes permet des implémentations avec au choix un seul ou plusieurs processus;

L'utilisation d'un identificateur SSRC différent pour chaque support d'information et l'envoi de tous les médias dans la même session RTP permettraient d'éviter les trois premiers problèmes mais pas les deux derniers.

A.5.3 Modifications de l'en-tête RTP propres au profil

L'en-tête de paquet de données RTP existant passe pour être complet pour l'ensemble des fonctions communes requises pour toutes les classes d'application que le protocole RTP est susceptible de prendre en charge. Toutefois, conformément au principe de conception ALF, l'en-tête peut être adapté par des modifications ou des ajouts qui sont définis dans une spécification de profil et qui n'entravent pas le fonctionnement des outils d'enregistrement et de contrôle indépendants du profil:

- le bit de marqueur et le champ de type de charge utile contiennent des informations propres au profil, mais ces champs ont été affectés dans l'en-tête fixe car ils doivent être vraisemblablement nécessaires à de nombreuses applications et auraient nécessité sinon un autre mot de 32 bits uniquement pour les contenir. Les profils peuvent redéfinir l'octet comportant ces champs pour répondre à diverses prescriptions, avec par exemple un nombre plus grand ou plus petit de marqueurs. Si l'octet comporte des marqueurs, l'un de ceux-ci occupera la position du bit le plus significatif, étant donné qu'il est possible que les contrôleurs indépendants du profil puissent établir une corrélation entre les pertes de paquets et le bit de marqueur;
- les informations supplémentaires qui sont nécessaires pour un format de charge utile donné, comme un codage vidéo par exemple, doivent figurer dans la section de charge utile du paquet. Ces informations pourraient se trouver dans un en-tête qui est toujours présent au début de la section de charge utile, ou pourraient être indiquées par une valeur réservée dans la séquence de données;

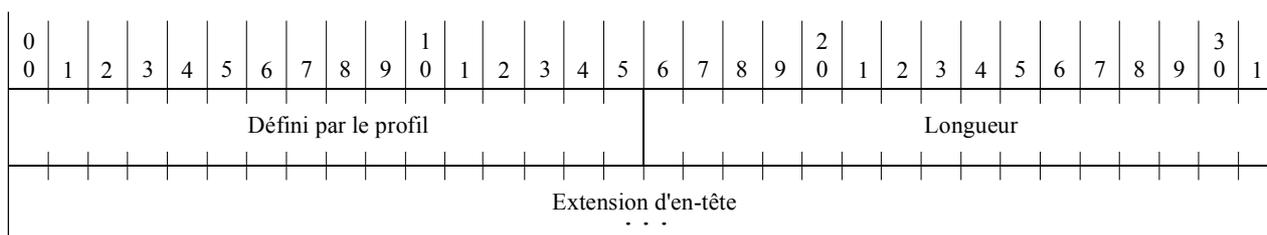
- si une classe d'applications particulière a besoin d'une fonctionnalité supplémentaire indépendante du format de charge utile, le profil de ces applications doit définir des champs fixes supplémentaires qui viennent immédiatement à la suite du champ SSRC de l'en-tête fixe existant. Ces applications pourront accéder rapidement et directement à ces champs supplémentaires tandis que les enregistreurs ou les contrôleurs indépendants du profil pourront continuer à traiter les paquets RTP en n'interprétant que les douze premiers octets.

Si une fonctionnalité supplémentaire devient nécessaire pour tous les profils, une nouvelle version du protocole RTP doit alors être définie afin d'effectuer une modification permanente de l'en-tête fixe.

A.5.3.1 Extension de l'en-tête RTP

Un mécanisme d'extension est fourni pour permettre d'expérimenter des réalisations particulières dans lesquelles sont intégrées de nouvelles fonctions qui sont indépendantes du format de charge utile et qui nécessitent des informations supplémentaires dans l'en-tête de paquet de données RTP. Ce mécanisme est conçu de sorte que l'extension d'en-tête puisse être ignorée par les autres implémentations en interfonctionnement qui n'ont pas été étendues.

On notera que cette extension d'en-tête n'est prévue que pour une utilisation limitée. Pour la plupart des utilisations potentielles de ce mécanisme, il vaudrait mieux ne pas recourir à ce mécanisme et d'utiliser les méthodes décrites au paragraphe précédent. Par exemple, la réalisation d'une extension de l'en-tête fixe qui soit propre au profil est moins coûteuse car cette extension n'est pas conditionnelle et elle n'est pas placée dans une position variable. Les informations supplémentaires requises pour un format de charge utile donné ne doivent pas figurer dans une telle extension d'en-tête, mais elles doivent au contraire figurer dans la section de charge utile du paquet:



T1527570-97

Si le bit X de l'en-tête RTP vaut un, une extension d'en-tête de longueur variable est rattachée à l'en-tête RTP, à la suite de la liste CSRC si elle est présente. L'extension d'en-tête contient un champ de 16 bits qui indique le nombre de mots de 32 bits figurant dans l'extension, non compris les quatre octets que compte l'en-tête d'extension (zéro est donc une longueur valide). Une seule extension peut être rattachée à l'en-tête RTP. Pour permettre à diverses implémentations en interfonctionnement d'être expérimentées indépendamment avec des extensions d'en-tête différentes, ou pour permettre à une implémentation donnée d'être expérimentée avec plusieurs types d'extension d'en-tête, les 16 premiers bits de l'extension d'en-tête sont laissés libres pour pouvoir indiquer s'il s'agit d'identificateurs ou de paramètres. Le format de ces 16 bits doit être défini par la spécification du profil des implémentations. La présente spécification RTP ne définit pas d'extension d'en-tête.

A.6 Protocole de commande RTP (RTCP)

Le protocole de commande RTP (RTCP), basé sur la transmission périodique de paquets de commande à tous les participants à une session, utilise le même mécanisme de distribution que celui utilisé pour les paquets de données. Le protocole sous-jacent doit assurer le multiplexage des paquets de données et de commande, par exemple en utilisant des numéros d'accès distincts et le protocole UDP. Le protocole RTCP exécute quatre fonctions:

- 1) la première fonction consiste à fournir un retour sur la qualité de distribution des données. Cette fonction fait partie intégrante du rôle du protocole RTP étant donné qu'il s'agit d'un protocole de transport, et cette fonction est liée aux fonctions de commande de flux et d'encombrement appartenant aux autres protocoles de transport. Le retour peut être directement utile pour commander des codages adaptatifs [A-6], [A-7], mais des expériences menées avec la multidiffusion IP ont montré qu'il est également très important d'obtenir un retour des récepteurs pour diagnostiquer les défauts dans la distribution. L'envoi de rapports de retour de réception à tous les participants permet à celui qui observe les problèmes de déterminer s'il s'agit de problèmes locaux ou globaux. Avec un mécanisme de distribution comme la multidiffusion IP, il est également possible à une entité qui ne participe pas à la session – Un fournisseur de service de réseau par exemple – De recevoir les informations de retour et de jouer le rôle d'un contrôleur tiers pour diagnostiquer les problèmes de réseau. Cette fonction de retour est réalisée par les rapports d'émetteur et de récepteur RTCP, qui sont décrits au A.6.3, Rapports d'émetteur et de récepteur;
- 2) les paquets RTCP contiennent de façon permanente un identificateur au niveau transport caractérisant une source RTP, cet identificateur s'appelle nom canonique ou CNAME (voir A.6.4.1, CNAME – Élément SDES identificateur d'extrémité canonique). Étant donné que l'identificateur SSRC peut changer en cas de découverte d'un conflit ou de redémarrage d'un programme, les récepteurs exigent que le CNAME ne perde pas la trace de chaque participant. Ils exigent aussi que le CNAME associe plusieurs flux de données provenant d'un participant donné en un ensemble de sessions RTP connexes, par exemple pour synchroniser les données audio et vidéo;
- 3) pour les deux premières fonctions, il est nécessaire que tous les participants envoient des paquets RTCP; il faut donc que la fréquence d'envoi soit commandée pour que le protocole RTP puisse s'étendre à un grand nombre de participants. Envoyant ses paquets de commande à tous les autres, chaque participant peut observer de façon indépendante le nombre de participants. Ce nombre sert à calculer la fréquence d'envoi des paquets, comme l'explique A.6.2, Intervalle de transmission RTCP;
- 4) une quatrième fonction, facultative, consiste à acheminer les informations de commande de session minimales, par exemple l'identification du participant qui doit être affichée à l'interface utilisateur. Cette fonction devrait surtout trouver son utilité dans les sessions "à commande souple" où les participants rejoignent et quittent la session sans contrôle de membre et sans négociation des paramètres. Le protocole RTCP sert de canal pour atteindre tous les participants, mais il n'est pas nécessairement prévu pour prendre en charge toutes les prescriptions de communication de commande d'une application. Un protocole de commande de session de niveau supérieur, qui sort du cadre de la présente Recommandation, peut être nécessaire.

Les fonctions 1 à 3 sont obligatoires en cas d'utilisation du protocole RTP dans l'environnement de multidiffusion IP et elles sont recommandées pour tous les autres environnements. Il est conseillé aux concepteurs d'application RTP d'éviter les mécanismes qui ne peuvent fonctionner qu'en mode unidiffusion et qui ne pourront être étendus à de plus grands nombres de participants.

A.6.1 Format de paquet RTCP

La présente spécification définit plusieurs types de paquet RTCP qui comportent diverses informations de commande:

- **SR**: rapport d'émetteur (*sender report*), pour les statistiques de transmission et de réception des participants qui sont des émetteurs actifs;
- **RR**: rapport de récepteur, pour les statistiques de réception des participants qui ne sont pas des émetteurs actifs;
- **SDES**: éléments de description (*source description*) de source, y compris le CNAME;

- **BYE**: indication de fin de participation;
- **APP**: fonctions propres à l'application.

Chaque paquet RTCP commence par une partie fixe semblable à celle des paquets de données RTP, suivie d'éléments structurés qui peuvent être de longueur variable selon le type de paquet mais la fin du paquet se trouve toujours sur une frontière à 32 bits. Le respect de la prescription d'alignement et l'inclusion d'un champ de longueur dans la partie fixe permettent "d'empiler" les paquets RTCP. Plusieurs paquets RTCP peuvent être concaténés sans séparateur pour former un paquet RTCP composé qui est envoyé dans un même paquet du protocole de couche inférieure, par exemple du protocole UDP. Les paquets RTCP contenus dans le paquet composé ne sont pas dénombrés explicitement puisqu'il est prévu que les protocoles de couche inférieure fournissent une longueur globale pour déterminer la fin du paquet composé.

Chaque paquet RTCP contenu dans le paquet composé peut être traité indépendamment, sans prescription ni sur l'ordre ni sur la combinaison des paquets. Toutefois, la réalisation des fonctions du protocole est soumise aux contraintes suivantes:

- les statistiques de réception (dans les paquets SR ou RR) doivent être transmises aussi souvent que le permettront les contraintes de largeur de bande afin de maximiser la résolution des statistiques; chaque paquet RTCP composé transmis périodiquement doit donc comporter un paquet de rapport;
- les nouveaux récepteurs ont besoin de recevoir le CNAME d'une source le plus tôt possible pour identifier la source et pour commencer à associer les médias à des fins de postsynchronisation par exemple, chaque paquet RTCP composé doit également comporter l'élément SDES CNAME;
- le nombre de types de paquet qui peuvent apparaître en premier dans le paquet composé doit être limité pour augmenter le nombre de bits constants dans le premier mot et la probabilité de valider avec succès les paquets RTCP et de ne pas valider les paquets de données RTP ou les autres paquets non connexes dont l'adresse est incorrecte.

Il faut donc que tous les paquets RTCP soient envoyés dans un paquet composé contenant au moins deux paquets distincts, avec le format recommandé suivant:

préfixe de chiffrement: si le paquet composé doit être chiffré et seulement dans ce cas, il doit avoir un préfixe aléatoire de 32 bits, différent pour chaque paquet composé transmis.

SR ou RR: le premier paquet RTCP contenu dans le paquet composé doit toujours être un paquet de rapport afin de faciliter la validation d'en-tête décrite au A.2. Cela est vrai même si aucune donnée n'est envoyée ni reçue, auquel cas un paquet RR vide est envoyé, et même si le seul autre paquet RTCP contenu dans le paquet composé est un paquet BYE.

RR supplémentaires: si le nombre de sources faisant l'objet d'un rapport sur les statistiques de réception dépasse 31 – Nombre maximal pour un seul paquet SR ou RR – Alors des paquets RR supplémentaires doivent suivre le paquet de rapport initial.

SDES: un paquet SDES contenant un élément CNAME doit figurer dans chaque paquet RTCP composé. D'autres éléments de description de source peuvent facultativement être inclus s'ils sont requis par une application donnée, sous réserve de satisfaire les contraintes de largeur de bande (voir A.6.2.2, Attribution de largeur de bande pour la description de source).

BYE ou APP: d'autres types de paquet RTCP, y compris ceux qu'il reste à définir, peuvent venir à la suite dans un ordre quelconque, à l'exception du paquet BYE qui doit être le dernier paquet envoyé avec un identificateur SSRC/CSRC donné. Chaque type de paquet peut apparaître plusieurs fois.

Il est recommandé, chaque fois que possible, que les traducteurs et les mélangeurs combinent les paquets RTCP provenant de plusieurs sources et transmis sous forme d'un seul paquet composé afin de limiter le surdébit de paquet (voir A.7, Traducteurs et mélangeurs RTP). La Figure A.1 montre un exemple de paquet composé RTCP pouvant être produit par un mélangeur. Dans le cas où la

longueur totale d'un paquet composé dépasserait l'unité de transmission maximale (MTU, *maximum transmission unit*) du conduit de réseau, ce paquet peut être segmenté en plusieurs paquets composés plus courts à transmettre dans des paquets distincts du protocole sous-jacent. On notera que chacun des paquets composés doit commencer par un paquet SR ou RR.

Une implémentation peut ignorer les paquets RTCP entrants dont elle ne connaît pas les types. Des types de paquets RTCP supplémentaires peuvent être enregistrés auprès de l'Autorité chargée de l'assignation des numéros Internet (IANA, *Internet assigned numbers authority*).

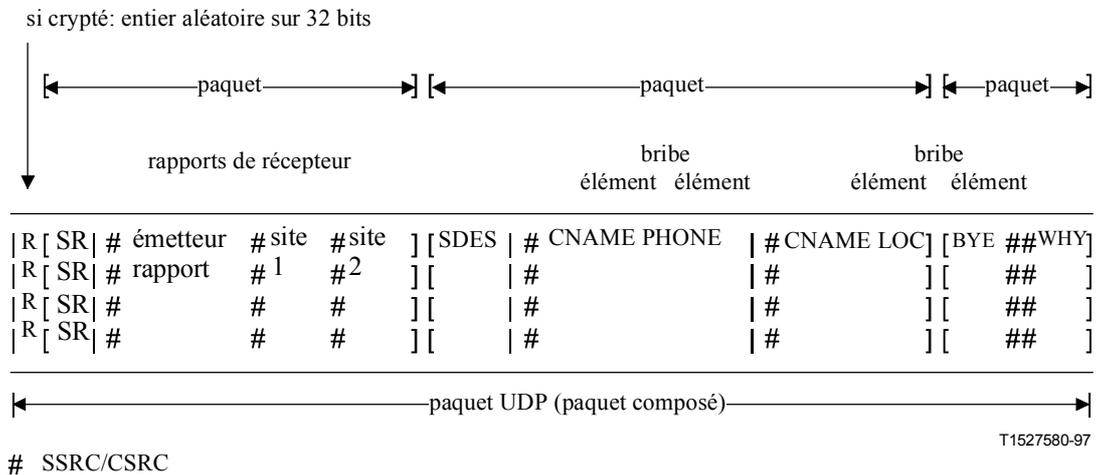


Figure A.1/H.225.0 – Exemple de paquet composé RTCP

A.6.2 Intervalle de transmission RTCP

Le protocole RTP est conçu de telle sorte qu'une application puisse s'adapter automatiquement à des tailles de session allant de quelques participants à des milliers de participants. Dans une audioconférence par exemple, le trafic des données est forcément autolimité car deux personnes au plus parleront en même temps; avec une distribution en mode multidiffusion, le débit des données sur une liaison donnée quelconque restera donc relativement constant quel que soit le nombre de participants. En revanche, le trafic de commande n'est pas autolimité. Si les rapports de réception de chaque participant sont envoyés à débit constant, le trafic de commande augmentera linéairement en fonction du nombre de participants. Il faut donc réduire ce débit.

Pour chaque session, on suppose que le trafic de données ne doit pas dépasser une limite globale appelée "largeur de bande de session" qui doit être répartie entre les participants. Cette largeur de bande peut être réservée et la limite imposée par le réseau, ou il peut simplement s'agir d'un partage raisonnable. La largeur de bande de session peut être choisie sur la base de certains coûts ou d'une connaissance a priori de la largeur de bande de réseau disponible pour la session. Elle est en quelque sorte indépendante du codage du média, mais le choix du codage peut être limité par la largeur de bande de la session. Une application de gestion de session devra fournir le paramètre de largeur de bande de session lorsqu'elle invoquera une application de média, mais les applications de média peuvent aussi fixer un paramètre par défaut en fonction de la largeur de bande correspondant aux données d'un seul émetteur et au codage choisi pour la session. L'application peut aussi imposer des limites de largeur de bande sur la base de règles s'appliquant à la multidiffusion ou sur la base d'autres critères.

Les calculs de largeur de bande pour le trafic de commande et de données tiennent compte des protocoles de réseau et de transport de couche inférieure (UDP et IP par exemple) étant donné qu'il s'agit des informations que le système de réservation des ressources a besoin de connaître. L'application devrait également savoir lequel de ces protocoles est utilisé. Dans le calcul, il n'est pas

tenu compte des en-têtes au niveau liaison car le paquet doit être encapsulé avec différents en-têtes au niveau liaison tout au long de sa transmission.

Le trafic de commande doit être limité à un faible pourcentage connu de la largeur de bande de la session: faible de façon à ne pas dégrader la première fonction du protocole de transport qui est d'acheminer les données; connu de sorte que le trafic de commande puisse figurer dans la spécification de largeur de bande donnée à un protocole de réservation des ressources, et que chaque participant puisse calculer indépendamment sa part. Il est proposé que le pourcentage de la largeur de bande de la session attribué aux paquets RTCP soit fixe et égal à 5%. Cette valeur et la valeur d'autres constantes intervenant dans le calcul de l'intervalle ne sont pas des valeurs seuils, mais il faut que tous les participants à la session utilisent les mêmes valeurs afin de calculer le même intervalle. Ces constantes doivent donc être fixes pour un profil donné.

L'algorithme décrit au A.7 a été conçu pour répondre aux objectifs présentés ci-dessus. Il calcule l'intervalle entre les envois de paquets RTCP composés pour pouvoir répartir la largeur de bande de trafic de commande autorisée entre les participants. Grâce à cela, une application qui fournit des réponses rapides pour les petites sessions dans lesquelles, par exemple, l'identification de tous les participants est importante, peut néanmoins s'adapter automatiquement aux grandes sessions. L'algorithme présente les caractéristiques suivantes:

- l'ensemble des émetteurs se voit attribuer au moins 1/4 de la largeur de bande du trafic de commande de sorte que dans les sessions comportant beaucoup de récepteurs et peu d'émetteurs, les participants souhaitant rejoindre la session à titre d'émetteur recevront un CNAME plus rapidement;
- il est nécessaire que l'intervalle calculé entre les paquets RTCP soit supérieur à au moins cinq secondes afin d'éviter que des rafales de paquets RTCP dépassent la largeur de bande autorisée lorsque le nombre de participants est faible et que la courbe du trafic n'est pas lisse conformément à la loi des grands nombres;
- on fait varier de façon aléatoire l'intervalle entre les paquets RTCP d'une valeur valant entre [0,5, 1,5] fois l'intervalle calculé de manière à éviter la synchronisation non voulue de tous les participants [A-8]. Le premier paquet RTCP envoyé après avoir rejoint une session est également retardé d'une valeur aléatoire autour de la moitié de l'intervalle RTCP minimal au cas où l'application soit lancée simultanément dans plusieurs sites, par exemple à la suite d'une annonce de session;
- la taille moyenne de tous les paquets RTCP composés reçus et envoyés est évaluée dynamiquement, afin de permettre une adaptation automatique aux variations de quantité d'informations de commande transmises.

Cet algorithme peut être utilisé pour les sessions dans lesquelles tous les participants sont autorisés à envoyer des paquets. Dans ce cas, le paramètre de largeur de bande de session est le produit de la largeur de bande attribuée à un émetteur par le nombre de participants, et la largeur de bande RTCP vaut 5% de cette largeur.

A.6.2.1 Mise à jour du nombre de membres de la session

Le calcul de l'intervalle entre les paquets RTCP est fonction de l'évaluation du nombre de sites participant à la session. De nouveaux sites sont comptés lorsqu'ils sont entendus, et une rubrique est créée pour chacun dans un tableau indexé par l'identificateur SSRC ou CSRC (voir A.8.2, Résolution des collisions et détection des boucles) afin de garder la trace de ces sites. Les nouvelles rubriques ne peuvent être considérées comme valides qu'après la réception de plusieurs paquets comportant le nouvel identificateur SSRC (voir A.6.1). Une rubrique peut être supprimée du tableau après la réception d'un paquet RTCP BYE comportant l'identificateur SSRC correspondant.

Un participant peut marquer un autre site comme étant inactif, ou le supprimer s'il n'est plus valide, si aucun paquet RTP ou RTCP n'a été reçu au bout d'un petit nombre d'intervalles de rapport RTCP (5 est proposé). Cela fournit une certaine robustesse contre la perte de paquet. Il faut que tous les

sites calculent approximativement la même valeur pour l'intervalle de rapport RTCP pour que cette temporisation fonctionne correctement.

Une fois qu'un site a été validé, si par la suite il est marqué inactif, il faut continuer à conserver l'état de ce site et à compter ce site dans le nombre total des sites utilisant en partage la largeur de bande RTCP pendant une période suffisamment longue pour couvrir les partitions typiques du réseau. Le but est d'éviter l'excès de trafic, lorsque la partition se résorbe, dû à un intervalle de rapport RTCP trop petit. Une temporisation de 30 minutes est proposée. On notera que cette valeur est toujours supérieure à 5 fois la plus grande valeur prévue pour l'intervalle de rapport RTCP (environ 2 à 5 minutes).

A.6.2.2 Attribution de largeur de bande pour la description de source

La présente Recommandation définit plusieurs éléments de description de source (SDES, *source description*) qui viennent s'ajouter à l'élément CNAME obligatoire, par exemple NAME (nom personnel) et EMAIL (adresse électronique). Elle permet en outre de définir de nouveaux types de paquet propres à l'application. Les applications doivent être prudentes lorsqu'elles attribuent une largeur de bande de commande pour ces informations supplémentaires car cette nouvelle largeur de bande attribuée ralentira la fréquence d'envoi des rapports de réception et du CNAME, d'où une dégradation de la performance du protocole. Il est conseillé de ne pas utiliser plus de 20% de la largeur de bande RTCP attribuée à un seul participant pour transporter les informations supplémentaires. Le but n'est donc pas d'inclure tous les éléments SDES dans chaque application. Les éléments qui sont inclus doivent se voir attribuer une fraction de la largeur de bande qui est fonction de leur utilité. Plutôt que d'évaluer dynamiquement ces fractions, il est recommandé de traduire statiquement les pourcentages en nombres d'intervalles de rapport sur la base de la longueur type d'un élément.

Une application peut par exemple être conçue pour envoyer les éléments CNAME, NAME et EMAIL et seulement ceux-là. L'élément NAME peut se voir accorder un rang de priorité plus élevé que l'élément EMAIL car l'élément NAME est affiché en permanence au niveau de l'interface utilisateur de l'application, alors que l'élément EMAIL n'est affiché qu'à la demande. A chaque intervalle RTCP, un paquet RR et un paquet SDES comportant l'élément CNAME sont envoyés. Pour une petite session fonctionnant avec l'intervalle minimal, cet envoi se produit toutes les 5 secondes en moyenne. Un intervalle sur trois (toutes les 15 secondes), un élément supplémentaire est inclus dans le paquet SDES. Sept fois sur huit, il s'agit de l'élément NAME, et une fois sur huit (toutes les deux minutes), il s'agit de l'élément EMAIL.

Lorsque plusieurs applications fonctionnent simultanément grâce à la possession d'un seul élément CNAME par participant permettant d'assurer un lien entre ces applications, par exemple dans une conférence multimédia composée d'une session RTP pour chaque support, les informations SDES supplémentaires peuvent être transmises dans une seule session RTP. Dans les autres sessions, seul l'élément CNAME est transmis.

A.6.3 Rapports d'émetteur et de récepteur

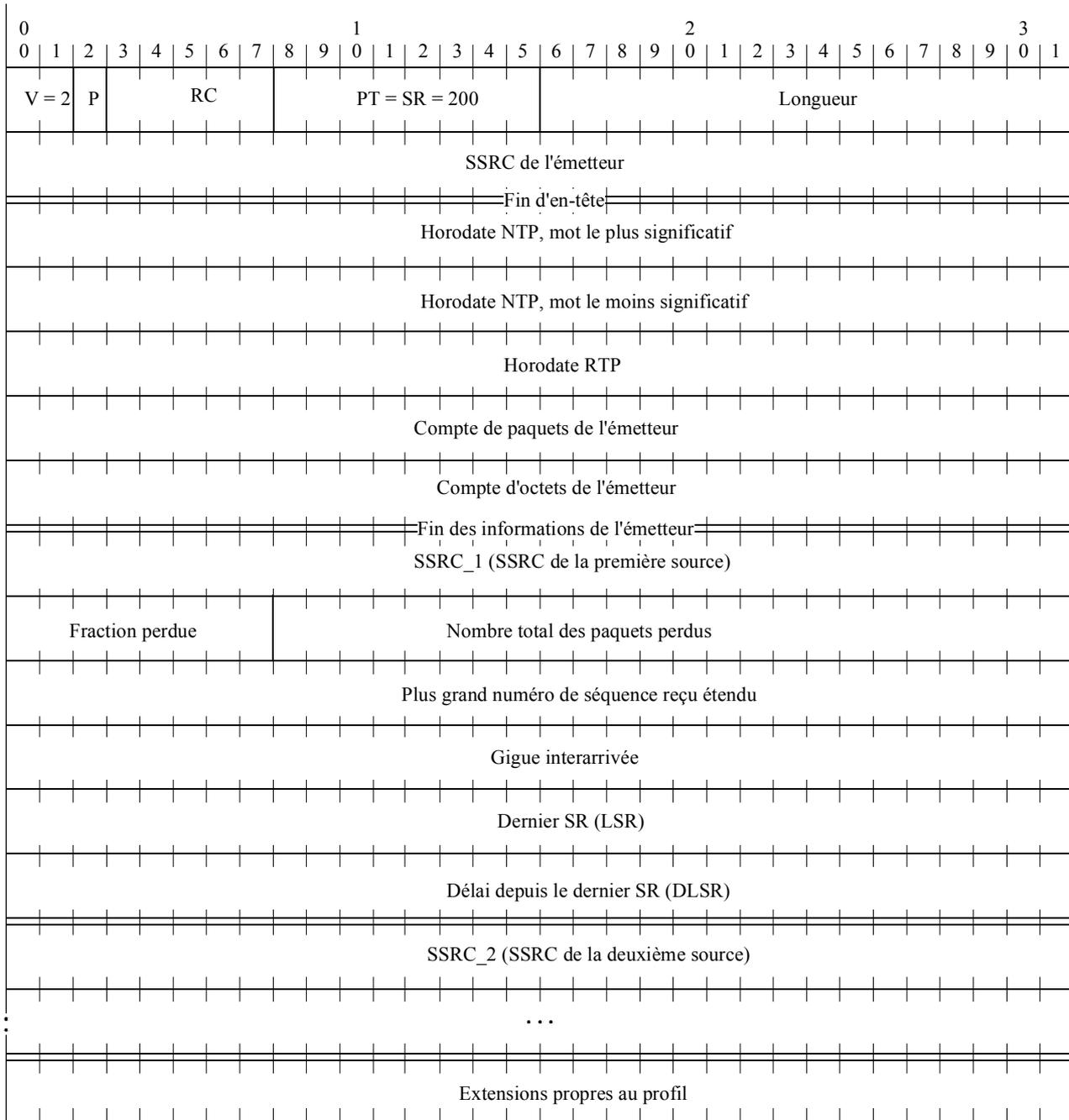
Les récepteurs RTP fournissent un retour de qualité de réception à l'aide de paquets de rapport RTCP; il existe deux types de rapport selon que le récepteur est aussi émetteur ou non. La seule différence entre le rapport d'émetteur (SR, *sender report*) et le rapport de récepteur (RR, *receiver report*), mis à part le code de type de paquet, réside dans le fait que le rapport d'émetteur contient une section d'informations d'émetteur sur 20 octets à l'intention des émetteurs actifs. Un site envoie un paquet SR s'il a envoyé des paquets de données depuis l'envoi du dernier ou de l'avant-dernier rapport, dans les autres cas, il envoie un paquet RR.

Les deux types de paquet SR et RR comportent zéro, un ou plusieurs blocs de rapport de réception, un pour chacune des sources de synchronisation en provenance desquelles ce récepteur a reçu des paquets de données RTP depuis le dernier rapport. Aucun rapport n'est transmis pour les sources contributives énumérées dans la liste CSRC. Chaque bloc de rapport de réception fournit des

statistiques concernant les données reçues en provenance de la source indiquée dans ce bloc. Etant donné qu'un paquet SR ou RR peut contenir au plus 31 blocs de rapport de réception, des paquets RR supplémentaires peuvent être empilés après le paquet SR ou RR initial, si nécessaire, pour faire figurer les rapports de réception de toutes les sources entendues depuis l'envoi du dernier rapport.

Les paragraphes qui suivent définissent les formats des deux rapports, la façon dont ils peuvent être étendus d'une manière propre au profil si une application nécessite des informations de retour supplémentaires, et la façon dont les rapports peuvent être utilisés. L'établissement de rapports de réception par les traducteurs et les mélangeurs est détaillé au A.7, Traducteurs et mélangeurs RTP.

A.6.3.1 SR: paquet RTCP rapport d'émetteur



T1527590-97

Le paquet de rapport d'émetteur est constitué de trois sections, éventuellement suivies d'une quatrième section d'extensions propre au profil, si elle est définie. La première section – l'en-tête – a une longueur de 8 octets. La signification des champs est la suivante:

version (V): 2 bits. Ce champ identifie la version du protocole RTP, qui est la même dans les paquets RTCP et les paquets RTP. La version définie par la présente Recommandation est deux (2).

bouffage (P, *padding*): 1 bit. Si le bit de bouffage est positionné, ce paquet RTCP contient à la fin quelques octets supplémentaires de bouffage qui ne font pas partie des informations de commande. Le dernier octet de bouffage donne le nombre d'octets de bouffage qu'il faut ignorer. Le bouffage peut être requis par certains algorithmes de chiffrement qui nécessitent des tailles de bloc fixes. Dans un paquet RTCP composé, le bouffage ne doit être requis qu'au niveau du dernier paquet simple car le paquet composé est chiffré comme un tout.

compte de rapports de réception (RC, *reception report count*): 5 bits. Ce champ donne le nombre de blocs de rapport de réception contenus dans ce paquet. La valeur zéro est valide.

type de paquet (PT): 8 bits. Ce champ contient la constante 200 qui indique qu'il s'agit d'un paquet RTCP SR.

longueur: 16 bits. Il s'agit de la longueur de ce paquet RTCP en nombre de mots de 32 bits moins un, y compris l'en-tête et le bouffage éventuel. (Grâce au décalage de un, zéro est une longueur valide et les éventuelles boucles infinies lors de l'analyse d'un paquet RTCP composé sont évitées; par ailleurs le fait de compter le nombre de mots de 32 bits permet d'éviter le contrôle de validité sur les multiples de 4.)

SSRC: 32 bits. Il s'agit de l'identificateur de source de synchronisation correspondant à l'émetteur de ce paquet SR.

La deuxième section, les informations de l'émetteur, a une longueur de 20 octets et figure dans chaque paquet de rapport d'émetteur. Elle récapitule les transmissions de données faites par cet émetteur. La signification des champs est la suivante:

horodate NTP: 64 bits. Ce champ indique l'heure – lue sur une horloge murale – à laquelle ce rapport a été envoyé; la combinaison de cette horodate et des horodates renvoyées dans les rapports de réception des autres récepteurs permet de mesurer le temps de transmission aller-retour vers ces récepteurs. Les récepteurs doivent savoir que la précision de la mesure de l'horodate peut être beaucoup moins bonne que la résolution de l'horodate NTP. L'incertitude sur la mesure de l'horodate n'est pas indiquée car elle peut ne pas être connue. Un émetteur qui peut garder la trace du temps écoulé mais qui ne connaît pas l'heure indiquée par l'horloge murale peut se servir du temps écoulé jusqu'à ce qu'il rejoigne la session. Ce temps est supposé être inférieur à 68 ans, le bit le plus élevé vaudra donc zéro. Il est permis d'utiliser l'horloge d'échantillonnage pour évaluer l'heure indiquée par l'horloge murale. Un émetteur qui ne connaît pas l'heure indiquée par l'horloge murale et qui n'a pas la notion du temps écoulé peut mettre l'horodate NTP à zéro.

horodate RTP: 32 bits. Cette horodate correspond à l'horodate NTP (ci-dessus), mais elle a la même unité et le même décalage aléatoire que les horodates RTP des paquets de données. Cette correspondance peut servir à la synchronisation intra et intermédia des sources dont les horodates NTP sont synchronisées; elle peut aussi servir aux récepteurs indépendants du média à évaluer la fréquence nominale de l'horloge RTP. On notera que dans la plupart des cas, cette horodate doit être différente de l'horodate RTP d'un paquet de données adjacent quelconque. Elle est plutôt calculée à partir de l'horodate NTP correspondante et en fonction de la relation qui existe entre l'horodate RTP et le temps réel et qui est mise à jour périodiquement par le contrôle de l'heure indiquée par l'horloge murale à un instant d'échantillonnage.

compte des paquets de l'émetteur: 32 bits. Il s'agit du nombre total de paquets de données RTP envoyés par l'émetteur depuis le début de la transmission et jusqu'au moment où ce paquet SR a été produit. Le compte est réinitialisé si l'émetteur change d'identificateur SSRC.

compte des octets de l'émetteur: 32 bits. Il s'agit du nombre total d'octets de charge utile (c'est-à-dire non compris l'en-tête et le bourrage) transmis par l'émetteur dans des paquets de données RTP depuis le début de la transmission et jusqu'au moment où ce paquet SR a été produit. Le compte est réinitialisé si l'émetteur change d'identificateur SSRC. Ce champ peut servir à évaluer le débit moyen de données de charge utile.

La troisième section contient zéro, un ou plusieurs blocs de rapport de réception selon le nombre d'autres sources que cet émetteur a entendues depuis le dernier rapport. Chaque bloc de rapport de réception comporte des statistiques sur la réception des paquets RTP provenant d'une même source de synchronisation. Les récepteurs n'indiquent pas de statistiques lorsqu'une source change d'identificateur SSRC en raison d'une collision. Ces statistiques sont les suivantes:

SSRC_n (identificateur de la source): 32 bits. Il s'agit de l'identificateur SSRC de la source sur laquelle portent les informations figurant dans ce bloc de rapport de réception.

fraction perdue: 8 bits. Il s'agit de la fraction des paquets de données RTP provenant de la source SSRC_n qui ont été perdus depuis l'envoi du paquet SR ou RR précédent, exprimée par un nombre à virgule fixe, la virgule binaire étant placée au bord gauche du champ. (Cela revient à prendre la partie entière après avoir multiplié la fraction perdue par 256.) Cette fraction est définie comme le nombre de paquets perdus divisé par le nombre de paquets prévus (voir la définition dans le paragraphe qui suit). Une implémentation est montrée au A.6.3. Si la perte est négative en raison de paquets en double, la fraction perdue est mise à zéro. On notera qu'un récepteur ne peut pas dire si des paquets ont été perdus après le dernier reçu, et qu'aucun bloc de rapport de réception ne doit être transmis pour une source donnée si tous les paquets envoyés par cette source pendant le dernier intervalle de rapport ont été perdus.

nombre total des paquets perdus: 24 bits. Il s'agit du nombre total de paquets de données RTP provenant de la source SSRC_n qui ont été perdus depuis le début de la réception. Par définition, c'est le nombre de paquets prévus moins le nombre de paquets effectivement reçus, où le nombre de paquets reçus comprend les paquets en retard et les paquets en double. Par conséquent, les paquets qui arrivent en retard ne sont pas comptés comme des paquets perdus, et la perte peut être négative s'il y a des paquets en double. Le nombre de paquets prévus est défini comme le dernier numéro de séquence reçu étendu (défini plus loin) moins le numéro de séquence initial reçu. Ce nombre peut se calculer par la méthode donnée au A.6.3.

plus grand numéro de séquence reçu étendu: 32 bits. Les 16 bits inférieurs contiennent le plus grand numéro de séquence reçu dans un paquet de données RTP provenant de la source SSRC_n, et les 16 bits les plus significatifs étendent ce numéro de séquence par le nombre correspondant de cycles de numéros de séquence, qui peut être mis à jour au moyen de l'algorithme au A.13. On notera que des récepteurs distincts à l'intérieur d'une même session produiront des extensions du numéro de séquence différentes si les heures auxquelles ils ont démarré sont très différentes.

gigue entre arrivées: 32 bits. Il s'agit d'une évaluation de la variance statistique du temps entre arrivées des paquets de données RTP, mesurée en unités d'horodate et exprimée par un entier non signé. La gigue entre arrivées J est définie comme l'écart moyen (valeur absolue lissée) de la différence D d'espacement de deux paquets au niveau du récepteur par rapport à l'émetteur. Comme le montre l'équation ci-dessous, la différence D est équivalente à la différence de "temps de transit relatif" pour les deux paquets; le temps de transit relatif est la différence entre l'horodate RTP d'un paquet et l'heure indiquée par l'horloge du récepteur au moment de l'arrivée du paquet, mesurées dans la même unité.

Si " S_i " est l'horodate RTP du paquet i , et " R_i " l'heure d'arrivée en unités d'horodate RTP du paquet i , alors pour les deux paquets i et j , D peut s'exprimer comme suit:

$$D(i + j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

Chaque fois qu'un paquet de données i est reçu en provenance de la source $SSRC_n$, la gigue entre arrivées est recalculée en fonction de la différence D s'appliquant à ce paquet et au paquet $i - 1$ le précédant dans l'ordre d'arrivée (pas nécessairement le paquet le précédant dans la séquence), selon l'équation:

$$J = J + \frac{|D(i-1, i) - J|}{16}$$

Chaque fois qu'un rapport de réception est transmis, la valeur courante de J est échantillonnée.

Le calcul de la gigue est prescrit ici pour permettre aux contrôleurs indépendants du profil de faire des interprétations valides des rapports venant de différentes implémentations. Cet algorithme est l'estimateur optimal du premier ordre et le paramètre de gain $1/16$ donne un bon facteur de réduction du bruit tout en conservant une vitesse de convergence raisonnable [A-9]. Le paragraphe A.8 donne un exemple d'implémentation.

horodate du dernier paquet SR (LSR, last SR timestamp): 32 bits. Il s'agit des 32 bits du milieu parmi les 64 de l'horodate NTP (comme expliqué au A.4, Ordre des octets, alignement et format temporel) qui figurent dans le dernier paquet RTCP SR (rapport d'émetteur) reçu en provenance de la source $SSRC_n$. Si aucun paquet SR n'a encore été reçu, le champ est mis à zéro.

délai depuis le dernier paquet SR (DLSR, delay since last SR): 32 bits. Il s'agit du délai, exprimé dans l'unité $1/65\,536$ secondes, entre la réception du dernier paquet SR envoyé par la source $SSRC_n$ et l'envoi de ce bloc de rapport de réception. Si aucun paquet SR n'a encore été reçu de la source $SSRC_n$, le champ DLSR est mis à zéro.

Désignons par $SSRC_r$ le récepteur qui envoie ce rapport de réception. La source $SSRC_n$ peut calculer le temps de transmission total vers le récepteur $SSRC_r$ en enregistrant l'heure A à laquelle ce bloc de rapport de réception est reçu. Elle calcule le temps total aller-retour $A - LSR$ en utilisant le champ LSR (horodate du dernier paquet SR), et en soustrayant ensuite le champ DLSR pour aboutir au temps de transmission aller-retour ($A - LSR - DLSR$). Cela est illustré sur la Figure A.2.

Ce délai peut servir de mesure approximative de la distance à un groupe de récepteurs, même si certaines liaisons présentent des délais très asymétriques.

A.6.3.2 RR: paquet RTCP rapport de récepteur

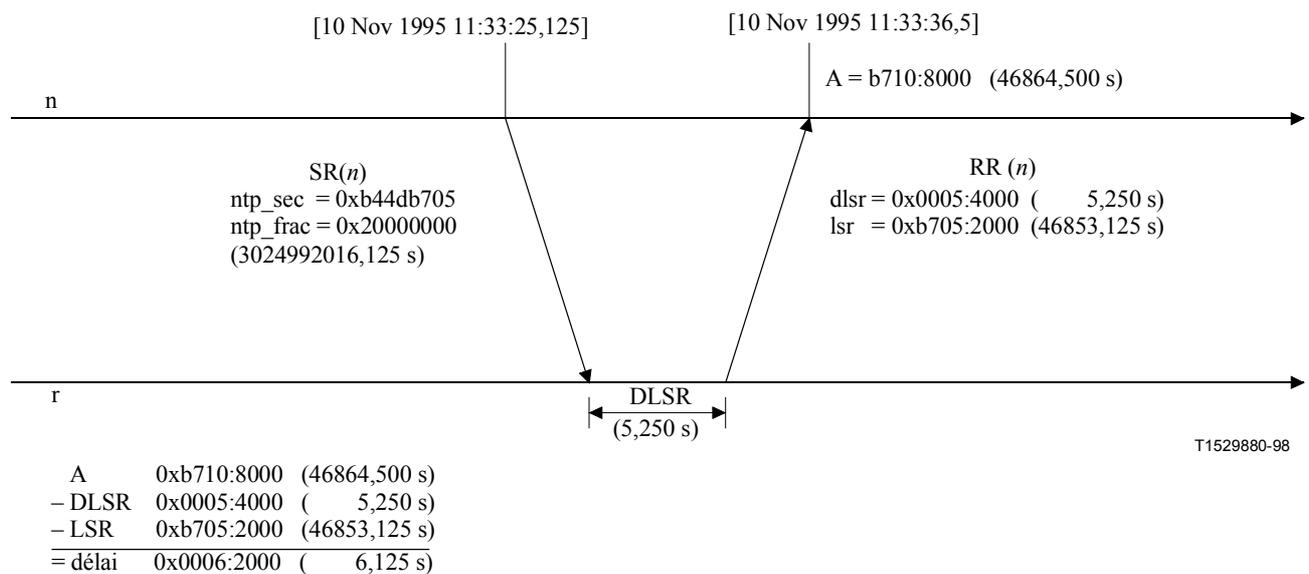
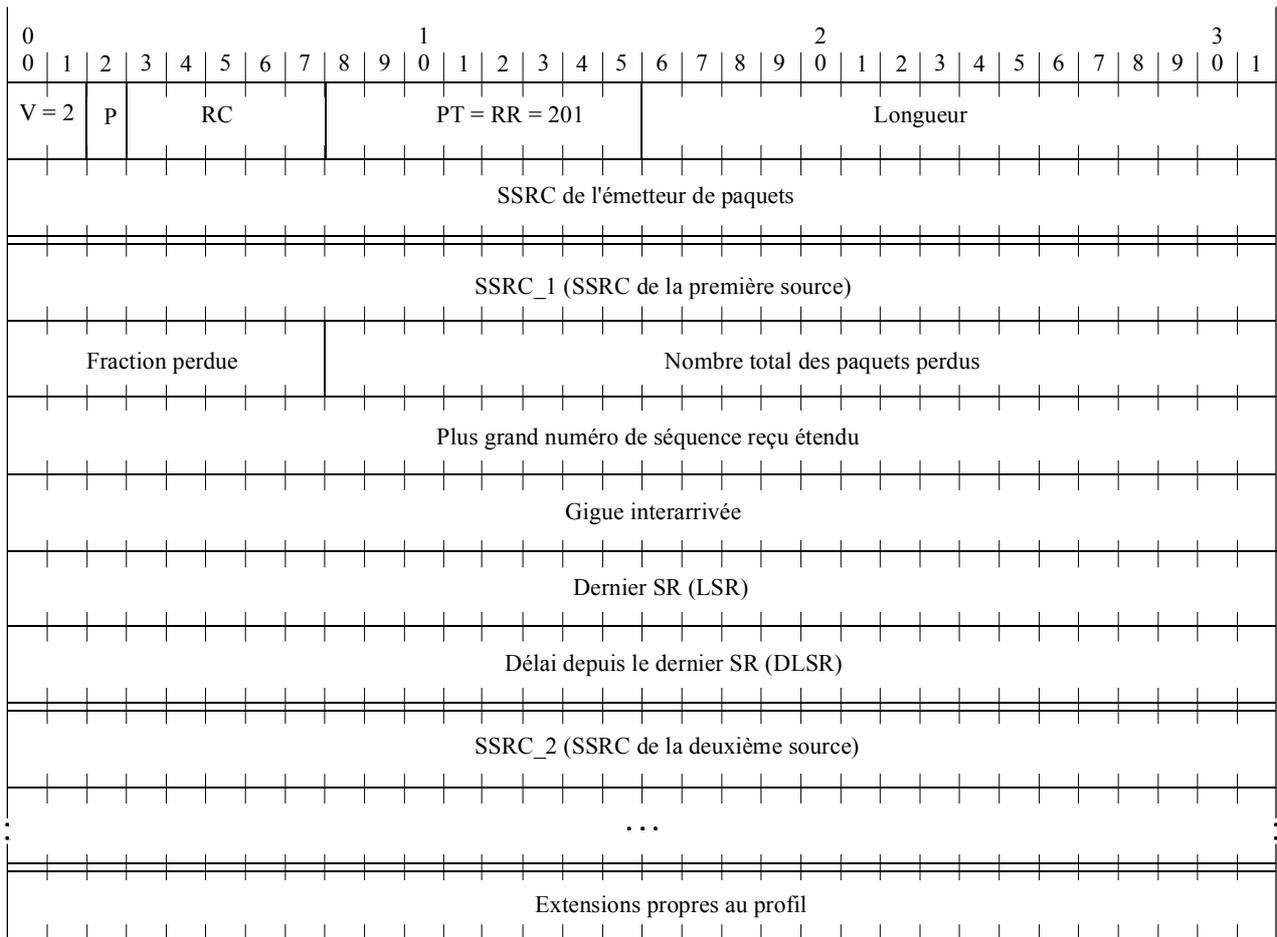


Figure A.2/H.225.0 – Exemple de calcul de temps de transmission aller-retour



T1527600-97

Le format du paquet RR (rapport de récepteur) est identique à celui du paquet SR aux exceptions près suivantes: le champ de type de paquet contient la constante 201 et les cinq mots d'informations de l'émetteur sont omis (il s'agit des horodates NTP et RTP, du compte des paquets de l'émetteur et du compte des octets de l'émetteur). Les champs restants ont la même signification que pour le paquet SR.

Un paquet RR vide (RC = 0) est placé en tête d'un paquet RTCP composé lorsqu'il n'y a ni transmission ni réception de données à signaler.

A.6.3.3 Extension des rapports d'émetteur et de récepteur

Un profil doit définir des extensions aux rapports d'émetteur et de récepteur qui sont propres au profil ou à l'application si des informations supplémentaires concernant l'émetteur ou les récepteurs doivent faire l'objet de rapports réguliers. Il vaut mieux utiliser cette méthode plutôt que définir un autre type de paquet RTCP car cette méthode nécessite un préfixe moins long:

- moins d'octets dans le paquet (ni en-tête RTCP ni champ SSRC);
- analyse plus simple et plus rapide car les applications qui fonctionnent sous ce profil sont programmées pour toujours attendre les champs d'extension à une position directement accessible après les rapports de réception.

Si des informations d'émetteur supplémentaires sont requises, elles doivent d'abord être incluses dans l'extension aux rapports d'émetteur, mais elles ne figurent pas dans les rapports de récepteur. Si des informations concernant les récepteurs doivent être incluses, ces données doivent être structurées sous forme d'un tableau de blocs qui est parallèle au tableau des blocs de rapport de réception existant; c'est-à-dire que le nombre de blocs est indiqué par le champ RC.

A.6.3.4 Analyse des rapports d'émetteur et de récepteur

Normalement, le retour sur la qualité de réception doit être utile non seulement à l'émetteur mais aussi aux autres récepteurs et aux contrôleurs tiers. L'émetteur peut modifier ses transmissions en fonction du retour; les récepteurs peuvent déterminer si les problèmes sont locaux, régionaux ou globaux; les gestionnaires de réseau peuvent utiliser des contrôleurs indépendants du profil qui ne reçoivent que les paquets RTCP et pas les paquets de données RTP correspondants pour évaluer la performance de leurs réseaux concernant la distribution en mode multidiffusion.

Les comptes totaux sont utilisés à la fois dans les informations d'émetteur et dans les blocs de rapport de récepteur pour calculer les différences entre deux rapports quelconques afin de pouvoir effectuer des mesures à la fois sur des périodes courtes et sur des périodes longues, et afin de fournir une certaine robustesse contre la perte d'un rapport. La différence entre les deux derniers rapports reçus peut servir à évaluer la qualité récente de la distribution. L'horodate NTP est incluse de façon à pouvoir calculer les taux à partir de ces différences sur l'intervalle séparant deux rapports. Cette horodate étant indépendante de la fréquence d'horloge pour le codage des données, il est possible d'implémenter des contrôleurs de la qualité indépendants du codage et du profil.

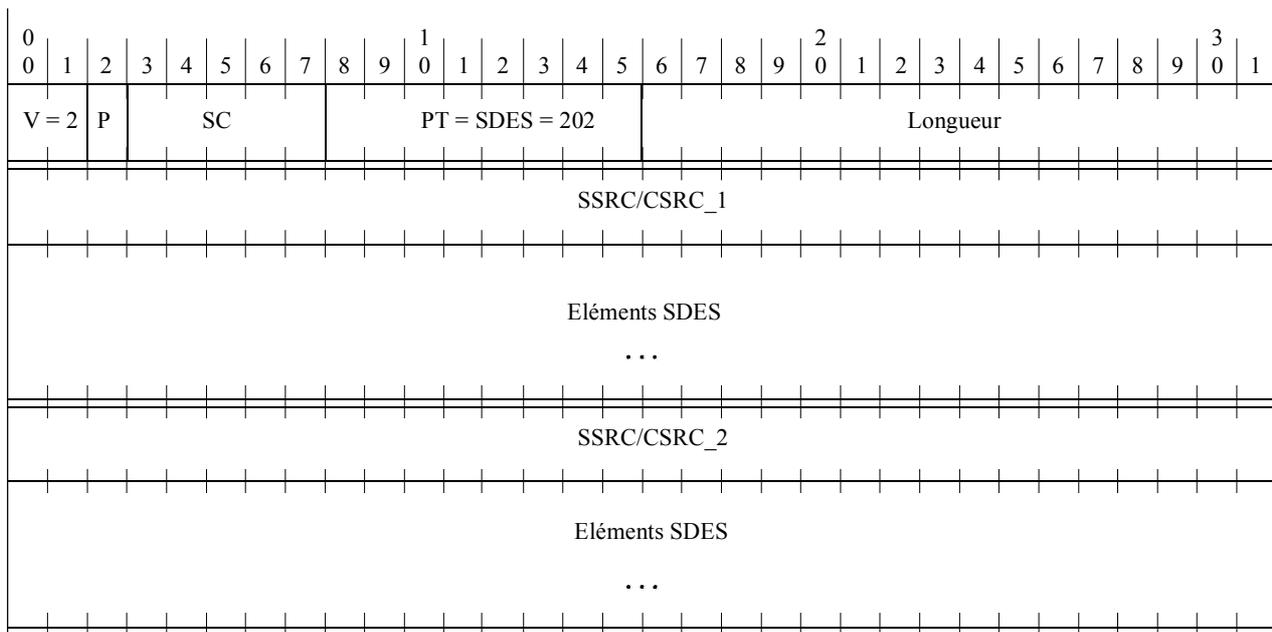
Comme exemple de calcul, citons le taux de paquets perdus sur l'intervalle séparant deux rapports de réception. La différence entre les nombres totaux de paquets perdus donne le nombre de paquets perdus pendant l'intervalle. La différence entre les derniers numéros de séquence étendus reçus donne le nombre de paquets prévus pendant l'intervalle. Le rapport entre ces deux nombres correspond à la fraction de paquets perdus sur l'intervalle. Ce rapport doit être égal au champ fraction perdue si les deux rapports de réception sont consécutifs, mais pas dans les autres cas. Le taux de perte par seconde peut être obtenu en divisant la fraction de perdus par la différence des horodates NTP, exprimées en secondes. Le nombre de paquets reçus est égal au nombre de paquets prévus moins le nombre de paquets perdus. Le nombre de paquets prévus peut aussi servir à évaluer la validité statistique de toute estimation de perte. Par exemple, un paquet perdu sur cinq a moins de sens que 200 paquets perdus sur 1000.

Grâce aux informations d'émetteur, un contrôleur tiers peut calculer le débit moyen des données de charge utile et le débit moyen des paquets sur un intervalle sans recevoir les données. Le rapport entre ces deux débits donne la taille moyenne de la charge utile. Si l'on peut supposer que la perte de paquets est indépendante de la taille de paquet, le produit entre le nombre de paquets reçus par un récepteur donné et la taille moyenne de charge utile (ou la taille de paquet correspondante) donne le débit apparent au niveau de ce récepteur.

Outre les comptes totaux qui permettent de mesurer la perte de paquets sur des longues périodes à partir des différences entre les rapports, le champ fraction perdue fournit une mesure sur une courte période à partir d'un seul rapport. Cela devient plus important lorsque la taille d'une session augmente tellement que les informations d'état de réception ne peuvent pas être conservées pour tous les récepteurs ou lorsque l'intervalle entre les rapports devient si long que seul un rapport peut avoir été reçu en provenance d'un récepteur donné.

Le champ gigue entre arrivées fournit une deuxième mesure sur une courte période de l'encombrement du réseau. La perte de paquet permet de repérer les encombrements persistants alors que la mesure de gigue permet de repérer les encombrements transitoires. La mesure de gigue peut signaler un encombrement avant que celui-ci ne conduise à une perte de paquet. Etant donné que le champ gigue entre arrivées n'est qu'un instantané de la gigue au moment d'un rapport, il peut être nécessaire d'analyser un certain nombre de rapports d'un même récepteur au cours du temps ou un certain nombre de rapports de plusieurs récepteurs, par exemple à l'intérieur d'un même réseau.

A.6.4 SDES: paquet RTCP de description de source



T1527610-97

Le paquet SDES est une structure à trois niveaux constituée d'un en-tête et de zéro, une ou plusieurs tranches, chacune étant composée d'éléments décrivant la source identifiée dans cette tranche. Les éléments sont décrits séparément dans des paragraphes ultérieurs.

version (V), bourrage (P), longueur: ces champs sont identiques à ceux décrits pour le paquet SR (voir A.6.3.1, SR: paquet RTCP rapport d'émetteur).

type de paquet (PT): 8 bits. Ce champ contient la constante 202 indiquant qu'il s'agit d'un paquet RTCP SDES.

compte de la source (SC, source count): 5 bits. Il s'agit du nombre de tranches SSRC/CSRC figurant dans ce paquet SDES. La valeur zéro est valide mais inutile.

Chaque tranche est constituée d'un identificateur SSRC/CSRC suivi d'une liste de zéro, un ou plusieurs éléments, comportant des informations concernant la source SSRC/CSRC. Chaque tranche commence sur une frontière à 32 bits. Chaque élément comprend: un champ de type sur 8 bits, un compte des octets sur 8 bits indiquant la longueur du texte (non compris ces deux octets d'en-tête), et le texte proprement dit. On notera que la longueur du texte ne peut pas être supérieure à 255 octets, mais cela est cohérent avec la nécessité de limiter la consommation de largeur de bande RTCP.

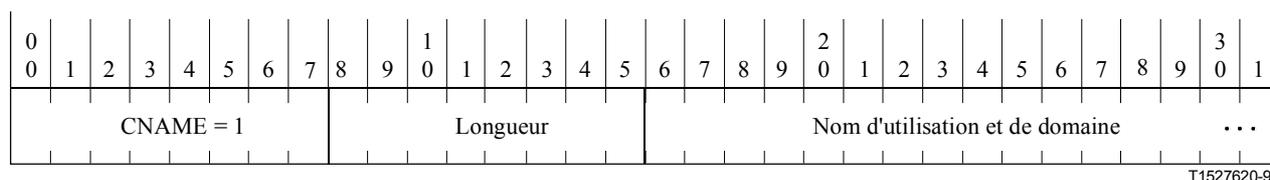
Le texte est codé conformément au codage UTF-2 spécifié dans l'Annexe F de l'ISO/CEI 10646-1 [A-10]. Ce codage est également connu sous les appellations suivantes: UTF-8 et UTF-FSS. Il est décrit dans *File System Safe UCS Transformation Format (FSS_UTF)*, *X/Open Preliminary Specification, Document Number P316* et *Unicode Technical Report n° 4*. US-ASCII est une partie de ce codage qui ne nécessite aucun codage supplémentaire. La présence de codages multioctets est indiquée par le positionnement du bit le plus significatif d'un caractère sur la valeur un.

Les éléments sont contigus, c'est-à-dire qu'ils ne sont pas complétés chacun par des bits de bourrage jusqu'à une frontière à 32 bits. Le texte n'est pas terminé par des octets vides car certains codages multioctets incluent des octets vides. La liste des éléments de chaque tranche se termine par un ou plusieurs octets vides, le premier de ces octets étant interprété comme le type d'élément zéro qui indique la fin de la liste, le reste servant de bourrage jusqu'à la frontière à 32 bits suivante. Une tranche avec zéro élément (quatre octets vides) est valide mais inutile.

Les systèmes d'extrémité envoient un seul paquet SDES contenant leur propre identificateur de source (le même que l'identificateur SSRC dans l'en-tête RTP fixe). Un mélangeur envoie un seul paquet SDES contenant une tranche pour chaque source contributive de laquelle il reçoit des informations SDES, ou plusieurs paquets SDES complets dans le format indiqué ci-dessus si le nombre de sources contributives est supérieur à 31 (voir A.2.3, Mélangeur et traducteurs).

Les éléments SDES définis ici sont décrits dans les sous-paragraphes suivants. Seul l'élément CNAME est obligatoire. Certains éléments présentés ici peuvent être utiles uniquement pour certains profils, mais les types d'élément sont tous attribués à partir d'un espace commun pour promouvoir l'utilisation en partage et pour simplifier les applications indépendantes du profil. Il est possible de définir des éléments supplémentaires dans un profil, il suffit pour cela d'enregistrer les numéros de type auprès de l'IANA.

A.6.4.1 CNAME: élément SDES identificateur d'extrémité canonique



L'identificateur CNAME a les propriétés suivantes:

- comme l'identificateur SSRC attribué de façon aléatoire peut changer en cas de découverte d'un conflit ou de redémarrage d'un programme, l'élément CNAME doit assurer le lien entre l'identificateur SSRC et un identificateur pour la source qui reste constant;
- tout comme l'identificateur SSRC, l'identificateur CNAME doit aussi être unique parmi tous les participants à une même session RTP;
- pour assurer un lien entre les divers outils multimédias utilisés par un même participant dans un ensemble de sessions RTP connexes, le CNAME doit être fixe pour ce participant;
- afin de faciliter le contrôle par un tiers, le CNAME doit pouvoir être utilisé aussi bien par un programme que par une personne pour localiser la source.

Le CNAME doit donc être dérivé algorithmiquement et non pas entré manuellement, lorsque c'est possible. Pour répondre à ces prescriptions, il convient d'utiliser le format donné ci-après sauf si un profil spécifie une autre syntaxe ou une autre sémantique. L'élément CNAME doit avoir le format "utilisateur@serveur", ou "serveur" s'il n'y a pas de nom d'utilisateur comme dans les systèmes à un seul utilisateur. Pour les deux formats, "serveur" est soit le nom entier du domaine du serveur à partir duquel proviennent les données en temps réel, formatées conformément aux règles énoncées dans les documents RFC 1034 [A-11], RFC 1035 [A-12] et au 2.1/RFC 1123 [A-13]; soit la représentation ASCII normalisée de l'adresse numérique du serveur à l'interface utilisée pour la communication RTP. Par exemple, la représentation ASCII normalisée d'une adresse IP Version 4 est "décimale pointée", on dit aussi "dotted quad" en anglais. On prévoit que d'autres types d'adresse auront des représentations ASCII qui doivent être globalement uniques. Le nom entier du domaine est plus pratique pour un observateur humain et peut éviter d'avoir à envoyer un élément NAME en plus, mais il peut être difficile voire impossible à obtenir de façon fiable dans certains environnements. Dans ces environnements, les applications doivent plutôt utiliser la représentation ASCII de l'adresse.

Pour un système multi-utilisateurs, on peut citer en exemple "doe@sleepy.megacorp.com" ou "doe@192.0.2.89". Pour un système sans nom d'utilisateur, les exemples précédents deviennent "sleepy.megacorp.com" et "192.0.2.89".

Le nom d'utilisateur doit revêtir une forme qu'un programme comme "finger" ou "talk" puisse utiliser, c'est-à-dire que généralement il s'agira du nom d'entrée plutôt que du nom personnel. Le

nom de serveur n'est pas nécessairement identique à celui qui figure dans l'adresse électronique du participant.

Cette syntaxe ne fournira pas un identificateur unique pour chaque source si une application autorise un utilisateur à produire plusieurs sources à partir d'un seul serveur. Pour une telle application, il faudra avoir recours à l'identificateur SSRC pour pouvoir identifier une source donnée, ou le profil de cette application devra spécifier une syntaxe supplémentaire pour l'identificateur CNAME.

Si chaque application crée son CNAME de façon indépendante, les CNAME résultants pourront ne pas être identiques à ceux requis pour assurer un lien entre les divers outils multimédias appartenant à un participant dans un ensemble de sessions RTP connexes. Si le lien entre les divers média est requis, il peut être nécessaire qu'un outil de coordination configure de façon externe le CNAME de chaque outil avec la même valeur. Les concepteurs d'application doivent savoir que l'attribution d'adresses de réseau privées comme l'attribution Net-10 proposée dans le document RFC 1597 [A-14] peut créer des adresses de réseau qui ne sont pas globalement uniques. On aurait alors des CNAME qui ne sont pas uniques si des serveurs avec des adresses privées et ne présentant pas de connectivité IP directe avec le réseau Internet public transmettaient leurs paquets RTP au réseau Internet public via un traducteur au niveau RTP. (Voir aussi le document RFC 1627 [A-15].) Pour traiter ce cas, les applications peuvent fournir un moyen de configurer un CNAME unique, mais la responsabilité repose sur le traducteur qui doit traduire les CNAME d'adresses privées en adresses publiques, lorsqu'il est nécessaire de ne pas exposer les adresses privées.

A.6.4.2 NAME: élément SDES nom d'utilisateur

Voir l'Appendice I.

A.6.4.3 EMAIL: élément SDES adresse électronique

Voir l'Appendice I.

A.6.4.4 PHONE: élément SDES numéro de téléphone

Voir l'Appendice I.

A.6.4.5 LOC: élément SDES situation géographique de l'utilisateur

Voir l'Appendice I.

A.6.4.6 TOOL: élément SDES nom d'application ou d'outil

Voir l'Appendice I.

A.6.4.7 NOTE: élément SDES avis/état

Voir l'Appendice I.

A.6.4.8 PRIV: élément SDES extensions privées

Voir l'Appendice I.

A.6.5 BYE: paquet RTCP au revoir

Voir l'Appendice I.

A.6.6 APP: paquet RTCP défini par l'application

Voir l'Appendice I.

A.7 Traducteurs et mélangeurs RTP

Outre les systèmes d'extrémité, le protocole RTP prend en charge des "traducteurs" et des "mélangeurs", qui peuvent être considérés comme des "systèmes intermédiaires" au niveau RTP. Cette prise en charge ajoute une certaine complexité au protocole, mais des expériences réalisées avec des applications audio et vidéo en mode multidiffusion sur Internet ont clairement démontré que ces fonctions sont nécessaires. Les exemples d'utilisation de traducteurs et de mélangeurs donnés dans le présent paragraphe reposent sur la présence de barrières et de connexions à faible largeur de bande, qui risquent de perdurer.

A.7.1 Description générale

Un traducteur/mélangeur RTP raccorde plusieurs "nuages" au niveau transport. Généralement, chaque nuage est défini par un protocole de réseau et de transport commun (IP/UDP par exemple), une adresse de multidiffusion ou deux adresses d'unidiffusion et un accès de destination au niveau transport. (Les traducteurs de protocole au niveau réseau, IP version 4 à IP version 6 par exemple, peuvent figurer dans un nuage de manière invisible pour le protocole RTP.) Un système peut servir de traducteur ou de mélangeur pour un certain nombre de sessions RTP, mais chacun est considéré comme une entité distincte sur le plan logique.

Afin d'éviter la création d'une boucle lorsqu'un traducteur ou un mélangeur est installé, il faut respecter les règles suivantes:

- chacun des nuages raccordés par des traducteurs et des mélangeurs participant à une session RTP doit soit différer de tous les autres par au moins un de ces paramètres (protocole, adresse, accès), soit être isolé des autres au niveau réseau;
- une règle découlant de la première est qu'il ne doit pas y avoir plusieurs traducteurs ou mélangeurs raccordés en parallèle sauf si, par un certain arrangement, ils créent une partition de l'ensemble des sources à transmettre.

D'une manière similaire, tous les systèmes d'extrémité RTP qui peuvent communiquer par l'intermédiaire d'un ou plusieurs traducteurs ou mélangeurs RTP utilisent le même espace SSRC, c'est-à-dire que les identificateurs SSRC doivent être uniques parmi tous ces systèmes d'extrémité. Le paragraphe A.8.2 (Résolution des collisions et détection des boucles) décrit l'algorithme de résolution des collisions qui permet d'assurer l'unicité des identificateurs SSRC et de détecter les boucles.

Il peut exister de nombreuses variétés de traducteurs et de mélangeurs conçus à différentes fins et pour diverses applications: par exemple pour ajouter ou supprimer un chiffrement, pour modifier le codage des données des protocoles sous-jacents, ou encore pour transformer une adresse de multidiffusion en une ou plusieurs adresses d'unidiffusion. La différence entre un traducteur et un mélangeur est qu'un traducteur traite séparément les flux de données provenant de plusieurs sources, alors qu'un mélangeur combine ces flux de données pour ne former qu'un seul nouveau flux de données:

traducteur: il transmet les paquets RTP en gardant intact leur identificateur SSRC; les récepteurs peuvent ainsi identifier chaque source même si les paquets provenant de toutes les sources passent dans le même traducteur et comportent l'adresse de source de réseau du traducteur. Certains types de traducteur transmettront les données sans les modifier, alors que d'autres pourront modifier le codage des données et donc l'horodate et le type de charge utile de données RTP. Si plusieurs paquets de données sont codés à nouveau pour ne former plus qu'un paquet, ou inversement, le traducteur doit attribuer de nouveaux numéros de séquence aux paquets sortants. Des pertes au niveau du flux de paquets entrant peuvent provoquer des trous correspondants dans les numéros de séquence en sortie. Les récepteurs ne peuvent pas détecter la présence d'un traducteur sauf s'ils connaissent par un autre moyen l'adresse de transport ou le type de charge utile qui a été utilisé par la source d'origine;

mélangeur: il reçoit les flux de paquets de données RTP provenant d'une ou plusieurs sources, modifie éventuellement le format des données, combine les flux d'une certaine manière et transmet ensuite le flux combiné. Etant donné que le rythme entre les diverses sources d'entrée ne doit être généralement pas synchronisé, le mélangeur fera des ajustements de rythme entre les flux et produira son propre rythme pour le flux combiné, il s'agit donc de la source de synchronisation du flux combiné. Tous les paquets de données transmis par un mélangeur doivent être donc marqués de l'identificateur SSRC propre au mélangeur. Afin de conserver l'identité des sources d'origine, qui ont contribué au paquet mélangé, le mélangeur doit insérer leurs identificateurs SSRC dans la liste d'identificateurs CSRC venant à la suite de l'en-tête RTP fixe du paquet. Un mélangeur qui joue également le rôle de source contributive pour un certain paquet doit inclure explicitement son propre identificateur SSRC dans la liste CSRC correspondant à ce paquet.

Pour certaines applications, il peut être accepté qu'un mélangeur n'identifie pas les sources dans la liste CSRC au prix d'une incapacité à détecter des boucles faisant intervenir ces sources.

Pour des applications de type audio par exemple, l'avantage d'un mélangeur par rapport à un traducteur réside dans le fait que la largeur de bande de sortie est limitée à celle d'une source, même lorsque plusieurs sources sont actives du côté entrée. Cela peut être important pour les liaisons à faible largeur de bande. L'inconvénient est que les récepteurs du côté sortie ne commandent pas les sources qui sont transmises et celles qui sont écartées, sauf si un mécanisme est implémenté pour la commande à distance du mélangeur. La régénération des informations de synchronisation par les mélangeurs signifie aussi que les récepteurs ne peuvent pas faire de synchronisation intermédia des flux de données d'origine. Un mélangeur multimédia pourrait le faire.

La Figure A.3 illustre l'effet d'un ensemble de mélangeurs et de traducteurs sur les identificateurs SSRC et CSRC. Sur la figure, les systèmes d'extrémité (E) sont représentés par des rectangles, les traducteurs (T) par des losanges et les mélangeurs (M) par des ovales. La notation "M1:48 (1, 17)" désigne un paquet provenant du mélangeur M1, identifié par l'identificateur SSRC (aléatoire) du mélangeur M1 valant 48 et par les deux identificateurs CSRC, 1 et 17, qui sont les identificateurs SSRC des paquets venant de E1 et de E2.

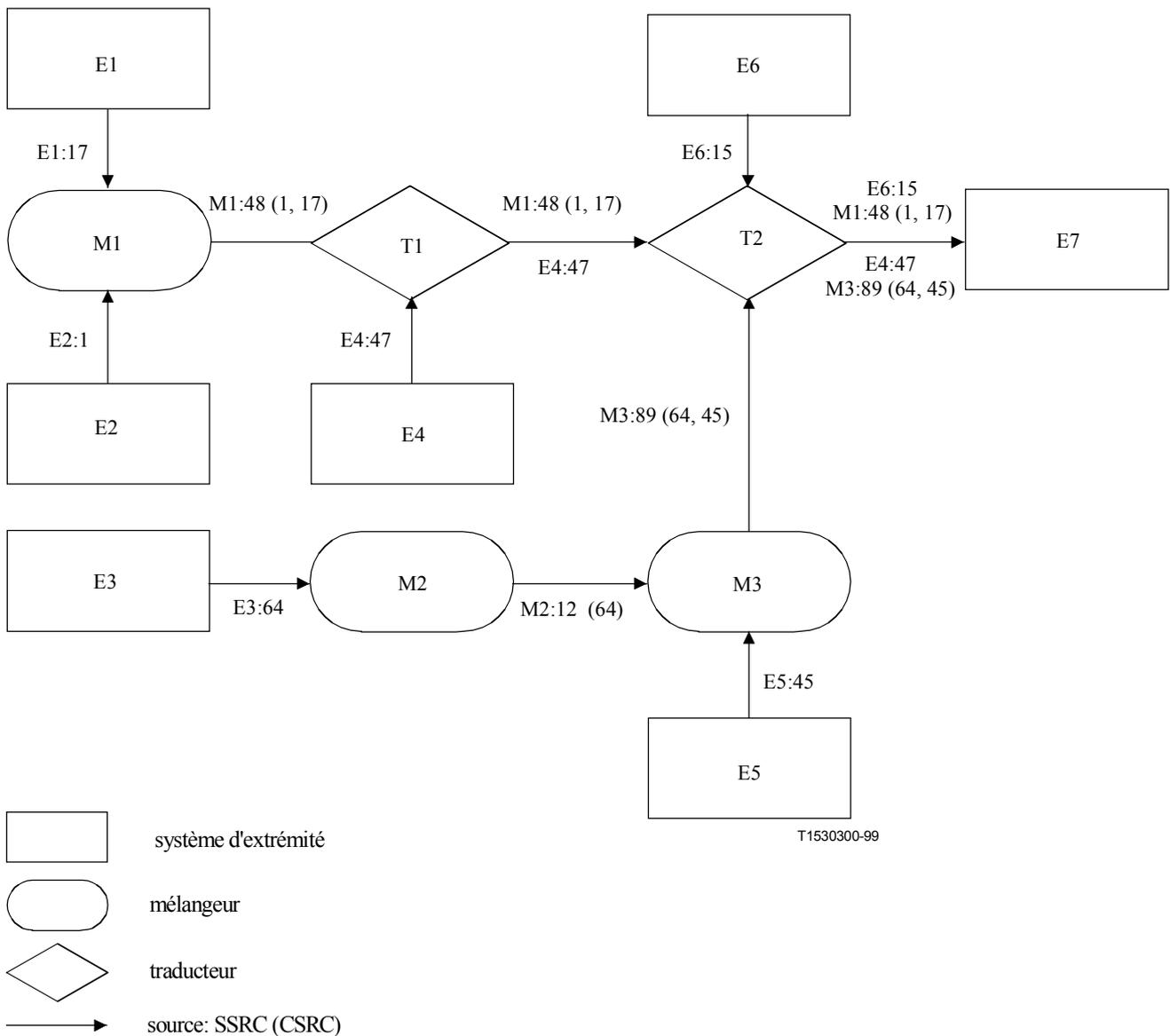


Figure A.3/H.225.0 – Exemple de réseau RTP avec des systèmes d'extrémité, des mélangeurs et des traducteurs

A.7.2 Traitement RTCP dans les traducteurs

Outre la transmission des paquets de données, éventuellement modifiés, les traducteurs et les mélangeurs doivent aussi traiter les paquets RTCP. Dans de nombreux cas, ils décomposeront les paquets RTCP composés reçus en provenance de systèmes d'extrémité afin de regrouper les informations SDES et de modifier les paquets SR ou RR. La retransmission de ces informations peut être déclenchée par l'arrivée de paquets ou à la fin d'une temporisation d'intervalle RTCP du traducteur ou du mélangeur.

Un traducteur qui ne modifie pas les paquets de données, par exemple un traducteur qui ne fait que transformer une adresse de multidiffusion en adresses d'unidiffusion, peut tout simplement transmettre les paquets RTCP sans les modifier non plus. Un traducteur qui transforme la charge utile doit effectuer les transformations correspondantes dans les informations SR et RR de sorte que celles-ci continuent à refléter les caractéristiques des données et la qualité de réception. Ces traducteurs ne doivent pas se contenter de transmettre les paquets RTCP. En général, un traducteur

ne doit pas regrouper en un seul paquet les paquets SR et RR provenant de plusieurs sources car cela réduirait la précision des mesures de temps de transmission basées sur les champs LSR et DLSR.

Informations d'émetteur SR: un traducteur ne génère pas ses propres informations d'émetteur, mais transmet les paquets SR envoyés par un nuage aux autres nuages. L'identificateur SSRC est laissé intact mais les informations d'émetteur doivent être modifiées si c'est nécessaire pour la traduction. Si un traducteur modifie le codage des données, il doit modifier le champ "compte des octets de l'émetteur". S'il combine plusieurs paquets de données en un seul paquet de sortie, il doit modifier le champ "compte des paquets de l'émetteur". S'il modifie la fréquence d'horodatage, il doit modifier le champ "horodate RTP" du paquet SR.

Blocs de rapport de réception SR/RR: un traducteur transmet les rapports de réception envoyés par un nuage aux autres nuages. On notera que ces rapports vont dans le sens opposé à celui des données. L'identificateur SSRC est laissé intact. Si un traducteur combine plusieurs paquets de données en un seul paquet de sortie, et que, par conséquent, il modifie les numéros de séquence, il doit faire la manipulation inverse pour les champs de perte de paquets et le champ "dernier numéro de séquence étendu", ce qui peut être compliqué. Dans le cas extrême, il peut ne pas y avoir de manière sensée de traduire les rapports de réception, auquel cas le traducteur peut ne pas transmettre de rapport de réception du tout ou transmettre un rapport synthétique sur la base de sa propre réception. La règle générale consiste à faire ce qui a un sens pour une traduction donnée.

Un traducteur ne nécessite pas d'identificateur SSRC propre, mais il peut choisir d'en attribuer un s'il veut envoyer des rapports concernant ce qu'il a reçu. Ces rapports doivent être envoyés à tous les nuages raccordés, chaque rapport correspondant à la traduction du flux de données envoyé à un nuage, étant donné que les rapports de réception sont normalement multidiffusés à tous les participants.

SDES: les traducteurs transmettent généralement sans modification les informations SDES envoyées par un nuage aux autres nuages, mais ils peuvent par exemple décider de filtrer les informations SDES autres que les informations de CNAME si la largeur de bande est limitée. Il faut transmettre les CNAME pour que la détection des collisions d'identificateurs SSRC puisse fonctionner. Un traducteur qui génère ses propres paquets RR doit envoyer les informations SDES CNAME le concernant aux mêmes nuages auxquels il envoie ces paquets RR.

BYE: les traducteurs transmettent les paquets BYE sans les modifier. Les traducteurs possédant leur propre identificateur SSRC doivent produire des paquets BYE comportant cet identificateur SSRC s'ils sont sur le point d'arrêter de transmettre des paquets.

APP: les traducteurs transmettent les paquets APP sans les modifier.

A.7.3 Traitement RTCP dans les mélangeurs

Etant donné qu'un mélangeur génère un nouveau flux de données, il ne transmet pas les paquets SR ou RR mais il génère de nouvelles informations dans les deux sens.

Informations d'émetteur SR: un mélangeur ne transmet pas les informations d'émetteur provenant des sources qu'il mélange car les caractéristiques des flux de données de la source sont perdues dans le mélange. En tant que source de synchronisation, le mélangeur génère ses propres paquets SR contenant les informations d'émetteur concernant le flux de données mélangé et les envoie dans le même sens que le flux de données mélangé.

Blocs de rapport de réception SR/RR: un mélangeur génère ses propres rapports de réception pour les sources de chaque nuage et ne les envoie qu'au nuage correspondant. Il n'envoie pas ces rapports de réception aux autres nuages et ne transmet pas non plus les rapports de réception provenant d'un nuage aux autres nuages car les sources ne sont pas des SSRC (uniquement des CSRC).

SDES: les mélangeurs transmettent généralement sans modification les informations SDES envoyées par un nuage aux autres nuages, mais ils peuvent par exemple décider de filtrer les informations SDES autres que les informations de CNAME si la largeur de bande est limitée. Il faut

transmettre les CNAME pour que la détection des collisions d'identificateurs SSRC puisse fonctionner. (Un identificateur d'une liste CSRC produit par un mélangeur peut entrer en collision avec un identificateur SSRC produit par un système d'extrémité.) Un mélangeur doit envoyer les informations SDES CNAME le concernant aux mêmes nuages auxquels il envoie les paquets SR ou RR.

Etant donné que les mélangeurs ne transmettent pas les paquets SR ou RR, ils extraient généralement les paquets SDES d'un paquet RTCP composé. Pour réduire au minimum le surdébit, des tranches de paquets SDES peuvent être regroupées en un seul paquet SDES qui est ensuite empilé sur un paquet SR ou RR provenant du mélangeur. Le débit de paquets RTCP peut être différent de chaque côté du mélangeur.

Un mélangeur qui n'insère pas d'identificateur CSRC peut aussi s'abstenir de transmettre les informations SDES CNAME. Dans ce cas, les espaces d'identificateur SSRC des deux nuages sont indépendants. Comme il a été mentionné plus haut, ce mode de fonctionnement crée le danger d'une incapacité à détecter des boucles.

BYE: il est nécessaire que les mélangeurs transmettent les paquets BYE. Ils doivent produire des paquets BYE avec leurs propres identificateurs SSRC s'ils sont sur le point d'arrêter de transmettre des paquets.

APP: le traitement des paquets APP par les mélangeurs est propre à l'application.

A.7.4 Mélangeurs en cascade

Une session RTP peut faire intervenir tout un ensemble de mélangeurs et de traducteurs comme le montre la Figure A.3. Si deux mélangeurs sont en cascade, comme M2 et M3 sur la figure, les paquets reçus par un mélangeur peuvent déjà avoir été mélangés et peuvent comporter une liste CSRC de plusieurs identificateurs. Le deuxième mélangeur doit établir la liste CSRC pour le paquet sortant à partir des identificateurs CSRC des paquets d'entrée déjà mélangés et des identificateurs SSRC des paquets d'entrée non mélangés. Cette liste est montrée en sortie du mélangeur M3 appelé M3:89 (64, 45) sur la Figure A.3. Comme dans le cas de mélangeurs qui ne sont pas en cascade, si la liste CSRC résultante comporte plus de 15 identificateurs, les identificateurs en surplus ne peuvent pas être inclus.

A.8 Attribution et utilisation des identificateurs SSRC

L'identificateur SSRC figurant dans l'en-tête RTP et dans divers champs des paquets RTCP est un nombre aléatoire sur 32 bits qui doit absolument être unique sur le plan mondial dans une session RTP donnée. Il est très important que ce nombre soit choisi avec soin afin que les participants qui sont sur le même réseau ou qui démarrent en même temps ne risquent pas de choisir le même nombre.

Il ne suffit pas d'utiliser l'adresse de réseau local (une adresse IPv4 par exemple) pour l'identificateur car l'adresse peut ne pas être unique. Etant donné que les traducteurs et les mélangeurs RTP permettent l'interfonctionnement de plusieurs réseaux avec des espaces d'adresses différents, les schémas d'attribution des adresses dans deux espaces pourraient conduire à un taux de collision beaucoup plus élevé que si l'attribution était aléatoire.

Plusieurs sources fonctionnant sur un même serveur entreraient également en conflit.

Il ne suffit pas d'obtenir un identificateur SSRC simplement en invoquant la fonction de randomisation () sans en initialiser l'état avec soin. Le paragraphe A.8.2 présente un exemple de procédure pour produire un identificateur aléatoire.

A.8.1 Probabilité de collision

Etant donné que les identificateurs sont choisis au hasard, il est possible que plusieurs sources choisissent le même nombre. La probabilité de collision est la plus élevée lorsque toutes les sources

démarrer simultanément, par exemple lorsqu'elles sont déclenchées automatiquement par un événement de gestion de session. Si N est le nombre de sources et L la longueur de l'identificateur (ici 32 bits), la probabilité pour que deux sources tirent de façon indépendante la même valeur peut être approchée pour N grand [20] par $1 - \exp^{-\frac{N^2}{2^{L+1}}}$. Pour $N = 1000$, la probabilité vaut environ 10^{-4} .

La probabilité de collision est généralement beaucoup plus faible que celle indiquée ci-dessus qui correspond au cas le plus défavorable. Lorsqu'une nouvelle source rejoint une session RTP dans laquelle toutes les autres sources ont déjà un identificateur unique, la probabilité de collision correspond à la fraction de nombres utilisés en dehors de l'espace. Si N est de nouveau le nombre de sources et L la longueur de l'identificateur, la probabilité de collision vaut $\frac{N}{2^L}$. Pour $N = 1000$, cette probabilité vaut environ $2 \cdot 10^{-7}$. La probabilité de collision est encore réduite si une nouvelle source reçoit des paquets des autres participants avant d'envoyer son propre paquet (de données ou de commande). Si la nouvelle source garde la trace des autres participants (par l'identificateur SSRC), alors, avant d'envoyer son premier paquet, elle peut vérifier que son identificateur n'entre en conflit avec aucun de ceux qu'elle a reçus; si ce n'est pas le cas, elle en choisit un autre.

A.8.2 Résolution des collisions et détection des boucles

Bien que la probabilité de collision d'identificateurs SSRC soit faible, toutes les implémentations du protocole RTP doivent pouvoir détecter les collisions et prendre des mesures appropriées pour les résoudre. Si une source découvre à un instant quelconque qu'une autre source utilise le même identificateur SSRC que le sien, elle doit envoyer un paquet RTCP BYE pour l'ancien identificateur et en choisir un autre au hasard. Si un récepteur découvre que deux autres sources sont en collision, il peut conserver les paquets de l'une et ignorer les paquets de l'autre lorsque cette collision peut être détectée par des adresses de transport de source ou des CNAME différents. On attend des deux sources qu'elles résolvent la collision afin que la situation ne s'éternise pas.

Etant donné que l'unicité globale des identificateurs aléatoires est assurée pour chaque session RTP, ces identificateurs peuvent être utilisés pour détecter les boucles qui peuvent être engendrées par les mélangeurs et les traducteurs. Une boucle entraîne la duplication des informations de données et de commande, soit non modifiées soit éventuellement mélangées, comme dans les exemples suivants:

- un traducteur peut transmettre un paquet par erreur au groupe de multidiffusion qui lui a envoyé le paquet, soit directement soit par l'intermédiaire d'une chaîne de traducteurs. Dans ce cas, le même paquet apparaît plusieurs fois, provenant de différentes sources de réseau;
- deux traducteurs mis en parallèle par erreur, c'est-à-dire avec les mêmes groupes de multidiffusion, transmettent tous les deux les paquets d'un groupe de multidiffusion à un autre. Les traducteurs unidirectionnels produiront deux copies; les traducteurs bidirectionnels forment une boucle;
- un mélangeur peut former une boucle en envoyant les paquets qu'il reçoit d'une destination de transport à cette même destination, soit directement soit par l'intermédiaire d'un autre mélangeur ou traducteur. Dans ce cas, une source peut apparaître à la fois comme une source SSRC dans un paquet de données et comme une source CSRC dans un paquet de données mélangé.

Une source peut découvrir une boucle concernant ses propres paquets ou les paquets d'une autre source (boucle causée par un tiers). Les boucles et les collisions produites par le choix aléatoire d'un identificateur de source se traduisent par des paquets arrivant avec le même identificateur SSRC mais avec une adresse de transport de source différente, qui peut être celle du système d'extrémité ayant produit ce paquet ou celle d'un système intermédiaire. Par conséquent, si une source change d'adresse de transport, elle doit aussi choisir un nouvel identificateur SSRC pour éviter d'être interprétée comme une source bouclée. Les boucles ou les collisions se produisant du côté distant d'un traducteur ou d'un mélangeur ne peuvent pas être détectées au moyen de l'adresse de transport

de source si toutes les copies des paquets passent dans le traducteur ou le mélangeur; toutefois, les collisions peuvent continuer à être détectées lorsque des tranches provenant de deux paquets RTCP SDES contiennent le même identificateur SSRC mais des CNAME différents.

Pour détecter et résoudre ces conflits, les implémentations du protocole RTP doivent intégrer un algorithme similaire à celui décrit ci-dessous. Cet algorithme ignore les paquets qui proviennent d'une nouvelle source ou d'une boucle et qui entrent en collision avec une source établie. Il résout les collisions avec l'identificateur SSRC du participant en envoyant un paquet RTCP BYE pour l'ancien identificateur et en choisissant un nouvel identificateur. Toutefois, lorsque la collision provient d'une boucle causée par les propres paquets du participant, l'algorithme ne choisira un nouvel identificateur qu'une seule fois et ignorera ensuite les paquets dont l'adresse est l'adresse de transport de la source bouclante. Cela est nécessaire pour éviter une avalanche de paquets BYE.

Dans cet algorithme, l'adresse de transport de source doit être la même pour les paquets RTP et RTCP provenant d'une même source. L'algorithme doit être modifié pour prendre en charge les applications qui ne satisfont pas cette contrainte.

Dans cet algorithme, il est nécessaire de conserver un tableau indexé par les identificateurs de source et contenant l'adresse de transport de source à partir de laquelle l'identificateur a été reçu (la première fois), ainsi qu'un autre état pour cette source. Chaque identificateur SSRC ou CSRC reçu dans un paquet de données ou de commande est recherché dans ce tableau en vue du traitement de ces informations de données ou de commande. Pour les paquets de commande, chaque élément avec son propre identificateur SSRC, par exemple une tranche SDES, nécessite une recherche distincte. (La présence d'un identificateur SSRC dans un bloc de rapport de réception est une exception.) Si l'identificateur SSRC ou CSRC n'est pas trouvé, une nouvelle rubrique est créée. Une rubrique du tableau est supprimée lorsqu'est reçu un paquet RTCP BYE contenant l'identificateur SSRC correspondant, ou après un temps relativement long pendant lequel aucun paquet n'est arrivé (voir A.6.2.1, Mise à jour du nombre de membres de la session).

Pour repérer les boucles causées par les paquets de données du participant, il est également nécessaire de conserver une liste séparée des adresses de transport de source (et pas des identificateurs) pour lesquelles on a repéré un conflit. On notera que cette liste doit être courte et la plupart du temps vide. Chaque élément de cette liste comporte l'adresse de source ainsi que l'instant auquel a été reçu le dernier paquet conflictuel. Un élément peut être supprimé de la liste lorsque aucun paquet conflictuel n'est arrivé en provenance de cette source pendant une période de l'ordre de dix intervalles de rapport RTCP (voir A.6.2, Intervalle de transmission RTCP).

Dans l'algorithme présenté, on suppose que l'état et l'identificateur de source du participant figurent dans le tableau des identificateurs de source. L'algorithme pourrait être restructuré pour qu'il commence par vérifier si l'identificateur de source du participant figure dans le tableau.

Si l'identificateur SSRC ou CSRC n'est pas trouvé dans le tableau des identificateurs de source:

alors créer une nouvelle rubrique contenant l'adresse de transport de source et l'identificateur SSRC ou CSRC ainsi qu'un autre état.

Continuer le traitement normal.

(L'identificateur est trouvé dans le tableau.)

Si l'adresse de transport de source indiquée dans le paquet correspond à l'adresse enregistrée dans le tableau pour cet identificateur:

alors continuer le traitement normal.

(Une collision d'identificateurs ou une boucle est indiquée.)

Si l'identificateur de la source n'est pas celui du participant:

alors si l'identificateur de la source provient d'une tranche RTCP SDES contenant un élément CNAME qui est différent du CNAME figurant dans la rubrique du tableau:

- alors (option) il s'agit d'une collision causée par un tiers;
- sinon (option) il s'agit d'une boucle causée par un tiers;
- abandonner le traitement du paquet de données ou de l'élément de commande.

(Il s'agit d'une collision ou d'une boucle causée par les propres données du participant.)

Si l'adresse de transport de source est trouvée dans la liste des adresses conflictuelles:

alors si l'identificateur de source ne provient pas d'une tranche RTCP SDES contenant un élément CNAME, OU si ce CNAME est celui du participant:

- alors (option) il s'agit d'une occurrence de bouclage du trafic du participant. Inscire l'heure courante dans une rubrique de la liste des adresses conflictuelles;
- abandonner le traitement du paquet de données ou de l'élément de commande.

Journaliser l'occurrence d'une collision.

Créer une nouvelle rubrique dans la liste des adresses conflictuelles et inscrire l'heure courante.

Envoyer un paquet RTCP BYE avec l'ancien identificateur SSRC.

Choisir un nouvel identificateur.

Créer dans le tableau des identificateurs de source une nouvelle rubrique comportant l'ancien identificateur SSRC ainsi que l'adresse de transport de source associée au paquet en cours de traitement.

Continuer le traitement normal.

Dans cet algorithme, les paquets associés à l'adresse d'une source qui vient juste d'entrer en conflit doivent être ignorés et les paquets provenant de la source d'origine doivent être conservés. (Si la source d'origine passe par un mélangeur et si plus tard la même source est reçue directement, il peut être judicieux que le récepteur commute sauf si cela entraîne la perte d'autres sources du mélange.) Si aucun paquet n'arrive en provenance de la source d'origine pendant une longue période, la rubrique du tableau arrivera en fin de temporisation et la nouvelle source pourra prendre le relais. Cela se produira si la source d'origine détecte la collision et change d'identificateur de source, mais dans le cas général, un paquet RTCP BYE doit être reçu en provenance de la source d'origine, ce qui permet de supprimer l'état sans avoir à attendre la fin de la temporisation.

Lorsqu'un nouvel identificateur SSRC est choisi par suite d'une collision, l'identificateur candidat doit d'abord être recherché dans le tableau des identificateurs de source pour voir s'il est déjà utilisé par une autre source. Si c'est le cas, il faut produire un autre identificateur candidat et répéter le processus.

Une boucle de paquets de données en direction d'une destination de multidiffusion peut causer un grave encombrement du réseau. Il est nécessaire que tous les mélangeurs et traducteurs implémentent un algorithme de détection des boucles – Comme celui qui est décrit ici – De façon à pouvoir rompre les boucles. Cet algorithme doit limiter l'excès de trafic à une quantité au plus égale au trafic d'origine, ce qui permet de poursuivre la session afin de pouvoir trouver ce qui a créé la boucle. Toutefois, dans les cas extrêmes où un mélangeur ou un traducteur ne rompt pas correctement la boucle, entraînant des niveaux de trafic élevés, il peut être nécessaire que les systèmes d'extrémité cessent toute transmission de paquets de données ou de commande. Cette décision peut dépendre de l'application. Une condition d'erreur doit être indiquée, le cas échéant. La transmission peut être tentée à nouveau périodiquement après une longue durée aléatoire (de l'ordre de quelques minutes).

A.9 Sécurité

On trouvera à l'Appendice I des renseignements sur certaines méthodes de sécurité Internet. La confidentialité et les méthodes d'échange de clé H.323 sont décrites dans l'UIT-T H.323.

A.10 Protocole RTP au-dessus des protocoles de réseau et de transport

Le présent paragraphe traite de problèmes propres à l'acheminement de paquets RTP dans des protocoles de réseau et de transport particuliers. Les règles suivantes s'appliquent sauf si elles sont remplacées par des définitions propres au protocole qui sortent du cadre de la présente Recommandation.

Le protocole RTP s'appuie sur le ou les protocoles sous-jacents pour assurer le démultiplexage des flux de données RTP et des flux de commande RTCP. Pour le protocole UDP et les protocoles similaires, le numéro d'accès RTP est un numéro pair et le numéro d'accès RTCP correspondant est le numéro (impair) qui vient juste après. Si le numéro d'accès RTP fourni pour une application est impair, l'application doit remplacer ce numéro par le numéro (pair) qui le suit immédiatement.

Les paquets de données RTP ne comportent ni champ de longueur ni autre délimitation, le protocole RTP attend donc du ou des protocoles sous-jacents qu'ils fournissent une indication de longueur. La longueur maximale des paquets RTP n'est limitée que par les protocoles sous-jacents.

Si des paquets RTP sont inclus dans un flux d'octets continu plutôt que dans des messages (paquets) au niveau du protocole sous-jacent, il faut définir une encapsulation des paquets RTP pour fournir un mécanisme de tramage. Le tramage est également nécessaire si le bourrage est possible au niveau du protocole sous-jacent de sorte que la longueur de la charge utile RTP ne puisse pas être déterminée. Le mécanisme de tramage n'est pas défini ici.

Un profil peut spécifier une méthode de tramage à utiliser même lorsque des paquets RTP sont transportés dans des protocoles qui ne fournissent pas de tramage afin que plusieurs paquets RTP puissent rentrer dans une seule unité de données protocolaire de couche inférieure, un paquet UDP par exemple. Si plusieurs paquets RTP peuvent être contenus dans un même paquet de réseau ou de transport, le préfixe d'en-tête est réduit et la synchronisation entre les différents flux peut être simplifiée.

A.11 Récapitulatif des constantes protocolaires

Le présent paragraphe récapitule sommairement les constantes définies dans la présente Recommandation.

Les constantes de type de charge utile (PT, *payload type*) RTP sont définies dans les profils plutôt que dans la présente Recommandation. Toutefois, il ne faut pas attribuer les valeurs réservées 200 et 201 (en décimal) à l'octet de l'en-tête RTP qui contient le ou les bits de marqueur ainsi que le type de charge utile pour que les paquets RTP puissent être distingués des types de paquet RTCP SR et RR dans la procédure de validation d'en-tête décrite au A.6.3. Pour la définition normalisée donnée dans la présente Recommandation (un bit de marqueur et 7 bits de type de charge utile), cette restriction signifie que les types de charge utile 72 et 73 sont réservés.

A.11.1 Types de paquet RTCP

Abréviation	Nom	Valeur
SR	Rapport d'émetteur	200
RR	Rapport de récepteur	201
SDES	Description de source	202
BYE	Au revoir	203
APP	Défini par l'application	204

Ces valeurs de type ont été choisies dans l'intervalle 200 à 204 afin d'améliorer le contrôle de validité d'en-tête des paquets RTCP par rapport aux paquets RTP ou aux autres paquets non connexes. Lorsque le champ de type de paquet RTCP est comparé à l'octet correspondant de l'en-tête RTP, cet intervalle correspond à un bit de marqueur valant 1 (ce qui n'est généralement pas le cas pour les paquets de données) et à un bit le plus élevé du champ de type de charge utile normalisé valant 1 (étant donné que les types de charge utile statiques sont généralement définis dans la moitié inférieure). Cet intervalle a également été choisi de façon à être numériquement éloigné des valeurs 0 et 255 étant donné que les séquences ne comportant que des zéros et que des uns sont des séquences de données courantes.

Etant donné que tous les paquets RTCP composés doivent commencer par un paquet SR ou RR, on a choisi pour les codes de ces types de paquet un couple pair/impair pour permettre au contrôle de validité RTCP de tester le plus grand nombre de bits par rapport à un masque et à une valeur.

D'autres constantes sont attribuées par l'IANA. Les personnes réalisant des expériences sont invitées à enregistrer les numéros dont elles ont besoin pour leurs expériences, et à radier ensuite les numéros qui se révèlent non nécessaires.

A.11.2 Types d'éléments SDES

Abréviation	Nom	Valeur
END	Fin de liste SDES	0
CNAME	Nom canonique	1
NAME	Nom d'utilisateur	2
EMAIL	Adresse électronique de l'utilisateur	3
PHONE	Numéro de téléphone de l'utilisateur	4
LOC	Situation géographique de l'utilisateur	5
TOOL	Nom d'application ou d'outil	6
NOTE	Avis concernant la source	7
PRIV	Extensions privées	8

D'autres constantes sont attribuées par l'IANA. Les personnes réalisant des expériences sont invitées à enregistrer les numéros dont elles ont besoin pour leurs expériences, et à radier ensuite les numéros qui se révèlent non nécessaires.

A.12 Spécifications de profil et de format de charge utile RTP

La spécification complète du protocole RTP pour une application donnée nécessitera un ou plusieurs documents associés de deux types décrits ici: spécifications de profil et de format de charge utile.

Le protocole RTP peut être utilisé pour diverses applications ayant des prescriptions qui diffèrent quelque peu. La flexibilité d'adaptation à ces prescriptions est assurée par les diverses possibilités de choix dans la partie principale de la spécification de protocole et par la possibilité de définir, dans un document de profil distinct, des extensions concernant un environnement particulier et une classe d'applications particulière. Une application ne fonctionnera généralement que conformément à un seul profil, le profil utilisé n'est donc pas indiqué explicitement. On trouvera à l'Annexe B un profil pour les applications audio et vidéo.

La spécification d'un format de charge utile, qui est le second type de document associé, définit la manière dont un type particulier de données de charge utile, des données vidéo codées H.261 par exemple, doit être transporté dans le protocole RTP. Le titre de ces documents est du type "Format de charge utile RTP pour le codage audio/vidéo XYZ". Etant donné qu'on peut avoir besoin d'un format de charge utile sous divers profils, on peut définir les formats indépendamment du profil. Les documents de profil doivent alors spécifier un mappage par défaut entre les formats et des valeurs de type de charge utile, si nécessaire. (Voir l'Annexe C.)

Dans la présente Recommandation, les éléments suivants ont été identifiés comme pouvant être définis dans un profil, mais cette liste ne se veut pas exhaustive:

en-tête de données RTP: l'octet de l'en-tête de données RTP qui contient le bit de marqueur et le champ de type de charge utile peut être redéfini par un profil pour répondre à des prescriptions différentes, par exemple une prescription avec un nombre de bits de marqueur différent (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

types de charge utile: en supposant qu'un champ de type de charge utile est présent, le profil définira généralement un ensemble de formats de charge utile (des codages de média par exemple) et un mappage statique par défaut entre ces formats et les valeurs de type de charge utile. Certains formats de charge utile peuvent être définis par référence à des spécifications de format de charge utile séparées. Pour chaque type de charge utile défini, le profil doit spécifier la fréquence de l'horloge d'horodatage RTP à utiliser (voir A.5.1, Champs de l'en-tête fixe RTP).

ajouts à l'en-tête de données RTP: des champs supplémentaires peuvent être rattachés à l'en-tête de données RTP fixe si une fonctionnalité supplémentaire indépendante du type de charge utile est requise par une classe d'applications conformes à un profil donné (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

extensions de l'en-tête de données RTP: il faut définir le contenu des 16 premiers bits de la structure d'extension de l'en-tête de données RTP si l'utilisation de ce mécanisme pour des extensions propres à la réalisation est accordée par le profil (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

types de paquets RTCP: de nouveaux types de paquet RTCP propres à une classe d'applications peuvent être définis et enregistrés auprès de l'IANA.

intervalle de rapport RTCP: un profil doit spécifier s'il doit être fait usage des valeurs proposées au A.6.2, Intervalle de transmission RTCP pour les constantes servant au calcul de l'intervalle de rapport RTCP. Il s'agit de la fraction RTCP de largeur de bande de la session, de l'intervalle de rapport minimal et du partage de la largeur de bande entre émetteurs et récepteurs. Un profil peut spécifier d'autres valeurs s'il a été démontré qu'elles pouvaient être évolutives.

extension SR/RR: il est possible de définir une section d'extension pour les paquets RTCP SR et RR si des informations supplémentaires concernant l'émetteur ou les récepteurs doivent faire l'objet de rapports réguliers (voir A.6.3.3, Extension des rapports d'émetteur et de récepteur).

utilisation d'éléments SDES: le profil peut spécifier des rangs de priorité relatifs pour les éléments RTCP SDES devant être transmis ou exclus (voir A.6.2.2, Attribution de largeur de bande pour la description de source); une autre syntaxe ou une autre sémantique pour l'élément CNAME (voir A.6.4.1, CNAME: élément SDES identificateur d'extrémité canonique); le format de

l'élément LOC (voir A.6.4.5, LOC: élément SDES situation géographique de l'utilisateur); la sémantique et l'utilisation de l'élément NOTE (voir A.6.4.7, NOTE – Élément SDES avis/état); ou de nouveaux types d'élément SDES à enregistrer auprès de l'IANA.

sécurité: un profil peut spécifier les algorithmes et les services de sécurité qui doivent être offerts par les applications, et peut fournir des indications sur leur utilisation (voir A.9, Sécurité).

mappage chaîne-clé: un profil peut spécifier le mappage entre un mot de passe fourni par l'utilisateur et une clé de chiffrement.

protocole sous-jacent: il peut être nécessaire d'utiliser un protocole particulier de couche Transport ou réseau sous-jacent pour transporter les paquets RTP.

mappage de transport: il est possible de spécifier un mappage entre d'une part les protocoles RTP et RTCP et d'autre part les adresses au niveau transport (accès UDP par exemple) qui soit différente du mappage normalisé défini à l'Annexe B.

encapsulation: il est possible de définir une encapsulation des paquets RTP pour pouvoir transporter plusieurs paquets de données RTP dans un seul paquet de couche inférieure ou pour fournir un tramage au-dessus des protocoles sous-jacents qui ne fournissent pas de tramage (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport).

A.13 Algorithmes

On trouvera le présent paragraphe à l'Appendice I. Tous ces exemples d'implémentations ne sont donnés qu'à titre d'information et ne sont donc pas inclus ici.

A.14 Bibliographie

On notera que les documents cités dans la présente bibliographie ne sont donnés qu'à titre d'information, et qu'ils ne sont pas requis pour l'implémentation de la présente annexe.

- [A-1] CLARK (D.D.) et TENNENHOUSE (D.L.): Architectural considerations for a new generation of protocols, in *SIGCOMM Symposium on Communications Architectures and Protocols*, (Philadelphie, Pennsylvanie), pp. 200-208, *IEEE*, septembre 1990. *Computer Communications Review*, Vol. 20 (4), septembre 1990.
- [A-2] COMER (D.E.): *Internetworking with TCP/IP*, Vol. 1, *Prentice Hall*, Englewood Cliffs, New Jersey, 1991.
- [A-3] POSTEL (J.): Internet protocol, RFC 791, *Internet Engineering Task Force*, septembre 1981.
- [A-4] MILLS (D.): Network time protocol (v3), RFC 1305, *Internet Engineering Task Force*, avril 1992.
- [A-5] EASTLAKE (D.), CROCKER (S.) et SCHILLER (J.): Randomness recommendations for security, RFC 1750, *Internet Engineering Task Force*, décembre 1994.
- [A-6] BOLOT (J.-C.), TURLETTI (T.) et WAKEMAN (I.): Scalable feedback control for multicast video distribution in the internet, in *SIGCOMM Symposium on Communications Architectures and Protocols*, pp. 58-67, *ACM*, Londres, août 1994.
- [A-7] BUSSE (I.), DEFFNER (B.) et SCHULZRINNE (H.): Dynamic QOS control of multimedia applications based on RTP, *Computer Communications*, janvier 1996.
- [A-8] FLOYD (S.) et JACOBSON (V.): The synchronization of periodic routing messages, in *SIGCOMM Symposium on Communications Architectures and Protocols* (D. P. Sidhu, ed.), pp. 33-44, *ACM*, (San Francisco, Californie), septembre 1993.
- [A-9] CADZOW (J.A.): *Foundations of digital signal processing and data analysis*, *Macmillan*, New York, 1987.

- [A-10] ISO/CEI 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*.
- [A-11] MOCKAPETRIS (P.): Domain names – Concepts and facilities, STD 13, RFC 1034, *Internet Engineering Task Force*, novembre 1987.
- [A-12] MOCKAPETRIS (P.): Domain names – Implementation and specification, STD 13, RFC 1035, *Internet Engineering Task Force*, novembre 1987.
- [A-13] BRADEN (R.): Requirements for internet hosts – Application and support, STD 3, RFC 1123, *Internet Engineering Task Force*, octobre 1989.
- [A-14] REKHTER (Y.), MOSKOWITZ (R.), KARREBERG (D.) et de GROOT (G.): Address allocation for private internets, RFC 1597, *Internet Engineering Task Force*, mars 1994.
- [A-15] LEAR (E.), FAIR (E.), CROCKER (D.) et KESSLER (T.): Network 10 considered harmful (some practices should not be codified), RFC 1627, *Internet Engineering Task Force*, juillet 1994.
- [A-16] CROCKER (D.): Standard for the format of ARPA internet text messages, STD 11, RFC 822, *Internet Engineering Task Force*, août 1982.
- [A-17] FELLER (W.): An Introduction to Probability Theory and its Applications, Vol. 1, *John Wiley and Sons*, third ed., New York, 1968.
- [A-18] BALENSON (D.): Privacy enhancement for internet electronic mail: Part III: algorithms, modes, and identifiers, RFC 1423, *Internet Engineering Task Force*, février 1993.
- [A-19] VOYDOCK (V.L.) et KENT (S.T.): Security mechanisms in high-level network protocols, *ACM Computing Surveys*, Vol. 15, pp. 135-171, juin 1983.
- [A-20] RIVEST (R.): The MD5 message-digest algorithm, RFC 1321, *Internet Engineering Task Force*, avril 1992.

ANNEXE B

Profil RTP

On se reportera à l'introduction de l'Annexe A; toutes les mises en garde contenues dans l'Annexe A sont également applicables à la présente annexe. On trouvera dans l'Appendice II, à titre d'information, la référence au document IETF complet; cependant, la présente annexe contient toutes les informations nécessaires à l'implémentation de l'UIT-T H.323.

B.1 Introduction

Le présent profil définit certains aspects du protocole RTP non spécifiés dans l'Annexe A. Ce profil est destiné à être utilisé dans le cadre de conférences audio et vidéo avec gestion minimale des sessions. En particulier, aucune prise en charge pour la négociation des paramètres ou la gestion de participation n'est assurée. Ce profil est censé être utile dans les sessions où aucune négociation ou gestion de participation n'est utilisée (par exemple en utilisant les types de charge utile statiques et les indications d'appartenance fournies par le protocole RTCP), ce profil peut également être utile en association avec un protocole de commande de haut niveau.

L'utilisation du présent profil est valide pour certaines applications; il n'y a pas d'indication explicite par numéro d'accès, identificateur de protocole, etc.

D'autres profils peuvent reposer sur des choix différents pour les éléments spécifiés ici.

B.2 Formes de paquets RTP et RTCP et comportement des protocoles

Le paragraphe A.12 "Profils RTP et spécification des formats de charge utile" donne la liste d'un nombre d'éléments qui peuvent être spécifiés ou modifiés à l'intérieur d'un profil. Le présent paragraphe traite de ces éléments. En général, ce profil suit les aspects par défaut ou recommandés de la spécification RTP.

en-tête de données RTP: le format normalisé de l'en-tête de données RTP fixe est utilisé (un bit marqueur).

types de charge utile: les types de charge utile statiques sont définis au B.6, Définitions des types de charge utile.

additions aux en-têtes de données RTP: aucun champ supplémentaire n'est ajouté à l'en-tête de données RTP.

extensions des en-têtes de données RTP: aucune extension d'en-tête RTP n'est définie, mais les applications fonctionnant dans le cadre de ce profil peuvent utiliser de telles extensions d'en-tête. Ainsi, dans les applications, on ne supposera pas que le bit X est toujours égal à zéro et ces applications devront pouvoir ignorer l'extension d'en-tête. Si une extension d'en-tête est définie dans le futur, cette définition devra spécifier le contenu des premiers 16 bits de manière telle que plusieurs extensions différentes puissent être identifiées.

types de paquet RTCP: aucun type additionnel de paquet RTCP n'est défini dans la présente spécification de profil.

intervalle de rapport RTCP: constantes proposées, servant au calcul de l'intervalle entre rapports RTCP.

extension SR/RR: aucune section d'extension n'est définie pour les paquets SR ou RR du protocole RTCP.

utilisation SDES: les applications peuvent utiliser tout élément SDES décrit. L'information CNAME est envoyée à chaque intervalle de rapport, en revanche, les autres éléments doivent être envoyés seulement tous les cinq intervalles de rapport.

sécurité: les services de sécurité par défaut RTP ne sont pas les services par défaut dans le présent profil.

mappage chaîne-touche: voir l'Appendice II concernant ces informations.

protocole sous-jacent: la présence d'un protocole sous-jacent est autorisée et décrit dans l'Appendice IV sous réserve du respect de certaines conditions.

mappage de transport: le mappage normalisé du RTP et RTCP avec les adresses niveau de transport est utilisé.

encapsulation: aucune encapsulation des paquets RTP n'est spécifiée.

B.3 Types de charge utile

Voir l'Appendice II pour des renseignements concernant l'enregistrement des nouveaux types de charge utile.

Il convient de noter que tous les codages à utiliser par le RTP doivent se voir assigner un type de charge utile statique. Les moyens non RTP hors du domaine d'application de la présente annexe (par exemple, pour les services d'annuaire et les protocoles d'invitation) peuvent être utilisés pour établir un mappage dynamique entre un type de charge utile extrait de la gamme 96-127 et un codage. Pour faciliter la tâche de la personne chargée de la réalisation, le présent profil contient des descriptions des codages qui actuellement n'ont pas de type de charge utile statique qui leur est assigné.

L'espace pour les types de charge utile disponibles est relativement petit. Ainsi, les nouveaux types de charge utile statiques sont assignés seulement si les conditions suivantes sont satisfaites:

- le codage intéresse la communauté Internet dans son ensemble;
- il présente des avantages comparé au codage actuel ou est rendu nécessaire pour assurer l'interopérabilité avec les systèmes de conférence ou les systèmes multimédias existants, largement utilisés;
- la description est suffisante pour construire un décodeur.

B.4 Audio

B.4.1 Recommandations indépendantes du codage

Pour les applications qui n'envoient aucun paquet au cours des silences, le premier paquet de signal de parole (premier paquet après une période de silence) est repéré par la valeur du bit marqueur dans l'en-tête de données RTP. Pour les applications sans suppression de silence, le bit doit être mis à zéro.

Le signal d'horloge RTP utilisé pour produire l'horodate RTP est indépendant du nombre de canaux et du codage; il est égal au nombre de périodes d'échantillonnage par seconde. Pour les codages de canal N, chaque période d'échantillonnage (par exemple, 1/8000 s) produit N échantillons. (Cette terminologie est normalisée, mais pas assez précise, étant donné que le nombre total d'échantillons par seconde est alors égal au taux d'échantillonnage que multiplie le décompte du canal.

Si l'on utilise plusieurs canaux audio, les canaux sont numérotés de gauche à droite, à partir de 1. Dans les paquets audio RTP, les informations provenant des canaux de numéro inférieur précèdent celles des canaux de numéro plus élevé.

Pour plus de deux canaux, la convention doit être la suivante:

l	gauche	<i>(left)</i>
r	droit	<i>(right)</i>
c	centre	<i>(centre)</i>
S	ambiophonie	<i>(surround)</i>
F	avant	<i>(front)</i>
R	arrière	<i>(rear)</i>

Canaux	Description	Canaux					
		1	2	3	4	5	6
2	stéréophonique	l	r				
3		l	r	c			
4	tétraphonique	Fl	Fr	Rl	Rr		
4		l	c	r	S		
5		Fl	Fr	Fc	Sl	Sr	
6		l	lc	c	r	rc	S

Les échantillons pour tous les canaux appartenant à un même instant d'échantillonnage doivent se trouver dans le même paquet. L'entrelacement d'échantillons provenant des différents canaux dépend du codage. Des directives générales sont données au B.4.2, Directives pour les codages audio à échantillonnage.

La fréquence d'échantillonnage doit être extraite de l'ensemble 8000, 11 025, 16 000, 22 050, 24 000, 32 000, 44 100 et 48 000 Hz. (Les ordinateurs Apple Macintosh ont des fréquences d'échantillonnage propres de 22 254,54 et 11 127,27 qui peuvent être converties à 22 050 et 11 025 avec une qualité

acceptable en éliminant 4 ou 2 échantillons dans une trame de 20 ms.) Cependant, la plupart des codages audio sont définis pour un ensemble plus restreint de fréquences d'échantillonnage. Les récepteurs doivent pouvoir accepter de l'information audio multicanal, mais peuvent choisir seulement de reproduire un seul canal.

Les recommandations ci-dessous sont des paramètres de fonctionnement par défaut. Les applications doivent pouvoir accepter d'autres valeurs. Les gammes indiquées ici sont simplement destinées à donner des indications aux rédacteurs d'applications, afin de permettre à un ensemble d'applications conformes à ces directives d'interfonctionner sans négociation supplémentaire. Ces directives ne sont pas destinées à limiter les paramètres de fonctionnement pour les applications qui peuvent négocier un jeu de paramètres "interfonctionnables", par exemple par l'intermédiaire d'un protocole de gestion de conférence.

Pour l'audio en mode paquet, l'intervalle de mise en paquet par défaut doit avoir une durée de 20 ms, sauf indication contraire lors de la description du codage. L'intervalle de mise en paquet détermine le délai minimal de bout en bout; les paquets plus longs donnent lieu à moins de préfixes d'en-tête mais à des délais plus élevés et rendront plus facilement repérable la perte de paquets. Pour les applications non interactives (par exemple, pour les cours ou les liaisons avec des contraintes sévères de largeur de bande) un délai de mise en paquet plus élevé peut être tout à fait approprié. Un récepteur doit accepter des paquets représentant de 0 à 200 ms de données audio. Cette restriction permet un dimensionnement raisonnable des tampons pour le récepteur.

B.4.2 Directives pour les codages audio à échantillonnage

Dans les codages à échantillonnage, chaque échantillon audio est représenté par un nombre fixe de bits. Dans les données audio comprimées, les codes pour chaque échantillon peuvent dépasser les limites d'octets. Un paquet audio RTP peut contenir un nombre quelconque d'échantillons audio, étant soumis à la contrainte que le nombre de bits par échantillon multiplié par le nombre d'échantillons par paquet donne un nombre entier d'octets. Les codages fractionnaires donnent moins d'un octet par échantillon.

La durée d'un paquet audio est déterminée par le nombre d'échantillons dans le paquet.

Pour les codages à échantillonnage produisant un ou plusieurs octets par échantillon, les échantillons provenant de différents canaux échantillonnés au même instant d'échantillonnage sont mis en paquets dans des octets consécutifs. Par exemple, pour un codage à deux canaux, la séquence d'octets est (canal de gauche, premier échantillon), (canal de droite, premier échantillon), (canal de gauche, deuxième échantillon), (canal de droite, deuxième échantillon) et ainsi de suite. Pour des codages multioctets, les octets sont transmis dans l'ordre d'octet du réseau (c'est-à-dire, l'octet de plus fort poids d'abord).

La mise en paquets des codages à échantillonnage produisant moins d'un octet par échantillon est propre au codage.

B.4.3 Directives pour les codages audio à trame

Les codages à trame codent un bloc de longueur fixe audio en un autre bloc de données comprimées, en général de longueur fixe aussi. Pour les codages à trame, l'expéditeur peut choisir de combiner plusieurs de ces trames en un seul message. Le récepteur peut indiquer le nombre de trames contenues dans un message étant donné que la durée de trame est définie dans le cadre du codage.

Pour les codecs à trame, l'ordre des canaux est défini pour l'ensemble du bloc. C'est-à-dire, pour l'audio à deux canaux, les échantillons droit et gauche sont codés de manière indépendante, la trame codée pour le canal de gauche précédant celle pour le canal de droite.

Tous les codecs audio à trame doivent être en mesure de coder et de décoder plusieurs trames consécutives dans un seul paquet. Comme la taille de trame pour les codecs à trame est donnée, il

n'est pas nécessaire d'utiliser une désignation distincte pour le même codage, mais avec différents numéros de trame par paquet.

B.4.4 Codages audio

Les caractéristiques des codages audio normalisés sont indiquées dans le Tableau B.1 et leurs types de charge utile sont donnés dans le Tableau B.2.

Tableau B.1/H.225.0 – Caractéristiques des codages audio

Codage	Echantillon/trame	Bits/échantillon	ms/trame
G722	Echantillon	8	
G722.1	Trame	NA	20
G728	Trame	NA	2,5
PCMA	Echantillon	8	
PCMU	Echantillon	8	
G723.1	Trame	NA	30
G729	Trame	NA	10
GSM	Trame	NA	20
ISO/CEI 14496-3	Trame	NA	NA

Tableau B.2/H.225.0 – Types de charge utile (PT) pour les codages normalisés audio et vidéo

PT	Nom du codage	Audio/vidéo (A/V)	Fréquence d'horloge (Hz)	Canaux (audio)
0	PCMU	A	8 000	1
8	PCMA	A	8 000	1
9	G722	A	8 000	1
Dynamique	G722.1	A	16 000	1
4	G723.1	A	8 000	1
15	G728	A	8 000	1
18	G729	A	8 000	1
31	H261	V	90 000	NA
34	H263	V	90 000	NA
3	GSM	A	8 000	1
Dynamique	ISO/CEI 14496-2	V	90 000	NA
Dynamique	ISO/CEI 14496-3	A	90 000	1-5.1 (Note 2)
96-127	Dynamique	?		

NOTE 1 – Les types de charge utile non énumérés dans ce tableau sont réservés. Voir l'Appendice II pour de plus amples détails.

NOTE 2 – Le nombre de voies audio indiqué par "5.1" pour l'ISO/CEI 14496-3 signifie 5 voies plus une voie de caisson d'extrêmes graves.

Voir l'Appendice II pour des informations concernant les codages non indiqués dans le Tableau B.1. La prise en charge de ces codages ne fait pas partie de l'UIT-T H.323.

B.4.4.1 Codage G722

Le codage G722 est spécifié dans l'UIT-T G.722, "Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s".

B.4.4.2 Codage G728

Le codage G728 est défini dans l'UIT-T G.728, "Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code".

B.4.4.3 Codage PCMA

Le codage PCMA (codage MIC en loi A) est spécifié dans l'UIT-T G.711. Les données audio sont codées à raison de huit bits par échantillon, après l'application d'une mise à l'échelle logarithmique.

B.4.4.4 Codage PCMU

Le codage PCMU (codage MIC en loi μ) est spécifié dans l'UIT-T G.711. Les données audio sont codées sur huit bits par échantillon, après l'application d'une mise à l'échelle logarithmique.

B.5 Vidéo

Les codages vidéo ci-dessous sont actuellement définis, ainsi que leurs noms abrégés utilisés pour l'identification. Voir l'Appendice II pour tout codage non décrit ici. Ces codages ne font pas partie de l'UIT-T H.323.

H261

Le codage est spécifié dans l'UIT-T H.261. La mise en paquets et les propriétés propres au RTP sont décrites dans l'Annexe C.

H263

Le codage est spécifié dans l'UIT-T H.263. La mise en paquets et les propriétés propres au RTP sont décrites dans l'Annexe E.

La procédure suivante doit être suivie par des entités H.323 souhaitant transmettre des flux vidéo H.263 (1996 ou 1998):

- dans un message **OpenLogicalChannel** H.245, un expéditeur qui souhaite utiliser le format de charge utile léguée, largement utilisé dans l'industrie pour les flux H.263 (1996), ne doit signaler que les éléments H.263 (1996) et doit omettre la mise en paquets multimédias des paramètres de voie logique H.225.0;
- dans un message **OpenLogicalChannel** H.245, un expéditeur qui souhaite utiliser le format de charge utile défini dans le commentaire RFC 2190 pour les flux H.263 (1996) doit spécifier comme suit **h2250LogicalChannelParameters.rtpPayloadType**: type de charge utile par protocole RTP dans les paramètres de voie logique H.225.0: {rfc-number = 2190, payloadType = 34};
- en général, dans un message **OpenLogicalChannel** H.245, l'expéditeur doit spécifier le format de charge utile conformément à la sémantique définie dans l'UIT-T H.245. Cette procédure doit en particulier être suivie pour signaler le format de charge utile H.263+ (1998) qui est défini dans l'Annexe A, ainsi que ses éventuels successeurs.

B.6 Définitions des types de charge utile

Le Tableau B.2 définit ces valeurs de type de charge utile statique du profil pour le champ PT de l'en-tête de données RTP. De plus, les valeurs de type de charge utile dans la plage 96-127 peuvent être définies de manière dynamique par l'intermédiaire du protocole de gestion de la conférence, qui sort du domaine d'application de la présente Recommandation. Par exemple, un directeur de session

peut spécifier que pour une session donnée, le type de charge utile 96 indique un codage PCMU, une fréquence d'échantillonnage de 8000 Hz, 2 canaux. La gamme de types de charge utile indiquée "réservée" a été mise à part de sorte que les paquets RTCP et RTP peuvent être distingués de manière fiable (voir A.11, Récapitulatif des constantes protocolaires).

Une source RTP émet un seul type de charge utile RTP à un instant donné; l'entrelacement de plusieurs types de charge utile RTP en une session RTP n'est pas autorisé, mais plusieurs sessions RTP peuvent être utilisées en parallèle pour envoyer plusieurs médias. Les types de charge utile actuellement définis dans le présent profil acheminent soit de l'audio ou de la vidéo mais pas les deux. Cependant, on est autorisé à définir des types de charge utile qui combinent plusieurs supports, par exemple audio et vidéo, avec une distinction appropriée dans le format de charge utile. Les participants à une session décident au moyen de mécanismes hors du domaine d'application de la présente Recommandation, de l'ensemble de types de charge utile autorisé dans une session donnée. Cet ensemble peut, par exemple, être défini par les capacités des applications utilisées, négocié par un protocole de gestion de conférence ou être fixé par accord entre les participants.

Tous les codages vidéo actuels utilisent une fréquence d'horodatage de 90 000 Hz, la même que la fréquence d'horodatage de la présentation MPEG. Cette fréquence conduit à des incréments d'horodatage entiers exacts pour fréquence de trame type de 24 (TVHD), 25 (PAL) et 29,97 (NTSC) et 30 Hz (TVHD) et des fréquences images de 50, 59,94 et 60 Hz. La fréquence de 90 kHz est recommandée pour les futurs codages vidéo utilisés dans le cadre du présent profil, mais d'autres fréquences sont possibles. Cependant il n'est pas suffisant d'utiliser une fréquence de trame vidéo (en général comprise entre 15 et 30 Hz), car sa résolution adéquate ne correspond pas aux impératifs de synchronisation type, lorsqu'on calcule l'horodatage RTP correspondant à l'horodatage NTP dans un paquet SR RTPC (voir l'Annexe A). La résolution d'horodatage doit aussi être suffisante pour l'évaluation de gigue figurant dans les rapports de récepteur.

Les codages vidéo normalisés et leurs types de charge utile sont donnés dans le Tableau B.2.

B.7 Assignation des accès

Comme spécifié dans la définition du protocole RTP, les données RTP doivent être acheminées sur un numéro pair d'accès UDP et les paquets RTCP correspondants doivent être acheminés sur le numéro d'accès immédiatement supérieur (impair).

Les applications fonctionnant dans le cadre du présent profil peuvent utiliser toutes les paires d'accès UDP de ce type. Par exemple, la paire d'accès peut être attribuée de façon aléatoire par un programme de gestion de session. Une seule paire de numéros d'accès fixe ne peut pas être requise car les applications utilisant le présent profil fonctionneront vraisemblablement sur le même serveur, et il existera certains systèmes d'exploitation qui ne permettent pas à plusieurs processus d'utiliser le même accès UDP avec différentes adresses de multidiffusion.

Toutefois, les numéros d'accès 5004 et 5005 ont été enregistrés pour être utilisés dans le présent profil pour les applications qui choisissent de les utiliser comme paire par défaut. Les applications qui fonctionnent dans le cadre de plusieurs profils peuvent utiliser cette paire d'accès comme indication pour choisir le présent profil, si elles ne sont pas soumises aux contraintes décrites dans le précédent paragraphe. Les applications doivent avoir une valeur par défaut et peuvent nécessiter la spécification explicite de la paire d'accès. Les numéros d'accès particuliers ont été choisis pour se trouver au-dessus de 5000 afin d'être compatibles avec la pratique d'attribution des numéros d'accès dans le cadre du système d'exploitation Unix, où les numéros d'accès inférieurs à 1024 ne peuvent être utilisés que par des processus privilégiés et les numéros d'accès compris entre 1024 et 5000 sont automatiquement assignés par le système d'exploitation.

Format de charge utile RTP pour les flux vidéo H.261

On se reportera à l'Annexe A; toutes les mises en garde contenues dans l'Annexe A sont également applicables à la présente annexe. On trouvera dans l'Appendice III à titre d'information la référence du Document IETF complet; cependant, la présente annexe contient toutes les informations nécessaires à l'implémentation de l'UIT-T H.323.

C.1 Introduction

L'UIT-T H.261 [12] spécifie les codages utilisés par les codecs de visioconférence conformes aux normes de l'UIT-T. Bien que ces codages aient été à l'origine spécifiés pour des circuits du RNIS à débit de données fixe, les expériences ont montré qu'ils pouvaient être également utilisés sur des réseaux à commutation par paquets tel Internet.

L'objet de la présente annexe est de spécifier un format de charge utile RTP pour l'encapsulation des flux vidéo H.261 (voir l'Annexe A).

C.2 Structure du flux de paquets

C.2.1 Description générale de l'UIT-T H.261

Le codage H.261 est organisé en hiérarchie de groupements. Le flux vidéo est composé de séquences d'images, ou trames, qui sont elles-mêmes organisées en ensembles de groupes de blocs (GOB, *group of blocks*). Il convient de noter que les "images" H.261 sont appelées "trames" dans la présente Recommandation. Chaque GOB contient trois lignes de 11 macroblocs (MB, *macro block*). Chaque macrobloc transporte des informations sur un groupe de 168×16 pixels: l'information de luminance est spécifiée pour 4 blocs de 8×8 pixels, tandis que l'information de chrominance est donnée par deux composantes de différence de couleur "rouge" et "bleu" avec une résolution de seulement 8×8 pixels. Ces composantes et les codecs représentant leurs valeurs échantillonnées sont définis dans l'UIT-R BT.601-5 [C-3].

Ce groupement est utilisé pour spécifier l'information à chaque niveau de la hiérarchie:

- au niveau de la trame, on spécifie l'information telle que le décalage temporel par rapport à la trame précédente, le format d'image et divers indicateurs;
- au niveau du groupe de blocs, on spécifie le numéro du groupe de blocs et le quantificateur par défaut qui doit être utilisé pour les macroblocs;
- au niveau du macrobloc, on spécifie les blocs qui sont présents et qui ne doivent être pas modifiés, et facultativement un quantificateur et les vecteurs de mouvement.

Les blocs qui ont été modifiés sont codés par calcul de la transformée discrète en cosinus (DCT, *discrete cosine transform*) de leurs coefficients, qui sont ensuite quantifiés et codés avec un code Huffman (codes de longueur variable).

Le codage de Huffman H.261 inclut une séquence spéciale de début de groupe de blocs, composée de 15 zéros suivis d'un seul 1, qui ne peut pas être imitée par les autres mots de code. Cette séquence est incluse au début de chaque en-tête de groupe de blocs (et également au début de chaque en-tête de trame) pour indiquer la séparation entre deux groupes de blocs, et est en réalité utilisée pour indiquer que le groupe de blocs courant est terminé. Le codage inclut aussi une séquence de bourrage, composée de 7 zéros suivis par 4 chiffres 1; cette séquence de bourrage peut être seulement introduite entre le codage de macroblocs ou juste avant le séparateur de groupe de blocs.

C.2.2 Mise en paquets

Les codecs H.261 conçus pour fonctionner sur des circuits RNIS produisent un flux binaire composé de plusieurs niveaux de codage spécifiés dans l'UIT-T H.261 et les Recommandations associées. Les bits résultant du codage de Huffman sont disposés en trames de 512 bits, contenant 2 bits de synchronisation, 492 bits de données, 18 bits de code de correction d'erreur. Les trames de 512 bits sont ensuite entrelacées avec un flux audio et transmises sur des circuits $p \times 64$ kbit/s conformément à l'UIT-T H.221 [C-1].

Dans le cas d'une transmission sur Internet, on prendra en considération la sortie du codage de Huffman. Tous les bits produits lors de l'étape de codage de Huffman doivent être inclus dans le paquet. Les trames de 512 bits ne doivent être pas transportées, étant donné que la protection contre les erreurs binaires peut être obtenue par d'autres moyens. De même, les signaux audio et vidéo ne doivent être pas multiplexés dans les mêmes paquets, étant donné que les protocoles UDP et RTP disposent d'un moyen bien plus efficace de multiplexage.

La transmission directe du résultat du codage de Huffman sur un flux non fiable de datagrammes UDP se traduirait toutefois, par une très faible fiabilité. Le résultat de la structure hiérarchique du flux binaire H.261 est tel qu'il est nécessaire de recevoir l'information présente dans l'en-tête de trame pour décoder les groupes de blocs, ainsi que l'information présente dans l'en-tête du groupe de blocs pour décoder les macroblocs. Si aucune précaution n'était prise, cela signifierait qu'il faut recevoir tous les paquets qui transportent une image afin de décoder de manière convenable ses composantes.

Si chaque image pouvait être transportée dans un seul paquet, cette exigence ne créerait pas de problème. Cependant, une image vidéo ou même un groupe de blocs lui-même peut parfois être trop volumineux pour être contenu dans un seul paquet. Par conséquent, le macrobloc est pris comme unité de fragmentation. Les paquets doivent débuter et se terminer sur une limite de macrobloc, c'est-à-dire qu'un macrobloc ne peut pas être fragmenté sur plusieurs paquets. Plusieurs macroblocs peuvent être transportés sur un seul paquet lorsqu'ils s'insèrent convenablement dans leur taille maximale des paquets autorisés. Cette pratique est recommandée afin de réduire le débit de paquet à l'émission et les préfixes de paquet.

Pour permettre à chaque paquet d'être traité indépendamment en vue d'une resynchronisation efficace en présence de pertes de paquets, certaines informations d'état extraites de l'en-tête de trame et de l'en-tête du groupe de blocs sont acheminées avec chaque paquet pour permettre de décoder les macroblocs contenus dans le paquet. Cette information d'état inclut le numéro du groupe de blocs en vigueur au début du paquet, le prédicteur d'adresse de macrobloc (c'est-à-dire la dernière adresse MBA codée dans le précédent paquet), la valeur du quantificateur au début de ce paquet (GQUANT, MQUANT ou zéro dans le cas d'un commencement de groupe de blocs) et les données du vecteur de mouvement de référence (MVD, *motion vector data*) pour calculer les vraies données MVD contenues dans ce paquet. Le flux binaire ne peut être fragmenté entre un en-tête GOB et le macrobloc MB 1 de ce groupe de blocs.

De plus, comme le macrobloc comprimé peut ne pas être contenu dans un nombre entier d'octets, l'en-tête données contient deux entiers à trois bits, SBIT et EBIT, pour indiquer le nombre de bits inutilisés dans le premier et le dernier octet des données H.261 respectivement.

C.3 Spécification du système de mise en paquets

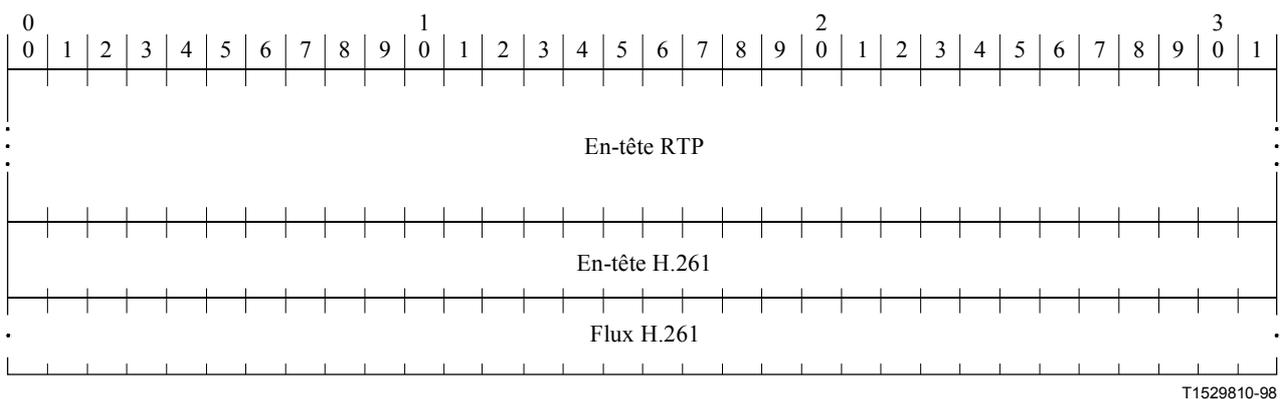
C.3.1 Utilisation du protocole RTP

L'information H.261 est transportée comme des données de charge utile à l'intérieur du protocole RTP. Les champs suivants de l'en-tête RTP sont spécifiés:

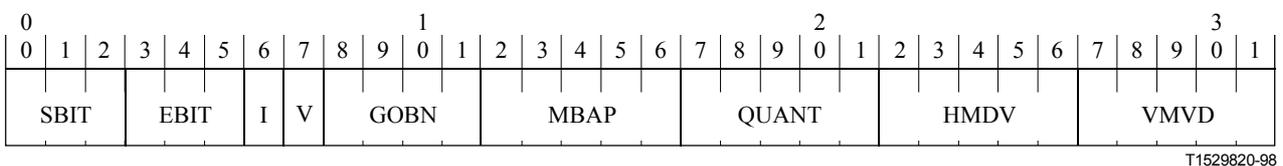
- le type de charge utile doit spécifier le format de la charge utile H.261 (voir l'Annexe B);

- l'horodate RTP code l'instant d'échantillonnage de la première image vidéo contenue dans le paquet de données RTP. L'horodate RTP doit être la même dans des paquets successifs lorsqu'une image vidéo de la même trame vidéo occupe plusieurs paquets. Pour les flux vidéo H.261, l'horodate RTP est établie à partir d'une horloge à 90 kHz. Cette fréquence d'horloge est un multiple de la fréquence de trame H.261 (c'est-à-dire 30 000/1001 ou approximativement 29,97 Hz). De cette manière, pour chaque instant image, l'horloge est simplement incrémentée par le multiple et cela supprime l'imprécision dans le calcul de l'horodate. De plus, la valeur initiale de l'horodate est aléatoire (imprévisible) pouvant rendre plus difficiles les attaques en texte clair dans le chiffrement (voir RTP) (Annexe A). Il convient de noter que si plusieurs trames sont codées dans un paquet (par exemple lorsqu'il y a de faibles modifications entre deux images), il est nécessaire de calculer les temps d'affichage pour les trames après la première en utilisant l'information de synchronisation contenue dans l'en-tête de trame H.261. Cela est nécessaire car l'horodate RTP donne seulement le temps d'affichage de la première trame dans le paquet;
- le bit de marqueur de l'en-tête RTP est mis à "un" dans le dernier paquet d'une trame vidéo. Dans les autres cas il doit être égal à "zéro". Ainsi, il n'est pas nécessaire d'attendre un paquet suivant (qui contient le code de démarrage qui termine la trame actuelle) pour savoir qu'une nouvelle trame doit être affichée.

Les données H.261 suivront l'en-tête RTP comme suit:



L'en-tête H.261 est défini comme suit:



Les champs dans l'en-tête H.261 ont les significations suivantes:

Position bit de démarrage (SBIT, start bit position): 3 bits – Nombre de bits qui doivent être ignorés dans le premier octet de données.

Position fin de bit (EBIT, end bit position): 3 bits – Nombre de bits qui doivent être ignorés dans le dernier octet de données.

Données codées INTRAtrame (I, INTRA-frame encoded data): 1 bit – Mis à 1 si le flux contient seulement des blocs codés INTRAtrame. Mis à 0 si ce flux peut ou peut ne pas contenir des blocs codés INTRAtrame. La signification de ce bit peut ne pas changer au cours d'une même session.

Drapeau vecteur de mouvement (V, motion vector flag): 1 bit – Mis à 0 si les vecteurs de mouvement ne sont pas utilisés dans ce flux. Mis à 1 si les vecteurs de mouvement peuvent ou peuvent ne pas être utilisés dans ce flux. La signification de ce bit peut ne pas changer au cours d'une même session.

Numéro de GOB (GOBN): 4 bits – Code le numéro de groupe de bloc utilisé au début du paquet. Mis à 0 si le paquet commence avec un en-tête GOB.

Prédicteur d'adresse de macrobloc (MBAP, macroblock address predictor): 5 bits – Code le prédicteur d'adresse de macrobloc (c'est-à-dire la dernière adresse MBA codée dans le paquet précédent). La valeur de ce prédicteur est comprise entre 0 et 32 (pour prévoir les adresses MBA valides 1-33), et comme ce flux binaire ne peut pas être fragmenté entre un en-tête GOB et un macrobloc 1, le prédicteur au début du paquet ne peut jamais avoir la valeur 0. Par conséquent, la gamme de valeurs est 1-32, qui est biaisée de -1 pour pouvoir être contenue dans 5 bits. Ainsi, si MBAP = 0, la valeur du prédicteur MBA est 1. Il est mis à zéro si le paquet commence par un en-tête GOB.

Quantificateur (QUANT): 5 bits – La valeur du quantificateur (MQANT ou GQUANT) en cours avant le début du paquet. Mis à 0 si le paquet commence par un en-tête GOB.

Données du vecteur de mouvement horizontal (HMVD, horizontal motion vector data): 5 bits – Données relatives au vecteur de mouvement horizontal de référence (MVD, motion vector data). Mis à 0 si le fanion V est à 0 ou si le paquet commence par un en-tête GOB. Les valeurs HMVD sont des nombres en complément à 2 sur 5 bits appartenant à l'intervalle [-16, +15], où -16 n'est pas utilisé.

Données de vecteur de mouvement vertical (VMVD, vertical motion vector data): 5 bits – Données relatives au vecteur de mouvement vertical de référence (MVD). Mis à 0 si le fanion V est égal à 0 ou si le paquet commence par un en-tête GOB. Les valeurs VMVD sont des nombres en complément à 2 sur 5 bits appartenant à l'intervalle [-16, +15], où -16 n'est pas utilisé.

Il convient de noter que les fanions I et V sont des fanions dissimulés, c'est-à-dire qu'ils peuvent être déduits à partir du flux binaire. Ils sont inclus pour permettre aux décodeurs de procéder à des optimisations qui ne seraient pas possibles si ces fanions dissimulés n'étaient pas fournis avant le décodage du flux binaire. Par conséquent, ces bits ne peuvent pas être modifiés pendant la durée du flux. Une implémentation conforme peut toujours fixer $V = 1$ et $I = 0$.

Les données des vecteurs de mouvement horizontal et vertical doivent être mises à zéro lorsque le MTYPE du dernier macrobloc codé dans le paquet précédent était "mouvement non compensé".

C.3.2 Recommandations relatives au fonctionnement des codecs matériels

Les dispositifs de mise en paquets pour les codecs matériels peuvent de manière triviale être représentés en dehors des limites des groupes de blocs en utilisant la séquence de début de groupe de blocs incluse dans les données H.261. (Il convient de noter que les codeurs à logiciel connaissent déjà les limites.) La mise en paquets la moins coûteuse est celle qui consiste à effectuer cette opération au niveau du groupe de blocs pour tous les groupes de blocs qui peuvent être contenus dans un paquet. Cependant, lorsqu'un groupe de blocs est trop grand, le dispositif de mise en paquets doit l'examiner pour opérer une fragmentation en macroblocs. (Il convient de noter que seul le codage de Huffman peut être analysé et qu'il n'est pas nécessaire de décompresser totalement le flux, ainsi cela ne nécessite que relativement peu de traitement; des exemples de réalisation se trouvent dans l'Appendice III. Il est recommandé d'utiliser la fragmentation au niveau des macroblocs lorsque cela est réalisable afin d'obtenir une mise en paquets plus efficace. Ce système de fragmentation réduit le débit de paquets en sortie et par conséquent réduit les surdébits.

Du côté du récepteur, le flux de données peut être dépaquetisé et acheminé à l'entrée d'un codec matériel. Si ce codec fonctionne à débit fixe, la synchronisation peut être maintenue en insérant une séquence de bourrage entre les macroblocs (c'est-à-dire entre les paquets) lorsque le débit d'arrivée des paquets est inférieur au débit.

C.3.3 Perte des paquets

Sur Internet, la plupart des pertes de paquets sont dues à l'encombrement du réseau et moins aux erreurs de transmission. Avec le protocole UDP, l'expéditeur ne dispose pas d'un mécanisme lui permettant de savoir si un paquet a été bien reçu. Il appartient à l'application, c'est-à-dire au codeur et au décodeur, de traiter la perte des paquets. Chaque paquet RTP inclut un champ numéro de séquence qui peut être utilisé pour détecter la perte de paquets.

L'UIT-T H.261 utilise une redondance temporelle de la vidéo pour effectuer la compression. Ce codage différentiel (ou codage INTERtrame) est sensible à la perte de paquets. Après une perte de paquets, des parties de l'image peuvent rester erronées jusqu'à ce que les macroblocs correspondants aient été codés en mode INTRAtrame (c'est-à-dire codés indépendamment des trames antérieures). Il y a plusieurs façons de palier la perte des paquets:

- 1) une des façons consiste à utiliser le codage INTRAtrame et le bourrage conditionnel au niveau du macrobloc. C'est-à-dire, seuls les macroblocs qui changent (au-delà d'un certain seuil) sont transmis;
- 2) une autre façon consiste à régler le débit de rafraîchissement du codage INTRAtrame en fonction de la perte de paquets observée par les récepteurs. L'UIT-T H.261 spécifie qu'un macrobloc est codé INTRAtrame au moins toutes les 132 fois qu'il est transmis. Cependant, le taux de rafraîchissement INTRAtrame peut être augmenté afin d'accélérer la récupération lorsque le taux de perte mesuré est important;
- 3) la façon la plus rapide de corriger une image erronée est de demander le rafraîchissement d'image codée INTRAtrame après la détection d'une perte de paquets. Une façon de réaliser cette opération est pour le décodeur d'envoyer au codeur la liste des paquets perdus. Le codeur peut décider de coder chaque macrobloc de chaque groupe de blocs de la trame vidéo suivante dans le mode INTRAtrame (c'est-à-dire le codage INTRAtrame intégral), ou si le codeur peut déduire des numéros de séquence des paquets les macroblocs qui sont concernés par la perte, il peut économiser de la largeur de bande en envoyant seulement ces macroblocs dans le mode INTRAtrame. Ce mode est particulièrement efficace pour une connexion point à point ou lorsque le nombre de décodeurs est faible. Le paragraphe suivant spécifie comment la fonction de rafraîchissement peut être implémentée.

C.3.4 Utilisation des paquets de commande H.261 spécifiques facultatifs

La présente Recommandation définit deux paquets de commande RTCP spécifiques à l'UIT-T H.261, les paquets "*Full INTRA-frame Request*" (demande mode INTRAtrame intégral) et "*Negative Acknowledgement*" (acquiescement négatif), décrits dans le présent paragraphe. Leur objet est d'accélérer le rafraîchissement de la vidéo dans les situations où leur utilisation est réalisable. La prise en charge de ces paquets de commande spécifiques H.261 par l'expéditeur H.261 est facultative; en particulier, les premières expériences ont montré que l'utilisation de cette caractéristique pouvait avoir des effets négatifs lorsque le nombre de sites est très important. Aussi, ces paquets de commande doivent-ils être utilisés avec précaution.

Les paquets de commande spécifiques H.261 diffèrent des paquets normaux RTCP dans le sens où ils ne sont pas transmis vers l'adresse de transport de destination RTCP normale pour la session RTP (qui est souvent une adresse de multidiffusion). Ainsi, ces paquets de commande sont envoyés directement par unidiffusion du décodeur au codeur. L'accès de destination pour ces paquets de commande est le même accès que le codeur utilise comme accès source pour la transmission des paquets RTP (données). Par conséquent, ces paquets peuvent être considérés comme des paquets de commande "inverses".

En conséquence, ces paquets de commande peuvent seulement être utilisés lorsque aucun mélangeur ou expéditeur RTP n'intervient dans le trajet du codeur au décodeur. Si des systèmes intermédiaires de ce type venaient à intervenir, l'adresse du codeur ne serait plus présente comme adresse source au

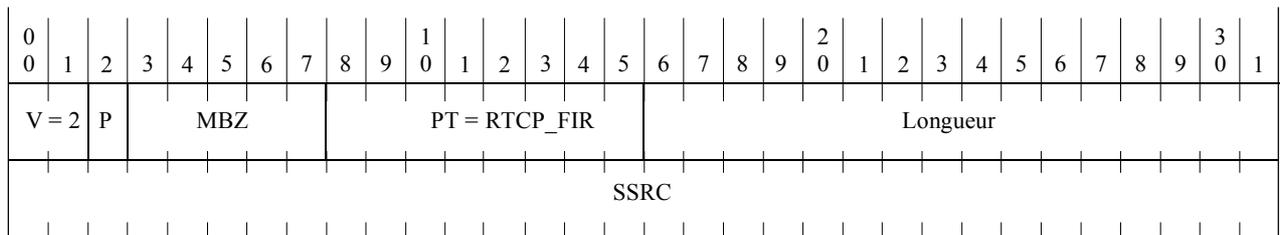
niveau réseau dans les paquets reçus par le décodeur, et en fait, il ne serait pas possible pour le décodeur d'envoyer des paquets directement au codeur.

Certains protocoles de multidiffusion fiables utilisent des paquets de commande NACK similaires transmis suivant le canal de distribution de multidiffusion normal, mais ils utilisent en général des délais aléatoires pour éviter les problèmes liés à l'implosion de paquets NACK. L'objectif de ces protocoles est d'assurer une remise des paquets de multidiffusion fiable au détriment des délais, ce qui est tout à fait justifié pour des applications comme par exemple un tableau blanc utilisé en partage.

Par ailleurs, la transmission vidéo interactive est plus sensible au temps de transmission et n'exige pas une fiabilité totale. Pour les applications vidéo, il semble plus efficace d'envoyer des paquets de commande NACK dès que possible, c'est-à-dire dès qu'une perte est détectée, sans ajouter de délai aléatoire. Dans ce cas, la multidiffusion des paquets de commande NACK produirait un trafic inutile entre les récepteurs étant donné que seul le codeur les doit utiliser. Mais cette méthode est seulement efficace lorsque le nombre de récepteurs est faible. Par exemple, si les paquets de commande spécifiques H.261 sont utilisés dans des connexions point à point ou dans des connexions point à multipoint lorsqu'il y a moins de 10 participants à la conférence.

C.3.5 Définition des paquets de commande

C.3.5.1 Paquet demande mode INTRAtrame intégral (FIR, *full INTRA-frame request*)

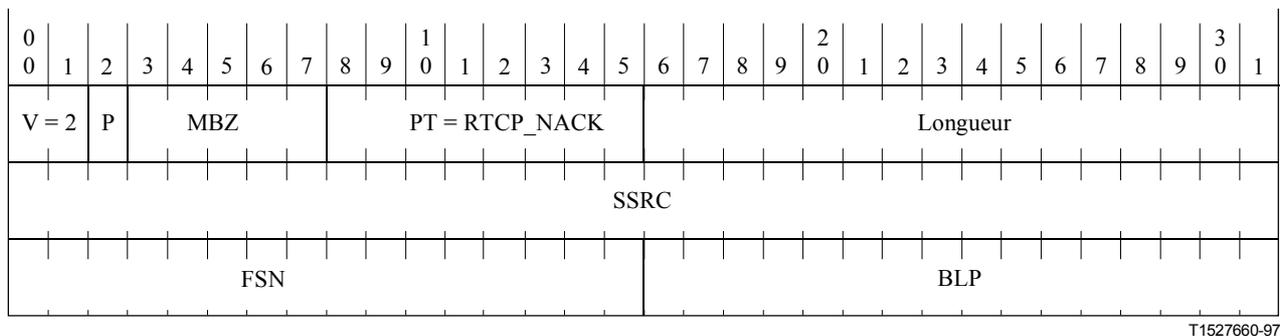


T1527650-97

Ce paquet indique qu'un récepteur demande une image totalement codée afin de soit commencer le décodage avec une image entière, soit de rafraîchir cette image et d'accélérer la récupération après une succession de paquets perdus. Le récepteur demande à la source de forcer l'image suivante à passer dans le mode de codage INTRAtrame intégral, c'est-à-dire sans utilisation du codage différentiel. Les différents champs sont définis dans la spécification du protocole RTP (Annexe A). L'identificateur SSRC est l'identificateur de source de synchronisation pour l'expéditeur de ce paquet. La valeur de l'identificateur de type de paquet (PT) est la constante RTCP_FIR (192).

C.3.5.2 Paquet acquittement négatif (NACK, *negative acknowledgements*)

Le format du paquet NACK est le suivant:



Les différents champs T, P, PT, longueur et SSRC sont définis dans la spécification du protocole RTP (Annexe A). La valeur de l'identificateur type de paquet (PT) est la constante RTCP_NACK (193). L'identificateur SSRC est l'identificateur de source de synchronisation pour l'expéditeur de ce paquet.

Les deux champs restants ont les significations suivantes:

Premier numéro de séquence (FSN, first sequence number): 16 bits. Identifie le premier numéro de la séquence perdue.

Masque binaire des paquets perdus suivants (BLP, bitmask of following lost packets): 16 bits. Un bit est mis à 1 si le paquet correspondant a été perdu et mis à 0 dans le cas contraire. Le champ BLP est mis à 0 si et seulement si aucun paquet autre que celui ayant fait l'objet d'un acquittement négatif (NACK) (en utilisant le champ FSN) a été perdu. Le champ BLP est mis à 0x00001 si le paquet correspondant au numéro FSN et le paquet suivant ont été perdus, etc.

C.4 Bibliographie

- [C-1] UIT-T H.221 (1999), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels.*
- [C-2] UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à $p \times 64$ kbit/s.*
- [C-3] UIT-R BT.601-5 (1995), *Paramètres de codage en studio de la télévision numérique pour des formats standards d'image 4:3 (normalisé) et 16:9 (écran panoramique).*

ANNEXE D

Format de charge utile RTP pour les flux vidéo H.261A

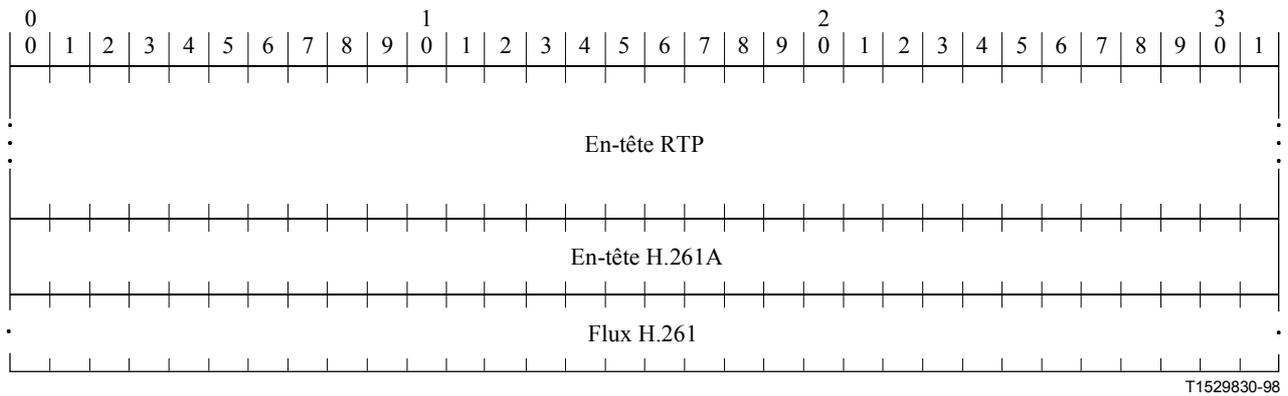
D.1 Introduction

Pour améliorer l'interfaçage des flux vidéo H.323 vers le RCC via des passerelles, l'UIT-T H.323 définit un format modifié de la charge utile vidéo H.261 RTP ce qui permet de faciliter la gestion de mémoire tampon et l'interopérabilité avec les codecs RCC distants. La prise en charge du type de charge utile H.261A est signalée à l'aide des ensembles de capacités H.245 ainsi que dans le message **openLogicalChannel** (ouverture de canal logique) à l'aide des types de charge utile dynamique RTP.

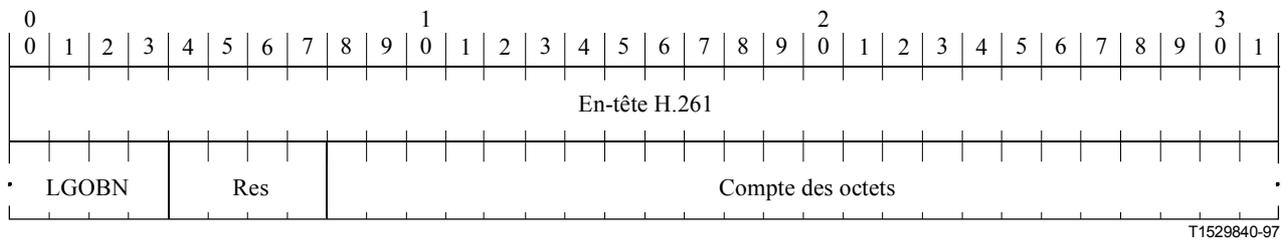
D.2 Mise en paquets RTP H.261A

Cette version est une extension de la version décrite dans l'Annexe C avec un mot supplémentaire de 32 bits qui est attaché à l'en-tête H.261. Les procédures décrites dans l'Annexe C s'appliquent également à la présente annexe.

Les données H.261A viendront après l'en-tête RTP, comme indiqué ci-après:



L'en-tête H.261A est défini comme suit:



Les champs de l'en-tête H.261A ont la signification suivante:

En-tête H.261: 32 bits. Cet en-tête est décrit à l'Annexe C.

Numéro du dernier groupe de blocs (LGOBN, last GOB number): 4 bits. Numéro du dernier groupe de blocs figurant dans le paquet RTP (le numéro maximal de groupe de blocs est 12 pour l'UIT-T H.261).

Réservé (RES): réservé.

Compte des octets: 24 bits. Indique le nombre total d'octets qui ont été envoyés dans la partie flux H.261 des paquets RTP. Si le dernier octet d'un paquet n'est que partiellement rempli (comme indiqué par EBIT), il n'est pas compté dans le compte total d'octets. Ce compte d'octets modulo 2^{24} commence à une valeur aléatoire et n'est jamais réinitialisé.

Les deux champs supplémentaires peuvent être utilisés lorsque des paquets sont perdus ou remis dans le désordre. Le compte d'octets peut servir à déterminer le nombre de bits de bourrage qui doivent être nécessaires dans le flux RCC et ce compte facilite la gestion de mémoire tampon. Le numéro du dernier groupe de blocs permet de déterminer plus simplement les groupes de blocs qui ont été perdus en raison d'une perte de paquets.

ANNEXE E

Mise en paquets de données vidéo

La présente annexe décrit en détail la mise en paquets RTP pour les codecs vidéo. Le Tableau E.1 présente des références aux définitions des formats de mise en paquets vidéo non définis dans la présente Recommandation. Les autres paragraphes de la présente annexe définissent d'autres formats de mise en paquets vidéo.

Tableau E.1/H.225.0 – Formats de mise en paquets vidéo à définition externe

Norme de codage	Définition de la mise en paquets
ISO/CEI 14496-2 (vidéo MPEG-4)	IETF RFC 3016, <i>RTP Payload Format for MPEG-4 Audio/Visual Streams (format de charge utile des flux MPEG-4 en protocole RTP)</i>

E.1 H.263

Un format de charge utile RTP pour flux vidéo H.263 est spécifié dans le commentaire RFC 2190 de l'IETF pour les flux binaires de données vidéo H.263 qui ne contiennent pas les nouveaux éléments adoptés dans la version 2 (de 1988) de l'UIT-T H.263 (éléments utilisant le type PLUSTYPE en faisant appel à des annexes postérieures à l'Annexe H/H.263). Un format additionnel de charge utile, prenant en charge les caractéristiques évoluées des flux binaires selon la version 2 de l'UIT-T H.263, doit être spécifié ultérieurement. Un format de mise en paquets par capacité léguée, largement utilisé dans l'industrie mais non conforme à la spécification RFC 2190 de l'IETF, ne pourra être utilisé que si l'extrémité homologue a signalé sa prise en charge de ce format, lors de l'échange des capacités.

Le paragraphe B.5 décrit la procédure à utiliser pour signaler les flux vidéo H.263.

ANNEXE F

Mise en paquets audio et en paquets multiplexés

La présente annexe décrit en détail la mise en paquets RTP pour les codecs audio. Le Tableau F.1 présente des références aux définitions des formats de mise en paquets audio non définis dans la présente Recommandation. Le Tableau F.2 présente des références aux définitions des formats de mise en paquets multiplexés. Les autres paragraphes de la présente annexe définissent d'autres formats de mise en paquets audio.

Tableau F.1/H.225.0 – Formats de mise en paquets audio à définition externe

Norme de codage	Définition de la mise en paquets
ISO/CEI 14496-3 (audio MPEG-4)	IETF RFC 3016, <i>RTP Payload Format for MPEG-4 Audio/Visual Streams (format de charge utile des flux MPEG-4 en protocole RTP)</i>

Tableau F.2/H.225.0 – Formats de mise en paquets de flux multiplexés à définition externe

Norme de codage	Définition de la mise en paquets
Flux multiplexés H.222 (flux de transport MPEG-2)	IETF RFC 2250, <i>RTP Payload Format for MPEG1/MPEG2 Video</i> (Format de charge utile pour flux vidéo MPEG-1/MPEG-2 en protocole RTP)

F.1 G.723.1

La présente Recommandation spécifie une représentation codée qui peut être utilisée pour compresser la composante de signal vocal de services multimédias à un très faible débit. La taille d'une trame G.723.1 peut être de 24 octets (6,3 kbit/s), 20 octets (5,3 kbit/s) ou 4 octets. Les trames à 4 octets sont appelées trames SID descripteur d'insertion de silence (SID, *silence insertion descriptor*) et servent à spécifier les paramètres de bruit de confort. Aucune restriction n'est imposée quant à la manière dont les trames à 4, 20 et 24 octets sont mélangées. Les deux bits de plus faible poids du premier octet de la trame déterminent la taille de la trame et le type de codec (se reporter au Tableau 5/G.723.1 et au Tableau 6/G.723.1 pour de plus amples informations sur l'ordre des bits). Il

est possible de commuter entre les deux débits à n'importe quelle frontière de trame de 30 ms. Les deux débits (5,3 kbit/s et 6,4 kbit/s) sont obligatoires pour le codeur comme pour le décodeur. Le codeur en question a été optimisé pour représenter les signaux vocaux avec une qualité de type circuit quasi longue distance aux débits susmentionnés tout en ayant une complexité limitée.

Tous les bits du flux binaire codé sont toujours transmis du bit de plus faible poids au bit de plus fort poids. Noter qu'il s'agit ici de l'ordre des bits présentés à la couche de transport et non de l'ordre des bits sur le réseau filaire.

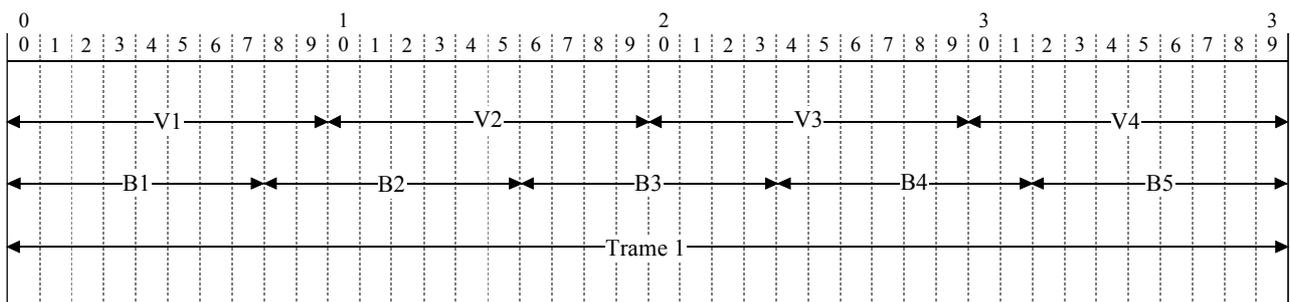
La mise en paquets G.723.1 est conforme à l'Annexe B sauf pour l'intervalle de mise en paquets (30 ms et non 20 ms, qui est la valeur par défaut):

- 1) le premier paquet de signal de parole (premier paquet après une période de silence) est repéré par la valeur du bit marqueur dans l'en-tête de données RTP;
- 2) la fréquence d'échantillonnage (fréquence d'horloge RTP) est de 8 000 Hz;
- 3) l'intervalle de mise en paquets doit avoir une durée de 30 ms (une trame) et non de 20 ms, qui est la durée par défaut;
- 4) les codecs doivent être en mesure de coder et de décoder plusieurs trames consécutives d'un même paquet;
- 5) un récepteur doit accepter les paquets représentant de 0 à 180 ms de données audio et non de 0 à 200 ms, qui est l'intervalle par défaut.

F.2 G.728

- 1) *Mise en paquets de trames:*

une trame G.728 (4 vecteurs: V1-V4, 10 bits chacun, V1 est le plus ancien – Le premier à être lu) est organisée en 5 octets (B1-B5). Compte tenu de la figure ci-dessous, le principe applicable à l'ordre des bits est le suivant: "maintien de l'ordre de leur poids". Les bits des vecteurs plus anciens ont un plus fort poids que les bits des vecteurs plus récents. Le bit de plus fort poids (MSB, *most significant bit*) de la trame devient le bit MSB de B1 et le bit de plus faible poids (LSB, *least significant bit*) de la trame devient le bit LSB de B5. Pour plus de clarté: les bits de plus fort poids de la trame de vecteurs deviennent les bits de plus fort poids de B1-B5 (les bits de plus fort poids de l'octet B de plus petit numéro).



T1529850-98

Par exemple:

B1 contient les 8 bits de plus fort poids de V1, le bit MSB de V1 devenant le bit MSB de B1.

B2 contient les 2 bits de plus faible poids de V2 – Le bit de plus fort poids parmi ces 2 bits devenant le bit MSB de V2 – et les 6 bits de plus fort poids de V2, dont le bit de plus fort poids est aussi le bit MSB de l'octet B2.

B1 doit être mis en premier dans le paquet (octet de plus fort poids du protocole RTP) et B5 en dernier.

2) *Mise en paquets de plusieurs trames:*

en cas d'envoi d'une seule trame par paquet RTP, le préfixe peut être long. Par conséquent, l'envoi d'un paquet contenant plusieurs trames est autorisé de la manière suivante:

un paquet RTP G.728 devra contenir un nombre entier de trames.

Les anciennes trames (à lire en premier) devront être placées en premier dans le paquet RTP.

L'horodate tiendrait compte du temps de saisie du premier échantillon, dans le premier vecteur (V1) de la première trame (les informations les plus anciennes du paquet).

3) Le bit de marqueur devra conserver la même signification que celle qui lui est donnée dans la présente Recommandation.

F.3 G.729

La présente Recommandation spécifie une représentation codée qui peut être utilisée pour compresser la composante de signal de parole de services multimédias à un débit de 8 kbit/s. Le codeur en question a été optimisé pour représenter les signaux vocaux avec une qualité de type circuit longue distance ou circuit filaire à 8 kbit/s. Il est intrinsèquement robuste contre les erreurs binaires aléatoires et aussi contre les suppressions aléatoires de salves de trames. Il représente les signaux vocaux avec une qualité élevée en cas de fonctionnement dans un environnement avec bruit. Une version de l'algorithme G.729 à complexité réduite est spécifiée dans l'Annexe A/G.729. Une version à virgule flottante de ces deux algorithmes est spécifiée dans l'Annexe C/G.729. Les algorithmes de codage de la parole donnés dans le corps de l'UIT-T G.729, dans l'Annexe A/G.729 et dans l'Annexe C/G.729 sont entièrement compatibles, de sorte qu'il n'est pas nécessaire de continuer à les distinguer.

Un algorithme de détection d'activité vocale (VAD, *voice activity detector*) et de génération de bruit de confort (CNG, *comfort noise generator*) donné dans l'Annexe B/G.729 est recommandé. Cet algorithme est appliqué à l'Annexe F/G.729 (6,4 kbit/s avec VAD/CNG), à l'Annexe G/G.729 (11,8 kbit/s avec VAD/CNG), à l'Annexe B/G.729 (G.729 et Annexe A/G.729 avec VAD/CNG) et à l'Annexe I/G.729. Une trame G.729 ou Annexe A/G.729 contient 10 octets; une trame Annexe D/G.729 contient 8 octets; une trame Annexe E/G.729 contient 15 octets; et la trame de bruit de confort selon Annexes B/G.729, F/G.729 et G/G.729 occupe 2 octets, comme indiqué sur la Figure F.1.

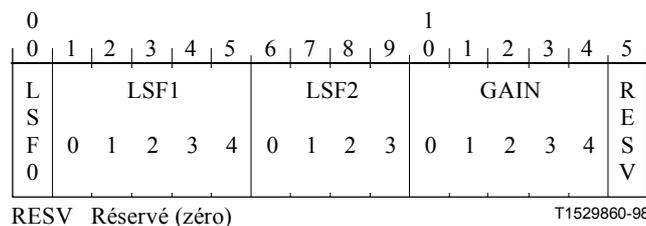


Figure F.1/H.225.0 – Format de mise en paquets de générateur CNG selon les Annexes B/G.729, F/G.729 et G/G.729

Les paramètres transmis d'une trame de 10 ms G.729, Annexe A/G.729 ou Annexe C/G.729, constituée de 80 bits, sont définis dans le Tableau 8/G.729. Le mappage de ces paramètres est donné dans la Figure F.2. Les bits sont numérotés dans l'ordre Internet, c'est-à-dire que le bit de plus fort poids est le bit 0.

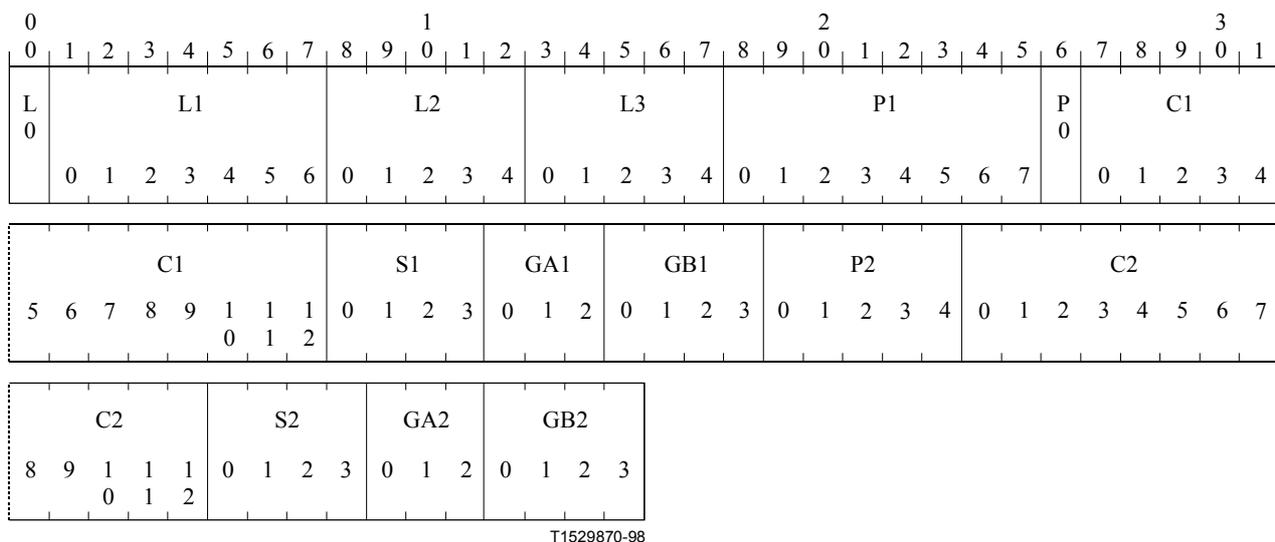


Figure F.2/H.225.0 – Format de mise en paquets selon G.729, Annexe A/G.729 et Annexe C/G.729

L'Annexe D/G.729 définit une extension de débit G.729 à 6,4 kbit/s pour réduire momentanément la capacité des voies, par exemple, afin de gérer des conditions de surcharge. L'Annexe E/G.729 définit une extension de débit G.729 à 11,8 kbit/s afin d'obtenir une meilleure performance avec une large gamme de signaux d'entrée, comme la parole avec bruit de fond et musique. Par ailleurs, l'Annexe E/G.729 comporte deux modes opératoires: adaptatif différé et adaptatif anticipé, qui sont signalés par les deux premiers éléments binaires de l'en-tête de paquet.

Les bits d'une trame G.729-6.4 sont formatés comme indiqué sur la Figure F.3 (voir Tableau D.1/G.729). Les bits sont numérotés dans l'ordre Internet, c'est-à-dire que le bit de plus fort poids est le bit 0. Un total de 64 bits est utilisé.

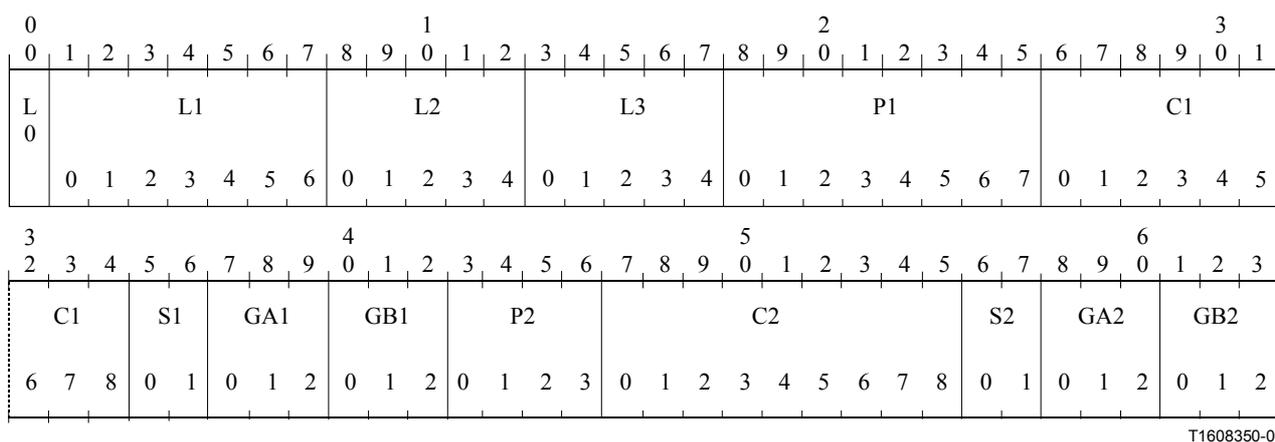
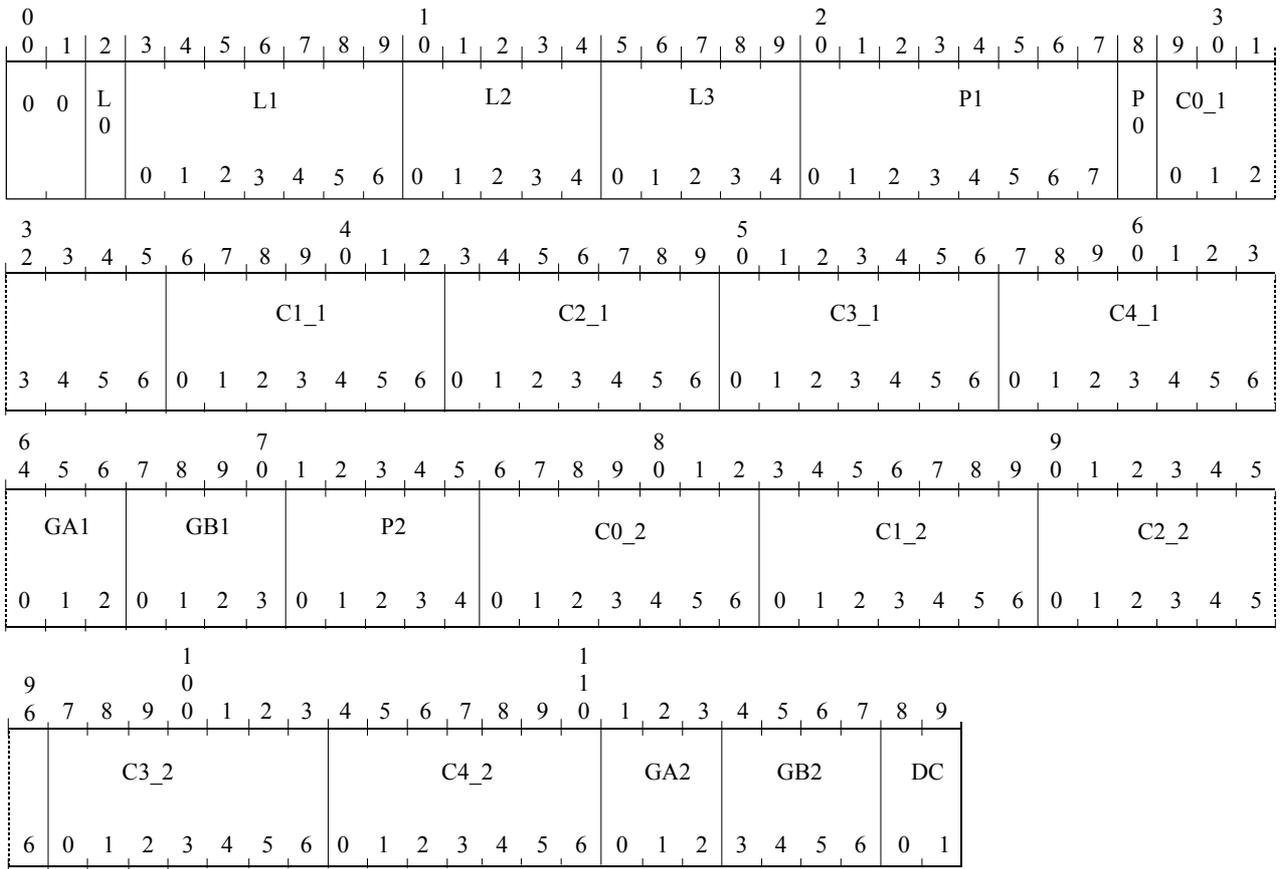


Figure F.3/H.225.0 – Format de mise en paquets G.729-6.4

Le débit net est de 11,8 kbit/s pour l'algorithme de l'Annexe E/G.729 et un total de 118 bits est utilisé. Les bits d'une trame G.729-12 sont formatés comme indiqué dans les Figures F.4 et F.5 (voir Tableau E.1/G.729). Les Figures F.4 et F.5 décrivent les champs pour, respectivement, le mode adaptatif anticipé et le mode adaptatif différé de l'algorithme selon l'Annexe E/G.729. Les deux bits de plus faible poids sont inclus comme éléments binaires indifférents et servent à compléter un nombre entier d'octets pour la trame.



T1608360-00

Figure F.4/H.225.0 – Format de mise en paquets G.729-12 pour le mode adaptatif anticipé

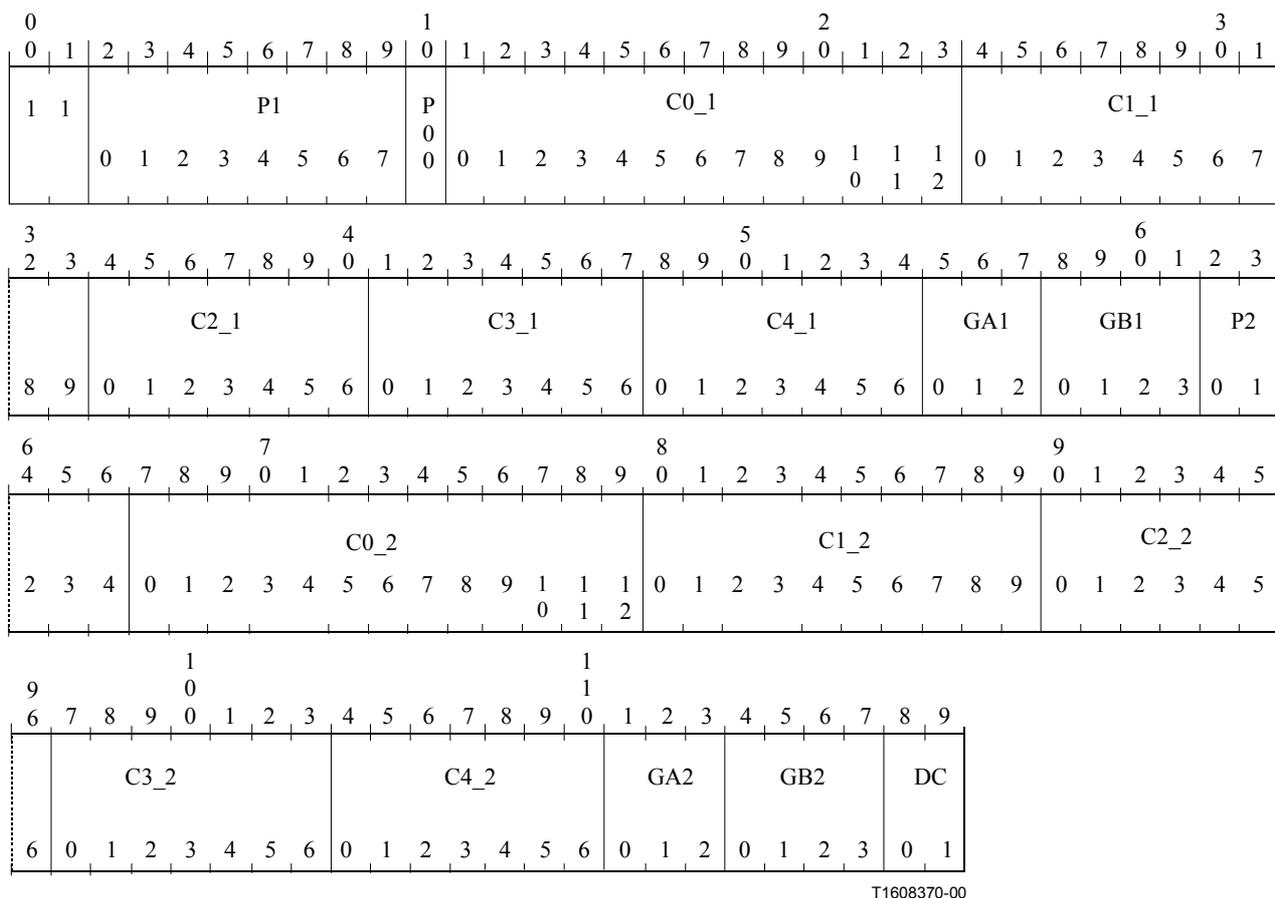


Figure F.5/H.225.0 – Format de mise en paquets G.729-12 pour le mode adaptatif différé

Un paquet RTP peut être constitué de zéro, une ou plusieurs trames G.729 ou Annexe A, C, D ou E/G.729, suivies de zéro ou une charge utile Annexe B/G.729. La présence d'une trame de bruit de confort peut être déduite de la longueur de la charge utile RTP.

- 1) Le premier paquet de signal de parole (premier paquet après une période de silence) est repéré par la valeur du bit marqueur dans l'en-tête RTP.
- 2) La fréquence d'échantillonnage (fréquence d'horloge RTP) est de 8000 Hz.
- 3) L'intervalle de mise en paquets par défaut doit normalement avoir une durée de 20 ms. Bien qu'une telle durée soit une valeur fortement recommandée, il est parfois souhaitable, dans certaines situations, d'envoyer des paquets à intervalle de 10 ms. Soit par exemple une transition d'élément voisé à élément non voisé au cours des premières 10 ms du paquet. Si un intervalle de 20 ms était imposé à la mise en paquets, l'émetteur devrait attendre que la parole soit de nouveau activée.
- 4) Les codecs doivent être en mesure de coder et de décoder de une à dix trames consécutives d'un même paquet.
- 5) Un récepteur doit accepter les paquets représentant de 0 à 200 ms de données audio.

F.4 Suppression de silence

D'après l'UIT-T H.225, les codeurs peuvent envoyer des trames de silence avant l'arrêt de transmission pendant les périodes de silence. Etant donné que tous les codeurs audio n'ont pas de signalisation dans la bande pour le silence, il convient de définir un mécanisme général au niveau RTP. Par exemple, un paquet RTP vide pourrait être envoyé. Cela appelle un complément d'étude.

F.5 Codecs GSM

Il existe trois types de codecs GSM vocaux: les codecs GSM plein débit (FR, *full rate*) [F-1], les codecs GSM demi-débit (HR, *half rate*) [F-2] et les codecs GSM plein débit amélioré (EFR, *enhanced full rate*) [F-3]. Chaque codec produit trois types de trames de trafic vocal différents, à savoir:

- trames vocales – Contiennent les données vocales effectives.
- trames inactives – Indiquent l'absence d'activité vocale; tous les bits de données sont mis à 1.
- trames descriptives de silence (SID, *silence descriptor*) – Indiquent le début d'une période de silence; les données décrivent le bruit de fond. Les trames SID sont marquées dans la bande avec un schéma de bits fixe.

F.5.1 Groupage des trames par paquets

Avec les trois codecs GSM, les bits de trames de trafic sont groupés par paquets dans le bit de plus fort poids (MSB) de la trame RTP. Un paquet RTP peut contenir une ou plusieurs trames de trafic vocal GSM. Toutes les extrémités doivent être en mesure de recevoir et d'identifier une trame inactive. Une trame vocale GSM inactive est remplie de uns binaires.

Si une extrémité met le paramètre *comfortNoise* à Vrai, il doit envoyer des trames SID telles que définies dans les spécifications du bruit de confort et de transmission discontinue (DTX, *discontinuous transmission*) d'un codec GSM donné. Pendant une période de silence, une nouvelle trame SID, avec (éventuellement) une information de bruit actualisée, est envoyée périodiquement, c'est-à-dire toutes les 24^e trames. Après une période de silence, le bit marqueur doit être mis à 1 dans l'en-tête RTP.

Codec plein débit

Le codec GSM plein débit envoie une trame de 260 bits (32,5 octets) toutes les 20 ms. Cette information doit être compactée dans la trame RTP avec un préfixe de quatre bits (0xD ou 1101 binaire), appelé signature. Par conséquent, la charge utile du codec GSM FR (*full rate*) doit être de 33 octets. La trame SID (*silence descriptor*) est marquée dans la bande par un mot de code SID enregistré dans les paramètres du codec décrits dans la référence [F-4] ci-dessous. La taille de la charge utile d'une trame SID est de 33 octets. La signature d'une trame SID plein débit doit être identique à celle d'une trame vocale plein débit (0xD). Les signaux vocaux plein débit codés RTP doivent avoir un débit binaire de 13 200 bits/s, sans compter le bit supplémentaire de groupage par paquets.

Codec mi-débit

Le codec GSM mi-débit envoie une trame de 112 bits (14 octets) toutes les 20 ms. Cette information doit être compactée dans un en-tête RTP sans préfixes/signatures. La trame SID est marquée dans la bande par un mot de code SID dans les paramètres du codec décrits dans la référence [F-4] ci-dessous. La taille de la charge utile d'une trame SID est de 14 octets. Les signaux vocaux codés RTP doivent avoir un débit binaire de 5600 bits/s, sans compter le bit supplémentaire de groupage par paquets.

Codec plein débit amélioré

Le codec GSM plein débit amélioré (EFR, *enhanced full rate*) envoie une trame de 244 bits (30,5 octets) toutes les 20 ms. Cette information doit être compactée dans un en-tête RTP avec un préfixe de quatre bits (0xC ou 1100 binaire), appelé "signature". Par conséquent, la charge utile du codec GSM EFR doit être de 31 octets. La trame SID est marquée dans la bande par un mot de code SID enregistré dans les paramètres du codec décrits dans la référence [F-4] ci-dessous. La taille de la charge utile d'une trame SID est de 31 octets. Les signaux vocaux plein débit amélioré codés RTP

doivent avoir un débit linéaire de 12 400 bits/s, sans compter le bit supplémentaire de groupage par paquets.

F.5.2 Références informatives

- [F-1] GSM 06.10 (ETS 300 961), *Digital cellular telecommunications system; Full rate speech; Transcoding.*
- [F-2] GSM 06.60 (ETS 300 726), *Digital cellular telecommunications system; Enhanced Full Rate (EFR) speech transcoding.*
- [F-3] GSM 06.20 (ETS 300 969), *Digital cellular telecommunications system; Half rate speech; Half rate speech transcoding.*
- [F-4] ETSI, TIPHON 03 001 (TS 101 318), *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Using GSM speech codecs within ITU-T Recommendation H.323.*
- [F-5] GSM 06.31 (ETS 300 963), *Digital cellular telecommunications system; Full rate speech; Comfort noise aspect for full rate speech traffic channels.*
- [F-6] GSM 06.81 (ETS 300 729), *Digital cellular telecommunications system; Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels.*
- [F-7] GSM 06.41 (ETS 300 972), *Digital cellular telecommunications system; Half rate speech; Discontinuous Transmission (DTX) for half rate speech traffic channels.*
- [F-8] GSM 06.12 (ETS 300 963), *Full rate speech; Comfort noise aspect for full rate speech traffic channels.*
- [F-9] GSM 06.62 (ETS 300 728), *Digital cellular telecommunications system; Comfort noise aspects for Enhanced Full Rate (EFR) speech traffic channels.*
- [F-10] GSM 06.22 (ETS 300 971), *Digital cellular telecommunications system; Half rate speech; Comfort noise aspect for the half Rate speech traffic channels.*
- [F-11] GSM 08.60 (ETS 300 737), *Digital cellular telecommunications system; (Phase 2+) (GSM); In-band control of remote transcoders and rate adaptors for Enhanced Full Rate (EFR) and full rate traffic channels.*

F.6 G.722.1

L'algorithme de codage de la parole qui est défini dans l'UIT-T G.722.1 code les signaux audio large bande (de 50 Hz à 7 kHz) en un flux de 24 kbit/s ou de 32 kbit/s, au moyen de trames de 20 ms à une fréquence d'échantillonnage de 16 kHz. Le débit peut être modifié à toute limite de trame de 20 ms, bien que la notification de ce changement ne soit pas fournie dans la bande avec le flux binaire. Lors d'une exploitation à 24 kbit/s, 480 bits (60 octets) sont produits à chaque trame; lors d'une exploitation à 32 kbit/s, 640 bits (80 octets) sont produits à chaque trame. Ces deux débits permettent un alignement des octets sans que des bits de bourrage soient nécessaires.

Le nombre de bits contenus dans une trame est fixe. A l'intérieur de cette trame, l'UIT-T G.722.1 utilise un codage de longueur variable (comme le codage de Huffman) afin de représenter la plupart des paramètres codés. A l'exception du paramètre Bits de commande de catégorisation, tous les autres paramètres du flux binaire sont représentés par des codes de longueur variable et donc par un nombre variable de bits. La Figure F.6 illustre ce point et l'ordre des champs paramétriques transmis. Tous les codes de longueur variable et les bits de commande de catégorisation sont émis en séquence du bit placé le plus à gauche (de plus fort poids – MSB) jusqu'au bit placé le plus à droite (de plus faible poids, LSB). L'utilisation du codage de Huffman signifie qu'il n'est pas possible d'identifier les divers paramètres/champ de codeur contenus dans le flux binaire sans décodage complet, au préalable, de la trame entière.

La Figure F.7 décrit la façon dont le flux binaire G.722.1 s'applique sur une charge utile RTP verrouillée en octets. Le flux binaire codeur est subdivisé en séquences d'octets (60 ou 80 octets selon le débit), chaque octet étant à son tour appliqué sur un octet RTP.

Un paquet RTP ne doit contenir que des trames G.722.1 ayant le même débit. Le marqueur temporel RTP doit être exprimé en unités de 16.10^{-3} s.

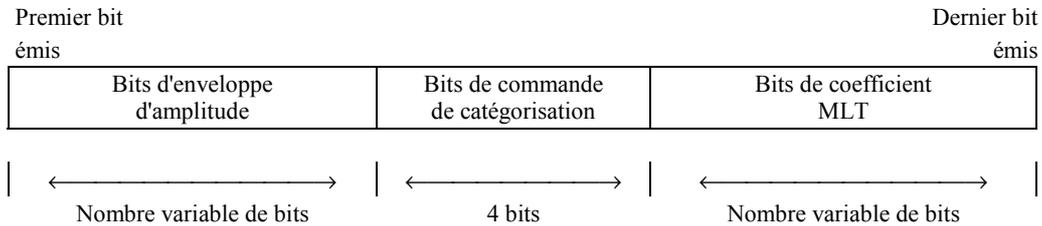


Figure F.6/H.225.0 – Ordre de transmission des principaux champs du flux binaire G.722.1

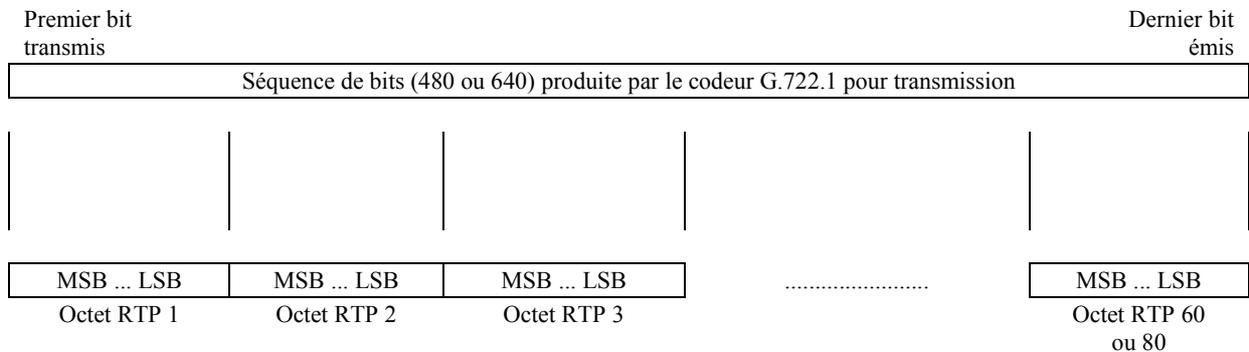


Figure F.7/H.225.0 – Mappage sur le protocole RTP du flux codé G.722.1

F.7 Vocodeur TIA/EIA-136 à codage ACELP

Ce vocodeur est optimisé pour les systèmes TIA/EIA-136 des réseaux cellulaires numériques et de communications personnelles (PCS) à accès AMRT. Il comporte les capacités de détection d'activité vocale (VAD, *voice activity detection*), de substitution de trame perdue et de production de bruit de confort (CNG, *comfort noise generation*). La fréquence d'échantillonnage est de 8000 Hz et la longueur de trame vocale comprimée est de 20 ms. Le vocodeur produit un vecteur vocal de 148 bits (de s0 à s147) pour chaque trame vocale de 20 ms. s0 est le bit de plus fort poids (MSB). Voir la section 4 de la référence [F.7-1] pour plus de détails.

F.7.1 Format de trame TIA/EIA-136 à codage ACELP

Un fanion indiquant un signal vocal, SP, doit être produit par le vocodeur et mis à "1" pour indiquer une trame vocale ou à "0" pour indiquer une trame de silence (bruit de confort). Ce fanion SP doit être inséré à la position binaire 148. La position binaire 149 est réservée au fanion indicateur de trame erronée ou de bruit de confort (BFI_CN, *bad frame or comfort noise*) et la position binaire 150 est le fanion d'actualisation du bruit de confort (CNU, *comfort noise update*). La position binaire 151 doit toujours être mise à 0.

Les combinaisons logiques de ces trois fanions sont décrites ci-dessous.

La trame d'émission de 152 bits (19 octets) est décrite par la Figure F.8 ci-dessous. Les octets sont formés à partir du bit LSB en allant vers le bit MSB. Le bit LSB est émis en premier.

Bit 0 (MSB)	1 ... 146	147	148	149	150	bit 151 (LSB)
s0	s1 ... s146	S147	SP	BFI_CN	CNU	Toujours 0
Vecteur vocal/Bruit de confort			Fanion	Fanion	Fanion	Bit de bourrage

Figure F.8/H.225.0 – Trame vocale de vocodeur ACELP

F.7.2 Trames de suppression de silence des vocodeurs TIA/EIA-136 à codage ACELP

En mode silencieux, le vocodeur produit une instance de la trame de bruit ambiant. Cette trame est utilisée par le vocodeur à l'extrémité réceptrice afin de régénérer le bruit ambiant de l'extrémité émettrice. Le vecteur contenant les paramètres de bruit de confort (CN, *comfort noise*) se compose de 38 bits seulement, auxquels sont ajoutés les 3 fanions et 7 bits de bourrage (formant une séquence de zéros) afin de constituer une trame de 6 octets.

La trame CN de 48 bits (6 octets) est décrite par la Figure F.9 ci-dessous. Les octets sont formés à partir du bit LSB en allant vers le bit MSB. Le bit LSB est émis en premier.

Bit 0 (MSB)	1 ... 37	38	39	40	41	41-47 (LSB)
Cn0	cn1 ... cn37	S147	SP	BFI_CN	CNU	Toujours 0
Vecteur vocal/Bruit de confort			Fanion	Fanion	Fanion	Bit de bourrage

Légende:

SP Indicateur vocal

BFI_CN Indicateur de trame erronée/Indicateur de bruit de confort

CNU Actualisation du bruit de confort

Les valeurs logiques de ces fanions et leurs significations sont définies ci-dessous.

SP: 1 = trame vocale; 0 = trame non vocale (de bruit de confort)

BFI_CN:

Si SP = 1

Et si BFI_CN = 1

Alors il s'agit d'une trame vocale erronée

Sinon (BFI_CN = 0), il s'agit d'une trame vocale correcte

Si SP = 0

Et si BFI_CN = 1

Alors il s'agit d'une trame vocale erronée

Sinon (BFI_CN = 0), il s'agit d'une trame vocale correcte

CN:

Si SP = 0

Et si BFI_CN = 0

Et si CN = 1

Alors il s'agit d'une trame d'actualisation de bruit de confort

Sinon il s'agit d'une trame CN non valide

NOTE – Un vocodeur mobile sans fil doit mettre à 0 le fanion BFI_CN. La station de base réceptrice peut mettre ce fanion à 1 si elle ne possède pas la capacité de corriger les erreurs dues au canal radioélectrique.

Figure F.9/H.225.0 – Trame de suppression de silence dans un vocodeur ACELP

F.7.3 Mise en paquets dans les vocodeurs TIA/EIA-136 à codage ACELP

La mise en paquets dans les vocodeurs à codage IS-ACELP doit être conforme à l'Annexe B.

- 1) La durée de mise en paquets doit être un multiple entier de 20 ms.
- 2) Chaque paquet peut comporter une ou plusieurs trames.
- 3) Il convient que les codecs possèdent la capacité de coder et décoder plusieurs trames consécutives à l'intérieur d'un même paquet.
- 4) Tous les bits du flux binaire codé sont toujours émis à partir du bit de plus faible poids vers le bit de plus fort poids.

F.7.4 Références normatives des vocodeurs TIA/EIA-136 à codage ACELP

[F.7-1] TIA/EIA-136, part 410, *TDMA Cellular/PCS – Radio Interface, Enhanced Full Rate Voice Codec* (ACELP) (Réseaux cellulaires/PCS à AMRT – Interface radio – Vocodeurs à plein débit amélioré) (ACELP). Anciennement IS-641.

F.8 Vocodeur TIA/EIA-136 à codage US1

Ce vocodeur est optimisé pour les systèmes TIA/EIA-136 des réseaux cellulaires numériques et de communications personnelles (PCS). La référence [F.8-1] contient une description détaillée de ce vocodeur.

F.8.1 Format de trame TIA/EIA-136 à codage US1

La fréquence d'échantillonnage est de 8000 Hz et la longueur de trame vocale comprimée est de 20 ms. Le vocodeur produit 244 éléments binaires ordonnés par trame vocale. Trois fanions (BFI, SID et TAF) sont ajoutés au vecteur vocal. Un seul bit de bourrage (à la position binaire 247) est ajouté pour former un nombre entier d'octets (31). Le dernier bit est désigné comme étant le bit de plus faible poids (LSB). Ce vocodeur prend également en charge le mode de silence à transmission discontinue (DTX).

La structure des trames vocales d'émission est décrite dans la Figure F.10.

MSB – bit 0	1 ... 243	244	245	246	247 (LSB)
s0	s1 ... s243	BFI	SID	TAF	Toujours 0
Vecteur vocal		Fanion	Fanion	Fanion	Bit de bourrage

Figure F.10/H.225.0 – Trame vocale de vocodeur US1

F.8.2 Trames de suppression de silence des vocodeurs TIA/EIA-136 à codage US1 (TX-DTX)

En mode silencieux, le vocodeur émet des trames spéciales appelées SID (*silence descriptor*) à un rythme spécifié à la section 1.3 de la référence [F.8-1].

Une trame SID contient le même nombre de bits que les trames vocales normales mais l'affectation des bits est différente. Voir la référence [F.8-1] pour les détails. La trame SID contient des paramètres de bruit de confort (CN, *comfort noise*) et un mot de code SID de 95 bits, constitué d'une séquence de zéros. Les autres bits inutilisés dans la charge utile du vecteur de 244 bits sont également mis à "0". (Voir Figure F.11.)

MSB – bit 0	1 ... 243	244	245	246	247 (LSB)
cn0	cn1 ... cn243	BFI	SID	TAF	Toujours 0
Vecteur de bruit de confort		Fanion	Fanion	Fanion	Bit de bourrage

Figure F.11/H.225.0 – Trame d'émission de bruit de confort entre une station de base et une ligne terrestre (US1)

La logique des fanions BFI, SID et TAF est similaire à celle des fanions équivalents du vocodeur TIA/EIA-136 à codage ACELP, décrit au F.7.

F.8.3 Mise en paquets dans les vocodeurs TIA/EIA-136 à codage US1

La mise en paquets doit être conforme à l'Annexe B.

- 1) La durée de mise en paquets doit être un multiple entier de 20 ms.
- 2) Chaque paquet peut comporter zéro, une ou plusieurs trames.
- 3) Il convient que les codecs possèdent la capacité de coder et décoder plusieurs trames consécutives à l'intérieur d'un même paquet.
- 4) Tous les bits du flux binaire codé sont toujours émis à partir du bit de plus faible poids vers le bit de plus fort poids.

F.8.4 Références normatives des vocodeurs TIA/EIA-136 à codage US1

[F.8-1] TIA/EIA-136, part 430, *TDMA Cellular/PCS – Radio Interface, US1 Full Rate Voice Codec* (Réseaux cellulaires/PCS à AMRT – Interface radio – Vocodeurs à plein débit US1).

F.9 Codec à débit variable amélioré (EVRC) selon la norme IS-127

F.9.1 Description du codec EVRC IS-127

F.9.1.1 Considérations générales

Le codec à débit variable amélioré (EVRC, *enhanced variable rate codec*) selon la norme IS-127 de l'association TIA/EIA est optimisé pour les systèmes cellulaires numériques et PCS à CDMA selon la norme IS-125 de l'association TIA/EIA. La fréquence d'échantillonnage est de 8 000 Hz et la longueur d'une trame vocale est de 20 ms (soit 160 échantillons/trame). Le codec EVRC code les conversations actives à plein débit ou à mi-débit et code le bruit de fond (sans signaux vocaux) au débit 1/8. Il fournit des signaux vocaux de qualité interurbaine à un très bas débit moyen. L'on peut trouver une description détaillée du codec EVRC dans la norme provisoire IS-127 publiée par l'association TIA/EIA (voir la référence [F.9-1]).

F.9.1.2 Taux de compression

Le codeur EVRC comprime son signal d'entrée selon trois débits: le débit plein (taux 1/1), le mi-débit (taux 1/2) et le débit huitième (taux 1/8). Le plein débit et le mi-débit sont principalement utilisés pour coder des signaux de conversation active, alors que le débit huitième est utilisé pour coder le bruit de fond (mode silencieux). Toutes les trames ont une longueur de 20 ms, quel que soit le taux de compression.

F.9.1.3 Paquets effacés

Certaines trames vocales sont effacées afin de permettre une signalisation dans la bande ou un transport de trafic auxiliaire (voir section 1.4.1 de la référence [F.9-1]). Le paquet vocal ainsi produit reste simplement inutilisé et le décodeur le traite comme un paquet effacé. (Voir détails dans la référence [F.9-1].)

F.9.1.4 Demi-débit

Le codage en demi-débit est utilisé à la place du codage normal à plein débit lorsqu'un message de signalisation doit être ajouté à la voie de trafic.

F.9.1.5 Données néant dans la voie de trafic au débit 1/8

Un paquet au débit un huitième dont tous les bits sont mis à "1" est considéré comme contenant des données néant dans la voie de trafic. Ces paquets sont déclarés "effacés" et sont traités comme décrit à la section 5 de la référence [F.9-1].

Les bits d'information de débit et de codage de voie sont ajoutés aux bits de sortie du vocodeur pour transport par voie hertzienne, conformément à la norme IS-95 de l'association TIA/EIA.

Le Tableau F.3 ci-dessous montre les types de paquet, le nombre de bits par paquet, les débits bruts du vocodeur et les débits résultants (bits du vocodeur plus bits additionnels).

Tableau F.3/H.225.0 – Débits de paquets et de bits EVRC

Type de paquet (3 bits)	Débit	Bits/paquet	Débit brut du vocodeur	Débit résultant
1	Plein	171	8,55 kbit/s	9,6 kbit/s
2	Demi	80	4,0 kbit/s	4,8 kbit/s
3 (Note)	Quart (compatibilité avec option de service 1)	40		
4	Huitième	16	0,8 kbit/s	1,2 kbit/s
5	Effacé	0	–	–
6	Plein débit avec erreurs	171	–	–
7	Trame erronée (effacée)	0	–	–

NOTE – Les paquets de type 3 ne peuvent être produits que par les codeurs IS-96 plus anciens. Le décodeur IS-127 traitera ces paquets comme des paquets effacés.

F.9.2 Mise en paquets dans les vocodeurs EVRC à codage IS-127

F.9.2.1 Exigences générales

La mise en paquets d'émission doit être conforme à l'Annexe B.

- 1) La durée de mise en paquets doit être un multiple entier de 20 ms.
- 2) Chaque paquet d'émission peut comporter zéro, une ou plusieurs trames.
- 3) Il convient que les codecs possèdent la capacité de coder et décoder plusieurs trames consécutives à l'intérieur d'un même paquet.
- 4) Tous les bits du flux binaire codé sont toujours émis à partir du bit de plus faible poids vers le bit de plus fort poids.

F.9.2.2 Formats de trame

F.9.2.2.1 Trame à plein débit – F1

La trame d'émission à plein débit de 176 bits (22 octets) d'un vocodeur EVRC (F1) est décrite par la Figure F.12 ci-dessous. Les octets sont formés à partir du bit LSB en allant vers le bit MSB. Le bit LSB (position 175) est émis en premier.

Bit 0 (MSB)	Bits 1 à 170	Bits 171 à 175 (LSB)
s0	s1 ... s170	Toujours 0
Vecteur vocal		Bit de bourrage

Figure F.12/H.225.0 – Trame F1 à plein débit de vocodeur EVRC

F.9.2.2.2 Trame à mi-débit – F2

La trame d'émission à mi-débit de 80 bits (10 octets) d'un vocodeur EVRC (F2) est décrite par la Figure F.13 ci-dessous. Les octets sont formés à partir du bit LSB en allant vers le bit MSB. Le bit LSB (position 79) est émis en premier.

Bit 0 (MSB)	Bits 1 à 79 (LSB)
s0	s1 ... s79
Vecteur vocal	

Figure F.13/H.225.0 – Trame F2 à mi-débit de vocodeur EVRC

F.9.2.2.3 Trame à huitième de débit – F3

La trame d'émission à huitième de débit de 16 bits (2 octets) d'un vocodeur EVRC (F3) est décrite par la Figure F.14 ci-dessous. Les octets sont formés à partir du bit LSB en allant vers le bit MSB. Le bit LSB (position 15) est émis en premier.

Bit 0 (MSB)	Bits 1 à 15 (LSB)
s0	s1 ... s15
Vecteur vocal	

Figure F.14/H.225.0 – Trame F3 à huitième de débit de vocodeur EVRC

F.9.3 Références normatives des vocodeurs EVRC à codage IS-127

- [F.9-1] TIA/EIA IS-127 (1997), *Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems* (Codec à débit variable amélioré – Option 3 de service vocal pour systèmes numériques à large étalement du spectre).
- [F.9-2] TIA/EIA IS-95-B (1999), *Mobile Station-Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems* (Norme compatibilité station mobile-station de base pour systèmes cellulaires à large étalement du spectre).

F.10 Mise en paquets d'unités MUX-PDU à codage H.223

F.10.1 Introduction

Les unités MUX-PDU à codage H.223 sont utilisées par un protocole de multiplexage en mode paquet qui est conçu pour le transport d'un ou de plusieurs flux informationnels entre entités de couche supérieure comme des protocoles de transmission de données et de commande avec des codecs audio et vidéo, comme défini dans l'UIT-T H.223.

Chaque flux d'information est représenté par une voie logique H.245 unidirectionnelle qui est identifiée par un numéro de voie logique (LCN, *logical channel number*) sous forme d'un entier compris entre 0 et 65535. Le numéro LCN 0 désigne une voie logique attribuée à titre permanent à la voie de commande H.245. Toutes les autres voies logiques sont ouvertes et fermées dynamiquement

par l'émetteur au moyen des messages H.245 OpenLogicalChannel et CloseLogicalChannel. Tous les attributs nécessaires de la voie logique sont spécifiés dans le message OpenLogicalChannel. Pour les applications qui exigent une voie inverse, l'UIT-T H.245 définit également une procédure d'ouverture de canaux logiques bidirectionnels.

La structure générale du multiplexeur est décrite par la Figure 2/H.223. Le multiplexeur se compose de deux couches distinctes: une couche multiplex (MUX, *multiplex layer*) et une couche d'adaptation (AL, *adaptation layer*).

La prise en charge du type de charge utile H.223 est signalée au moyen des ensembles de capacités H.245 et dans le message H.245 OpenLogicalChannel au moyen de types de charge utile dynamique du protocole RTP.

F.10.2 Format de mise en paquets des unités MUX-PDU

L'unité MUX-PDU H.223 qui est spécifiée par la Figure 3/H.223 est transportée sous forme de données de charge utile à l'intérieur du protocole RTP. L'ordre de transmission des bits est spécifié au 3.2.2/H.223 et la convention de mappage des champs est spécifiée au 3.2.3/H.223.

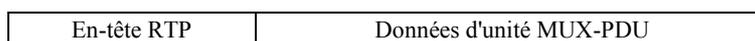
Bien qu'une unité MUX-PDU puisse occuper plus d'un seul paquet RTP, elle doit commencer par le premier octet d'une charge utile de paquet RTP.

Chaque paquet RTP contient un pointeur temporel qui est extrait de la référence d'horloge de l'expéditeur. Ce pointeur temporel doit représenter l'instant d'émission cible du premier octet de l'unité MUX-PDU H.223. La fonction principale de ce pointeur temporel est que le récepteur puisse estimer et réduire une éventuelle gigue due au réseau et puisse reproduire le flux H.223 à débit constant.

L'utilisation des champs de l'en-tête des paquets RTP doit être la suivante:

- 1) une charge utile de type dynamique est utilisée dans le protocole RTP;
- 2) le pointeur temporel RTP représente l'instant d'émission cible du premier octet de l'unité MUX-PDU contenue dans le paquet dans la voie H.223 à débit constant. Ce pointeur temporel est extrait de la fréquence d'horloge avec une valeur par défaut de 90 kHz. L'expéditeur peut modifier cette fréquence et la valeur choisie est signalée par le paramètre **BitRate** de la structure **H223Capability** contenue dans les messages H.245. Si une unité MUX-PDU occupe plus d'un seul paquet RTP, le pointeur temporel du protocole RTP doit être le même pour cette série de paquets. Il convient de calculer le pointeur temporel sur la base du nombre d'octets inclus dans les unités MUX-PDU émises;
- 3) le bit marqueur de l'en-tête RTP est mis à 1 dans le dernier paquet d'une unité MUX-PDU. Sinon, il doit être mis à 0. Il n'est donc pas nécessaire d'attendre le paquet suivant pour détecter la fin de l'unité MUX-PDU.

L'unité MUX-PDU H.223 fait suite à l'en-tête RTP, soit:



ANNEXE G

Communication entre domaines administratifs

G.1 Domaine d'application

Il est prévu que le réseau global H.323 se constituera de sous-ensembles d'équipement plus restreints correspondant à certain type d'organisation, par exemple sous la forme de domaines administratifs. Compte tenu du nombre important d'équipements H.323 susceptibles d'exister au sein de réseaux H.323, il est nécessaire de disposer d'un protocole efficace permettant d'établir des appels

entre domaines administratifs. L'exemple le plus simple est celui d'un utilisateur (point de terminaison) appartenant à un domaine administratif qui tente d'atteindre un utilisateur (point de terminaison) desservi par un autre domaine administratif. Bien qu'il convienne à un grand nombre de besoins de communication entre domaines administratifs, le protocole RAS H.225.0 n'est ni complet, ni efficace à cet effet.

La présente annexe décrit des méthodes permettant de traiter la résolution d'adresse, l'autorisation d'accès, et la communication de rapports d'utilisation entre des domaines administratifs de systèmes H.323 en vue de l'établissement d'appels entre domaines. Un domaine administratif se présente vis-à-vis d'autres domaines administratifs sous la forme d'un élément logique appelé "élément frontière". Un élément frontière peut occuper le même emplacement que toute autre entité (par exemple un portier). L'Annexe G n'impose pas qu'un domaine administratif divulgue les détails de son organisation ou de son architecture. Elle ne prescrit aucune architecture spécifique au sein d'un domaine administratif. Elle prend de plus en charge l'utilisation de tout modèle d'appel (acheminement par un portier ou directement vers le point de terminaison).

La procédure générale consiste, pour les éléments frontière, à échanger des informations portant sur les adresses que chacun d'eux est en mesure de résoudre. Il est possible de spécifier les adresses d'une manière générale ou sous des formes de plus en plus spécifiques. Des informations supplémentaires permettent aux éléments situés au sein d'un domaine administratif de déterminer quel est le domaine administratif le plus approprié comme destination de l'appel. Des éléments frontière peuvent gérer l'accès à leurs adresses exposées et exiger la production de rapports sur l'utilisation faite pendant les appels à ces adresses.

La Figure G.1 indique un certain nombre de points de référence qui représentent la signalisation entre divers éléments au sein d'un réseau H.323. Les domaines administratifs de cette figure appartiennent à un réseau global par paquets sans bords. Il convient de noter que cette figure ne fournit pas de définition explicite de l'architecture système H.323 et qu'elle a uniquement pour objet l'illustration des points de référence de signalisation.

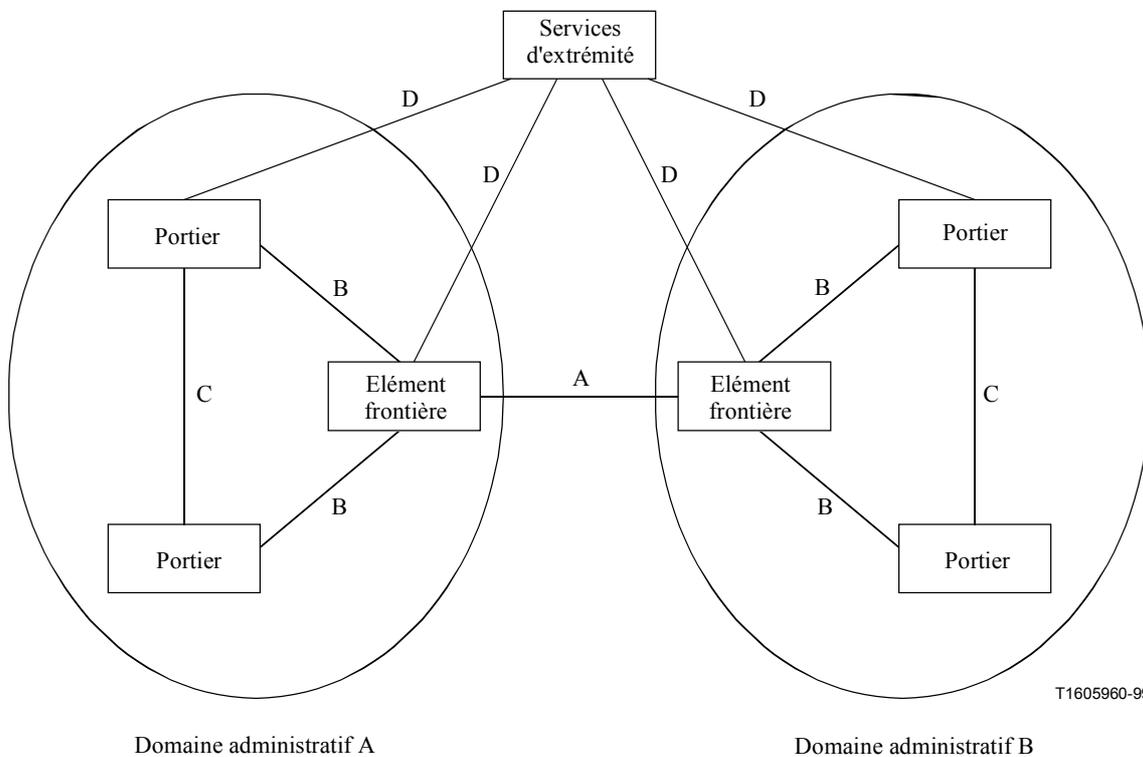


Figure G.1/H.225.0 – Points de référence système

La figure met en évidence les points de référence suivants:

A – entre éléments frontière;

B – entre élément frontière et portiers;

C – entre portiers;

D – entre éléments H.323 et services d'extrémité (en dehors du domaine d'application de la présente annexe).

La présente annexe s'intéresse principalement au point de référence A. L'utilisation du protocole décrit dans la présente annexe à des fins de communication entre portiers au sein d'un domaine administratif appelle une étude ultérieure. Le point de référence B appelle une étude ultérieure, car on estime à l'heure actuelle que l'élément frontière sera situé au même emplacement qu'un autre élément H.323.

Le paragraphe G.9 "Exemples de signalisation" fournit quelques exemples destinés à faciliter la compréhension.

G.2 Définitions

La présente annexe définit les termes suivants:

G.2.1 domaine administratif: un domaine administratif est un ensemble d'entités H.323 gérées par une même entité administrative. Il peut se constituer d'un ou de plusieurs portiers (c'est-à-dire, d'une ou plusieurs zones).

G.2.2 services d'extrémité: les services d'extrémité sont des fonctions concernant l'utilisateur, telles l'authentification ou l'autorisation, la comptabilité, la facturation, la tarification, etc.. Les services d'extrémité et le protocole permettant l'échange d'informations avec de tels services sont en dehors du domaine d'application de la présente annexe (lorsqu'ils diffèrent de ceux décrits dans la présente annexe).

G.2.3 élément frontière: l'élément frontière est un élément fonctionnel qui prend en charge l'accès public au sein d'un domaine, à des fins d'établissement d'appel ou d'autres services impliquant une communication multimédia avec d'autres éléments situés au sein du domaine administratif. L'élément frontière gère la vue externe du domaine administratif. Un élément frontière communique avec d'autres éléments frontière au moyen du protocole spécifié dans la présente annexe. Un élément frontière peut en outre, selon son implémentation, communiquer avec d'autres entités au sein de son domaine administratif. Cet élément peut se présenter en association avec d'autres éléments H.323, par exemple comme la combinaison d'un élément frontière, d'un portier et d'une passerelle. Un domaine administratif peut contenir un nombre quelconque d'éléments frontière.

G.2.4 résolveur d'adressage: service (se présentant éventuellement sous la forme d'un élément frontière) qui est en mesure de fournir une résolution pour toutes les adresses (c'est-à-dire, un type de point d'agrégation).

G.3 Abréviations

La présente annexe utilise les abréviations suivantes:

AD domaine administratif (*administrative domain*)

BE élément frontière (*border element*)

CH résolveur d'adressage (*clearing house*)

DST décalage de l'heure d'été (*daylight saving time*)

EP point de terminaison (*endpoint*)

GK	portier (<i>gatekeeper</i>)
GW	passerelle (<i>gateway</i>)
T	terminal

G.4 Références

- [G-1] UIT-T H.235 (2000), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)*.
- [G-2] UIT-T H.323 (2000), *Systèmes de communication multimédia en mode paquet*.
- [G-3] UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.
- [G-4] UIT-T X.680 (1997)/Amd.1 (1999) | ISO/CEI 8824-1:1998/Amd.1:1999, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base – Amendement 1: identificateurs d'objet relatifs*.
- [G-5] UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact*.

G.5 Modèles système

La présente annexe ne prescrit pas d'architecture système particulière entre des domaines administratifs ou au sein d'un domaine administratif. Les sous-paragraphes qui suivent fourniront quelques exemples d'architecture qui sont considérés comme des illustrations sans aucun caractère exhaustif.

On considère en général qu'un domaine administratif se constitue d'un nombre quelconque de zones et d'un nombre quelconque d'éléments frontière. Rappelons qu'un élément frontière est un élément fonctionnel qui peut coexister avec tout autre élément H.323. La Figure G.2 présente quelques exemples d'implémentation d'élément frontière en combinaison avec d'autres éléments.

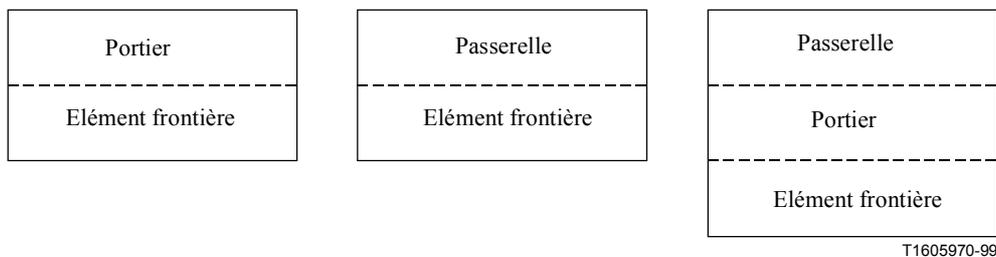


Figure G.2/H.225.0 – Exemples de localisation d'élément frontière

Les relations entre les domaines administratifs peuvent être organisées de diverses manières. Les sous-paragraphes qui suivent en fournissent des exemples.

G.5.1 Hiérarchique

La Figure G.3 présente une structure hiérarchique simple de domaines administratifs. Un élément frontière d'une telle structure consultera un élément frontière dans un domaine administratif supérieur de la hiérarchie pour résoudre une adresse.

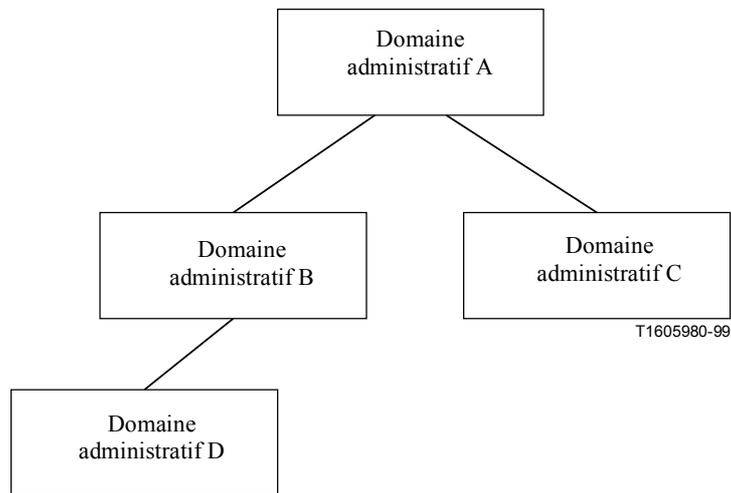


Figure G.3/H.225.0 – Structure hiérarchique simple

G.5.2 Répartition ou maillage total

Il est possible d'utiliser un modèle entièrement réparti ou maillé, comme indiqué dans la Figure G.4. Dans cet exemple, un élément frontière de chaque domaine administratif communique avec des éléments frontière appartenant aux autres domaines administratifs connus.

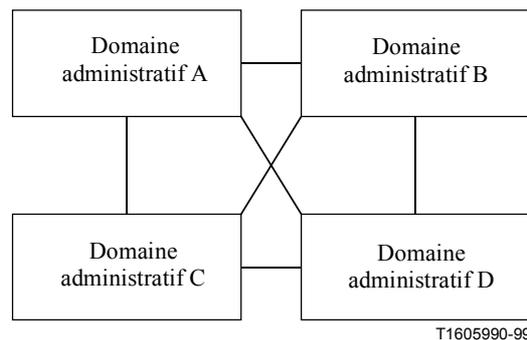


Figure G.4/H.225.0 – Exemple de structure répartie

G.5.3 Résolveur d'adressage

La Figure G.5 présente un exemple de structure de résolveur d'adressage. Dans cette structure, tout domaine administratif consulte le résolveur d'adressage pour résoudre des adresses.

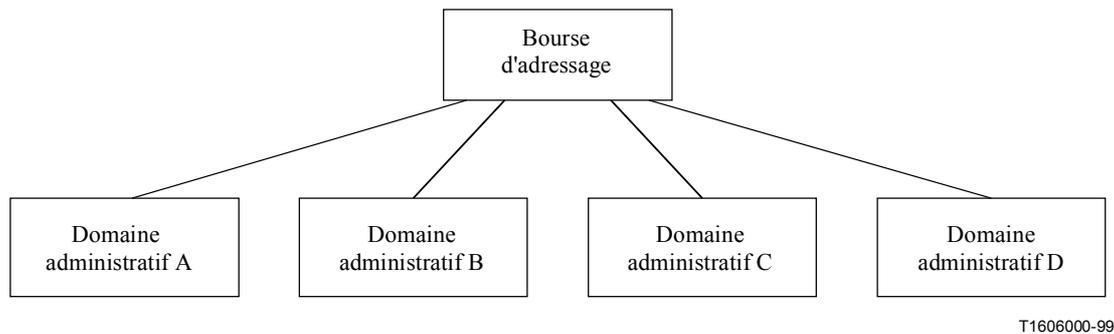


Figure G.5/H.225.0 – Exemple de structure de résolveur d'adressage

G.5.4 Point d'agrégation

La Figure G.6 présente un exemple de point d'agrégation. Le domaine administratif B de cet exemple est un point d'agrégation qui est en mesure de fournir la résolution d'adresse pour lui-même ainsi que pour les domaines administratifs C et D. Le domaine administratif B peut, par exemple, retransmettre à destination du domaine administratif C des demandes de résolution en provenance du domaine administratif A ou peut donner à ce dernier l'instruction de consulter directement le domaine administratif C en ce qui concerne certaines destinations. Si le domaine administratif B retransmet une demande du domaine administratif A à destination du domaine administratif C, il peut alors mémoriser dans son cache la réponse fournie par ce dernier.

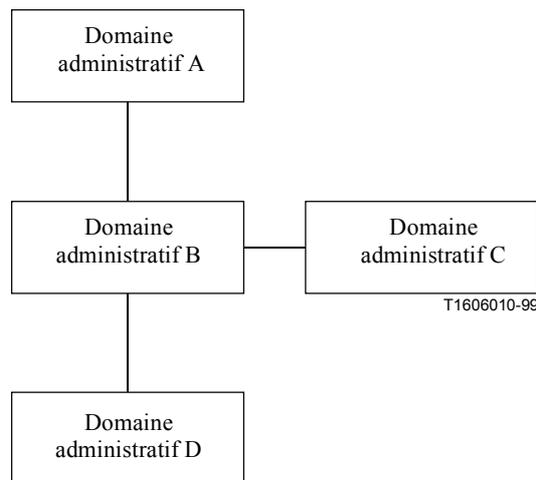


Figure G.6/H.225.0 – Exemple de point d'agrégation

G.5.5 Chevauchement de domaines administratifs

Il se peut que plusieurs domaines soient en mesure de résoudre une adresse donnée. Des domaines administratifs multiples peuvent, par exemple, contenir des passerelles capables d'établir un appel à destination d'un terminal du RTGC. Le choix du domaine administratif de destination adéquat est de la responsabilité du domaine administratif d'origine. L'algorithme utilisé pour ce choix est une affaire d'implémentation.

G.6 Conventions d'adressage

Il est important, pour la fourniture de l'interfonctionnement entre domaines, que les formats d'adressage utilisés dans des messages H.323 émis soient compris par le système récepteur. Un

élément frontière prendra en charge les deux types d'alias "identificateur de messagerie électronique" et "numéro d'abonné" (le numéro public étant utilisé avec un type de numéro public égal à "numéro international"). Il convient de noter que cette prescription implique la prise en charge de la version (1998) de l'UIT-T H.225.0 ou d'une version ultérieure. Lors de la communication avec d'autres éléments frontière, seuls les types d'alias "identificateur de messagerie électronique" et "numéro d'abonné" doivent être utilisés dans le champ "adresse de destination" d'un message LRQ ou SETUP, sauf accord préalable entre les domaines administratifs concernés. Si, par exemple, les domaines administratifs d'un groupe ont conclu un accord sur l'interprétation de numéros locaux privés, ces numéros peuvent alors être utilisés dans leurs messages mutuels.

G.7 Fonctionnement

G.7.1 Canevas et descripteurs d'adresse

Un canevas d'adresse (en abrégé un "canevas") définit un ensemble d'identificateurs d'adresse d'alias, des informations de tarification pour l'établissement d'appels à destination de ces adresses et le protocole devant être utilisé pour atteindre les adresses de cet ensemble. Un domaine administratif indique les appels qu'il est en mesure de résoudre en publiant des canevas. Les canevas sont regroupés par un identificateur appelé "descripteur". Une fois qu'un canevas a été associé à un descripteur, toute modification de ce canevas implique une modification du descripteur "groupe". Les informations de canevas peuvent permettre la mise en commun d'informations d'adressage si le schéma d'adressage est organisé de manière hiérarchique ou pouvant faire l'objet d'un acheminement. Une zone donnée peut, par exemple, traiter le numéro 1303538* qui équivaut à tous les numéros débutant par les chiffres 1303538. (Il convient de noter que le caractère "*" est significatif et que le canevas contient dans ce cas un fanion booléen indiquant si l'adresse est spécifique ou non. Les exemples qui suivent utilisent le caractère "*" pour indiquer un "joker", mais la représentation effective dans le canevas se fait au moyen du fanion booléen.)

Le canevas suivant est donné à titre d'exemple:

pour le numéro 1 555 123 4567	émettre un message de demande d'accès à destination de l'élément frontière A;
pour les chiffres 1 555 987*	émettre un message de demande d'accès à destination de l'élément frontière B;
pour le numéro 1 555 987 6543	émettre un message d'établissement à destination de la passerelle X;
pour *@example.org	émettre un message de demande d'accès à destination de l'élément frontière A;
pour les chiffres 1*	émettre un message de demande d'accès à destination de l'élément frontière B;
pour les chiffres privés 31*	émettre un message de demande d'accès à destination de l'élément frontière C;
pour les chiffres 44 171 112*	n'existe pas.

Un élément frontière obtient des canevas par l'un des mécanismes suivants:

- configuration statique;
- réception de descripteur émis par d'autres éléments frontière en réponse à des demandes générales;
- réception de réponses à des demandes spécifiques.

G.7.1.1 Configuration statique

Un élément frontière gèrera des canevas pour toutes les zones dont il est responsable. Ces canevas peuvent être fournis de manière explicite dans l'élément frontière ou obtenus en résumant des informations fournies par des portiers présents au sein de son domaine. L'élément frontière peut mettre ces informations à la disposition d'autres éléments frontière par le biais d'un mécanisme d'échange de demandes et de réponses. Un domaine administratif peut déterminer le niveau de détail fourni par ses éléments frontière, par exemple:

- un élément frontière souhaitant masquer sa structure interne peut fournir un descripteur unique (avec une indication d'émission de message "demande d'accès") qui représente la totalité de sa zone et fait référence à un portier pour le traitement des appels arrivées;
- un élément frontière qui n'est pas préoccupé par la divulgation de sa structure interne peut fournir un ensemble de canevas dont chacun indique le portier pour une zone située au sein du domaine;
- un élément frontière appartenant à un serveur pare-feu (ou utilisant le modèle avec acheminement par portier) peut fournir un canevas pour la totalité de la zone avec une indication d'émission de message "établissement";
- un élément frontière dont le domaine contient des trous (correspondant à des numéros qui ont été déplacés vers un autre domaine administratif) fournit des canevas avec une marque "émission de demande d'accès" indiquant l'élément frontière qui doit être utilisé pour contacter l'autre domaine administratif;
- un élément frontière résolveur d'adressage (possédant, par exemple une copie complète de 44) peut contenir un canevas avec une marque "émission de demande d'accès" pour chaque domaine administratif au sein de 44.

Les éléments frontière n'ont pas l'obligation de gérer une copie complète de la base de donnée. S'il dispose d'une telle copie, l'élément frontière doit alors utiliser des canevas de configuration statique dont les marques "émission de demande d'accès" indiquent un élément frontière résolveur d'adressage qui sera utilisé pour résoudre les autres demandes.

G.7.1.2 Réception de descripteurs

Un élément frontière peut demander à un autre élément frontière de lui fournir des canevas de configuration statique. Ce dernier décide de la réponse donnée à la demande.

L'élément frontière demandeur émet un message "demande de descripteur" indiquant les descripteurs qu'il souhaite recevoir. S'il est en mesure de transférer les descripteurs, l'élément frontière propriétaire répond alors au moyen d'un message "confirmation de descripteur" qui contient tous les canevas.

L'élément frontière demandeur peut mémoriser dans un cache une copie d'un canevas obtenu de cette manière et la conserver jusqu'à la fin de la durée de vie du canevas, moment auquel la copie sera effacée. S'il modifie ses canevas de configuration statique avant la fin de leur durée de vie, l'élément frontière propriétaire doit alors émettre un message "mise à jour de descripteur" à destination des éléments frontière dont il a connaissance. Lorsqu'il reçoit un message "mise à jour de descripteur", un élément frontière doit supprimer, ajouter ou modifier dans son cache tous les canevas concernés ou demander au propriétaire de lui fournir les copies des descripteurs indiqués.

Un élément frontière intermédiaire (situé entre les domaines administratifs d'origine et de destination, par exemple un résolveur d'adressage ou un point d'agrégation) peut publier ses propres descripteurs en fonction de ceux qu'il reçoit. Un résolveur d'adressage peut, par exemple, indiquer qu'elle est elle-même le contact pour une "demande d'accès", même si elle a reçu d'un autre élément frontière des descripteurs indiquant ce dernier comme contact.

Un élément frontière peut indiquer dans un canevas la nécessité pour l'auteur d'un appel de recevoir l'autorisation d'appeler un domaine administratif. Si un fanion **callSpecific** est placé dans un canevas et si le type de message spécifie la nécessité d'envoyer un message "demande d'accès", alors l'auteur de l'appel doit fournir des informations pour chaque appel dans le message de demande d'accès. Lorsqu'un élément frontière reçoit un message de demande d'accès ne contenant d'informations propres à l'appel, et si la stratégie adoptée consiste à exiger des informations pour chaque appel, l'élément frontière répondra par un message de "rejet d'accès" portant la mention explicative **needCallInformation**.

Un élément frontière peut émettre un message "mise à jour de descripteur" à destination d'autres éléments frontière connus individuellement ou émettre ce message en diffusion multiple. L'élément frontière doit déterminer le domaine de diffusion dans le cas de diffusion multiple d'un tel message. Le message "mise à jour de descripteur" peut contenir les descripteurs qui ont été modifiés. Il peut, par ailleurs, fournir uniquement les identificateurs des descripteurs modifiés, ce qui permet au destinataire de demander de nouvelles informations. Si un grand nombre de descripteurs a été modifié, les informations doivent alors être émises dans plusieurs messages "mise à jour de descripteur", de manière à éviter que la taille des messages devienne supérieure au maximum pris en charge par les paquets de transport.

G.7.1.3 Réception de réponses à des demandes spécifiques

Un élément frontière peut émettre un message "demande d'accès" à destination d'un autre élément frontière pour demander la résolution d'une adresse qualifiée de manière totale ou partielle. La demande d'accès est émise en général au moyen d'un protocole de transport non fiable (par exemple, le protocole UDP) mais elle peut utiliser également un protocole de transport fiable (par exemple, le protocole TCP).

Lorsqu'il reçoit la demande d'accès, l'élément frontière effectue une recherche dans sa base de données et fournit dans sa réponse le canevas le plus spécifique pour la destination demandée. Il renverra la totalité des canevas s'il en existe plusieurs qui satisfont à la demande. S'il est effectivement responsable pour l'adresse d'alias spécifiée, l'élément frontière répondra alors en général avec un canevas indiquant qu'un message "demande d'accès" ou un message "établissement" doit être émis. S'il joue le rôle d'un résolveur d'adressage, l'élément frontière de destination répondra alors normalement avec un canevas indiquant que le message "demande d'accès" doit être émis.

L'élément frontière de destination peut également ajouter à la réponse d'autres canevas dont il estime qu'ils peuvent être utiles dans le futur. L'ajout de ces canevas ne doit pas faire passer la taille de la réponse au-delà de la limite qui conduirait le réseau de transport à effectuer une fragmentation (par exemple, 576 octets pour le protocole IPv4 ou de 1200 octets pour le protocole IPv6).

Un élément frontière qui est, par exemple, couplé étroitement à un serveur pare-feu peut fournir deux canevas dans sa réponse à un message "demande d'accès": un canevas de courte durée de vie (de quelques minutes ou secondes) spécifiant l'emplacement vers lequel un message "établissement" doit être émis et des canevas supplémentaires spécifiant que des messages "demande d'accès" doivent être émis à destination de l'élément frontière pour d'autres adresses d'alias au sein du domaine administratif.

Un élément frontière peut mémoriser dans un cache, et jusqu'à l'expiration de sa durée de vie, un canevas reçu dans un message "confirmation d'adresse".

G.7.2 Découverte d'un élément frontière ou d'un ensemble d'éléments frontière

G.7.2.1 Statique

Un élément frontière peut disposer d'un ensemble administré d'autres éléments frontière qu'il peut contacter à des fins de résolution d'adresse. La détermination de cet ensemble administré peut se

faire par le biais d'accords bilatéraux entre domaines administratifs. Les domaines administratifs peuvent utiliser de manière optionnelle les services d'un résolveur d'adressage.

G.7.2.2 Dynamique

Sur les réseaux Internet, le système DNS définit les propriétaires d'adresses du type "identificateur de messagerie électronique". Il s'ensuit qu'un élément frontière effectuera, en l'absence d'informations plus complètes, une recherche d'enregistrements SRV sur le serveur DNS en utilisant la partie de l'identificateur de messagerie électronique situé à la droite du caractère "@" (il effectuera, par exemple, une recherche de l'enregistrement "_h2250-annexe-g_udp.example.org" sur le serveur DNS pour la résolution de l'adresse "person@example.org"). La réponse à cette recherche permettra de générer un canevas "envoi de demande d'accès" qui peut être utilisé par le processus de résolution. Les canevas générés à partir des demandes de serveur DNS ne doivent pas rester en cache pendant une durée supérieure à la durée de vie indiquée dans la réponse du serveur DNS.

G.7.2.3 Autres méthodes

L'utilisation d'autres méthodes de localisation d'un élément frontière appelle une étude ultérieure.

G.7.3 Procédures de résolution

G.7.3.1 Procédure de résolution au sein d'un domaine administratif

L'élément frontière trouve un canevas correspondant lorsqu'il fait l'objet d'une demande de résolution d'alias d'adresse (par exemple, de la part d'une passerelle ou d'un portier situé au même emplacement).

Si plusieurs canevas conviennent, les canevas adéquats sont sélectionnés et triés en fonction d'une stratégie locale. Les canevas peuvent, par exemple, être triés tout d'abord selon le critère de longueur du joker (pour donner la préférence aux canevas les plus spécifiques), puis selon le critère du type de protocole spécifié (l'émission d'une demande d'établissement est préférable à l'émission d'une demande d'accès).

L'élément frontière renverra tous les canevas conformes si plusieurs satisfont à la demande.

Si la procédure de sélection de canevas ne fournit pas de canevas avec une marque "émission d'établissement", l'élément frontière émettra alors, à destination de l'adresse spécifiée dans le canevas, un message "demande d'accès" contenant une adresse de destination spécifique. Il peut mémoriser dans son cache la réponse reçue de l'élément frontière et renvoyer au demandeur l'adresse vers laquelle doit être émis le message (SETUP) "établissement".

G.7.3.2 Procédure de résolution entre domaines administratifs

L'élément frontière effectue une recherche parmi les canevas mémorisés dans son cache et trouve un canevas correspondant à l'adresse de la demande lorsqu'il fait l'objet d'une demande d'accès.

Si plusieurs canevas conviennent, ils sont triés tout d'abord selon le critère de longueur du joker (pour donner la préférence aux canevas les plus spécifiques), puis selon le critère du type de protocole spécifié (l'émission d'une demande d'établissement est préférable à l'émission d'une demande d'accès). On rejette lors de chaque tri les canevas qui ne fournissent pas la correspondance la plus spécifique.

Si les canevas conformes sont marqués "émission de demande d'accès", l'élément frontière peut alors choisir de transmettre le message "demande d'accès" vers un ou plusieurs éléments frontière spécifiés dans un ou plusieurs canevas ou il peut choisir de renvoyer les canevas tels quels. L'élément frontière ne doit pas retransmettre le message "demande d'accès" vers un autre élément frontière si le compteur de bonds du message "demande d'accès" reçu a atteint la valeur nulle; il doit dans ce cas renvoyer à la place du message tout canevas conforme. L'élément frontière répondra au

moyen d'un message "rejet d'accès" indiquant le dépassement du nombre de bonds si le compteur de bonds a atteint la valeur nulle et si l'élément frontière ne dispose d'aucune information pouvant être fournie dans un message "confirmation d'accès".

A ce stade, l'élément frontière peut utiliser un élément frontière d'un troisième domaine administratif (par exemple un résolveur d'adressage) afin d'autoriser la demande d'accès. A cet effet, il envoie un message "de demande de validation", contenant des jetons d'accès fournis par l'élément frontière demandeur dans la demande d'accès. L'élément frontière récepteur valide les jetons et renvoie une "confirmation de validation".

L'élément frontière renvoie ensuite un message "confirmation d'accès" contenant les canevas qu'il a trouvé (ces canevas auront des champs "adresse" et "type de message" identiques) ainsi que tout autre canevas dont il estime qu'il sera utile.

L'élément frontière renverra tous les canevas conformes s'il en existe plusieurs qui satisfont à la demande.

Si la demande d'accès contient des informations spécifiques propres à l'appel, la validité des canevas renvoyés est limitée à l'appel demandé. Cette disposition est utilisée lorsqu'un domaine administratif souhaite autoriser l'accès pour chaque appel. Dans ce cas le domaine administratif peut demander l'inclusion d'informations propres à l'appel pour chaque demande d'accès transmise. Il positionne alors un fanion dans les canevas qui lui correspondent.

G.7.4 Echange d'information d'utilisation

Les domaines administratifs peuvent demander aux autres domaines de leur communiquer des informations d'utilisation lors de l'établissement d'appels particuliers. Des messages "d'indication d'utilisation" peuvent être communiqués à n'importe quel stade de l'établissement de l'appel. De plus, plusieurs indications d'utilisation peuvent être transmises pour le même appel, chacune contenant des informations plus récemment mises à jour.

Les échanges d'indications d'utilisation ne sont admis que s'il existe une relation de service entre les deux éléments frontière.

Des demandes d'indication d'utilisation seront envoyées, si un élément frontière l'exige, dans les canevas pour lesquels il fait office d'élément à contacter ou s'il l'indique dans l'un des messages de demande d'utilisation, de demande d'accès, de demande de validation et de confirmation de validation, envoyés en rapport avec l'appel faisant l'objet d'une demande d'indication d'utilisation.

G.8 Protocole

Les messages du protocole de la présente annexe peuvent être émis en utilisant un service de transport non fiable (par exemple, le protocole UDP) ou un service fiable (par exemple, le protocole TCP) vers une adresse bien connue. Sur les réseaux Internet, il convient d'utiliser le port 2099 bien connu, pour les protocoles TCP et UDP, sauf si un autre port a été indiqué à l'expéditeur. Les éléments frontière doivent être à l'écoute sur les ports TCP et UDP.

Si les messages sont émis au moyen d'un service de transport fiable, il est alors possible d'émettre des messages multiples en respectant les contraintes de taille définies pour l'unité de données du protocole (PDU) de transport fiable, dans la mesure où des messages entiers sont émis. (Cette unité PDU est définie par un paquet TPKT dans les implémentations du protocole IP, telles qu'elles sont résumées dans l'Appendice IV.)

En cas d'utilisation d'un service de transport non fiable, les messages de demande peuvent être retransmis. La valeur par défaut du temporisateur de retransmission doit être déterminée par une méthode adaptative sensible au délai (telle que celle qu'utilise le protocole TCP). Des temps d'attente exponentiels seront alors utilisés pour les retransmissions subséquentes. Le nombre de retransmissions ne dépassera pas 5. Les réponses ne seront pas retransmises.

Dans les implémentations du protocole IP, les messages comporteront un préfixe constitué d'en-têtes de paquet TPKT, permettant l'émission de messages multiples. Le champ longueur de paquet UDP contiendra la longueur totale de la charge utile, y compris tous les messages et leurs en-têtes TPKT.

G.8.1 Considérations relatives à la sécurité

Le fonctionnement de sécurité du protocole IP, tel qu'il est décrit dans la Norme IETF RFC 1825 ("architecture de sécurité pour le protocole Internet") et complété par la Norme IETF RFC 1826 ("en-tête d'authentification du protocole IP") ou par la Norme IETF RFC 1827 ["charge utile d'encapsulation de sécurité IP" (ESP, *IP encapsulating security payload*)] ou par les deux, sera mis en œuvre lorsque l'authentification, l'intégrité et le chiffrement sont souhaités pour des messages échangés entre éléments frontière.

Les procédures et expressions de l'UIT-T H.235 seront utilisées le cas échéant pour la prise en charge de la sécurité au niveau application. Ceci concerne plus précisément les formats de jeton et les échanges d'authentification. Les jetons et les Jetons (**token**) chiffrés reçus dans les messages de réponse seront utilisés dans une demande connexe ultérieure.

G.8.2 Définitions des messages

Tout message contient un ensemble de champs communs en plus des informations propres au message. Les champs communs sont les suivants:

Champ	Description
numéro de séquence [<i>sequenceNumber</i>]	Tout message de demande ou de mise à jour contient un numéro de séquence unique. Le message émis en réponse à un message de demande (message de confirmation ou de rejet) utilise le numéro de séquence du message de demande. Les messages retransmis auront le même numéro de séquence.
adresse de réponse [<i>replyAddress</i>]	Adresse à laquelle il faut envoyer la réponse à un message de demande. Tout message de demande contiendra une adresse de réponse, sauf si la demande a été envoyée par un service de transport bidirectionnel orienté connexion (par ex. TCP). Aucun message, hormis les messages de demande, ne doit contenir d'adresse de réponse.
Version	Version de protocole utilisée par l'émetteur du message.
comptage de bonds [<i>hopCount</i>]	Définition du nombre d'éléments frontière à travers lesquels ce message peut se propager. Un élément frontière commence par décrémenter le compteur de bonds (hopCount) lorsqu'il reçoit un message et qu'il décide de le retransmettre à destination d'un autre élément frontière. Si la nouvelle valeur du compteur est supérieure à 0, l'élément insère alors cette valeur dans le message à retransmettre. L'élément frontière ne retransmettra pas le message si le comptage a atteint la valeur 0. Si le message est une demande, l'élément frontière répondra alors au moyen d'un message de confirmation contenant toute information pertinente. Il répondra au moyen d'un message de rejet si aucune information n'est disponible.

valeur de vérification d'intégrité [<i>integrityCheckValue</i>]	Permet d'améliorer l'intégrité/authentification des messages RAS. La valeur de vérification d'intégrité fondée sur un mécanisme cryptographique est calculée par l'expéditeur, qui applique un algorithme d'intégrité négocié et la clé secrète à la totalité du message. Avant le calcul de la valeur integrityCheckValue , chaque octet de ce champ doit être mis à zéro. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ integrityCheckValue et transmet le message.
jetons [<i>token</i>]	Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.
Jetons (token) chiffrés [<i>cryptoTokens</i>]	Jetons chiffrés.
NonStandard	Informations non normalisées.

G.8.2.1 Descripteur [*descriptor*]

Le descripteur n'est pas un message, mais un élément de message utilisé comme étiquette pour un ensemble de canevas.

Le descripteur contient les informations suivantes:

Champ	Description
informations de descripteur [<i>descriptorInfo</i>]	Identificateur non ambigu du descripteur et de l'instant de sa dernière modification (se référer aux informations de descripteur ci-dessous).
Canevas [<i>templates</i>]	Ensemble de canevas qui définit les adresses pouvant être résolues par ce descripteur.
identificateur de portier [<i>gatekeeperID</i>]	Identificateur du type texte indiquant le propriétaire du descripteur (c'est-à-dire, le portier qui a créé ce message).

G.8.2.2 Informations de descripteur [*descriptorInfo*]

Les informations de descripteur identifient de manière non ambiguë le descripteur et indiquent l'instant de sa dernière modification.

Champ	Description
identificateur de descripteur [<i>descriptorID</i>]	Identificateur globalement non ambigu utilisé pour distinguer ce descripteur parmi un grand nombre d'autres descripteurs possibles.
dernière modification [<i>lastChanged</i>]	Date et heure de la dernière modification de ce descripteur.

G.8.2.3 Canevas [*template*]

Le canevas d'adresse décrit un ensemble d'une ou plusieurs adresses d'alias. Le canevas n'est pas un message, mais un élément utilisé par d'autres éléments comme bloc de construction. Le canevas se compose d'autres structures qui sont décrites dans les sous-paragraphe qui suivent.

Champ	Description
Modèle [<i>pattern</i>]	Liste de modèles (voir Modèle ci-dessous).
informations d'acheminement [<i>routeInfo</i>]	Liste d'informations d'acheminement pour ce canevas (se référer aux informations d'acheminement ci-dessous).
durée de vie [<i>timeToLive</i>]	Durée de validité de ce canevas, exprimée en secondes.

G.8.2.3.1 Informations d'acheminement [*routeInfo*]

La structure "informations d'acheminement" figurant dans l'élément "canevas" (**template**) (champ "informations de routage") (**routeInfo**) contient les champs suivants:

Champ	Description
type de message [<i>messageType</i>]	Type de message à émettre lors d'une tentative de résolution d'une adresse spécifique utilisant ce canevas. Les types possibles sont les suivants "émission de demande d'accès" [<i>sendAccessRequest</i>], "émission d'établissement" [<i>sendSetup</i>] ou "inexistant" [<i>nonExistent</i>] (indique que l'adresse n'existe pas).
spécificité d'appel [<i>callSpecific</i>]	Si ce champ est mis à la valeur Vrai, chaque appel vers cette route doit obligatoirement faire l'objet d'une autorisation, ce qui implique la nécessité pour le message de demande d'accès de contenir les informations propres à l'appel. Ce champ booléen a une signification uniquement quand le champ messageType prend la valeur <i>sendAccessRequest</i> ; sinon callSpecific sera mis à la valeur Faux.
spécification d'utilisation [<i>usageSpec</i>]	Lorsqu'il est présent ce champ spécifie les messages d'"indication d'utilisation" qui seront envoyés en ce qui concerne les appels vers cet acheminement.
informations de tarification [<i>priceInfo</i>]	Liste d'informations de tarification pour cet acheminement particulier (se référer aux informations de tarification ci-dessous). Il convient de noter que des structures de tarification différentes sont décrites dans des structures "informations de routage" multiples.
contacts	Informations de contact concernant l'élément qui acceptera le message tel qu'il est spécifié dans le champ "type de message" messageType des informations de routage. Les informations de contact peuvent être fournies sous la forme d'une liste de contacts possibles (se référer à la description des informations de contact ci-dessous).
type	Type de point de terminaison pouvant recevoir l'appel. Dans les cas de routage par portier, ce champ indique les points de terminaison desservis par le portier plutôt que le portier lui-même.

G.8.2.3.2 Informations de tarification [*priceInfo*]

Les informations de tarification constituent un élément figurant dans la structure "informations de routage" (champ "informations de tarification"). Les informations de tarification sont définies par les structures "spécification d'informations de tarification" [**PriceInfoSpec**] et "élément de prix" [**PriceElement**].

La structure "spécification d'informations de tarification" contient les champs suivants:

Champ	Description
devise [<i>currency</i>]	Désignation de devise conformément à l'ISO 4217.
échelle de devise [<i>currencyScale</i>]	Position de la virgule implicite, par rapport à la gauche d'un champ de prix. Si, par exemple, la devise est spécifiée en dollars US, un champ currencyScale égal à 2 indique que le montant indiqué dans le champ priceElement "élément de prix" est exprimé en cents US.
valable à partir de [<i>validFrom</i>]	Date et heure de début de validité de ces informations.
valable jusqu'à [<i>validUntil</i>]	Date et heure de fin de validité de ces informations.
heures depuis [<i>hoursFrom</i>]	Heure dans la journée à partir de laquelle ce tarif s'applique.
heures jusqu'à [<i>hoursUntil</i>]	Heure dans la journée à partir de laquelle ce tarif ne s'applique plus. Ce champ peut être inférieur au champ "heures depuis", ce qui indique un tarif d'étendue horaire nulle.
élément de prix [<i>priceElement</i>]	Liste optionnelle "d'éléments de prix" additifs constituant le tarif.
formule de prix [<i>priceFormula</i>]	Chaîne optionnelle contenant une formule de tarification utilisée en variante à la structure "élément de prix".

La structure "élément de prix" [PriceElement] contient les champs suivants:

Champ	Description
montant [<i>amount</i>]	Incrément de comptage. Le compteur est incrémenté d'une unité pour chaque quantum ou fraction de quantum .
quantum	Nombre d'unités auxquelles s'applique le "montant". Une valeur de 60, avec des "unités" units exprimées en secondes indique, par exemple, que l'appel est taxé toutes les minutes ou fractions de minute. Si le champ unités est mis à l'une des valeurs " initial ", " minimum " ou " maximum ", alors le champ "unités" est sans intérêt, et sa valeur ne doit pas être prise en compte par le destinataire.
unités [<i>units</i>]	Unité utilisée pour exprimer le quantum: <ul style="list-style-type: none"> • secondes – secondes de durée de l'appel; • paquets – nombre de paquets émis ou reçus; • octets – nombre d'octets émis ou reçus; • initial – taxation initiale au moment de la connexion; • minimum – taxation minimale d'un appel; • maximum – taxation maximale d'un appel.

G.8.2.3.3 Informations de contact [*ContactInformation*]

La structure "informations de contact" est un élément de la structure "informations de routage" (champ "**contacts**").

Champ	Description
adresse de transport [<i>transportAddress</i>]	Adresse (par exemple, une adresse de transport ou un identificateur URL) à destination de laquelle doit être émis le message spécifié dans le champ "type de message" messageType de la structure "informations d'acheminement". Dans tous les cas où cela est possible, l'utilisation d'une adresse de transport est obligatoire.
priorité [<i>priority</i>]	Lorsque plusieurs contacts figurent dans la liste, le champ "priorité" priority spécifie l'ordre des tentatives de contact multiple. Les contacts de la liste peuvent partager une même priorité, par exemple lorsqu'il n'existe pas de préférence pour l'ordre des tentatives de contact. Une priorité égale à 0 indique la priorité la plus élevée (premier choix).
qualité de service du transport [<i>transportQoS</i>]	Indique qui a la responsabilité de la réservation des ressources pour l'appel établi par ce contact.
Sécurité [<i>security</i>]	Mécanisme de sécurité indiquant l'ordre de préférence à observer pour communiquer avec le contact.
jetons d'accès [<i>accessTokens</i>]	Ensemble de jetons qui seront transmis dans le message adressé à ce contact ("Etablissement" ou "Demande d'accès"). Ces jetons seront également envoyés dans les messages "Indication d'utilisation" relatifs aux appels utilisant ce canevas.

G.8.2.3.4 Modèle

Une structure de modèle apparaît dans le canevas d'adresse. Le modèle permet de spécifier une adresse alias, une adresse alias joker ou une série d'adresses alias:

Champ	Description
Spécifique [<i>specific</i>]	Désigne une adresse alias spécifique.
Joker [<i>wildcard</i>]	Forme de définition hiérarchique qui représente des extensions possibles de la chaîne. Cette extension est possible à la fin d'un numéro E.164 ou en début de chaîne dans le cas d'une adresse de messagerie électronique. Par exemple si ce champ wilcard est "+1 303", le modèle peut représenter tout numéro dans la zone de code de Denver.
Série [<i>range</i>]	Il s'agit d'une série d'adresses, y compris la première et la dernière de la série.

G.8.2.4 Structures communes

Les structures définies dans le présent paragraphe apparaissent dans un certain nombre de messages.

G.8.2.4.1 Autre élément frontière [*AlternateBE*]

Champ	Description
adresse de contact [<i>contactAddress</i>]	Adresse de transport d'un autre élément frontière (à destination de laquelle peuvent être émis des messages définis dans l'Annexe G).
priorité [<i>priority</i>]	Lorsque des variantes multiples figurent dans la liste, le champ "priorité" priority spécifie l'ordre dans lequel ces variantes doivent être essayées. Les variantes dans la liste peuvent partager une même priorité, par exemple lorsqu'il n'existe pas de préférence pour le choix des variantes. Une priorité égale à 0 indique la priorité la plus élevée (premier choix).
identificateur d'élément [<i>elementIdentifier</i>]	Chaîne de caractères Unicode identifiant cet autre élément frontière.

G.8.2.4.2 Information sur un participant

Cette structure contient des informations relatives à un participant (à l'origine ou à destination) de l'appel.

Champ	Description
adresse logique [<i>logicalAddress</i>]	Adresses de format type adresse de messagerie électronique ou numéro E.164 permettant d'identifier le participant.
identificateur de domaine [<i>domainIdentifier</i>]	Adresse alias identifiant le domaine administratif à l'origine ou à destination de l'appel. Lorsque plusieurs domaines sont impliqués dans l'établissement d'un appel, il faut indiquer le domaine qui a fait office d'origine ou de destination du point de vue de l'expéditeur.
adresse de transport [<i>transportAddress</i>]	Adresse de transport du point d'extrémité.
type de point d'extrémité [<i>endpointType</i>]	Indications détaillées concernant le type de point d'extrémité et les capacités correspondantes.
information d'utilisateur [<i>userInfo</i>]	Information concernant l'utilisateur à l'origine de l'appel. Cette identification peut se faire sous un format d'adresse de messagerie électronique ou d'un numéro de réseau public et le cas échéant au moyen de justificatifs d'identité.
fuseau horaire [<i>timeZone</i>]	Indication du fuseau horaire du participant, dans la mesure où elle est utilisée à des fins de tarification. Si l'appel provient d'une passerelle, alors le fuseau horaire de la passerelle doit être transmis. Exprimé en secondes par rapport au temps universel coordonné.

G.8.2.4.3 Information sur l'appel

Information permettant d'identifier un appel particulier.

Champ	Description
identificateur d'appel [<i>callIdentifier</i>]	Permet d'identifier l'appel de façon univoque. Constituera l'identificateur d'appel associé au même appel que dans les messages RAS et de signalisation d'appel.

identificateur de conférence [<i>conferenceID</i>]	Permet d'identifier de façon univoque la conférence à laquelle participe l'appel. Constituera l'identificateur de conférence associé au même appel que dans les messages RAS et de signalisation d'appel.
---	---

G.8.2.4.4 Information d'utilisateur

Information permettant d'identifier l'utilisateur ou un participant quelconque à l'appel.

Champ	Description
Identificateur d'utilisateur [<i>userIdentifier</i>]	Identification univoque de l'utilisateur.
authentificateur d'utilisateur [<i>userAuthenticator</i>]	Jetons chiffrés permettant une authentification sécurisée.

G.8.2.4.5 Spécification d'utilisation

Cet élément décrit les paramètres dont la notification est nécessaire dans les messages "d'indication d'utilisation". Les appels auxquels cette spécification est applicable sont déterminés par le contexte du message qui contient l'élément "Spécification d'utilisation" **usageSpecification**.

Champ	Description
Envoyer à [<i>sendTo</i>]	Élément frontière vers lequel les messages d'indication d'utilisation doivent être envoyés. Puisque l'expéditeur doit avoir des relations de service avec cet élément frontière, il s'agit de l'identificateur d'élément renvoyé dans le message de confirmation de service.
Quand [<i>when</i>]	Spécifie les étapes de l'appel pour lesquelles les indications en question doivent être envoyées ainsi que la fréquence de ces envois: <ul style="list-style-type: none"> • jamais – arrêter d'envoyer des messages; • commencer – au début de l'appel; • cesser – à la fin de l'appel, ou après; • période – envoyer périodiquement les messages en question pendant la durée de l'appel. Période indiquée en secondes; • échec – notifier les tentatives d'appel ayant échoué.
champs obligatoires [<i>required</i>]	Liste d'identificateurs des champs qui <i>doivent</i> obligatoirement être présents dans les messages "UsageIndication". L'expéditeur du message information sur l'utilisation des ressources doit rejeter ou ignorer le message qui contient ce message, s'il ne peut fournir les informations relatives à ces champs.
champs théoriquement présents [<i>preferred</i>]	Liste d'identificateurs des champs qui <i>devraient</i> être présents dans les messages <i>UsageIndication</i> .

G.8.2.4.6 Mode sécurité

Cet élément décrit un profil de sécurité spécifique à utiliser dans les communications auxquelles l'Annexe G est consacrée.

Champ	Description
authentification [<i>authentication</i>]	Champ indiquant le mécanisme d'authentification devant être utilisé pour cette relation de service. Le mécanisme d'authentification doit être choisi dans l'ensemble fourni dans le message "demande de service".
Intégrité [<i>integrity</i>]	Indique le mécanisme d'intégrité devant être utilisé. Si ce champ est présent, tous les messages ultérieurs renseigneront le champ "valeur de vérification d'intégrité" integrityCheckValue ; le champ "mode d'authentification" AuthenticationMode décrit alors le mode de génération des clés secrètes (par échange de clés Diffie-Hellman ou <i>a priori</i>).
identificateurs d'algorithme [<i>algorithmOID</i>]	Indique l'algorithme de cryptage pour le mécanisme de sécurité.

G.8.2.5 Demande de service [*ServiceRequest*]

Un élément frontière peut établir une relation de service par l'émission d'un message "demande de service" à destination d'un autre élément frontière. La relation définit les mécanismes de sécurité devant être utilisés entre les éléments frontière et permet l'identification d'éléments frontière de remplacement ou de secours. Il convient de noter que cette relation est unidirectionnelle. La sécurité entre les deux éléments est utilisée pour des demandes émises par l'élément qui est à l'origine de la demande de service et pour des réponses émises par le récepteur de la demande de service. Des clés de session peuvent être créées pendant le processus d'établissement de la relation de service. Ces clés resteront valides pendant la durée de vie de la relation de service. Il est possible d'utiliser des jetons à cet effet, tel qu'indiqué dans l'UIT-T H.235.

Le destinataire d'une demande de service peut indiquer d'autres éléments frontière pouvant être utilisés par le récepteur de la demande pour une demande de service de secours. L'établissement d'une relation de service est obligatoire pour pouvoir échanger des messages d'indication d'utilisation. Sinon, il s'agit d'une procédure optionnelle, mais il se peut que la stratégie d'un élément frontière impose une telle relation.

Un élément frontière peut émettre un message "demande de service" à destination d'un élément frontière avec lequel il a déjà établi une relation dans le but de remplacer les caractéristiques de la relation existante par de nouvelles caractéristiques. Les relations de service peuvent avoir une durée de vie limitée. Un élément frontière peut alors régénérer la relation en envoyant une nouvelle demande de service.

Champ	Description
Identificateur d'élément [<i>elementIdentifier</i>]	Chaîne identifiant l'élément frontière qui envoie la demande.
Identificateur de domaine [<i>domainIdentifier</i>]	Indique le domaine administratif qui demande la relation de service.
capacité de sécurité [<i>securityCapability</i>]	Ensemble de mécanismes de sécurité que cet élément frontière peut prendre en charge.
durée de vie [<i>timeToLive</i>]	Durée de vie proposée, en secondes, pour la relation de service. En l'absence de cette indication, on suppose une durée de vie infinie.

G.8.2.6 Confirmation de service [*ServiceConfirmation*]

Lorsqu'il reçoit un message "demande de service", un élément frontière répond au moyen d'un message "confirmation de service" pour indiquer qu'il accepte l'établissement d'une relation de service. Si l'élément frontière possède déjà une relation de service avec l'élément frontière qui a émis le message "demande de service", l'émission du message "confirmation de service" indique que les caractéristiques de la relation initiale ne sont plus valides et ont été remplacées par les nouvelles caractéristiques.

Champ	Description
identificateur d'élément [<i>elementIdentifier</i>]	Chaîne identifiant l'élément frontière.
Variantes [<i>alternates</i>]	Liste d'autres éléments frontière pouvant être contactés en cas d'absence de réponse de cet élément frontière.
identificateur de domaine [<i>domainIdentifier</i>]	Désigne le domaine administratif qui répond à la demande.
mode sécurité [<i>securityMode</i>]	Indique le mécanisme de sécurité à utiliser pour cette relation de service. Le mécanisme de sécurité doit être choisi dans l'ensemble de mécanismes mentionnés dans le message de demande de service.
durée de vie [<i>timeToLive</i>]	Durée de vie exprimée en secondes de la relation de service déterminée par l'élément frontière serveur.

G.8.2.7 Rejet de service [*ServiceRejection*]

Lorsqu'il reçoit un message "demande de service", un élément frontière répond au moyen d'un message "rejet de service" pour indiquer qu'il refuse l'établissement d'une relation de service. Si l'élément frontière possède déjà une relation de service avec l'élément frontière qui a émis le message "demande de service", l'émission du message "rejet de service" indique que les nouvelles caractéristiques ont été rejetées, mais la relation initiale reste valide.

Champ	Description
motif [<i>reason</i>]	Motif du rejet de la demande de service par l'élément frontière. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• service indisponible [<i>serviceUnavailable</i>] – Cet élément frontière n'est pas disponible actuellement pour le service;• renvoi du service [<i>serviceRedirected</i>] – La tentative doit être renouvelée en utilisant la liste des autres éléments frontière;• sécurité [<i>security</i>] – Cet élément frontière ne peut pas prendre en charge les mécanismes de sécurité proposés dans le message "demande de service";• continuation [<i>continue</i>] – Indique le message suivant de demande de service à envoyer, afin de poursuivre la procédure à plusieurs étapes d'échange de clés;• non défini [<i>undefined</i>] – Le motif du rejet de la demande de service ne correspond à aucun des choix précédents.

Variantes
[*alternates*] Liste d'autres éléments frontière qui sont éventuellement en mesure d'établir une relation de service. Au moins une liste doit être fournie si le motif est "renvoi du service".

G.8.2.8 Libération du service [*ServiceRelease*]

L'un ou l'autre des éléments frontière engagés dans une relation de service peut y mettre fin par l'émission d'un message "libération du service".

Champ	Description
motif [<i>reason</i>]	Motif pour lequel cet élément frontière a mis fin à la relation de service. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• hors service [<i>outOfService</i>] – L'élément frontière passe dans l'état "hors service";• maintenance – L'élément frontière est mis hors service pour des raisons de maintenance;• terminé [<i>terminated</i>] – L'élément frontière a décidé de mettre fin à la relation;• expiration [<i>expired</i>] – La durée de vie de la relation de service s'est écoulée.

Variantes
[*alternates*] Liste d'autres éléments frontière qui sont éventuellement en mesure d'établir une relation de service.

G.8.2.9 Demande de descripteur [*DescriptorRequest*]

Le message "demande de descripteur" permet à une entité de demander à un élément frontière la fourniture de descripteurs spécifiques.

Champ	Description
identificateur de descripteur [<i>descriptorID</i>]	Identificateur d'un ou plusieurs descripteurs particuliers demandés par l'émetteur de ce message.

G.8.2.10 Confirmation de descripteur [*DescriptorConfirmation*]

Le message "confirmation de descripteur" constitue la réponse positive d'un élément frontière à une demande de descripteur, lorsque cet élément peut interpréter la demande et que les règles d'implémentation permettent l'échange d'informations.

Champ	Description
descripteur [<i>descriptor</i>]	Champ "descripteur" descriptor décrit ci-dessus.

G.8.2.11 Rejet de descripteur [*DescriptorRejection*]

Un élément frontière peut rejeter une demande de descripteur pour divers motifs.

Champ	Description
motif [<i>reason</i>]	Motif du rejet de la demande de descripteur. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• taille de paquet dépassée [<i>packetSizeExceeded</i>] – La taille de la réponse est supérieure à la taille maximale de paquet, de sorte que le demandeur devrait émettre la demande en utilisant un autre mécanisme de transport (il doit, par exemple, utiliser le protocole TCP à la place du protocole UDP);• identificateur non valide [<i>illegalID</i>] – Le destinataire de la demande de descripteur ne possède aucun enregistrement au sujet du descripteur concerné;• sécurité [<i>security</i>] – La demande de descripteur ne répond pas aux contraintes de sécurité du destinataire;• dépassement du compteur de bonds [<i>hopCountExceeded</i>] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible;• non disponible [<i>unavailable</i>] – Le destinataire ne peut fournir les descripteurs. Il convient alors d'utiliser une méthode de fourniture statique ou hors-bande;• pas de relation de service [<i>noServiceRelationship</i>] – Le destinataire échangera cette information seulement après établissement d'une relation de service;• non défini [<i>undefined</i>] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.
identificateur de descripteur [<i>descriptorID</i>]	Identificateur du descripteur particulier fourni dans cette réponse.

G.8.2.12 Demande d'identificateur de descripteur [*DescriptorIDRequest*]

La demande d'identificateur de descripteur permet à une entité d'interroger un élément frontière au sujet de la liste des identificateurs de descripteur de son domaine administratif.

G.8.2.13 Confirmation d'identificateur de descripteur [*DescriptorIDConfirmation*]

Un message "confirmation d'identificateur de descripteur" constitue la réponse positive d'un élément frontière à un message "demande d'identificateur de descripteur". Un élément frontière peut, une fois qu'il a reçu le message "demande d'identificateur de descripteur", émettre le message "demande de descripteur" pour demander la transmission des descripteurs.

Champ	Description
informations de descripteur [<i>descriptorInfo</i>]	Liste d'informations de descripteur dont chaque élément identifie sans ambiguïté le descripteur et l'instant de sa dernière modification.

G.8.2.14 Rejet d'identificateur de descripteur [*DescriptorIDRejection*]

Un élément frontière peut rejeter une demande d'identificateur de descripteur pour divers motifs.

Champ	Description
motif [<i>reason</i>]	Motif du rejet de la demande d'identificateur de descripteur. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• pas de descripteurs [<i>noDescriptors</i>] – L'élément frontière ne dispose pas de descripteurs dont il peut rendre compte;• sécurité [<i>security</i>] – La demande d'identificateur de descripteur ne répond pas aux contraintes de sécurité du destinataire;• dépassement du compteur de bonds [<i>hopCountExceeded</i>] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible;• non disponible [<i>unavailable</i>] – Le destinataire ne peut fournir les descripteurs. Il convient alors d'utiliser une méthode de fourniture statique ou hors-bande;• pas de relation de service [<i>noServiceRelationship</i>] – Le destinataire échangera cette information seulement après établissement d'une relation de service;• non défini [<i>undefined</i>] – Le motif du rejet de la demande d'identificateur de descripteur ne correspond à aucun des choix précédents.

G.8.2.15 Mise à jour de descripteur [*DescriptorUpdate*]

Le message "mise à jour de descripteur" constitue la notification d'un élément frontière indiquant que les informations d'adresse ont été modifiées. Un élément frontière peut également émettre ce message pendant l'initialisation. Un élément frontière qui reçoit le message "mise à jour de descripteur" peut demander des informations à l'élément identifié dans ce message.

Champ	Description
émetteur [<i>sender</i>]	Un élément qui reçoit la mise à jour de descripteur peut émettre une demande à destination de cette adresse (par exemple, une adresse de transport ou un identificateur URL).
informations de mise à jour [<i>updateInfo</i>]	Liste de mises à jour dont chaque élément fournit le descripteur ou l'identificateur de descripteur mis à jour. L'élément de la liste indique également si le descripteur a été modifié, ajouté ou supprimé.

G.8.2.16 Accusé de réception de mise à jour de descripteur [*DescriptorUpdateAck*]

Un élément frontière doit accuser réception d'un message "mise à jour de descripteur" en émettant le message "accusé de réception de mise à jour de descripteur". Le numéro de séquence utilisé dans l'accusé de réception doit être identique à celui reçu dans le message "mise à jour de descripteur". Un élément frontière ne doit pas fournir d'accusé de réception d'un message "mise à jour de descripteur" transmis par multidiffusion.

G.8.2.17 Demande d'accès [*AccessRequest*]

Un élément frontière peut émettre un message "demande d'accès" à destination d'un autre élément frontière pour demander la résolution d'une adresse d'alias spécifique.

Champ	Description
informations de destination [<i>destinationInfo</i>]	Adresse devant être résolue.
informations d'origine [<i>sourceInfo</i>]	Informations sur le participant qui a déclenché l'appel faisant l'objet d'une demande d'accès.
informations propres à l'appel [<i>callInfo</i>]	Champ permettant d'identifier l'appel particulier faisant l'objet d'une demande d'autorisation d'accès. En l'absence de cette indication la demande porte sur des appels non définis vers les destinations spécifiées.
spécification d'utilisation [<i>usageSpec</i>]	Champ indiquant les messages d'utilisation que le participant à l'origine de l'appel demande au participant qui répond et qui concernent l'appel faisant l'objet d'une demande d'accès dans ce message. Utilisé uniquement en présence du champ callInfo .

G.8.2.18 Confirmation d'accès [*AccessConfirmation*]

Un élément frontière renvoie dans le message "confirmation d'accès" les informations demandées dans le message "demande d'accès".

Champ	Description
canevas [<i>template</i>]	Liste des canevas correspondant aux attributs de la demande d'accès.
réponse partielle [<i>partialResponse</i>]	Ce message contient, en cas de réponse positive, certaines parties des informations disponibles. La totalité des informations n'a pas été émise pour respecter la taille maximale de paquet. La totalité des informations doit être récupérée en utilisant un autre type de transport (par exemple, le protocole TCP).

G.8.2.19 Rejet d'accès [*AccessRejection*]

Un élément frontière peut rejeter une demande de descripteur pour divers motifs.

Champ	Description
motif [<i>reason</i>]	<p>Motif du rejet de la demande de descripteur. Les valeurs possibles sont les suivantes:</p> <ul style="list-style-type: none">• pas de correspondance [<i>noMatch</i>] – La destination spécifiée dans la demande d'accès ne peut pas être résolue;• taille de paquet dépassée [<i>packetSizeExceeded</i>] – La taille de la réponse dépasse la taille maximale de paquet, de sorte que le demandeur doit émettre la demande en utilisant un autre mécanisme de transport (il doit, par exemple, utiliser le protocole TCP à la place du protocole PDU);• sécurité [<i>security</i>] – La demande de descripteur ne répond pas aux contraintes de sécurité du destinataire;• dépassement du compteur de bonds [<i>hopCountExceeded</i>] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible;• pas de relation de service [<i>noServiceRelationship</i>] – Le destinataire échangera cette information seulement après établissement d'une relation de service;• absence d'informations propres à l'appel [<i>CallInfoNeeded</i>] – La demande ne contenait pas d'informations spécifiques propres à l'appel;• non défini [<i>undefined</i>] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.

G.8.2.20 Demande en cours [*RequestInProgress*]

Un élément frontière peut renvoyer le message "demande en cours" dans le but d'indiquer que la durée nécessaire pour répondre à une demande peut dépasser la durée prévue normalement pour une réponse. Le numéro de séquence doit être identique à celui figurant dans la demande concernée par l'envoi de ce message.

Champ	Description
retard [<i>delay</i>]	Laps de temps, exprimé en millisecondes, prévu pour la réponse de l'élément frontière à la demande initiale.

G.8.2.21 Demande non normalisée [*NonStandardRequest*]

Le message "demande non normalisée" peut être émis par un élément frontière dans le cas d'un message de demande qui n'est pas défini dans l'Annexe G. Les informations non normalisées sont véhiculées dans l'élément "non normalisé" **nonStandard** défini dans le champ "informations communes" lequel est défini par l'Annexe G **AnnexGCommonInfo**.

G.8.2.22 Confirmation non normalisée [*NonStandardConfirmation*]

Le message "confirmation non normalisée" peut être émis par un élément frontière en réponse à un message "demande non normalisée". Les informations non normalisées sont véhiculées dans l'élément "non normalisé" défini dans le champ "informations communes".

G.8.2.23 Rejet non normalisé [*NonStandardRejection*]

Le message "rejet non normalisé" peut être émis par un élément frontière en réponse à un message "demande non normalisée". Les informations non normalisées sont véhiculées dans l'élément "non normalisé" défini dans le champ "informations communes".

Champ	Description
motif [<i>reason</i>]	Motifs du rejet de la demande. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• non prise en charge [<i>notSupported</i>] – Le destinataire sait qu'il s'agit d'une demande non normalisée, mais ne comprend pas ou ne prend pas en charge les données non normalisées;• pas de relation de service [<i>noServiceRelationship</i>] – Le destinataire échangera cette information seulement après établissement d'une relation de service;• non défini [<i>undefined</i>] – Le motif du rejet de la demande non normalisée ne correspond à aucun des choix précédents.

G.8.2.24 Réponse à un message non reconnu [*UnknownMessageResponse*]

Un élément frontière qui ne comprend pas un message reçu doit répondre à l'émetteur au moyen d'un message "réponse à un message non reconnu". L'élément frontière ne doit pas utiliser ce message si un autre message défini dans l'Annexe G fournit une réponse adéquate (un message "rejet de descripteur" constitue, par exemple, une réponse adéquate à une demande de descripteur contenant un identificateur de descripteur non valide).

Champ	Description
message non reconnu [<i>unknownMessage</i>]	Contenu du message non reconnu.
Motif [<i>reason</i>]	Motif d'utilisation du message "réponse à un message non reconnu". Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• non compris [<i>notUnderstood</i>] – Le message n'a pas été compris;• non défini [<i>undefined</i>] – Le motif de l'émission du message "réponse à un message non reconnu" ne correspond à aucun des choix précédents.

G.8.2.25 Demande d'utilisation

Ce message permet de demander au destinataire d'envoyer des messages d'indication d'utilisation concernant un appel particulier.

Champ	Description
informations propres à l'appel [<i>callInfo</i>]	Appel pour lequel la demande d'indications est formulée.
spécification d'utilisation [<i>usageSpec</i>]	Spécifie quand les indications doivent parvenir et quel doit être leur contenu.

G.8.2.26 Confirmation d'utilisation

Le message de confirmation d'utilisation est envoyé en réponse à un message de demande d'utilisation, afin d'indiquer que le destinataire a accepté la demande et enverra des indications d'utilisation.

G.8.2.27 Rejet d'utilisation

Le message de rejet d'utilisation est envoyé en réponse à un message de demande d'utilisation, afin d'indiquer que le destinataire a rejeté la demande et n'enverra pas subséquemment d'indications d'utilisation.

Champ	Description
motif [<i>reason</i>]	Motif du rejet de la demande d'utilisation par l'élément frontière. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none">• appel non valide [<i>invalidCall</i>];• sécurité [<i>security</i>];• non disponible [<i>unavailable</i>];• pas de relation de service [<i>noServiceRelationship</i>];• non défini [<i>undefined</i>].

G.8.2.28 Indication d'utilisation

Ce message notifie des indications détaillées sur l'appel ainsi que des information concernant l'utilisation des ressources. Il est envoyé en rapport avec le dernier élément de spécification d'utilisation **UsageSpecification** reçu par l'élément frontière et concernant l'appel.

Champ	Description
informations propres à l'appel [<i>callInfo</i>]	Appel auquel l'indication se rapporte.
jetons d'accès [<i>accessTokens</i>]	Jetons d'accès relatifs à l'appel. Ces jetons ont été reçus dans le canvas d'adresse utilisé pour l'appel et communiqué dans le message de demande d'accès/d'établissement relatif au même appel.
rôle de l'expéditeur [<i>senderRole</i>]	Rôle de l'expéditeur de l'indication: <ul style="list-style-type: none">• origine – participant à l'origine de l'appel;• destination – participant destinataire;• non normalisé – autre.

état d'appel d'utilisation [<i>usageCallStatus</i>]	Etat actuel de l'appel: <ul style="list-style-type: none"> • préconnexion; • appel en cours; • appel achevé.
adresse de la source [<i>sourceAddress</i>]	Numéro E.164 ou adresse de messagerie électronique du demandeur. Dans le cas d'un numéro E.164 il s'agit de l'identification ANI/CLI.
adresse de la destination [<i>destAddress</i>]	Numéro E.164 ou adresse de messagerie électronique de l'appelé.
instant de début [<i>startTime</i>]	Instant du début de l'appel exprimé en temps universel coordonné (UTC) . Concerne uniquement les appels ayant franchi l'étape de l'établissement.
instant de fin [<i>endTime</i>]	Instant de fin de l'appel exprimé en temps universel coordonné (UTC) . Concerne uniquement les appels aboutis.
cause de fin [<i>terminationCause</i>]	Motif de la fin de l'appel. Concerne uniquement les appels aboutis.
information d'utilisation [<i>usageInformation</i>]	Ensemble de champs d'information. Chaque champ est représenté par un "champ d'utilisation" susceptible d'être normalisé ou non. Les champs d'utilisation normalisés doivent faire l'objet d'un complément d'étude.

G.8.2.29 Confirmation d'indication d'utilisation

Le message de confirmation d'indication d'utilisation est envoyé en réponse à un message de d'indication d'utilisation, afin d'indiquer que le destinataire a accepté l'indication notifiée.

G.8.2.30 Rejet d'indication d'utilisation

Le message de rejet d'indication d'utilisation est envoyé en réponse à un message de d'indication d'utilisation, afin d'indiquer que le destinataire a rejeté l'indication notifiée et n'en tiendra pas compte.

Champ	Description
motif [<i>reason</i>]	Motif du rejet de la demande d'utilisation par l'élément frontière. Les valeurs possibles sont les suivantes: <ul style="list-style-type: none"> • appel non valide [<i>invalidCall</i>]; • sécurité [<i>security</i>]; • pas de relation de service [<i>noServiceRelationship</i>]; • non défini [<i>undefined</i>].

G.8.2.31 Demande de validation

Un élément frontière qui termine un appel peut envoyer un message de demande de validation à un autre élément frontière pour vérifier la validité de l'origine de l'appel.

Champ	Description
information de destination [<i>destinationInfo</i>]	Informations détaillées concernant la destination de l'appel.
informations d'origine [<i>sourceInfo</i>]	Informations concernant le point d'extrémité à l'origine de l'appel.
informations propres à l'appel [<i>callInfo</i>]	Ce champ identifie l'appel particulier qui fait l'objet d'une demande d'autorisation d'accès.
Spécification d'utilisation [<i>usageSpec</i>]	Lorsqu'il est présent ce champ indique à l'élément frontière adressant les demandes de message qu'il doit recevoir un message d'indication d'utilisation concernant l'appel validé.
jetons d'accès [<i>accessTokens</i>]	Jetons reçus du participant à l'origine de l'appel attestant l'autorisation d'accès relative à l'appel considéré.

G.8.2.32 Confirmation de validation

Message indiquant que l'appel est validé. L'élément frontière demandeur peut terminer l'appel. L'élément frontière qui effectue la validation peut indiquer des alias afin de terminer l'appel.

Champ	Description
information de destination [<i>destinationInfo</i>]	Autres paramètres relatifs à la destination à utiliser par l'élément frontière destinataire.
spécification d'utilisation [<i>usageSpec</i>]	Lorsqu'il est présent ce champ indique à l'élément frontière adressant les demandes de confirmation qu'il doit recevoir un message d'indication d'utilisation concernant l'appel validé.

G.8.2.33 Rejet de validation

Message indiquant que l'appel n'est pas valide. L'élément frontière demandeur peut ne pas établir l'appel.

Champ	Description
motif [<i>reason</i>]	<p>Motif du rejet de la demande. Les valeurs possibles sont les suivantes:</p> <ul style="list-style-type: none"> • jeton non valide [<i>tokenNotValid</i>] – Les jetons d'accès fournis ne sont pas valides pour l'appel en question; • sécurité [<i>security</i>] – La demande de validation ne répond pas aux contraintes de sécurité du destinataire; • dépassement du compteur de bonds [<i>hopCountExceeded</i>] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible; • information d'origine manquante [<i>missingSourceInfo</i>] – L'information d'origine fournie ne suffit pas pour valider l'appel; • information de destination manquante [<i>missingDestInfo</i>] – L'information de destination fournie ne suffit pas pour valider l'appel;

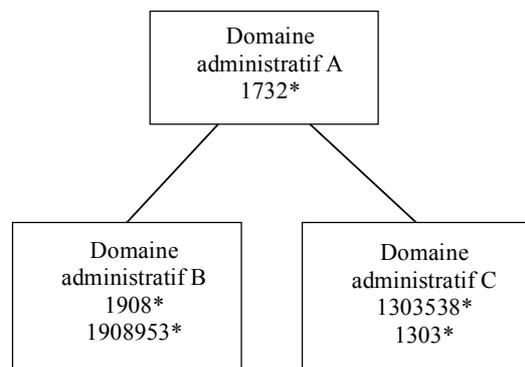
- pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service;
- non défini [*undefined*] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.

G.9 Exemples de signalisation

Les exemples de signalisation qui suivent illustrent le mode opératoire de base. On suppose que les domaines administratifs ont conclu des accords bilatéraux, de sorte que les éléments frontière ont reçu des informations mutuelles (concernant, par exemple, des ports TCP). Nombre des exemples ci-dessous font état de messages RAS LRQ/LCF qui sont échangés entre un portier et un élément frontière à l'intérieur du même domaine administratif. Ces exemples sont présentés à titre purement indicatif, puisque le protocole relatif au point de référence B n'a pas été déterminé (voir G.1).

G.9.1 Répartition ou maillage total

La Figure G.7 donne un exemple de réseau réparti.



T1606020-99

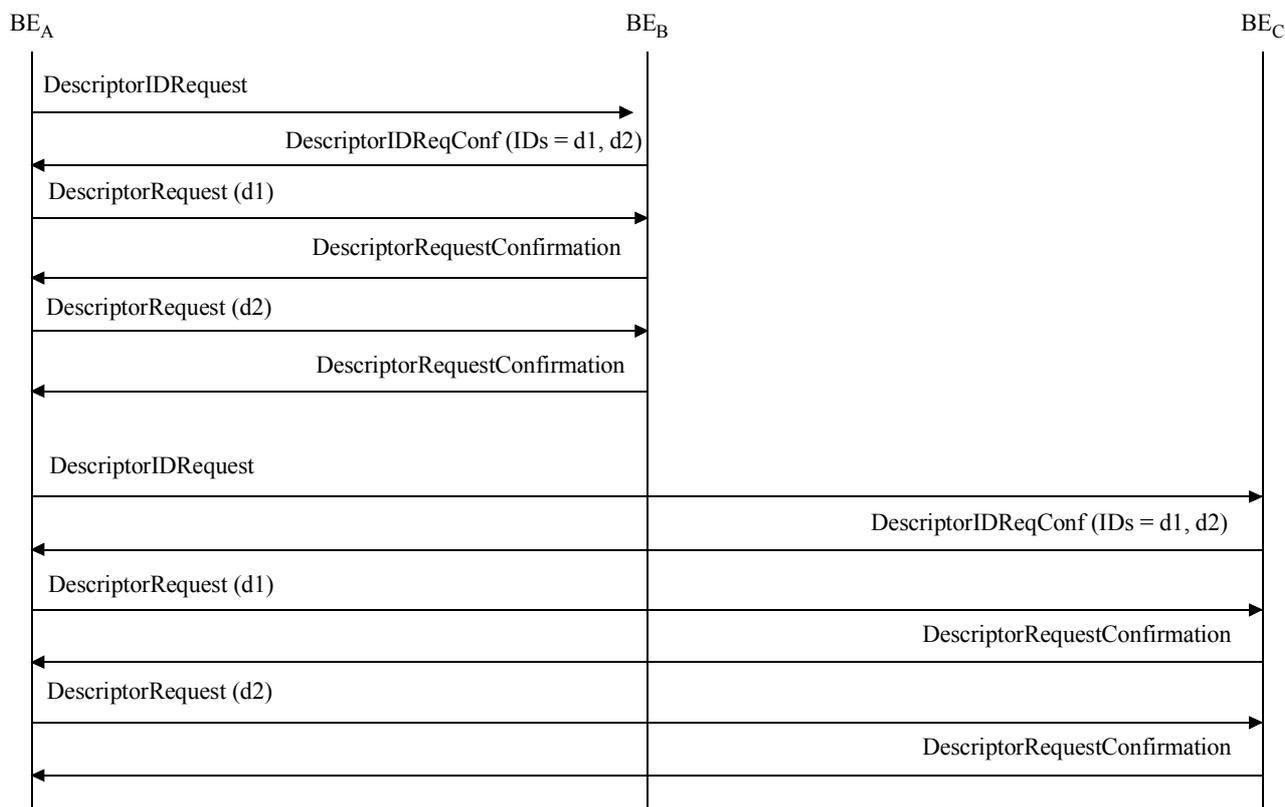
Figure G.7/H.225.0 – Réseau réparti pour les exemples de signalisation

On suppose, dans cet exemple, que les domaines administratifs possèdent chacun un élément frontière et que ces derniers sont configurés pour résoudre les adresses de la manière suivante:

Domaine administratif	Définitions de canevas	Commentaire
A	Descripteur "d1": modèle = 1732* adresse de transport = adresse de signal d'appel BE _A type de message = sendSetup	La signalisation pour tout appel dans le domaine administratif A se fera à travers l'élément frontière de ce domaine.
B	Descripteur "d1": modèle = 1908* adresse de transport = adresse annexe g BE _B type de message = sendAccessRequest Descripteur "d2": modèle = 1908953* adresse de transport = adresse de signalisation d'appel GW _{B1} type de message = sendSetup	Un message "demande d'accès" est nécessaire pour faire aboutir à leur adresse de signalisation d'appel de destination (c'est-à-dire, une passerelle) des messages concernant les numéros 1908*. Le message d'établissement peut être émis directement à destination de l'adresse particulière dans le cas d'appels concernant les numéros 1908953*.
C	Descripteur "d1": modèle = 1303538* adresse de transport = adresse de signal d'appel GK _{C1} type de message = sendSetup Descripteur "d2": modèle = 1303* adresse de transport = adresse annexe g BE _C type de message = sendAccessRequest	Les appels concernant les numéros 1303538*, seront acheminés par l'intermédiaire de ce portier particulier. Les appels concernant les numéros 1303* peuvent faire l'objet d'une signalisation directe vers la passerelle de destination, mais une demande d'accès doit être émise pour obtenir l'adresse de signalisation d'appel de la passerelle.

G.9.1.1 Echange d'informations de zone

L'organisation de chaque domaine administratif dans la structure répartie ou totalement maillée a connaissance de chacun des autres domaines administratifs, sans doute à la suite d'un certain nombre d'accords bilatéraux. Un élément frontière d'un domaine administratif peut demander à tout instant à un autre domaine administratif de lui fournir des informations d'adressage. La Figure G.8 présente un exemple de la signalisation correspondante.



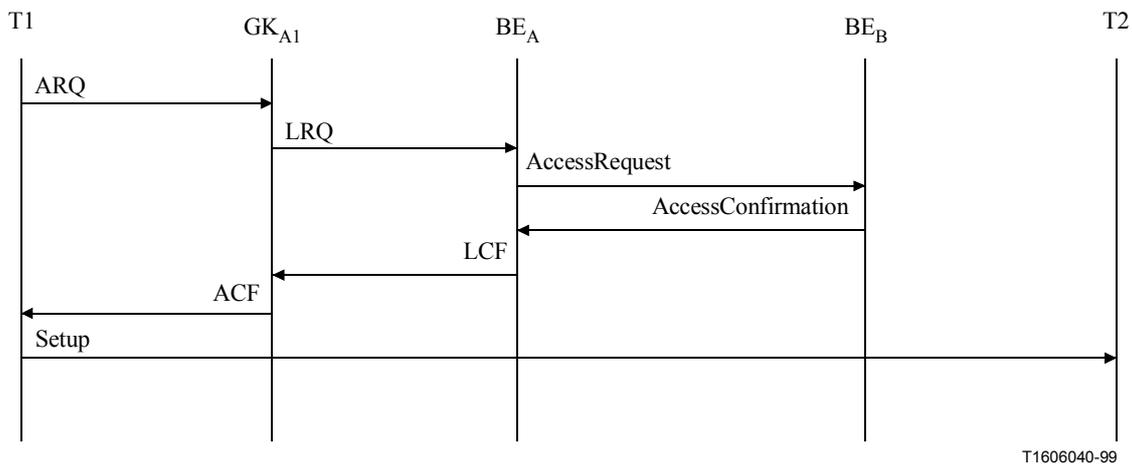
T1606030-99

Figure G.8/H.225.0 – Exemple d'échange de descripteurs

L'élément frontière BE_B interroge de même les éléments frontière BE_A et BE_C et l'élément frontière BE_C , les éléments frontière BE_A et BE_B .

G.9.1.2 Etablissement d'un appel

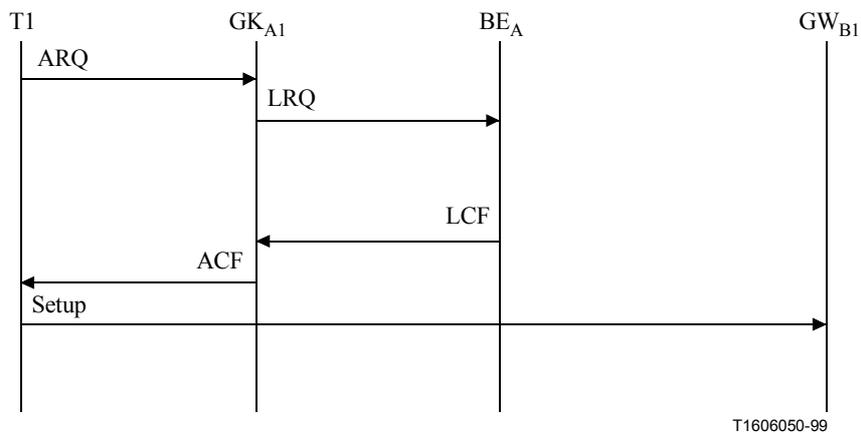
Supposons que le terminal T1 du domaine administratif A lance un appel vers le numéro 19085551515 (terminal T2). Le portier du terminal T1 émet un message LRQ lorsqu'il reçoit le message ARQ émis par le terminal T1. Un élément frontière BE_A du domaine administratif A a reçu au préalable des descripteurs de zone et connaît le traitement qui doit être appliqué à la requête. Comme indiqué dans la Figure G.9, l'élément frontière BE_A émet un message "demande d'accès" à destination de l'élément frontière BE_B qui est spécifié dans le descripteur BE_A reçu de l'élément frontière BE_B . Ce dernier renvoie une réponse contenant l'adresse de signalisation d'appel du terminal T2 (le terminal T2 de cet exemple peut être tout type de point de terminaison). Le terminal T1 émet ensuite le message SETUP (établissement) H.225.0 à destination de l'adresse de signalisation d'appel du terminal T2 en appliquant les procédures normales définies par l'UIT-T H.323 ou ses annexes.



T1606040-99

Figure G.9/H.225.0

Supposons maintenant que le terminal T1 lance un appel vers le numéro 19089532000. L'élément frontière BE_A de cet exemple a obtenu au préalable l'adresse de signalisation d'appel d'une passerelle située dans un domaine administratif qui acceptera l'appel. Comme indiqué dans la Figure G.10, l'élément frontière BE_A peut répondre au message LRQ sans aucun échange de message dans le domaine administratif B, ce qui permet au terminal T1 d'émettre le message SETUP directement à destination de la passerelle.



T1606050-99

Figure G.10/H.225.0

Supposons que, dans un autre exemple, le terminal T1 lance un appel vers le numéro 13035382899. Le domaine administratif C a publié sa capacité de prise en charge d'un appel à destination de ce numéro et acceptera la signalisation d'appel par le biais de son portier en implémentant le modèle de routage par portier. L'élément frontière BE_A peut répondre au message LRQ, comme indiqué dans la Figure G.11, au moyen d'un message LCF qui contient l'adresse de signalisation d'appel appartenant au domaine administratif C sans aucun échange de message au sein de ce domaine.

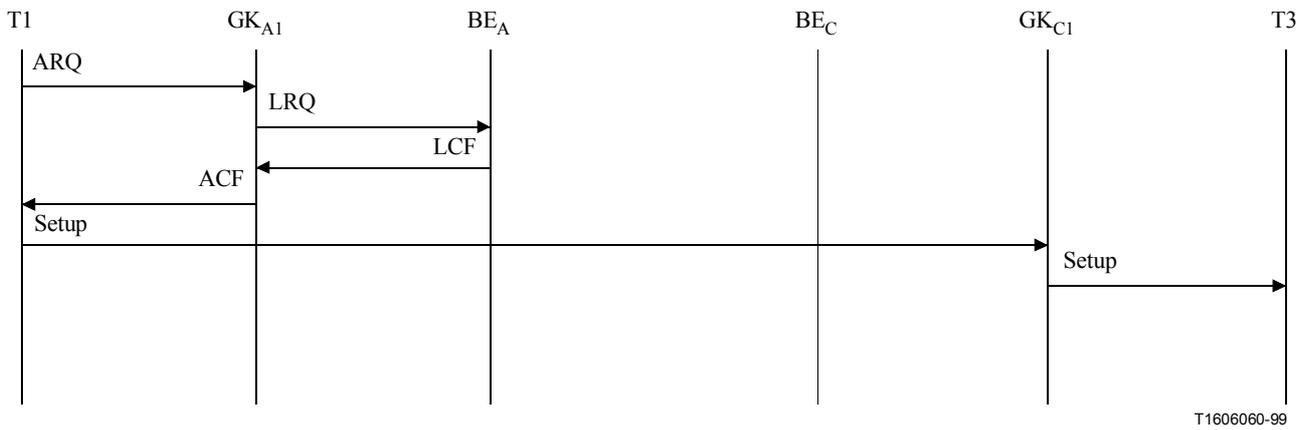


Figure G.11/H.225.0

Le portier du terminal T1 peut, en variante, implémenter le modèle avec routage par portier, comme indiqué dans la Figure G.12.

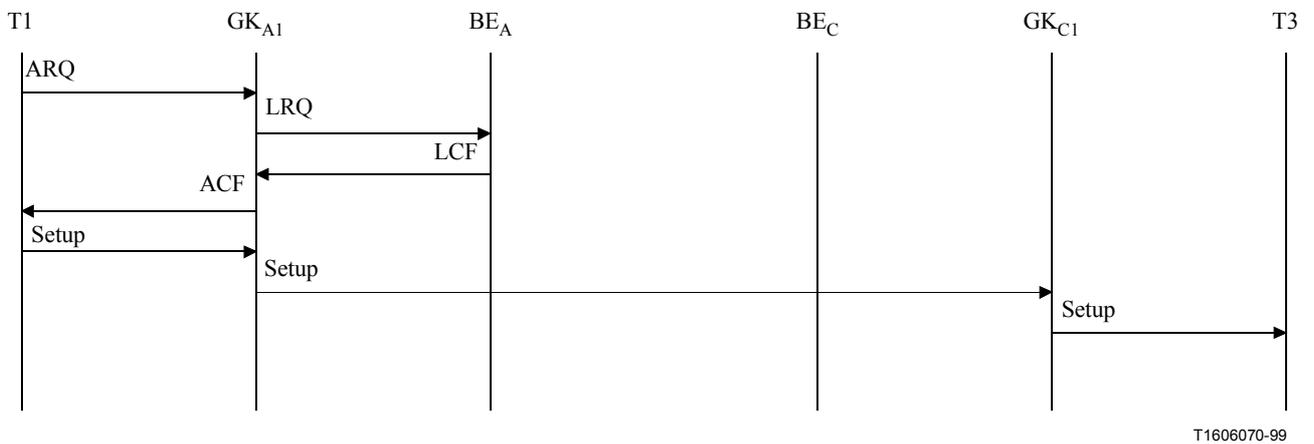
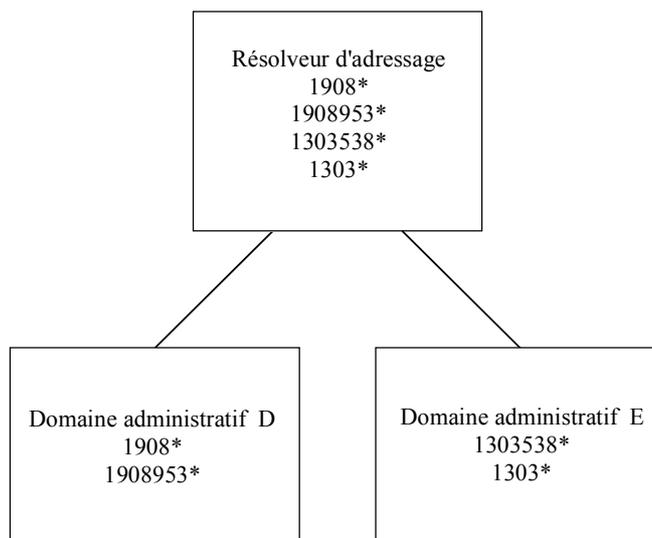


Figure G.12/H.225.0

G.9.2 Résolveur d'adressage

La Figure G.13 présente un exemple qui utilise un résolveur d'adressage. Cette figure sert de référence pour les exemples qui suivent. Le résolveur d'adressage conserve des informations d'adressage pour tous les domaines administratifs auxquels elle fournit des services.



T1606080-99

Figure G.13/H.225.0 – Exemple de configuration avec résolveur d'adressage

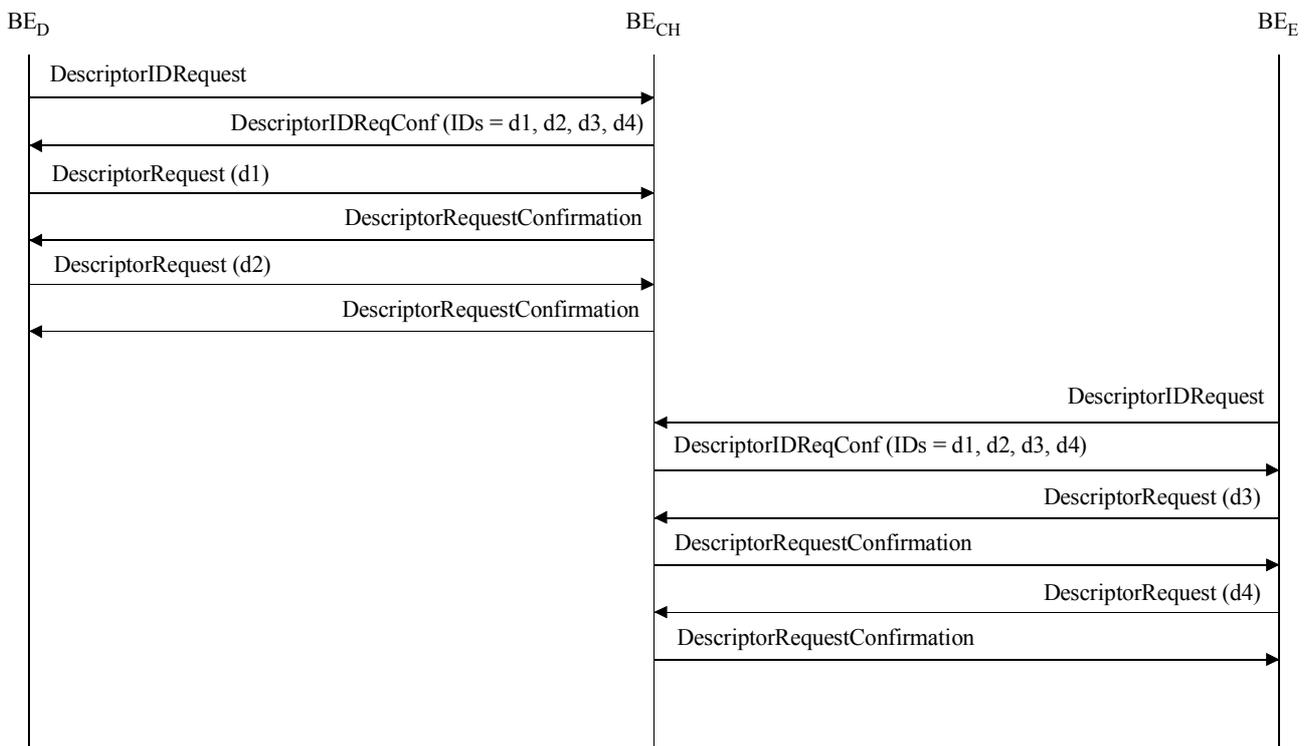
Les éléments frontière des domaines administratifs D et E et le résolveur d'adressage détiennent les informations suivantes dans l'exemple ci-dessous:

Domaine administratif	Définitions de canevas	Commentaire
D	Descripteur "d1": modèle = 1908* adresse de transport = adresse annexe g BE _D type de message = sendAccessRequest Descripteur "d2": modèle = 1908953* adresse de transport = adresse de signalisation d'appel GW _{D1} type de message = sendSetup	Pour les appels concernant les numéros 1908*, un message "demande d'accès" est nécessaire pour obtenir l'adresse de signalisation d'appel de destination (c'est-à-dire, une passerelle). Les appels concernant les numéros 1908953* seront acheminés par l'intermédiaire de cette passerelle particulière.
E	Descripteur "d1": modèle = 1303538* adresse de transport = adresse de signal d'appel GK _{E1} type de message = sendSetup Descripteur "d2": modèle = 1303* adresse de transport = adresse annexe g BE _E type de message = sendAccessRequest	Les appels concernant les numéros 1303538* seront acheminés par l'intermédiaire de ce portier particulier. Les appels concernant les numéros 1303* peuvent faire l'objet d'une signalisation directe vers la passerelle de destination, mais une demande d'accès doit être émise pour obtenir l'adresse de signalisation d'appel de la passerelle.

Domaine administratif	Définitions de canevas	Commentaire
CH	Descripteur "d1": modèle = 1908* adresse de transport = BE _D adresse annexe g type de message = sendAccessRequest Descripteur "d2": modèle = 1908953* adresse de transport = adresse de signalisation d'appel GW _{D1} type de message = sendSetup Descripteur "d3": modèle = 1303538* adresse de transport = adresse de signal d'appel GK _{E1} type de message = sendSetup Descripteur "d4": modèle = 1303* adresse de transport = adresse annexe g BE _E type de message = sendAccessRequest	Le résolveur d'adressage obtient des descripteurs d'autres domaines administratifs et conserve ces informations durant l'échange de descripteurs.

G.9.2.1 Echange d'informations de zone

Un résolveur d'adressage échange dans cet exemple des informations avec des domaines administratifs qui se sont abonnés à ses services. Le résolveur d'adressage conserve les informations qu'elle reçoit de chaque domaine administratif et les relaye à destination d'autres domaines administratifs. Le résolveur d'adressage est vue par le domaine administratif D comme le domaine administratif E, mais les domaines administratifs D et E n'ont pas nécessairement connaissance de leur existence mutuelle. Voir Figure G.14.



T1606090-99

Figure G.14/H.225.0 – Exemple d'échange de descripteur avec un résolveur d'adressage

G.9.2.2 Etablissement d'un appel

Supposons que le terminal T1 du domaine administratif E lance un appel vers le numéro 19085551515. L'élément frontière du domaine administratif E a reçu du résolveur d'adressage des descripteurs indiquant que cette dernière doit être consultée pour un tel appel. L'élément émet une demande d'accès à destination de l'élément frontière du résolveur d'adressage. Les descripteurs que l'élément frontière du résolveur d'adressage a reçu de l'élément frontière du domaine administratif D lui permettent d'émettre une demande d'accès à destination de l'élément frontière du domaine administratif D. Lorsque l'élément frontière du résolveur d'adressage renvoie la confirmation à destination de l'élément frontière du domaine administratif E, cette confirmation contient les informations émises par l'élément frontière du domaine administratif D. Le portier du terminal T1 renvoie un message ACF contenant l'adresse de signalisation d'appel du terminal T1 qui doit être utilisée pour émettre le message SETUP à destination du terminal T2. Voir Figure G.15.

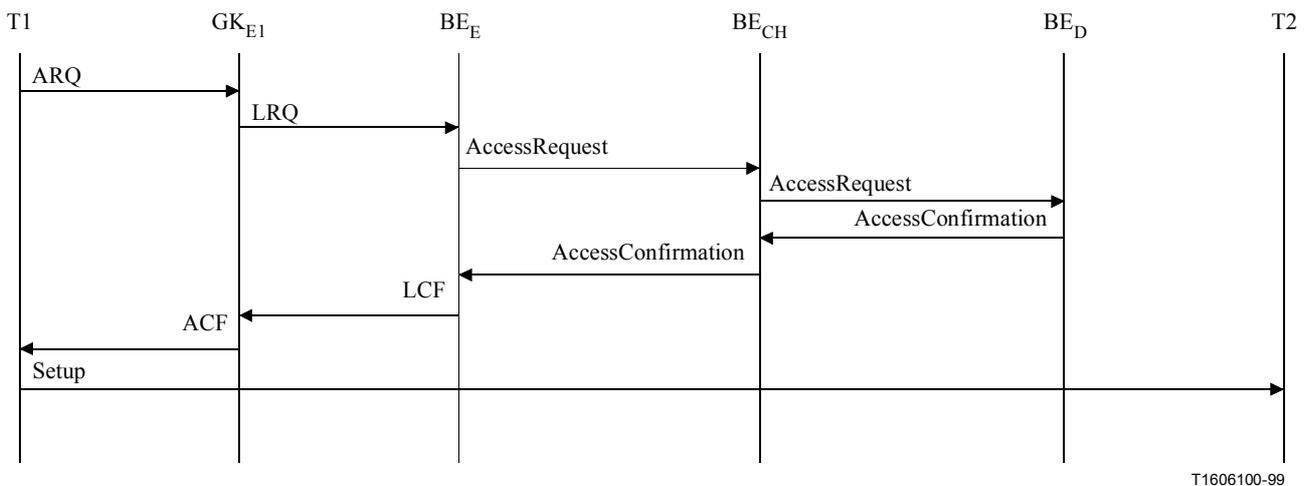


Figure G.15/H.225.0

Le portier du terminal T1 peut, en variante, router la signalisation d'appel comme indiqué dans la Figure G.16.

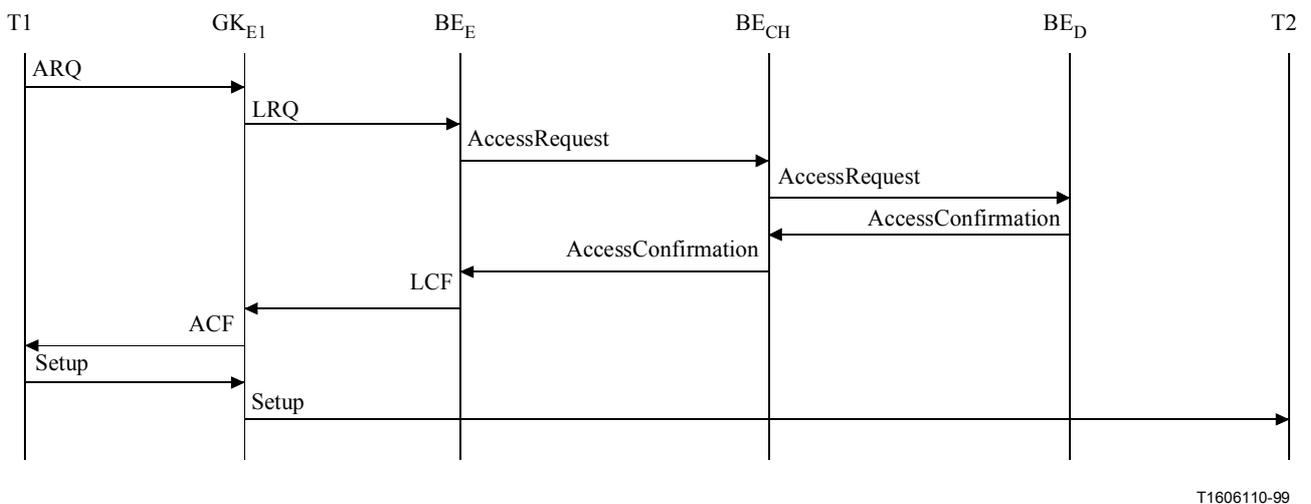


Figure G.16/H.225.0

Une autre possibilité pour le résolveur d'adressage consiste à répondre à l'élément frontière du domaine administratif E en lui fournissant les informations de contact pour l'élément frontière du domaine administratif D, comme indiqué dans la Figure G.17.

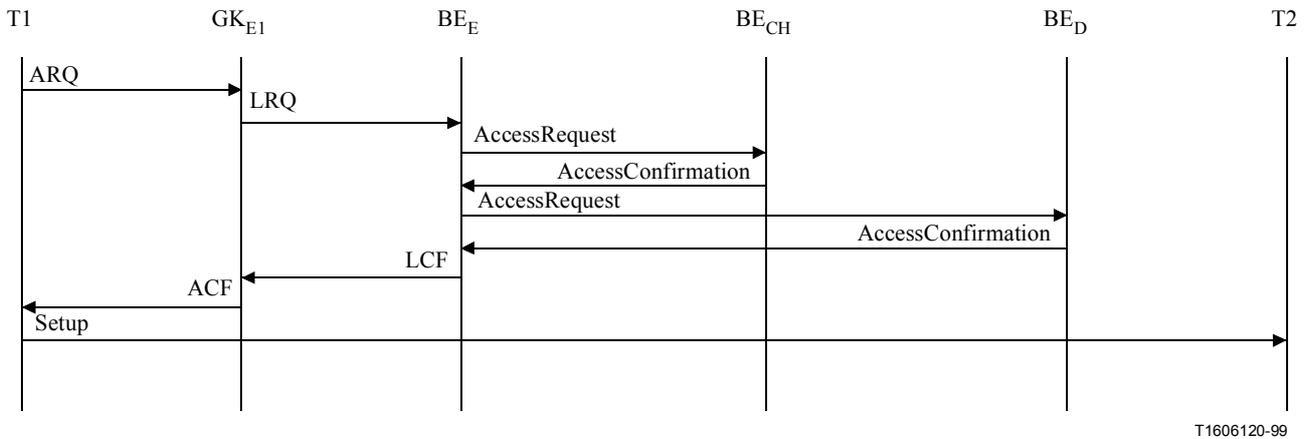


Figure G.17/H.225.0

Supposons maintenant que le terminal T1 lance un appel vers le numéro 19089532000. Les descripteurs échangés au préalable permettent à l'élément frontière de renvoyer au terminal T1 l'adresse de signalisation d'appel sans consulter le résolveur d'adressage, comme indiqué dans la Figure G.18.

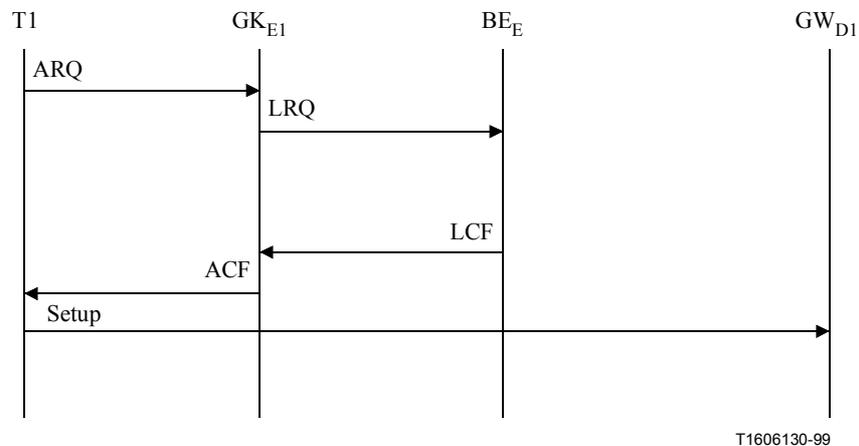


Figure G.18/H.225.0

Examinons maintenant un scénario dans lequel le terminal T1 lance un appel vers le numéro 13035382899. L'élément frontière du domaine administratif E a publié au préalable le fait que les appels à destination des numéros 1303538* peuvent être routés directement vers un portier dans le domaine administratif E sans utiliser un message ACCESS REQUEST (demande d'accès), comme indiqué dans la Figure G.19. (La publication n'indique pas que l'entité est un portier, mais uniquement qu'un message SETUP peut être émis à destination d'une adresse spécifiée.) L'élément frontière du domaine administratif D a reçu ces informations du résolveur d'adressage, si nous admettons que cette dernière n'a pas l'obligation, dans cet exemple, de fournir la résolution d'adresse pour ces appels.

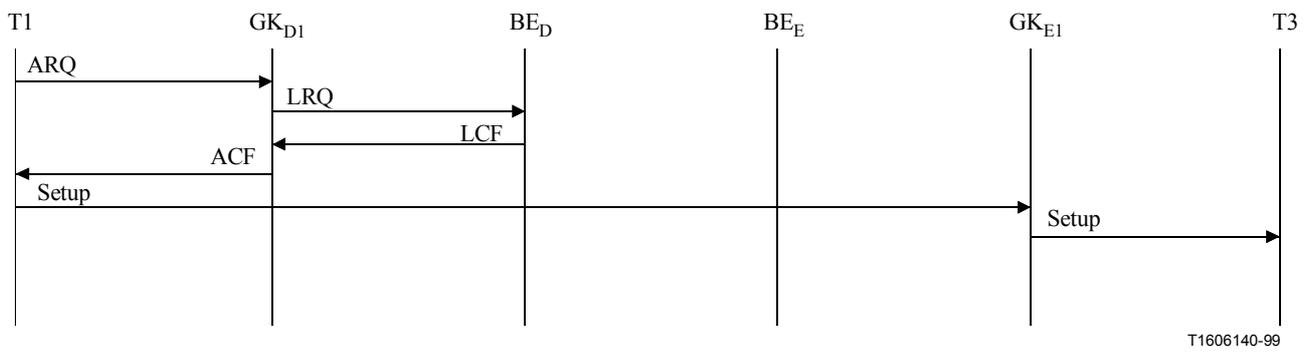


Figure G.19/H.225.0

Rappelons qu'un élément frontière peut être combiné avec un portier et peut également router des appels dans le modèle avec routage par portier. La Figure G.20 présente une variante d'exemple de signalisation. Il est également possible d'utiliser l'élément frontière comme un portier effectuant le routage dans un domaine si les descripteurs sont configurés en conséquence.

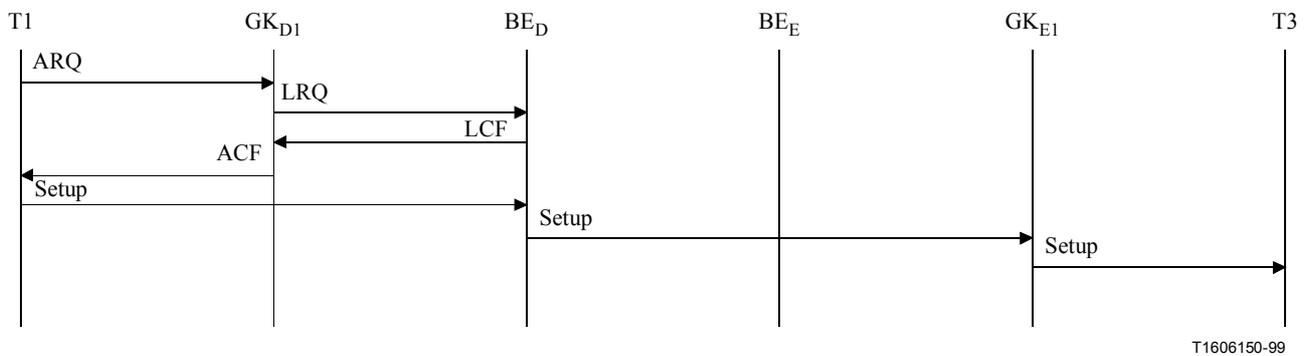
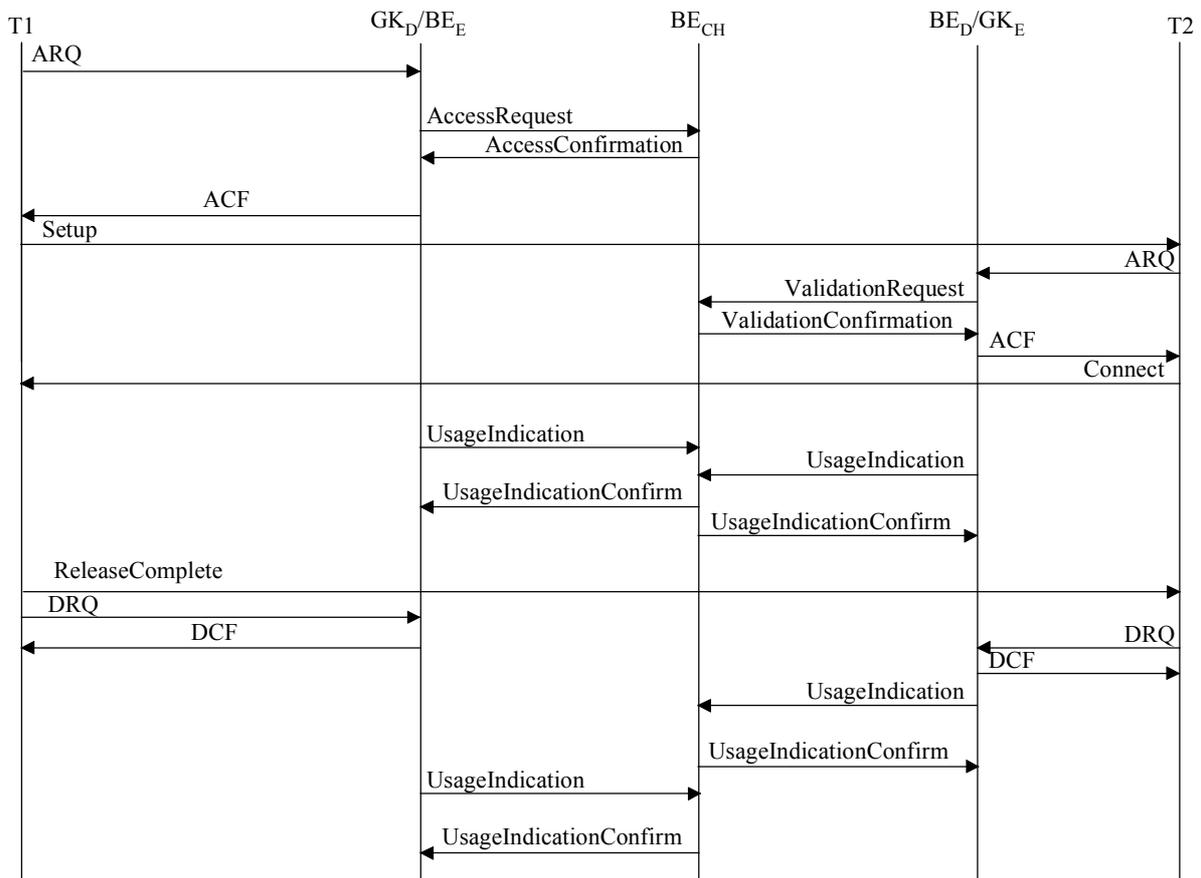


Figure G.20/H.225.0

Dans l'exemple de la Figure G.21, le résolveur d'adressage valide l'appel pour le domaine administratif de destination. Le résolveur d'adressage demande en outre aux éléments frontière d'origine et de destination d'envoyer des indications d'utilisation concernant l'appel.



T1607750-00

Figure G.21/H.225.0

Syntaxe des messages

ANNEXG-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

AuthenticationMechanism,
 TimeStamp,
 ClearToken
 FROM H235-SECURITY-MESSAGES

AliasAddress,
 TransportAddress,
 ReleaseCompleteReason,
 ConferenceIdentifier, CallIdentifier, CryptoH323Token, CryptoToken,

EndpointType,
 GatekeeperIdentifier,
 GloballyUniqueID,
 NonStandardParameter,
 NumberDigits,
 PartyNumber,
 TransportQOS,
 VendorIdentifier,
 IntegrityMechanism,
 ICV
 FROM H323-MESSAGES;

```

Message ::= SEQUENCE
{
    body AnnexGMessageBody,
    common AnnexGCommonInfo,
    ...
}

AnnexGMessageBody ::= CHOICE
{
    serviceRequest          ServiceRequest,
    serviceConfirmation     ServiceConfirmation,
    serviceRejection        ServiceRejection,
    serviceRelease          ServiceRelease,
    descriptorRequest       DescriptorRequest,
    descriptorConfirmation  DescriptorConfirmation,
    descriptorRejection     DescriptorRejection,
    descriptorIDRequest     DescriptorIDRequest,
    descriptorIDConfirmation DescriptorIDConfirmation,
    descriptorIDRejection   DescriptorIDRejection,
    descriptorUpdate        DescriptorUpdate,
    descriptorUpdateAck     DescriptorUpdateAck,
    accessRequest           AccessRequest,
    accessConfirmation      AccessConfirmation,
    accessRejection         AccessRejection,
    requestInProgress       RequestInProgress,
    nonStandardRequest      NonStandardRequest,
    nonStandardConfirmation NonStandardConfirmation,
    nonStandardRejection    NonStandardRejection,
    unknownMessageResponse UnknownMessageResponse,
    usageRequest            UsageRequest,
    usageConfirmation       UsageConfirmation,
    usageIndication         UsageIndication,
    usageIndicationConfirmation UsageIndicationConfirmation,
    usageIndicationRejection UsageIndicationRejection,
    usageRejection          UsageRejection,
    validationRequest       ValidationRequest,
    validationConfirmation  ValidationConfirmation,
    validationRejection     ValidationRejection,
    ...
}

AnnexGCommonInfo ::= SEQUENCE
{
    sequenceNumber          INTEGER (0..65535),
    version                 AnnexGVersion,
    hopCount                INTEGER (1..255),
    replyAddress            SEQUENCE OF TransportAddress OPTIONAL,
    -- doit être présent dans la demande
    integrityCheckValue     ICV OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    nonStandard             SEQUENCE OF NonStandardParameter OPTIONAL,
    ...
}

--
-- messages de l'Annexe G
--

```

```

ServiceRequest ::= SEQUENCE
{
  elementIdentifier      ElementIdentifier OPTIONAL,
  domainIdentifier       AliasAddress OPTIONAL,
  securityMode           SEQUENCE OF SecurityMode OPTIONAL,
  timeToLive            INTEGER (1..4294967295) OPTIONAL,
  ...
}

SecurityMode ::= SEQUENCE
{
  authentication        AuthenticationMechanism OPTIONAL,
  integrity              IntegrityMechanism OPTIONAL,
  algorithmOIDs         SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
  ...
}

ServiceConfirmation ::= SEQUENCE
{
  elementIdentifier      ElementIdentifier,
  domainIdentifier       AliasAddress,
  alternates            AlternateBEInfo OPTIONAL,
  securityMode          SecurityMode OPTIONAL,
  timeToLive            INTEGER (1..4294967295) OPTIONAL,
  ...
}

ServiceRejection ::= SEQUENCE
{
  reason                ServiceRejectionReason,
  alternates            AlternateBEInfo OPTIONAL,
  ...
}

ServiceRejectionReason ::= CHOICE
{
  serviceUnavailable    NULL,
  serviceRedirected     NULL,
  security              NULL,
  continue              NULL,
  undefined             NULL,
  ...
}

ServiceRelease ::= SEQUENCE
{
  reason                ServiceReleaseReason,
  alternates            AlternateBEInfo OPTIONAL,
  ...
}

ServiceReleaseReason ::= CHOICE
{
  outOfService          NULL,
  maintenance           NULL,
  terminated            NULL,
  expired              NULL,
  ...
}

```

```

DescriptorRequest ::= SEQUENCE
{
    descriptorID      SEQUENCE OF DescriptorID,
    ...
}

DescriptorConfirmation ::= SEQUENCE
{
    descriptor        SEQUENCE OF Descriptor,
    ...
}

DescriptorRejection ::= SEQUENCE
{
    reason            DescriptorRejectionReason,
    descriptorID      DescriptorID OPTIONAL,
    ...
}

DescriptorRejectionReason ::= CHOICE
{
    packetSizeExceeded  NULL,    -- utiliser un autre type de transport
    illegalID           NULL,    -- pas de descripteur pour l'identificateur
                                -- fourni
    security             NULL,    -- la demande n'a pas satisfait aux
                                -- prescriptions de sécurité
    hopCountExceeded   NULL,
    noServiceRelationship  NULL,
    undefined           NULL,
    ...
}

DescriptorIDRequest ::= SEQUENCE
{
    ...
}

DescriptorIDConfirmation ::= SEQUENCE
{
    descriptorInfo     SEQUENCE OF DescriptorInfo,
    ...
}

DescriptorIDRejection ::= SEQUENCE
{
    reason             DescriptorIDRejectionReason,
    ...
}

DescriptorIDRejectionReason ::= CHOICE
{
    noDescriptors      NULL,    -- pas de descripteurs pour le compte rendu
    security           NULL,    -- la demande n'a pas satisfait aux
                                -- prescriptions de sécurité
    hopCountExceeded   NULL,
    noServiceRelationship  NULL,
    undefined           NULL,
}

```

```
} ...
```

```
DescriptorUpdate ::= SEQUENCE  
{  
    sender          AliasAddress,  
    updateInfo      SEQUENCE OF UpdateInformation,  
    ...  
}
```

```
UpdateInformation ::= SEQUENCE  
{  
    descriptorInfo  CHOICE {  
        descriptorID  DescriptorID,  
        descriptor    Descriptor,  
        ...  
    },  
    updateType      CHOICE  
    {  
        added         NULL,  
        deleted       NULL,  
        changed       NULL,  
        ...  
    },  
    ...  
}
```

```
DescriptorUpdateAck ::= SEQUENCE  
{  
    ...  
}
```

```
AccessRequest ::= SEQUENCE  
{  
    destinationInfo PartyInformation,  
    sourceInfo       PartyInformation OPTIONAL,  
    callInfo         CallInformation OPTIONAL,  
    usageSpec        UsageSpecification OPTIONAL, ...  
}
```

```
AccessConfirmation ::= SEQUENCE  
{  
    templates        SEQUENCE OF AddressTemplate,  
    partialResponse  BOOLEAN,  
    ...  
}
```

```
AccessRejection ::= SEQUENCE  
{  
    reason           AccessRejectionReason,  
    ...  
}
```

```
AccessRejectionReason ::= CHOICE  
{  
    noMatch          NULL,      -- aucun canevas ne convient pour  
                                -- l'information de destination  
    packetSizeExceeded NULL,    -- utiliser un autre type de transport
```

```

security          NULL,      -- la demande n'a pas satisfait aux
                        -- prescriptions de sécurité
hopCountExceeded  NULL,
needCallInformation  NULL,    -- l'information d'appel
                        -- doit être spécifiée
noServiceRelationship  NULL,
undefined          NULL,
...
}

UsageRequest ::= SEQUENCE
{
    callInfo          CallInformation,
    usageSpec         UsageSpecification,
    ...
}

UsageConfirmation ::= SEQUENCE
{
    ...
}

UsageRejection ::= SEQUENCE
{
    reason            UsageRejectReason,
    ...
}

UsageIndication ::= SEQUENCE
{
    callInfo          CallInformation,
    accessTokens      SEQUENCE OF AccessToken OPTIONAL,
    senderRole        Role,
    usageCallStatus   UsageCallStatus,
    srcInfo           PartyInformation OPTIONAL,
    destAddress       PartyInformation,
    startTime         TimeStamp OPTIONAL,
    endTime           TimeStamp OPTIONAL,
    terminationCause  TerminationCause OPTIONAL,
    usageFields       SEQUENCE OF UsageField,
    ...
}

UsageField ::= SEQUENCE
{
    id                OBJECT IDENTIFIER,
    value             OCTET STRING,
    ...
}

UsageRejectReason ::= CHOICE
{
    invalidCall       NULL,
    unavailable        NULL,
    security           NULL,
    noServiceRelationship  NULL,
    undefined         NULL,
    ...
}

```

```

UsageIndicationConfirmation ::= SEQUENCE
{
    ...
}

UsageIndicationRejection ::= SEQUENCE
{
    reason          UsageIndicationRejectionReason,
    ...
}

UsageIndicationRejectionReason ::= CHOICE
{
    unknownCall          NULL,
    incomplete           NULL,
    security             NULL,
    noServiceRelationship NULL,
    undefined            NULL,
    ...
}

ValidationRequest ::= SEQUENCE
{
    accessToken          SEQUENCE OF AccessToken OPTIONAL,
    destinationInfo     PartyInformation OPTIONAL,
    sourceInfo          PartyInformation OPTIONAL,
    callInfo            CallInformation,
    usageSpec           UsageSpecification OPTIONAL,
    ...
}

ValidationConfirmation ::= SEQUENCE
{
    destinationInfo     PartyInformation OPTIONAL,
    usageSpec           UsageSpecification OPTIONAL,
    ...
}

ValidationRejection ::= SEQUENCE
{
    reason          ValidationRejectionReason,
    ...
}

ValidationRejectionReason ::= CHOICE
{
    tokenNotValid          NULL,
    security              NULL,          -- la demande n'a pas satisfait aux
                                     -- prescriptions de sécurité

    hopCountExceeded      NULL,
    missingSorceInfo      NULL,
    missingDestInfo       NULL,
    noServiceRelationship NULL,
    undefined             NULL,
    ...
}

RequestInProgress ::= SEQUENCE
{
    delay          INTEGER (1..65535),
    ...
}

```

```

NonStandardRequest ::= SEQUENCE
{
    ...
}

NonStandardConfirmation ::= SEQUENCE
{
    ...
}

NonStandardRejection ::= SEQUENCE
{
    reason          NonStandardRejectionReason,
    ...
}

NonStandardRejectionReason ::= CHOICE
{
    notSupported          NULL,
    noServiceRelationship NULL,
    undefined             NULL,
    ...
}

UnknownMessageResponse ::= SEQUENCE
{
    unknownMessage  OCTET STRING,
    reason          UnknownMessageReason,
    ...
}

UnknownMessageReason ::= CHOICE
{
    notUnderstood     NULL,
    undefined         NULL,
    ...
}

--
-- structures communes à plusieurs messages
--

AddressTemplate ::= SEQUENCE
{
    pattern          SEQUENCE OF Pattern,
    routeInfo       SEQUENCE OF RouteInformation,
    timeToLive      INTEGER (1..4294967295),
    ...
}

```

```

Pattern ::= CHOICE
{
    specific      AliasAddress,
    wildcard      AliasAddress,
    range         SEQUENCE {
        startOfRange PartyNumber,
        endOfRange   PartyNumber
    },
    ...
}

RouteInformation ::= SEQUENCE
{
    messageType CHOICE
    {
        sendAccessRequest NULL,
        sendSetup          NULL,
        nonExistent        NULL,
        ...
    },
    callSpecific  BOOLEAN,
    usageSpec     UsageSpecification OPTIONAL,
    priceInfo     SEQUENCE OF PriceInfoSpec OPTIONAL,
    contacts      SEQUENCE OF ContactInformation,
    type          EndpointType OPTIONAL,
                -- doit être présent si messageType = sendSetup
    ...}

ContactInformation ::= SEQUENCE{ transportAddress  AliasAddress,      priority
                                INTEGER (0..127), transportQoS TransportQoS OPTIONAL,
                                security          SEQUENCE OF SecurityMode OPTIONAL,
                                accessTokens      SEQUENCE OF AccessToken OPTIONAL,
                                ...
}

PriceInfoSpec ::= SEQUENCE
{
    currency      IA5String (SIZE(3)),          -- par exemple "USD"
    currencyScale INTEGER(-127..127),
    validFrom     GlobalTimeStamp OPTIONAL,
    validUntil    GlobalTimeStamp OPTIONAL,
    hoursFrom     IA5String (SIZE(6)) OPTIONAL, -- heure UTC "HHMMSS"
    hoursUntil    IA5String (SIZE(6)) OPTIONAL, -- heure UTC "HHMMSS"
    priceElement  SEQUENCE OF PriceElement OPTIONAL,
    priceFormula  IA5String (SIZE(1..2048)) OPTIONAL,
    ...
}

PriceElement ::= SEQUENCE
{
    amount        INTEGER(0..4294967295),      -- incrément de compteur
    quantum       INTEGER(0..4294967295),      -- pour chaque quantum ou
                                                -- fraction de quantum

    units CHOICE
    {
        seconds    NULL,
        packets    NULL,
        bytes      NULL,
        initial     NULL,
        minimum     NULL,
        maximum     NULL,
        ...
    },
    ...
}

```

```

Descriptor ::= SEQUENCE
{
    descriptorInfo      DescriptorInfo,
    templates           SEQUENCE OF AddressTemplate,
    gatekeeperID        GatekeeperIdentifier OPTIONAL,
    ...
}

DescriptorInfo ::= SEQUENCE
{
    descriptorID        DescriptorID,
    lastChanged         GlobalTimeStamp,
    ...
}

AlternateBEInfo ::= SEQUENCE
{
    alternateBE          SEQUENCE OF AlternateBE,
    alternateIsPermanent BOOLEAN,
    ...
}

AlternateBE ::= SEQUENCE
{
    contactAddress      AliasAddress,
    priority             INTEGER (1..127),
    elementIdentifier    ElementIdentifier OPTIONAL,
    ...
}

AccessToken ::= CHOICE
{
    token               ClearToken,
    cryptoToken         CryptoH323Token,
    ...
}

CallInformation ::= SEQUENCE
{
    callIdentifier      CallIdentifier,
    conferenceID        ConferenceIdentifier,
    ...
}

UsageCallStatus ::= CHOICE
{
    preConnect          NULL,           -- la communication n'a pas commencé
    callInProgress      NULL,           -- Appel en cours
    callEnded           NULL,           -- Fin d'appel
    ...
}

UserInformation ::= SEQUENCE
{
    userIdentifier      AliasAddress,
    userAuthenticator   SEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}

```

```

UsageSpecification ::= SEQUENCE
{
    sendTo          ElementIdentifier,
    when SEQUENCE
    {
        never       NULL OPTIONAL,
        start       NULL OPTIONAL,
        end         NULL OPTIONAL,
        period      INTEGER(1..65535) OPTIONAL, -- en secondes
        failures    NULL OPTIONAL,
        ...
    },
    required       SEQUENCE OF OBJECT IDENTIFIER,
    preferred      SEQUENCE OF OBJECT IDENTIFIER,
    ...
}

PartyInformation ::= SEQUENCE
{
    logicalAddresses  SEQUENCE OF AliasAddress,
    domainIdentifier  AliasAddress OPTIONAL,
    transportAddress  AliasAddress OPTIONAL,
    endpointType      EndpointType OPTIONAL,
    userInfo          UserInformation OPTIONAL,
    timeZone          TimeZone OPTIONAL,
    ...
}

Role ::= CHOICE
{
    originator       NULL,
    destination      NULL,
    nonStandardData  NonStandardParameter,
    ...
}

TimeZone ::= INTEGER (-43200..43200)
-- nombre de secondes relatives au
-- temps UTC y compris le DST le cas échéant

TerminationCause ::= SEQUENCE
{
    releaseCompleteReason  ReleaseCompleteReason,
    causeIE                 INTEGER (1..65535) OPTIONAL,
    nonStandardData         NonStandardParameter OPTIONAL,
    ...
}

AnnexGVersion ::= OBJECT IDENTIFIER
-- doit être positionné sur la valeur
-- {itu-t (0) recommendation (0) h(8) h225.0(2250)
-- Annex (1) G (7) version (0) 1 (0)}

DescriptorID ::= GloballyUniqueID

ElementIdentifier ::= BMPString (SIZE(1..128))

```

```

GlobalTimeStamp ::= IA5String (SIZE(14)) -- sous la forme YYYYMMDDHHmmSS
-- avec YYYY = année, MM = mois, DD = jour,
-- HH = heure, mm = minute, SS = seconde
-- (par exemple, 19981219120000 pour minuit
-- 19 décembre 1998)

```

```

END -- des messages de l'Annexe G

```

ANNEXE H

Syntaxe des messages H.225.0 (ASN.1)

La présente Recommandation définit les protocoles pour les messages RAS (essentiellement un protocole de portier) et pour la signalisation d'appel (essentiellement des unités de données protocolaires qui résident dans un élément d'information d'utilisateur-utilisateur). Ces protocoles sont définis ensemble dans l'arbre ASN.1 suivant. La définition sémantique des messages et des divers éléments figure dans des paragraphes ultérieurs.

```

H323-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

```

```

IMPORTS

```

```

    SIGNED{ },
    ENCRYPTED{ },
    HASHED{ },
    ChallengeString,
    TimeStamp,
    RandomVal,
    Password,
    EncodedPwdCertToken,
    ClearToken,
    CryptoToken,
    AuthenticationMechanism
FROM H235-SECURITY-MESSAGES
    DataProtocolCapability,
    T38FaxProfile
FROM MULTIMEDIA-SYSTEM-CONTROL
    PackagesDescriptor,
    SignalsDescriptor
FROM MEDIA-GATEWAY-CONTROL;

```

```

H323-UserInformation ::= SEQUENCE -- racine pour tout l'ASN.1
-- associé à l'UIT-T Q.931
{
    h323-uu-pdu    H323-UU-PDU,
    user-data     SEQUENCE
    {
        protocol-discriminator    INTEGER (0..255),
        user-information           OCTET STRING (SIZE(1..131)),
        ...
    } OPTIONAL,
    ...
}

```

```

H323-UU-PDU ::= SEQUENCE
{
    h323-message-body    CHOICE
    {
        setup                Setup-UUIE,
        callProceeding      CallProceeding-UUIE,
        connect              Connect-UUIE,
        alerting             Alerting-UUIE,
        information          Information-UUIE,
        releaseComplete      ReleaseComplete-UUIE,
        facility              Facility-UUIE,
        ...,
        progress             Progress-UUIE,
        empty                 NULL,      -- utilisé lorsqu'un message Facility est
                                        -- envoyé, mais l'élément Facility-UUIE
                                        -- ne doit pas être invoqué
                                        -- (possible en cas de transport
                                        -- de messages de services complémentaires
                                        -- dans les versions antérieures à
                                        -- la présente version 4/H.225.0)

        status                Status-UUIE,
        statusInquiry        StatusInquiry-UUIE,
        setupAcknowledge      SetupAcknowledge-UUIE,
        notify                Notify-UUIE
    },
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    h4501SupplementaryService SEQUENCE OF OCTET STRING OPTIONAL,
                                        -- chaque séquence de chaîne d'octets est définie
                                        -- comme une unité APDU H4501SupplementaryService
                                        -- telle que définie dans le Tableau 3/H.450.1

    h245Tunneling          BOOLEAN,
                                        -- si la valeur est Vrai, l'encapsulation des
                                        -- messages H.245 est activée.

    h245Control            SEQUENCE OF OCTET STRING OPTIONAL,
    nonStandardControl      SEQUENCE OF NonStandardParameter OPTIONAL,
    callLinkage             CallLinkage OPTIONAL,
    tunnelledSignallingMessage SEQUENCE
    {
        tunnelledProtocolID  TunnelledProtocol, -- ID de protocole de signalisation
                                        -- mis en tunnel

        messageContent        SEQUENCE OF OCTET STRING, -- séquence de message(s)
                                        -- entier(s)

        tunnellingRequired    NULL OPTIONAL,
        nonStandardData        NonStandardParameter OPTIONAL,
        ...
    } OPTIONAL,
    provisionalRespToH245Tunneling NULL OPTIONAL,
    stimulusControl          StimulusControl OPTIONAL,
    genericData              SEQUENCE OF GenericData OPTIONAL
    }

StimulusControl ::= SEQUENCE
{
    nonStandard                NonStandardParameter OPTIONAL,
    isText                      NULL OPTIONAL,
    h248Message                 OCTET STRING OPTIONAL,
    ...
}

```

Alerting-UUIE ::= SEQUENCE

```
{
  protocolIdentifier      ProtocolIdentifier,
  destinationInfo        EndpointType,
  h245Address            TransportAddress OPTIONAL,
  ...,
  callIdentifier          CallIdentifier,
  h245SecurityMode       H245Security OPTIONAL,
  tokens                 SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
  fastStart              SEQUENCE OF OCTET STRING OPTIONAL,
  multipleCalls          BOOLEAN,
  maintainConnection     BOOLEAN,
  alertingAddress        SEQUENCE OF AliasAddress OPTIONAL,
  presentationIndicator PresentationIndicator OPTIONAL,
  screeningIndicator     ScreeningIndicator OPTIONAL,
  fastConnectRefused     NULL OPTIONAL,
  serviceControl         SEQUENCE OF ServiceControlSession OPTIONAL,
  capacity               CallCapacity OPTIONAL,
  featureSet             FeatureSet OPTIONAL
}
```

CallProceeding-UUIE ::= SEQUENCE

```
{
  protocolIdentifier      ProtocolIdentifier,
  destinationInfo        EndpointType,
  h245Address            TransportAddress OPTIONAL,
  ...,
  callIdentifier          CallIdentifier,
  h245SecurityMode       H245Security OPTIONAL,
  tokens                 SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
  fastStart              SEQUENCE OF OCTET STRING OPTIONAL,
  multipleCalls          BOOLEAN,
  maintainConnection     BOOLEAN,
  fastConnectRefused     NULL OPTIONAL,
  featureSet             FeatureSet OPTIONAL
}
```

Connect-UUIE ::= SEQUENCE

```
{
  protocolIdentifier      ProtocolIdentifier,
  h245Address            TransportAddress OPTIONAL,
  destinationInfo        EndpointType,
  conferenceID           ConferenceIdentifier,
  ...,
  callIdentifier          CallIdentifier,
  h245SecurityMode       H245Security OPTIONAL,
  tokens                 SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
  fastStart              SEQUENCE OF OCTET STRING OPTIONAL,
  multipleCalls          BOOLEAN,
  maintainConnection     BOOLEAN,
  language               SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
  -- balise de langue RFC1766
  connectedAddress       SEQUENCE OF AliasAddress OPTIONAL,
  presentationIndicator PresentationIndicator OPTIONAL,
  screeningIndicator     ScreeningIndicator OPTIONAL,
  fastConnectRefused     NULL OPTIONAL,
  serviceControl         SEQUENCE OF ServiceControlSession OPTIONAL,
  capacity               CallCapacity OPTIONAL,
  featureSet             FeatureSet OPTIONAL
}
```

Information-UUIE ::=SEQUENCE

```
{
    protocolIdentifier      ProtocolIdentifier,
    ...,
    callIdentifier          CallIdentifier,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart               SEQUENCE OF OCTET STRING OPTIONAL,
    fastConnectRefused     NULL OPTIONAL,
    circuitInfo             CircuitInfo OPTIONAL
}
```

ReleaseComplete-UUIE ::= SEQUENCE

```
{
    protocolIdentifier      ProtocolIdentifier,
    reason                  ReleaseCompleteReason OPTIONAL,
    ...,
    callIdentifier          CallIdentifier,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    busyAddress             SEQUENCE OF AliasAddress OPTIONAL,
    presentationIndicator  PresentationIndicator OPTIONAL,
    screeningIndicator     ScreeningIndicator OPTIONAL,
    capacity                CallCapacity OPTIONAL,
    serviceControl          SEQUENCE OF ServiceControlSession OPTIONAL,
    featureSet              FeatureSet OPTIONAL
}
```

ReleaseCompleteReason ::= CHOICE

```
{
    noBandwidth             NULL, -- largeur de bande reprise ou
                                -- demande ARQ refusée
    gatekeeperResources     NULL, -- épuisé
    unreachableDestination  NULL, -- pas de conduit de transport vers
                                -- la destination
    destinationRejection   NULL, -- refusé à destination
    invalidRevision         NULL,
    noPermission            NULL, -- le portier de l'appelé refuse
    unreachableGatekeeper  NULL, -- le terminal ne peut pas atteindre
                                -- le portier pour la demande ARQ
    gatewayResources        NULL,
    badFormatAddress        NULL,
    adaptiveBusy            NULL, -- l'appel est refusé en raison
                                -- d'un encombrement dans le LAN
    inConf                  NULL, -- pas d'adresse dans l'élément
                                -- AlternativeAddress
    undefinedReason         NULL,
    ...,
    facilityCallDeflection  NULL, -- l'appel a été dévié au moyen d'un
                                -- message FACILITY
    securityDenied          NULL, -- positionnements de sécurité
                                -- incompatibles
    calledPartyNotRegistered NULL, -- utilisé par le portier lorsque
                                -- l'extrémité a l'autorisation
                                -- preGrantedARQ lui permettant de
                                -- se passer des messages ARQ/ACF
    callerNotRegistered     NULL, -- utilisé par le portier lorsque
                                -- l'extrémité a l'autorisation
                                -- preGrantedARQ lui permettant de
                                -- se passer des messages ARQ/ACF
    newConnectionNeeded     NULL, -- indique que le message SETUP n'a
                                -- pas été accepté pour cette
                                -- connexion, mais qu'il peut l'être
                                -- pour une nouvelle.
}
```

```

nonStandardReason          NonStandardParameter,
replaceWithConferenceInvite ConferenceIdentifier, -- appel abandonné en
-- raison d'une invitation
-- ultérieure à une
-- conférence
-- (voir 8.4.3.8/H.323)

genericDataReason          NULL,
neededFeatureNotSupported  NULL,
tunnelledSignallingRejected NULL
}

Setup-UUIE ::= SEQUENCE
{
    protocolIdentifier      ProtocolIdentifier,
    h245Address             TransportAddress OPTIONAL,
    sourceAddress           SEQUENCE OF AliasAddress OPTIONAL,
    sourceInfo              EndpointType,
    destinationAddress      SEQUENCE OF AliasAddress OPTIONAL,
    destCallSignalAddress   TransportAddress OPTIONAL,
    destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,          -- Note 1
    destExtraCRV            SEQUENCE OF CallReferenceValue OPTIONAL,    -- Note 1
    activeMC                BOOLEAN,
    conferenceID            ConferenceIdentifier,
    conferenceGoal          CHOICE
    {
        create              NULL,
        join                NULL,
        invite              NULL,
        ...,
        capability-negotiation NULL,
        callIndependentSupplementaryService NULL
    },
    callServices            QseriesOptions OPTIONAL,
    callType                CallType,
    ...,
    sourceCallSignalAddress TransportAddress OPTIONAL,
    remoteExtensionAddress  AliasAddress OPTIONAL,
    callIdentifier          CallIdentifier,
    h245SecurityCapability  SEQUENCE OF H245Security OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart               SEQUENCE OF OCTET STRING OPTIONAL,
    mediaWaitForConnect     BOOLEAN,
    canOverlapSend          BOOLEAN,
    endpointIdentifier      EndpointIdentifier OPTIONAL,
    multipleCalls           BOOLEAN,
    maintainConnection      BOOLEAN,
    connectionParameters   SEQUENCE -- paramètres additionnels
-- de passerelle
    {
        connectionType      ScnConnectionType,
        numberOfScnConnections INTEGER (0..65535),
        connectionAggregation ScnConnectionAggregation,
        ...
    } OPTIONAL,
    language                SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
-- Balise de langue RFC1766
    presentationIndicator   PresentationIndicator OPTIONAL,
    screeningIndicator       ScreeningIndicator OPTIONAL,
    serviceControl          SEQUENCE OF ServiceControlSession OPTIONAL,
    symmetricOperationRequired NULL OPTIONAL,
    capacity                 CallCapacity OPTIONAL,
    circuitInfo             CircuitInfo OPTIONAL,
    desiredProtocols        SEQUENCE OF SupportedProtocols OPTIONAL,

```

```

neededFeatures          SEQUENCE OF FeatureDescriptor OPTIONAL,
desiredFeatures         SEQUENCE OF FeatureDescriptor OPTIONAL,
supportedFeatures       SEQUENCE OF FeatureDescriptor OPTIONAL,
parallelH245Control     SEQUENCE OF OCTET STRING OPTIONAL,
additionalSourceAddresses SEQUENCE OF ExtendedAliasAddress OPTIONAL
}

ScnConnectionType ::= CHOICE
{
    unknown          NULL, -- à choisir lorsque le type de connexion est inconnu
    bChannel         NULL, -- chaque connexion individuelle du RCC est à 64 kbit/s.
                    -- Noter que si le RCC achemine des données
                    -- utilisables à 56 kbit/s, la largeur de bande
                    -- réellement attribuée au RCC reste à 64 kbit/s.
    hybrid2x64       NULL, -- chaque connexion est un appel hybride à 128 kbit/s
    hybrid384        NULL, -- chaque connexion est un appel hybride H0
                    -- (à 384 kbit/s)
    hybrid1536       NULL, -- chaque connexion est un appel hybride H11
                    -- (à 1536 kbit/s)
    hybrid1920       NULL, -- chaque connexion est un appel hybride H12
                    -- (à 1920 kbit/s)
    multirate        NULL, -- largeur de bande fournie par le RCC en mode
                    -- multidébit.
                    -- Dans ce cas, l'octet de débit de transfert
                    -- d'informations contenu
                    -- dans la capacité support doit être mis à
                    -- l'option multidébit et
                    -- l'octet multiplicateur de débit doit indiquer
                    -- le nombre de canaux B.
    ...
}

ScnConnectionAggregation ::= CHOICE
{
    auto             NULL, -- mécanisme d'agrégation inconnu
    none             NULL, -- appel établi avec une seule connexion RCC
    h221             NULL, -- utilisation du verrouillage de trames H.221
                    -- pour agréger les connexions
    bonded_mode1     NULL, -- utilisation du mode d'encapsulation 1
                    -- de l'ISO/CEI 13871
                    -- Utiliser l'élément bonded_mode1 pour
                    -- signaler que le mode
                    -- d'encapsulation à utiliser est inconnu
    bonded_mode2     NULL, -- utilisation du mode d'encapsulation 2
                    -- de l'ISO/CEI 13871
    bonded_mode3     NULL, -- utilisation du mode d'encapsulation 3
                    -- de l'ISO/CEI 13871
    ...
}

PresentationIndicator ::= CHOICE
{
    presentationAllowed      NULL,
    presentationRestricted    NULL,
    addressNotAvailable       NULL,
    ...
}

ScreeningIndicator ::= ENUMERATED
{
    userProvidedNotScreened (0),
        -- le numéro a été fourni par un utilisateur distant
        -- et n'a pas été filtré par un portier
}

```

```

userProvidedVerifiedAndPassed (1),
    -- le numéro a été fourni par l'équipement utilisateur
    -- (ou par un réseau distant) et a été filtré par un portier
userProvidedVerifiedAndFailed (2),
    -- le numéro a été fourni par l'équipement utilisateur
    -- (ou par un réseau distant), et le portier a déterminé
    -- que l'information était incorrecte
networkProvided (3),
    -- le numéro a été fourni par un portier
...
}

Facility-UUIE ::= SEQUENCE
{
    protocolIdentifier          ProtocolIdentifier,
    alternativeAddress          TransportAddress OPTIONAL,
    alternativeAliasAddress     SEQUENCE OF AliasAddress OPTIONAL,
    conferenceID                ConferenceIdentifier OPTIONAL,
    reason                      FacilityReason,
    ...,
    callIdentifier              CallIdentifier,
    destExtraCallInfo           SEQUENCE OF AliasAddress OPTIONAL,
    remoteExtensionAddress      AliasAddress OPTIONAL,
    tokens                      SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens                SEQUENCE OF CryptoH323Token OPTIONAL,
    conferences                  SEQUENCE OF ConferenceList OPTIONAL,
    h245Address                  TransportAddress OPTIONAL,
    fastStart                    SEQUENCE OF OCTET STRING OPTIONAL,
    multipleCalls                BOOLEAN,
    maintainConnection           BOOLEAN,
    fastConnectRefused           NULL OPTIONAL,
    serviceControl               SEQUENCE OF ServiceControlSession OPTIONAL,
    circuitInfo                  CircuitInfo OPTIONAL,
    featureSet                    FeatureSet OPTIONAL,
    destinationInfo              EndpointType OPTIONAL,
    h245SecurityMode             H245Security OPTIONAL
}

ConferenceList ::= SEQUENCE
{
    conferenceID                ConferenceIdentifier OPTIONAL,
    conferenceAlias              AliasAddress OPTIONAL,
    nonStandardData              NonStandardParameter OPTIONAL,
    ...
}

FacilityReason ::= CHOICE
{
    routeCallToGatekeeper       NULL,          -- l'appel doit utiliser le modèle
    -- de portier
    -- le portier est une adresse de
    -- remplacement

    callForwarded                NULL,
    routeCallToMC                 NULL,
    undefinedReason                NULL,
    ...,
    conferenceListChoice          NULL,
    startH245                       NULL,          -- le destinataire doit se connecter
    -- à l'adresse H.245

    noH245                          NULL,          -- l'extrémité ne prend pas en charge H.245
    newTokens                       NULL,
    featureSetUpdate                NULL,
    forwardedElements              NULL,
    transportedInformation          NULL
}

```

Progress-UUIE ::= SEQUENCE

```
{
    protocolIdentifier    ProtocolIdentifier,
    destinationInfo      EndpointType,
    h245Address           TransportAddress OPTIONAL,
    callIdentifier        CallIdentifier,
    h245SecurityMode      H245Security OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart             SEQUENCE OF OCTET STRING OPTIONAL,
    ...,
    multipleCalls         BOOLEAN,
    maintainConnection    BOOLEAN,
    fastConnectRefused    NULL OPTIONAL
}
```

TransportAddress ::= CHOICE

```
{
    ipAddress SEQUENCE
    {
        ip          OCTET STRING (SIZE(4)),
        port        INTEGER(0..65535)
    },
    ipSourceRoute SEQUENCE
    {
        ip          OCTET STRING (SIZE(4)),
        port        INTEGER(0..65535),
        route       SEQUENCE OF OCTET STRING (SIZE(4)),
        routing     CHOICE
        {
            strict  NULL,
            loose   NULL,
            ...
        },
        ...
    },
    ipxAddress SEQUENCE
    {
        node        OCTET STRING (SIZE(6)),
        netnum      OCTET STRING (SIZE(4)),
        port        OCTET STRING (SIZE(2))
    },
    ip6Address SEQUENCE
    {
        ip          OCTET STRING (SIZE(16)),
        port        INTEGER(0..65535),
        ...
    },
    netBios         OCTET STRING (SIZE(16)),
    nsap            OCTET STRING (SIZE(1..20)),
    nonStandardAddress NonStandardParameter,
    ...
}
```

Status-UUIE ::= SEQUENCE

```
{
    protocolIdentifier ProtocolIdentifier,
    callIdentifier      CallIdentifier,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}
```

```

StatusInquiry-UUIE ::= SEQUENCE
{
    protocolIdentifier ProtocolIdentifier,
    callIdentifier      CallIdentifier,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}

SetupAcknowledge-UUIE ::= SEQUENCE
{
    protocolIdentifier ProtocolIdentifier,
    callIdentifier      CallIdentifier,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}

Notify-UUIE ::= SEQUENCE
{
    protocolIdentifier ProtocolIdentifier,
    callIdentifier      CallIdentifier,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    ...
}

-- Début de la section des éléments de message communs

EndpointType ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    vendor                VendorIdentifier OPTIONAL,
    gatekeeper            GatekeeperInfo OPTIONAL,
    gateway                GatewayInfo OPTIONAL,
    mcu                    McuInfo OPTIONAL, -- le contrôleur mc doit également
                                         -- être activé
    terminal                TerminalInfo OPTIONAL,
    mc                      BOOLEAN, -- ne doit pas s'auto-activer
    undefinedNode          BOOLEAN,
    ...,
    set                    BIT STRING (SIZE(32)) OPTIONAL,
                                         -- ne doit pas être utilisé avec séquences
                                         -- codées de mc ou de portier car les divers
                                         -- dispositifs activés sont définis dans
                                         -- les Annexes SET respectives.
    supportedTunnelledProtocols SEQUENCE OF TunnelledProtocol OPTIONAL
                                         -- liste des protocoles pouvant être mis en tunnel
}

GatewayInfo ::= SEQUENCE
{
    protocol                SEQUENCE OF SupportedProtocols OPTIONAL,
    nonStandardData          NonStandardParameter OPTIONAL,
    ...
}

SupportedProtocols ::= CHOICE
{
    nonStandardData          NonStandardParameter,
    h310                      H310Caps,
    h320                      H320Caps,
    h321                      H321Caps,
    h322                      H322Caps,
}

```

```

    h323                H323Caps,
    h324                H324Caps,
    voice               VoiceCaps,
    t120-only           T120OnlyCaps,
    ...,
    nonStandardProtocol NonStandardProtocol,
    t38FaxAnnexbOnly   T38FaxAnnexbOnlyCaps
}

H310Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

H320Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

H321Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

H322Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

H323Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

H324Caps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

VoiceCaps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

```

```

T120OnlyCaps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

NonStandardProtocol ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix,
    ...
}

T38FaxAnnexbOnlyCaps ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    dataRatesSupported   SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix,
    t38FaxProtocol       DataProtocolCapability,
    t38FaxProfile        T38FaxProfile,
    ...
}

McuInfo ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    protocol             SEQUENCE OF SupportedProtocols OPTIONAL
}

TerminalInfo ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...
}

GatekeeperInfo ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    ...
}

VendorIdentifier ::= SEQUENCE
{
    vendor               H221NonStandard,
    productId            OCTET STRING (SIZE(1..256)) OPTIONAL,    -- par vendeur
    versionId           OCTET STRING (SIZE(1..256)) OPTIONAL,    -- par produit
    ...
}

H221NonStandard ::= SEQUENCE
{
    t35CountryCode      INTEGER(0..255),
    t35Extension        INTEGER(0..255),
    manufacturerCode    INTEGER(0..65535),
    ...
}

```

```

TunnelledProtocol ::= SEQUENCE
{
    id CHOICE
    {
        tunnelledProtocolObjectID          OBJECT IDENTIFIER,
        tunnelledProtocolAlternateID       TunnelledProtocolAlternateIdentifier,
        ...
    },
    subIdentifier                          IA5String (SIZE (1..64)) OPTIONAL,
    ...
}

```

```

TunnelledProtocolAlternateIdentifier ::= SEQUENCE
{
    protocolType          IA5String (SIZE (1..64)),
    protocolVariant      IA5String (SIZE (1..64)) OPTIONAL,
    ...
}

```

```

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

```

```

NonStandardIdentifier ::= CHOICE
{
    object          OBJECT IDENTIFIER,
    h221NonStandard H221NonStandard,
    ...
}

```

```

AliasAddress ::= CHOICE
{
    dialedDigits IA5String (SIZE (1..128)) (FROM ("0123456789#*,")),
    h323-ID      BMPString (SIZE (1..256)), -- ISO/CEI 10646-1 de base
                                                    -- (Unicode)
    ...,
    url-ID      IA5String (SIZE(1..512)), -- Adresse de type URL
    transportID TransportAddress,
    email-ID    IA5String (SIZE(1..512)), -- Adresse email selon rfc822
    partyNumber PartyNumber,
    mobileUIM   MobileUIM
}

```

```

AddressPattern ::= CHOICE
{
    wildcard AliasAddress,
    range     SEQUENCE
    {
        startOfRange PartyNumber,
        endOfRange   PartyNumber
    },
    ...
}

```

```

PartyNumber ::= CHOICE
{
    e164Number          PublicPartyNumber,
                        -- le plan de numérotage est selon
                        -- l'UIT-T E.163 et l'UIT-T E.164.
    dataPartyNumber    NumberDigits,
                        -- champ inutilisé, valeur réservée.
}

```

```

telexPartyNumber      NumberDigits,
                      -- champ inutilisé, valeur réservée.
privateNumber         PrivatePartyNumber,
                      -- le plan de numérotage est selon
                      -- l'ISO/CEI 11571.
nationalStandardPartyNumber NumberDigits,
                      -- champ inutilisé, valeur réservée.
...
}

PublicPartyNumber ::= SEQUENCE
{
    publicTypeOfNumber      PublicTypeOfNumber,
    publicNumberDigits      NumberDigits
}

PrivatePartyNumber ::= SEQUENCE
{
    privateTypeOfNumber     PrivateTypeOfNumber,
    privateNumberDigits     NumberDigits
}

NumberDigits ::= IA5String (SIZE (1..128)) (FROM ("0123456789#*,"))

PublicTypeOfNumber ::= CHOICE
{
    unknown                NULL,
    unknown                NULL,
                          -- si ce champ est utilisé, les chiffres
                          -- donnent un préfixe indiquant le type de
                          -- numéro selon les recommandations nationales.
    internationalNumber    NULL,
    nationalNumber         NULL,
    networkSpecificNumber  NULL,
                          -- champ inutilisé, valeur réservée
    subscriberNumber       NULL,
    abbreviatedNumber      NULL,
                          -- valide seulement pour numéro d'appelé
                          -- à l'accès sortant: le réseau remplace
                          -- par le numéro approprié
...
}

PrivateTypeOfNumber ::= CHOICE
{
    unknown                NULL,
    level2RegionalNumber   NULL,
    level1RegionalNumber   NULL,
    pISNSpecificNumber     NULL,
    localNumber            NULL,
    abbreviatedNumber      NULL,
...
}

MobileUIM ::= CHOICE
{
    ansi-41-uim ANSI-41-UIM, -- réseaux hertziens selon normes américaines
    gsm-uim GSM-UIM,        -- réseaux hertziens selon normes européennes
...
}

TBCD-STRING ::= IA5String (FROM ("0123456789#*abc"))

```

ANSI-41-UIM ::= SEQUENCE

```
{
  imsi                TBCD-STRING (SIZE (3..16)) OPTIONAL,
  min                 TBCD-STRING (SIZE (3..16)) OPTIONAL,
  mdn                 TBCD-STRING (SIZE (3..16)) OPTIONAL,
  msisdn              TBCD-STRING (SIZE (3..16)) OPTIONAL,
  esn                 TBCD-STRING (SIZE (16)) OPTIONAL,
  msclid              TBCD-STRING (SIZE (3..16)) OPTIONAL,
  system-id CHOICE
  {
    sid                TBCD-STRING (SIZE (1..4)),
    mid                TBCD-STRING (SIZE (1..4)),
    ...
  },
  systemMyTypeCode    OCTET STRING (SIZE (1)) OPTIONAL,
  systemAccessType    OCTET STRING (SIZE (1)) OPTIONAL,
  qualificationInformationCode OCTET STRING (SIZE (1)) OPTIONAL,
  sesn                TBCD-STRING (SIZE (16)) OPTIONAL,
  soc                 TBCD-STRING (SIZE (3..16)) OPTIONAL,
  ...
  -- IMSI correspond à International Mobile Station Identification
  -- MIN correspond à Mobile Identification Number
  -- MDN correspond à Mobile Directory Number
  -- MSISDN correspond à Mobile Station ISDN Number
  -- ESN Correspond à Electronic Serial Number
  -- MSCID correspond à Mobile Switching Center number + Market ID
  -- ou System ID
  -- SID correspond à System Identification et MID correspond à Market
  -- Identification
  -- SystemMyTypeCode correspond au numéro d'identification du vendeur
  -- SystemAccessType correspond au type d'accès système comme enregistrement
  -- hors alimentation ou localisation d'origine de l'appel ou réponse à
  -- message court, etc.
  -- le Code d'information de qualification correspond à la validité.
  -- SESN Correspond à SIM Electronic Serial Number pour sécurité
  -- d'identification d'utilisateur
  -- SOC correspond à System Operator Code
}
```

GSM-UIM ::= SEQUENCE

```
{
  imsi                TBCD-STRING (SIZE (3..16)) OPTIONAL,
  tmsi                OCTET STRING (SIZE (1..4)) OPTIONAL,
  msisdn              TBCD-STRING (SIZE (3..16)) OPTIONAL,
  imei                TBCD-STRING (SIZE (15..16)) OPTIONAL,
  hplmn               TBCD-STRING (SIZE (1..4)) OPTIONAL,
  vplmn               TBCD-STRING (SIZE (1..4)) OPTIONAL,
  -- IMSI correspond à International Mobile Station Identification
  -- MSISDN correspond à Mobile Station ISDN Number
  -- IMEI Correspond à International Mobile Equipment Identification
  -- VPLMN ou HPLMN correspond à Visiting ou Home Public Land Mobile
  -- Network number
  ...
}
```

ExtendedAliasAddress ::= SEQUENCE

```
{
  address              AliasAddress,
  presentationIndicator PresentationIndicator OPTIONAL,
  screeningIndicator   ScreeningIndicator OPTIONAL,
  ...
}
```

```

Endpoint ::= SEQUENCE
{
    nonStandardData          NonStandardParameter OPTIONAL,
    aliasAddress             SEQUENCE OF AliasAddress OPTIONAL,
    callSignalAddress        SEQUENCE OF TransportAddress OPTIONAL,
    rasAddress               SEQUENCE OF TransportAddress OPTIONAL,
    endpointType             EndpointType OPTIONAL,
    tokens                   SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
    priority                 INTEGER(0..127) OPTIONAL,
    remoteExtensionAddress   SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,
    ...,
    alternateTransportAddresses AlternateTransportAddresses OPTIONAL
}

```

```

AlternateTransportAddresses ::= SEQUENCE
{
    annexE                   SEQUENCE OF TransportAddress OPTIONAL,
    ...
}

```

```

UseSpecifiedTransport ::= CHOICE
{
    tcp                       NULL,
    annexE                   NULL,
    ...
}

```

```

AlternateGK ::= SEQUENCE
{
    rasAddress               TransportAddress,
    gatekeeperIdentifier     GatekeeperIdentifier OPTIONAL,
    needToRegister          BOOLEAN,
    priority                 INTEGER (0..127),
    ...
}

```

```

AltGKInfo ::=SEQUENCE
{
    alternateGatekeeper      SEQUENCE OF AlternateGK,
    altGKisPermanent        BOOLEAN,
    ...
}

```

```

SecurityServiceMode ::= CHOICE
{
    nonStandard              NonStandardParameter,
    none                     NULL,
    default                  NULL,
    ...                      -- peut être étendu à d'autres modes spécifiques
}

```

```

SecurityCapabilities ::= SEQUENCE
{
    nonStandard              NonStandardParameter OPTIONAL,
    encryption               SecurityServiceMode,
    authenticon              SecurityServiceMode,
    integrity                SecurityServiceMode,
    ...
}

```

```

H245Security ::= CHOICE
{
    nonStandard          NonStandardParameter,
    noSecurity           NULL,
    tls                  SecurityCapabilities,
    ipsec                SecurityCapabilities,
    ...
}

QseriesOptions ::= SEQUENCE
{
    q932Full             BOOLEAN, -- si Vrai, indique entier support de Q.932
    q951Full             BOOLEAN, -- si Vrai, indique entier support de Q.951
    q952Full             BOOLEAN, -- si Vrai, indique entier support de Q.952
    q953Full             BOOLEAN, -- si Vrai, indique entier support de Q.953
    q955Full             BOOLEAN, -- si Vrai, indique entier support de Q.955
    q956Full             BOOLEAN, -- si Vrai, indique entier support de Q.956
    q957Full             BOOLEAN, -- si Vrai, indique entier support de Q.957
    q954Info             Q954Details,
    ...
}

Q954Details ::= SEQUENCE
{
    conferenceCalling    BOOLEAN,
    threePartyService    BOOLEAN,
    ...
}

GloballyUniqueID       ::= OCTET STRING (SIZE(16))
ConferenceIdentifier    ::= GloballyUniqueID
RequestSeqNum           ::= INTEGER (1..65535)
GatekeeperIdentifier    ::= BMPString (SIZE(1..128))
BandWidth               ::= INTEGER (0..4294967295) -- en centaines de bits
CallReferenceValue     ::= INTEGER (0..65535)
EndpointIdentifier      ::= BMPString (SIZE(1..128))
ProtocolIdentifier      ::= OBJECT IDENTIFIER
TimeToLive              ::= INTEGER (1..4294967295) -- en secondes
H248PackagesDescriptor ::= PackagesDescriptor
H248SignalsDescriptor  ::= SignalsDescriptor
FeatureDescriptor       ::= GenericData

CallIdentifier ::= SEQUENCE
{
    guid                 GloballyUniqueID,
    ...
}

EncryptIntAlg ::= CHOICE
{
    -- algorithmes de chiffrement de base pour intégrité de message RAS
    nonStandard          NonStandardParameter,
    isoAlgorithm         OBJECT IDENTIFIER, -- défini dans l'ISO/CEI 9979
    ...
}

NonIsoIntegrityMechanism ::= CHOICE
{
    -- mécanisme HMAC utilisé, pas de troncature, balisage peut-être nécessaire!
    hMAC-MD5             NULL,
    hMAC-isol0118-2-s    EncryptIntAlg, -- selon l'ISO/CEI 10118-2 par
    -- algorithme de chiffrement
    -- de base par blocs EncryptIntAlg
    -- (mécanisme MAC court)
}

```

```

hMAC-iso10118-2-1 EncryptIntAlg, -- selon l'ISO/CEI 10118-2 par
-- algorithme de chiffrement
-- de base par blocs EncryptIntAlg
-- (mecanisme MAC long)
hMAC-iso10118-3 OBJECT IDENTIFIER, -- selon l'ISO/CEI 10118-3 par
-- OID comme fonction de hachage
-- (OID = SHA-1,
-- RIPE-MD160,
-- RIPE-MD128)
...
}

IntegrityMechanism ::= CHOICE
{
-- for RAS message integrity
nonStandard NonStandardParameter,
digSig NULL, -- indique l'application d'une
-- signature
iso9797 OBJECT IDENTIFIER, -- numerique selon l'ISO/CEI 9797
-- avec OID comme algorithme de
-- chiffrement base (X-CBC MAC)
nonIsoIM NonIsoIntegrityMechanism,
...
}

ICV ::= SEQUENCE
{
algorithmOID OBJECT IDENTIFIER, -- algorithme utilise pour calculer
-- la signature,
icv BIT STRING -- la valeur calculee d'integrite
-- cryptographique,
-- ou la signature
}

FastStartToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, dhkey
PRESENT, generalID PRESENT
-- mis à "alias" -- })
EncodedFastStartToken ::= TYPE-IDENTIFIER.&Type (FastStartToken)
CryptoH323Token ::= CHOICE
{
cryptoEPPwdHash SEQUENCE
{
alias AliasAddress, -- alias de l'entite produisant le hachage
timeStamp TimeStamp, -- pointeur temporel utilise dans le
-- hachage
token HASHED { EncodedPwdCertToken -- generalID mis à
-- "alias" -- }
},
cryptoGKPwdHash SEQUENCE
{
gatekeeperId GatekeeperIdentifier, -- identite du portier produisant
-- le hachage
timeStamp TimeStamp, -- pointeur temporel utilise dans
-- le hachage
token HASHED { EncodedPwdCertToken -- generalID mis à
-- Gatekeeperid -- }
},
cryptoEPPwdEncr ENCRYPTED { EncodedPwdCertToken -- generalID mis à
-- Gatekeeperid -- },
cryptoGKPwdEncr ENCRYPTED { EncodedPwdCertToken -- generalID mis à
-- Gatekeeperid -- },
cryptoEPCert SIGNED { EncodedPwdCertToken -- generalID mis à
-- Gatekeeperid -- },
cryptoGKCert SIGNED { EncodedPwdCertToken -- generalID mis à alias -- },
cryptoFastStart SIGNED { EncodedFastStartToken },

```

```

    nestedcryptoToken  CryptoToken,
    ...
}
DataRate ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    channelRate          BandWidth,
    channelMultiplier    INTEGER (1..256) OPTIONAL,
    ...
}
CallLinkage ::= SEQUENCE
{
    globalCallId        GloballyUniqueID  OPTIONAL,
    threadId            GloballyUniqueID  OPTIONAL,
    ...
}
SupportedPrefix ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    prefix               AliasAddress,
    ...
}
CapacityReportingCapability ::= SEQUENCE
{
    canReportCallCapacity  BOOLEAN,
    ...
}
CapacityReportingSpecification ::= SEQUENCE
{
    when SEQUENCE
    {
        callStart        NULL OPTIONAL,
        callEnd          NULL OPTIONAL,
        ...
    },
    ...
}
CallCapacity ::= SEQUENCE
{
    maximumCallCapacity  CallCapacityInfo OPTIONAL,
    currentCallCapacity  CallCapacityInfo OPTIONAL,
    ...
}
CallCapacityInfo ::= SEQUENCE
{
    voiceGwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h310GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h320GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h321GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h322GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h323GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    h324GwCallsAvailable  SEQUENCE OF CallsAvailable OPTIONAL,
    t120OnlyGwCallsAvailable SEQUENCE OF CallsAvailable OPTIONAL,
    t38FaxAnnexbOnlyGwCallsAvailable SEQUENCE OF CallsAvailable OPTIONAL,
    terminalCallsAvailable SEQUENCE OF CallsAvailable OPTIONAL,
    mcuCallsAvailable     SEQUENCE OF CallsAvailable OPTIONAL,
    ...
}

```

```

CallsAvailable ::= SEQUENCE
{
    calls          INTEGER (0..4294967295),
    group          IA5String (SIZE (1..128)) OPTIONAL,
    ...
}

CircuitInfo ::= SEQUENCE
{
    sourceCircuitID      CircuitIdentifier OPTIONAL,
    destinationCircuitID CircuitIdentifier OPTIONAL,
    genericData          SEQUENCE OF GenericData OPTIONAL,
    ...
}

CircuitIdentifier ::= CHOICE
{
    cic          CicInfo,
    group        GroupID,
    ...
}

CicInfo ::= SEQUENCE
{
    cic          SEQUENCE OF OCTET STRING (SIZE (2..4)),
    pointCode    OCTET STRING (SIZE (2..5)),
    ...
}

GroupID ::= SEQUENCE
{
    member        SEQUENCE OF INTEGER (0..65535) OPTIONAL,
    group         IA5String (SIZE (1..128)),
    ...
}

ServiceControlDescriptor ::= CHOICE
{
    url          IA5String (SIZE(0..512)), -- indique
                                                    -- protocole/ressource
                                                    -- adressé par URL
    signal       H248SignalsDescriptor,
    nonStandard  NonStandardParameter,
    callCreditServiceControl CallCreditServiceControl,
    ...
}

ServiceControlSession ::= SEQUENCE
{
    sessionId     INTEGER (0..255),
    contents      ServiceControlDescriptor OPTIONAL,
    reason CHOICE
    {
        open      NULL,
        refresh   NULL,
        close     NULL,
        ...
    },
    ...
}

```

```

RasUsageInfoTypes ::= SEQUENCE
{
    nonStandardUsageTypes      SEQUENCE OF NonStandardParameter,
    startTime                  NULL OPTIONAL,
    endTime                    NULL OPTIONAL,
    terminationCause           NULL OPTIONAL,
    ...
}

RasUsageSpecification ::= SEQUENCE
{
    when SEQUENCE
    {
        start                  NULL OPTIONAL,
        end                    NULL OPTIONAL,
        inIrr                  NULL OPTIONAL,
        ...
    },
    callStartingPoint SEQUENCE
    {
        alerting               NULL OPTIONAL,
        connect                NULL OPTIONAL,
        ...
    } OPTIONAL,
    required                   RasUsageInfoTypes,
    ...
}

RasUsageInformation ::= SEQUENCE
{
    nonStandardUsageFields     SEQUENCE OF NonStandardParameter,
    alertingTime               TimeStamp OPTIONAL,
    connectTime                TimeStamp OPTIONAL,
    endTime                    TimeStamp OPTIONAL,
    ...
}

CallTerminationCause ::= CHOICE
{
    releaseCompleteReason      ReleaseCompleteReason,
    releaseCompleteCauseIE     OCTET STRING (SIZE(2..32)),
    ...
}

BandwidthDetails ::= SEQUENCE
{
    sender                     BOOLEAN,           -- TRUE=émetteur, FALSE=récepteur
    multicast                   BOOLEAN,          -- TRUE si flux multidiffusé
    bandwidth                   BandWidth,        -- Largeur de bande utilisée
    ...                          -- pour flux
    rtcpAddresses               TransportChannelInfo, -- Adresses RTCP pour flux média
    ...
}

CallCreditCapability ::= SEQUENCE
{
    canDisplayAmountString      BOOLEAN OPTIONAL,
    canEnforceDurationLimit     BOOLEAN OPTIONAL,
    ...
}

```

```

CallCreditServiceControl ::= SEQUENCE
{
    amountString          BMPString (SIZE (1..512)) OPTIONAL,  -- (Unicode)
    billingMode CHOICE
    {
        credit            NULL,
        debit             NULL,
        ...
    } OPTIONAL,
    callDurationLimit    INTEGER (1..4294967295) OPTIONAL,    -- en secondes
    enforceCallDurationLimit    BOOLEAN OPTIONAL,
    callStartingPoint CHOICE
    {
        alerting         NULL,
        connect          NULL,
        ...
    } OPTIONAL,
    ...
}

GenericData ::= SEQUENCE
{
    id                   GenericIdentifier,
    parameters          SEQUENCE (SIZE (1..512)) OF EnumeratedParameter OPTIONAL,
    ...
}

GenericIdentifier ::= CHOICE
{
    standard            INTEGER(0..16383,...),
    oid                 OBJECT IDENTIFIER,
    nonStandard         GloballyUniqueID,
    ...
}

EnumeratedParameter ::= SEQUENCE
{
    id                  GenericIdentifier,
    content             Content OPTIONAL,
    ...
}

Content ::= CHOICE
{
    raw                 OCTET STRING,
    text                IA5String,
    unicode             BMPString,
    bool                BOOLEAN,
    number8             INTEGER (0..255),
    number16            INTEGER (0..65535),
    number32            INTEGER (0..4294967295),
    id                  GenericIdentifier,
    alias               AliasAddress,
    transport           TransportAddress,
    compound             SEQUENCE (SIZE (1..512)) OF EnumeratedParameter,
    nested              SEQUENCE (SIZE (1..16)) OF GenericData,
    ...
}

FeatureSet ::= SEQUENCE
{
    replacementFeatureSet    BOOLEAN,
    neededFeatures           SEQUENCE OF FeatureDescriptor OPTIONAL,
    desiredFeatures          SEQUENCE OF FeatureDescriptor OPTIONAL,
}

```

```

    supportedFeatures      SEQUENCE OF FeatureDescriptor OPTIONAL,
    ...
}

TransportChannelInfo ::= SEQUENCE
{
    sendAddress            TransportAddress OPTIONAL,
    recvAddress           TransportAddress OPTIONAL,
    ...
}

RTPSession ::= SEQUENCE
{
    rtpAddress            TransportChannelInfo,
    rtcpAddress          TransportChannelInfo,
    cname                 PrintableString,
    ssrc                 INTEGER (1..4294967295),
    sessionId            INTEGER (1..255),
    associatedSessionIds SEQUENCE OF INTEGER (1..255),
    ...,
    multicast            NULL OPTIONAL,
    bandwidth            BandWidth OPTIONAL
}

RasMessage ::= CHOICE
{
    gatekeeperRequest    GatekeeperRequest,
    gatekeeperConfirm    GatekeeperConfirm,
    gatekeeperReject     GatekeeperReject,
    registrationRequest  RegistrationRequest,
    registrationConfirm  RegistrationConfirm,
    registrationReject   RegistrationReject,
    unregistrationRequest UnregistrationRequest,
    unregistrationConfirm UnregistrationConfirm,
    unregistrationReject UnregistrationReject,
    admissionRequest     AdmissionRequest,
    admissionConfirm     AdmissionConfirm,
    admissionReject      AdmissionReject,
    bandwidthRequest     BandwidthRequest,
    bandwidthConfirm     BandwidthConfirm,
    bandwidthReject      BandwidthReject,
    disengageRequest     DisengageRequest,
    disengageConfirm     DisengageConfirm,
    disengageReject      DisengageReject,
    locationRequest      LocationRequest,
    locationConfirm      LocationConfirm,
    locationReject       LocationReject,
    infoRequest          InfoRequest,
    infoRequestResponse  InfoRequestResponse,
    nonStandardMessage   NonStandardMessage,
    unknownMessageResponse UnknownMessageResponse,
    ...,
    requestInProgress    RequestInProgress,
    resourcesAvailableIndicate ResourcesAvailableIndicate,
    resourcesAvailableConfirm ResourcesAvailableConfirm,
    infoRequestAck       InfoRequestAck,
    infoRequestNak       InfoRequestNak,
    serviceControlIndication ServiceControlIndication,
    serviceControlResponse ServiceControlResponse
}

```

```

GatekeeperRequest ::= SEQUENCE -- (GRQ)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData       NonStandardParameter OPTIONAL,
    rasAddress            TransportAddress,
    endpointType          EndpointType,
    gatekeeperIdentifier  GatekeeperIdentifier OPTIONAL,
    callServices          QseriesOptions OPTIONAL,
    endpointAlias         SEQUENCE OF AliasAddress OPTIONAL,
    ...,
    alternateEndpoints    SEQUENCE OF Endpoint OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    authenticationCapability SEQUENCE OF AuthenticationMechanism OPTIONAL,
    algorithmOIDs         SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
    integrity             SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue  ICV OPTIONAL,
    supportsAltGK         NULL OPTIONAL,
    featureSet            FeatureSet OPTIONAL,
    genericData           SEQUENCE OF GenericData OPTIONAL
}

GatekeeperConfirm ::= SEQUENCE -- (GCF)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData       NonStandardParameter OPTIONAL,
    gatekeeperIdentifier  GatekeeperIdentifier OPTIONAL,
    rasAddress            TransportAddress,
    ...,
    alternateGatekeeper    SEQUENCE OF AlternateGK OPTIONAL,
    authenticationMode     AuthenticationMechanism OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    algorithmOID          OBJECT IDENTIFIER OPTIONAL,
    integrity             SEQUENCE OF IntegrityMechanism OPTIONAL,
    integrityCheckValue  ICV OPTIONAL,
    featureSet            FeatureSet OPTIONAL,
    genericData           SEQUENCE OF GenericData OPTIONAL
}

GatekeeperReject ::= SEQUENCE -- (GRJ)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData       NonStandardParameter OPTIONAL,
    gatekeeperIdentifier  GatekeeperIdentifier OPTIONAL,
    rejectReason          GatekeeperRejectReason,
    ...,
    altGKInfo             AltGKInfo OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue  ICV OPTIONAL,
    featureSet            FeatureSet OPTIONAL,
    genericData           SEQUENCE OF GenericData OPTIONAL
}

GatekeeperRejectReason ::= CHOICE
{
    resourceUnavailable    NULL,
    terminalExcluded       NULL,      -- échec de permission
                                -- et non de ressource
}

```

```

invalidRevision          NULL,
undefinedReason         NULL,
...,
securityDenial          NULL,
genericDataReason       NULL,
neededFeatureNotSupported NULL
}

RegistrationRequest ::= SEQUENCE -- (RRQ)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier      ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    discoveryComplete      BOOLEAN,
    callSignalAddress      SEQUENCE OF TransportAddress,
    rasAddress             SEQUENCE OF TransportAddress,
    terminalType           EndpointType,
    terminalAlias          SEQUENCE OF AliasAddress OPTIONAL,
    gatekeeperIdentifier   GatekeeperIdentifier OPTIONAL,
    endpointVendor         VendorIdentifier,
    ...,
    alternateEndpoints     SEQUENCE OF Endpoint OPTIONAL,
    timeToLive             TimeToLive OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    keepAlive              BOOLEAN,
    endpointIdentifier      EndpointIdentifier OPTIONAL,
    willSupplyUUIEs        BOOLEAN,
    maintainConnection     BOOLEAN,
    alternateTransportAddresses AlternateTransportAddresses OPTIONAL,
    additiveRegistration   NULL OPTIONAL,
    terminalAliasPattern   SEQUENCE OF AddressPattern OPTIONAL,
    supportsAltGK          NULL OPTIONAL,
    usageReportingCapability RasUsageInfoTypes OPTIONAL,
    multipleCalls          BOOLEAN OPTIONAL,
    supportedH248Packages SEQUENCE OF H248PackagesDescriptor OPTIONAL,
    callCreditCapability   CallCreditCapability OPTIONAL,
    capacityReportingCapability CapacityReportingCapability OPTIONAL,
    capacity               CallCapacity OPTIONAL,
    featureSet             FeatureSet OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}

RegistrationConfirm ::= SEQUENCE -- (RCF)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier      ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    callSignalAddress      SEQUENCE OF TransportAddress,
    terminalAlias          SEQUENCE OF AliasAddress OPTIONAL,
    gatekeeperIdentifier   GatekeeperIdentifier OPTIONAL,
    endpointIdentifier      EndpointIdentifier,
    ...,
    alternateGatekeeper     SEQUENCE OF AlternateGK OPTIONAL,
    timeToLive             TimeToLive OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    willRespondToIRR       BOOLEAN,
    preGrantedARQ         SEQUENCE
    {
        makeCall           BOOLEAN,
        useGKCallSignalAddressToMakeCall BOOLEAN,
    }
}

```

```

answerCall                BOOLEAN,
useGKCallSignalAddressToAnswer  BOOLEAN,
...,
irrFrequencyInCall        INTEGER (1..65535) OPTIONAL, -- en secondes;
                                                                -- non présent
                                                                -- si le portier
                                                                -- ne veut pas de
                                                                -- messages IRR
totalBandwidthRestriction  BandWidth OPTIONAL, -- limite totale
                                                                -- pour tous les
                                                                -- appels
                                                                -- concurrents
alternateTransportAddresses  AlternateTransportAddresses OPTIONAL,
useSpecifiedTransport      UseSpecifiedTransport OPTIONAL
} OPTIONAL,
maintainConnection         BOOLEAN,
serviceControl             SEQUENCE OF ServiceControlSession OPTIONAL,
supportsAdditiveRegistration  NULL OPTIONAL,
terminalAliasPattern       SEQUENCE OF AddressPattern OPTIONAL,
supportedPrefixes         SEQUENCE OF SupportedPrefix OPTIONAL,
usageSpec                 SEQUENCE OF RasUsageSpecification OPTIONAL,
featureServerAlias        AliasAddress OPTIONAL,
capacityReportingSpec     CapacityReportingSpecification OPTIONAL,
featureSet                FeatureSet OPTIONAL,
genericData              SEQUENCE OF GenericData OPTIONAL
}

```

RegistrationReject ::= SEQUENCE -- (RRJ)

```

{
  requestSeqNum           RequestSeqNum,
  protocolIdentifier      ProtocolIdentifier,
  nonStandardData        NonStandardParameter OPTIONAL,
  rejectReason           RegistrationRejectReason,
  gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
  ...,
  altGKInfo              AltGKInfo OPTIONAL,
  tokens                 SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
  integrityCheckValue    ICV OPTIONAL,
  featureSet             FeatureSet OPTIONAL,
  genericData            SEQUENCE OF GenericData OPTIONAL
}

```

RegistrationRejectReason ::= CHOICE

```

{
  discoveryRequired      NULL,
  invalidRevision        NULL,
  invalidCallSignalAddress  NULL,
  invalidRASAddress      NULL, -- l'adresse fournie est non valide
  duplicateAlias         SEQUENCE OF AliasAddress,
                                                                -- alias enregistré à une autre
                                                                -- extrémité
  invalidTerminalType    NULL,
  undefinedReason        NULL,
  transportNotSupported  NULL, -- un ou plusieurs des transports
                                                                -- non supportés
  ...,
  transportQOSNotSupported  NULL, -- QS d'extrémité non supportée
  resourceUnavailable     NULL, -- ressources de portier épuisées
  invalidAlias           NULL, -- alias incompatible avec
                                                                -- règles de portier
  securityDenial         NULL,
  fullRegistrationRequired  NULL, -- la permission d'enregistrement
                                                                -- a expiré
}

```

```

        additiveRegistrationNotSupported NULL,
        invalidTerminalAliases          SEQUENCE
    {
terminalAlias          SEQUENCE OF AliasAddress OPTIONAL,
terminalAliasPattern  SEQUENCE OF AddressPattern OPTIONAL,
supportedPrefixes     SEQUENCE OF SupportedPrefix OPTIONAL,
...
    },
        genericDataReason          NULL,
        neededFeatureNotSupported  NULL
    }
UnregistrationRequest ::= SEQUENCE -- (URQ)
{
    requestSeqNum          RequestSeqNum,
    callSignalAddress      SEQUENCE OF TransportAddress,
    endpointAlias          SEQUENCE OF AliasAddress OPTIONAL,
    nonStandardData        NonStandardParameter OPTIONAL,
    endpointIdentifier      EndpointIdentifier OPTIONAL,
    ...,
    alternateEndpoints     SEQUENCE OF Endpoint OPTIONAL,
    gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     SUnregRequestReason OPTIONAL,
    endpointAliasPattern    SEQUENCE OF AddressPattern OPTIONAL,
    supportedPrefixes      SEQUENCE OF SupportedPrefix OPTIONAL,
    alternateGatekeeper     SEQUENCE OF AlternateGK OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}
UnregRequestReason ::= CHOICE
{
    reregistrationRequired  NULL,
    ttlExpired              NULL,
    securityDenial          NULL,
    undefinedReason         NULL,
    ...,
    maintenance             NULL
}
UnregistrationConfirm ::= SEQUENCE -- (UCF)
{
    requestSeqNum          RequestSeqNum,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}
UnregistrationReject ::= SEQUENCE -- (URJ)
{
    requestSeqNum          RequestSeqNum,
    rejectReason           UnregRejectReason,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    altGKInfo              AltGKInfo OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue     ICV OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}

```

```

UnregRejectReason ::= CHOICE
{
    notCurrentlyRegistered    NULL,
    callInProgress            NULL,
    undefinedReason           NULL,
    ...,
    permissionDenied          NULL,    -- l'utilisateur demandeur n'est pas autorisé à
                                       -- annuler l'enregistrement de l'utilisateur
                                       -- spécifié
    securityDenial            NULL
}

AdmissionRequest ::= SEQUENCE -- (ARQ)
{
    requestSeqNum             RequestSeqNum,
    callType                  CallType,
    callModel                 CallModel OPTIONAL,
    endpointIdentifier        EndpointIdentifier,
    destinationInfo           SEQUENCE OF AliasAddress OPTIONAL, -- Note 1
    destCallSignalAddress     TransportAddress OPTIONAL,         -- Note 1
    destExtraCallInfo         SEQUENCE OF AliasAddress OPTIONAL,
    srcInfo                   SEQUENCE OF AliasAddress,
    srcCallSignalAddress      TransportAddress OPTIONAL,
    bandwidth                 BandWidth,
    callReferenceValue        CallReferenceValue,
    nonStandardData           NonStandardParameter OPTIONAL,
    callServices              QseriesOptions OPTIONAL,
    conferenceID              ConferenceIdentifier,
    activeMC                  BOOLEAN,
    answerCall                BOOLEAN, -- connexion d'un appel
    ...,
    canMapAlias               BOOLEAN, -- peut traiter une adresse pseudonyme
    callIdentifier            CallIdentifier,
    srcAlternatives           SEQUENCE OF Endpoint OPTIONAL,
    destAlternatives          SEQUENCE OF Endpoint OPTIONAL,
    gatekeeperIdentifier      GatekeeperIdentifier OPTIONAL,
    tokens                    SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens              SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue       ICV OPTIONAL,
    transportQOS              TransportQOS OPTIONAL,
    willSupplyUUIEs          BOOLEAN,
    callLinkage               CallLinkage OPTIONAL,
    gatewayDataRate           DataRate OPTIONAL,
    capacity                  CallCapacity OPTIONAL,
    circuitID                 CircuitIdentifier OPTIONAL,
    desiredProtocols          SEQUENCE OF SupportedProtocols OPTIONAL,
    desiredTunnelledProtocol  TunnelledProtocol OPTIONAL,
    featureSet                FeatureSet OPTIONAL,
    genericData               SEQUENCE OF GenericData OPTIONAL
}

CallType ::= CHOICE
{
    pointToPoint              NULL,    -- Point à point
    oneToN                    NULL,    -- pas d'interaction (à étudier)
    nToOne                    NULL,    -- pas d'interaction (à étudier)
    nToN                      NULL,    -- interactive (multipoint)
    ...
}

```

```

CallModel ::= CHOICE
{
    direct                NULL,
    gatekeeperRouted     NULL,
    ...
}

TransportQOS ::= CHOICE
{
    endpointControlled   NULL,
    gatekeeperControlled NULL,
    noControl            NULL,
    ...
}

AdmissionConfirm ::= SEQUENCE -- (ACF)
{
    requestSeqNum        RequestSeqNum,
    bandwidth            BandWidth,
    callModel            CallModel,
    destCallSignalAddress TransportAddress,
    irrFrequency         INTEGER (1..65535) OPTIONAL,
    nonStandardData      NonStandardParameter OPTIONAL,
    ...,
    destinationInfo     SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCallInfo   SEQUENCE OF AliasAddress OPTIONAL,
    destinationType     EndpointType OPTIONAL,
    remoteExtensionAddress SEQUENCE OF AliasAddress OPTIONAL,
    alternateEndpoints  SEQUENCE OF Endpoint OPTIONAL,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    transportQOS        TransportQOS OPTIONAL,
    willRespondToIRR    BOOLEAN,
    uuiesRequested      UUIEsRequested,
    language            SEQUENCE OF IA5String (SIZE (1..32)) OPTIONAL,
    alternateTransportAddresses AlternateTransportAddresses OPTIONAL,
    useSpecifiedTransport UseSpecifiedTransport OPTIONAL,
    circuitInfo         CircuitInfo OPTIONAL,
    usageSpec           SEQUENCE OF RasUsageSpecification OPTIONAL,
    supportedProtocols  SEQUENCE OF SupportedProtocols OPTIONAL,
    serviceControl      SEQUENCE OF ServiceControlSession OPTIONAL,
    multipleCalls       BOOLEAN OPTIONAL,
    featureSet          FeatureSet OPTIONAL,
    genericData         SEQUENCE OF GenericData OPTIONAL
}

UUIEsRequested ::= SEQUENCE
{
    setup                BOOLEAN,
    callProceeding       BOOLEAN,
    connect              BOOLEAN,
    alerting             BOOLEAN,
    information          BOOLEAN,
    releaseComplete      BOOLEAN,
    facility             BOOLEAN,
    progress             BOOLEAN,
    empty               BOOLEAN,
    ...,
    status               BOOLEAN,
    statusInquiry       BOOLEAN,
    setupAcknowledge     BOOLEAN,
    notify              BOOLEAN
}

```

AdmissionReject ::= SEQUENCE -- (ARJ)

```
{
  requestSeqNum      RequestSeqNum,
  rejectReason       AdmissionRejectReason,
  nonStandardData    NonStandardParameter OPTIONAL,
  ...,
  altGKInfo          AltGKInfo OPTIONAL,
  tokens             SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
  callSignalAddress  SEQUENCE OF TransportAddress OPTIONAL,
  integrityCheckValue ICV OPTIONAL,
  serviceControl     SEQUENCE OF ServiceControlSession OPTIONAL,
  featureSet         FeatureSet OPTIONAL,
  genericData        SEQUENCE OF GenericData OPTIONAL
}
```

AdmissionRejectReason ::= CHOICE

```
{
  calledPartyNotRegistered  NULL,      -- adresse intraduisible
  invalidPermission         NULL,      -- expiration de permission
  requestDenied             NULL,      -- absence de bande disponible
  undefinedReason           NULL,
  callerNotRegistered       NULL,
  routeCallToGatekeeper     NULL,
  invalidEndpointIdentifier NULL,
  resourceUnavailable        NULL,
  ...,
  securityDenial            NULL,
  qosControlNotSupported    NULL,
  incompleteAddress         NULL,
  aliasesInconsistent       NULL,      -- les pseudonymes multiples de la
                                   -- demande correspondent à des
                                   -- personnes différentes

  routeCallToSCN            SEQUENCE OF PartyNumber,
  exceedsCallCapacity       NULL,      -- la destination n'a pas la capacité
                                   -- pour cet appel

  collectDestination        NULL,
  collectPIN                NULL,
  genericDataReason         NULL,
  neededFeatureNotSupported NULL
}
```

BandwidthRequest ::= SEQUENCE -- (BRQ)

```
{
  requestSeqNum      RequestSeqNum,
  endpointIdentifier EndpointIdentifier,
  conferenceID       ConferenceIdentifier,
  callReferenceValue CallReferenceValue,
  callType           CallType OPTIONAL,
  bandWidth          BandWidth,
  nonStandardData    NonStandardParameter OPTIONAL,
  ...,
  callIdentifier     CallIdentifier,
  gatekeeperIdentifier GatekeeperIdentifier OPTIONAL,
  tokens             SEQUENCE OF ClearToken OPTIONAL,
  cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
  integrityCheckValue ICV OPTIONAL,
  answeredCall       BOOLEAN,
  callLinkage        CallLinkage OPTIONAL,
  capacity           CallCapacity OPTIONAL,
  usageInformation   RasUsageInformation OPTIONAL,
  bandwidthDetails   SEQUENCE OF BandwidthDetails OPTIONAL,
  genericData        SEQUENCE OF GenericData OPTIONAL
}
```

```

BandwidthConfirm ::= SEQUENCE -- (BCF)
{
    requestSeqNum          RequestSeqNum,
    bandwidth              BandWidth,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    capacity               CallCapacity OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}

BandwidthReject ::= SEQUENCE -- (BRJ)
{
    requestSeqNum          RequestSeqNum,
    rejectReason           BandRejectReason,
    allowedBandWidth       BandWidth,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    altGKInfo              AltGKInfo OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL
}

BandRejectReason ::= CHOICE
{
    notBound               NULL,          -- la permission de recherche
                                -- est périmée
    invalidConferenceID    NULL,          -- révision possible
    invalidPermission      NULL,          -- violation vraie de permission
    insufficientResources  NULL,
    invalidRevision        NULL,
    undefinedReason        NULL,
    ...,
    securityDenial         NULL
}

LocationRequest ::= SEQUENCE -- (LRQ)
{
    requestSeqNum          RequestSeqNum,
    endpointIdentifier      EndpointIdentifier OPTIONAL,
    destinationInfo        SEQUENCE OF AliasAddress,
    nonStandardData        NonStandardParameter OPTIONAL,
    replyAddress           TransportAddress,
    ...,
    sourceInfo             SEQUENCE OF AliasAddress OPTIONAL,
    canMapAlias            BOOLEAN,      -- capacité de traitement d'adresse
                                -- pseudonyme
    gatekeeperIdentifier    GatekeeperIdentifier OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    desiredProtocols       SEQUENCE OF SupportedProtocols OPTIONAL,
    desiredTunnelledProtocol TunnelledProtocol OPTIONAL,
    featureSet             FeatureSet OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL,
    hopCount               INTEGER (1..255) OPTIONAL,
    circuitInfo            CircuitInfo OPTIONAL
}

```

```

LocationConfirm ::= SEQUENCE -- (LCF)
{
    requestSeqNum          RequestSeqNum,
    callSignalAddress      TransportAddress,
    rasAddress             TransportAddress,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    destinationInfo        SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCallInfo      SEQUENCE OF AliasAddress OPTIONAL,
    destinationType        EndpointType OPTIONAL,
    remoteExtensionAddress SEQUENCE OF AliasAddress OPTIONAL,
    alternateEndpoints     SEQUENCE OF Endpoint OPTIONAL,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    alternateTransportAddresses AlternatesTransportAddresses OPTIONAL,
    supportedProtocols     SEQUENCE OF SupportedProtocols OPTIONAL,
    multipleCalls          BOOLEAN OPTIONAL,
    featureSet             FeatureSet OPTIONAL,
    genericData            SEQUENCE OF GenericData OPTIONAL,
    circuitInfo           CircuitInfo OPTIONAL,
    serviceControl        SEQUENCE OF ServiceControlSession OPTIONAL
}

LocationReject ::= SEQUENCE -- (LRJ)
{
    requestSeqNum          RequestSeqNum,
    rejectReason           LocationRejectReason,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,
    altGKInfo             AltGKInfo OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    featureSet            FeatureSet OPTIONAL,
    genericData           SEQUENCE OF GenericData OPTIONAL,
    serviceControl        SEQUENCE OF ServiceControlSession OPTIONAL
}

LocationRejectReason ::= CHOICE
{
    notRegistered          NULL,
    invalidPermission      NULL,      -- exclusion par administrateur ou le
    requestDenied          NULL,      -- dispositif ne peut trouver l'emplacement
    undefinedReason        NULL,
    securityDenial         NULL,
    aliasesInconsistent    NULL,      -- les pseudonymes multiples de la
                                -- demande correspondent à des
                                -- personnes différentes

    routeCalltoSCN         SEQUENCE OF PartyNumber,
    resourceUnavailable    NULL,
    genericDataReason      NULL,
    neededFeatureNotSupported NULL
}

DisengageRequest ::= SEQUENCE -- (DRQ)
{
    requestSeqNum          RequestSeqNum,
    endpointIdentifier     EndpointIdentifier,
    conferenceID           ConferenceIdentifier,
    callReferenceValue     CallReferenceValue,
    disengageReason        DisengageReason,
    nonStandardData        NonStandardParameter OPTIONAL,
    ...,

```

```

callIdentifier          CallIdentifier,
gatekeeperIdentifier   GatekeeperIdentifier OPTIONAL,
tokens                 SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue   ICV OPTIONAL,
answeredCall           BOOLEAN,
callLinkage            CallLinkage OPTIONAL,
capacity              CallCapacity OPTIONAL,
circuitInfo            CircuitInfo OPTIONAL,
usageInformation       RasUsageInformation OPTIONAL,
terminationCause      CallTerminationCause OPTIONAL,
serviceControl         SEQUENCE OF ServiceControlSession OPTIONAL,
genericData            SEQUENCE OF GenericData OPTIONAL
}

DisengageReason ::= CHOICE
{
    forcedDrop          NULL,      -- le portier force l'abandon
    normalDrop          NULL,      -- associé à abandon normal
    undefinedReason     NULL,
    ...
}

DisengageConfirm ::= SEQUENCE -- (DCF)
{
    requestSeqNum       RequestSeqNum,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    capacity            CallCapacity OPTIONAL,
    circuitInfo         CircuitInfo OPTIONAL,
    usageInformation    RasUsageInformation OPTIONAL,
    genericData         SEQUENCE OF GenericData OPTIONAL
}

DisengageReject ::= SEQUENCE -- (DRJ)
{
    requestSeqNum       RequestSeqNum,
    rejectReason        DisengageRejectReason,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    altGKInfo           AltGKInfo OPTIONAL,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    genericData         SEQUENCE OF GenericData OPTIONAL
}

DisengageRejectReason ::= CHOICE
{
    notRegistered       NULL,      -- non enregistré auprès du portier
    requestToDropOther  NULL,      -- impossibilité de demander l'abandon
    ...,
    securityDenial      NULL,      -- pour d'autres usagers
}

InfoRequest ::= SEQUENCE -- (IRQ)
{
    requestSeqNum       RequestSeqNum,
    callReferenceValue  CallReferenceValue,
    nonStandardData     NonStandardParameter OPTIONAL,
}

```

```

replyAddress      TransportAddress OPTIONAL,
...
callIdentifier    CallIdentifier,
tokens           SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens     SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue ICV OPTIONAL,
uuiesRequested   UUIEsRequested OPTIONAL,
callLinkage      CallLinkage OPTIONAL,
usageInfoRequested RasUsageInfoTypes OPTIONAL,
segmentedResponseSupported NULL OPTIONAL,
nextSegmentRequested INTEGER (0..65535) OPTIONAL,
capacityInfoRequested NULL OPTIONAL,
genericData      SEQUENCE OF GenericData OPTIONAL
}

InfoRequestResponse ::= SEQUENCE -- (IRR)
{
    nonStandardData      NonStandardParameter OPTIONAL,
    requestSeqNum        RequestSeqNum,
    endpointType         EndpointType,
    endpointIdentifier    EndpointIdentifier,
    rasAddress           TransportAddress,
    callSignalAddress    SEQUENCE OF TransportAddress,
    endpointAlias        SEQUENCE OF AliasAddress OPTIONAL,
    perCallInfo          SEQUENCE OF SEQUENCE
    {
        nonStandardData      NonStandardParameter OPTIONAL,
        callReferenceValue    CallReferenceValue,
        conferenceID         ConferenceIdentifier,
        originator           BOOLEAN OPTIONAL,
        audio                SEQUENCE OF RTPSession OPTIONAL,
        video                SEQUENCE OF RTPSession OPTIONAL,
        data                 SEQUENCE OF TransportChannelInfo OPTIONAL,
        h245                 TransportChannelInfo,
        callSignaling        TransportChannelInfo,
        callType             CallType,
        bandWidth            BandWidth,
        callModel            CallModel,
        ...
        callIdentifier        CallIdentifier,
        tokens               SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens         SEQUENCE OF CryptoH323Token OPTIONAL,
        substituteConfIDs    SEQUENCE OF ConferenceIdentifier,
        pdu                  SEQUENCE OF SEQUENCE
        {
            h323pdu          H323-UU-PDU,
            sent             BOOLEAN -- envoi de TRUE, réception de FALSE
        } OPTIONAL,
        callLinkage          CallLinkage OPTIONAL,
        usageInformation     RasUsageInformation OPTIONAL,
        circuitInfo          CircuitInfo OPTIONAL
    } OPTIONAL,
    ...
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    needResponse           BOOLEAN,
    capacity               CallCapacity OPTIONAL,
    irrStatus              InfoRequestResponseStatus OPTIONAL,
    unsolicited            BOOLEAN,
    genericData            SEQUENCE OF GenericData OPTIONAL
}

```

```

InfoRequestResponseStatus ::= CHOICE
{
    complete           NULL,
    incomplete         NULL,
    segment            INTEGER (0..65535),
    invalidCall        NULL,
    ...
}

InfoRequestAck ::= SEQUENCE -- (IACK)
{
    requestSeqNum      RequestSeqNum,
    nonStandardData    NonStandardParameter OPTIONAL,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

InfoRequestNak ::= SEQUENCE -- (INAK)
{
    requestSeqNum      RequestSeqNum,
    nonStandardData    NonStandardParameter OPTIONAL,
    nakReason          InfoRequestNakReason,
    altGKInfo          AltGKInfo OPTIONAL,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

InfoRequestNakReason ::= CHOICE
{
    notRegistered      NULL,      -- non enregistré auprès du portier
    securityDenial     NULL,
    undefinedReason    NULL,
    ...
}

NonStandardMessage ::= SEQUENCE
{
    requestSeqNum      RequestSeqNum,
    nonStandardData    NonStandardParameter,
    ...,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    featureSet         FeatureSet OPTIONAL,
    genericData        SEQUENCE OF GenericData OPTIONAL
}

UnknownMessageResponse ::= SEQUENCE -- (XRS)
{
    requestSeqNum      RequestSeqNum,
    ...,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    messageNotUnderstood OCTET STRING
}

```

```

RequestInProgress ::= SEQUENCE          -- (RIP)
{
    requestSeqNum          RequestSeqNum,
    nonStandardData        NonStandardParameter OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    delay                   INTEGER(1..65535),
    ...
}

ResourcesAvailableIndicate ::= SEQUENCE -- (RAI)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    endpointIdentifier     EndpointIdentifier,
    protocols               SEQUENCE OF SupportedProtocols,
    almostOutOfResources   BOOLEAN,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    ...,
    capacity                CallCapacity OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

ResourcesAvailableConfirm ::= SEQUENCE -- (RAC)
{
    requestSeqNum          RequestSeqNum,
    protocolIdentifier     ProtocolIdentifier,
    nonStandardData        NonStandardParameter OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    ...,
    genericData             SEQUENCE OF GenericData OPTIONAL
}

ServiceControlIndication ::= SEQUENCE -- (SCI)
{
    requestSeqNum          RequestSeqNum,
    nonStandardData        NonStandardParameter OPTIONAL,
    serviceControl         SEQUENCE OF ServiceControlSession,
    endpointIdentifier     EndpointIdentifier OPTIONAL,
    callSpecific SEQUENCE
    {
        callIdentifier     CallIdentifier,
        conferenceID       ConferenceIdentifier,
        answeredCall       BOOLEAN,
        ...
    } OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    featureSet              FeatureSet OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL,
    ...
}

```

```

ServiceControlResponse ::= SEQUENCE -- (SCR)
{
    requestSeqNum          RequestSeqNum,
    result                  CHOICE
    {
        started            NULL,
        failed             NULL,
        stopped            NULL,
        notAvailable       NULL,
        neededFeatureNotSupported NULL
        ...
    } OPTIONAL,
    nonStandardData        NonStandardParameter OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    featureSet              FeatureSet OPTIONAL,
    genericData             SEQUENCE OF GenericData OPTIONAL,
    ...
}
END      -- de la notation ASN.1

```

ANNEXE I

Groupage par paquets vidéo H.263+

La norme IETF RFC 2429 spécifie le format de charge utile RTP des flux de bits vidéo H.263 qui contiennent les nouvelles caractéristiques "H.263+" adoptées dans la version 2 (1998) de l'UIT-T H.263 (avec les caractéristiques utilisant PLUSTYPE ou les Annexes I/H.263 à l'Annexe T/H.263).

La capacité de prendre en charge le format de charge utile H.263 de la norme RFC 2190 spécifié dans l'Annexe E est exigée pour les flux de bits H.263 qui n'utilisent pas les nouvelles caractéristiques de la version 2 de l'UIT-T H.263, car cette prise en charge est nécessaire pour la compatibilité avec les implémentations précédentes. Toutefois, le nouveau format de charge utile spécifié dans la norme RFC 2429 doit être utilisé même pour des flux de bits qui ne contiennent pas les nouvelles caractéristiques de la version 2, à condition que le format de charge utile le plus récent corresponde aux capacités des terminaux de réception.

APPENDICE I

Algorithmes RTP/RTCP

Les renseignements auxquels il est fait référence peuvent être trouvés dans la proposition de norme Internet suivante:

- SCHULZRINNE (H.), CASNER (S.), FREDERICK (R.) et JACOBSON (V.): RFC 1889, RTP: A Transport Protocol for Real-Time Applications (RTP: un protocole de transport pour les applications en temps réel), *Internet Engineering Task Force*, 1996.

APPENDICE II

Profil RTP

Les renseignements auxquels il est fait référence peuvent être trouvés dans la proposition de norme Internet suivante:

- SCHULZRINNE (H.): RFC 1890, RTP Profile for Audio and Video Confernces with Minimal Control (profil RTP pour les conférences audio et vidéo avec commande minimale), *Internet Engineering Task Force*, 1996.

APPENDICE III

Mise en paquets H.261

Les renseignements auxquels il est fait référence peuvent être trouvés dans la proposition de norme Internet suivante:

- TURLETTI (T.), HUITEMA (C.): RFC 2032, RTP Payload Format for H.261 Video Streams (format de charge utile RTP pour les flux vidéo H.261), *Internet Engineering Task Force*, 1996.

APPENDICE IV

Fonctionnement du mode H.225.0 sur différentes piles protocolaires de réseau en mode paquet

Le présent appendice donne des détails complémentaires concernant le fonctionnement du mode H.225.0 sur diverses piles protocolaires réelles de réseau à commutation par paquets. Les réseaux à commutation par paquets utilisés dans la présente Recommandation doivent fournir un mode de fonctionnement fiable et un mode de fonctionnement non fiable, comprenant un moyen de repérer les frontières de paquet.

IV.1 TCP/IP/UDP

Il convient de noter que le protocole UDP peut fragmenter et réassembler de grands paquets vidéo, mais que l'échec de la mise en paquets des macroblocs peut conduire à la perte d'un groupe de blocs entier.

La multidiffusion IP peut être utilisée pour la distribution GRQ par opposition à la diffusion de couche d'accès média.

Applications avec remise non fiable	Canal H.245 et canal de signalisation d'appel
UDP	TPKT — — TCP
IP	
Couche Liaison	
Couche Physique	

Un TPKT est un format de paquet tel que défini dans la norme IETF RFC 1006. Il sert à délimiter des messages individuels (unités PDU) dans le flux TCP, qui assure lui-même un flux continu d'octets sans limites précises. Un TPKT est constitué d'un champ de numéro de version d'un octet, suivi d'un champ réservé d'un octet, d'un champ de longueur de deux octets et des données effectives. Le champ de numéro de version doit contenir la valeur "3" et le champ réservé doit contenir la valeur "0". Le champ de longueur doit contenir la longueur de tout le paquet, y compris le numéro de version, le champ réservé et le champ de longueur sous forme de mot gros-boutiste de 16 bits.

IV.1.1 Recherche du portier

IV.1.1.1 Recherche au moyen d'une adresse de multidiffusion ou d'un accès connu

Après l'exécution des procédures de recherche et d'enregistrement du portier décrites au paragraphe 7/H.323, les extrémités doivent utiliser l'adresse de multidiffusion ou l'accès connu suivant lorsqu'elle recherche le portier en fonction de leur configuration de réseau:

- adresse UDP pour communication en multidiffusion avec des portiers: 224.0.1.41;
- accès UDP pour communication en multidiffusion avec des portiers: 1718;
- accès UDP pour communication RAS unidiffusion lorsqu'il n'existe aucun "autre accord": 1719.

A noter que les termes "autre accord" peuvent désigner l'enregistrement d'une extrémité auprès d'un portier.

A noter que, lors des implémentations, il faut faire attention à la portée de la multidiffusion de manière à ne pas inonder Internet avec des messages de recherche.

A supposer qu'un portier ait une adresse IP de type 134.134.12.1, par exemple, la signalisation suivante peut être établie:

- Arrivée d'un message LRQ ou GRQ à l'adresse 134.134.12.1: accès 1719;
- arrivée d'un message LRQ ou GRQ à l'adresse 134.134.12.1 : accès 1718 (à noter que cette situation peut se produire avec des portiers de version 1);
- arrivée d'un message LRQ ou GRQ à l'adresse 224.0.1.41: accès 1718.

Le portier peut transmettre un message LRQ aux adresses suivantes:

- 224.0.1.41: accès 1718 (multidiffusion à destination de tous les portiers);
- X.X.X.X: accès 1719 (à destination d'un portier donné);

L'accès 1719 ne doit être utilisé que lorsqu'une demande est envoyée en mode unidiffusion. Cela permet au destinataire de savoir s'il doit envoyer un refus (xRJ) à l'expéditeur (il doit le faire dans tous les cas).

L'accès 1718 ne doit être utilisé que lorsqu'une demande est envoyée en mode multidiffusion. Le destinataire doit envoyer une réponse appropriée, selon le message. Pour un message LRQ, aucun refus n'est nécessaire. Le destinataire ne répond pas aux demandes de multidiffusion. Pour un message GRQ, un message GRJ dirigé doit être envoyé à la source dont provient le message GRQ.

IV.1.1.2 Recherche au moyen d'un système DNS (à titre d'information)

IV.1.1.2.1 Adresse URL pour les portiers

D'abord, il convient de noter qu'un portier est identifié par une adresse de transport et un identificateur gatekeeperIdentifier, qui est une chaîne. Un portier étant une ressource particulière sur Internet, il est raisonnable de le spécifier avec un identificateur uniforme de ressources (URL,

uniform resource locator). Le protocole utilisé par le portier étant le protocole RAS, l'adresse URL d'un portier pourrait être donnée par:

ras://gkID@domainname

gkID est l'identificateur gatekeeperIdentifier et domainname est un nom de domaine DNS qui identifie le domaine du portier. A noter qu'il ne s'agit pas nécessairement d'un nom de domaine complet (FQDN, *fully qualified domain name*) avec un enregistrement A – Il n'est pas exigé que ce nom de domaine ait une interface de transport physique avec un numéro IP enregistré dans le système DNS. Toutefois, s'il s'agit d'un nom FQDN, il est raisonnable d'exiger que le numéro IP soit celui du portier auquel l'adresse URL se rapporte. Dans ce cas, l'adjonction d'un numéro d'accès facultatif à l'adresse URL est autorisée:

ras://gkID@domainname:port_no.

Si aucun numéro d'accès n'est donné, la valeur connue 1719 est prise comme valeur par défaut.

Le cas le plus intéressant se présente lorsqu'il ne s'agit pas d'un nom FQDN et que le nom de domaine ne se rapporte pas à une adresse de transport énumérée dans le système DNS. Le nom de domaine peut alors renvoyer à une simple "zone d'autorité du portier". Le paragraphe qui suit explique la manière de rechercher le portier dans ce cas.

IV.1.1.2.2 Recherche de l'adresse URL

L'adresse URL ne permet pas de résoudre le problème de la localisation du portier, elle permet simplement de disposer d'un format normalisé pour les informations à rechercher. Le problème est le suivant: comment produire une adresse de transport et un identificateur gatekeeperIdentifier pour la signalisation RAS, étant donné le nom de domaine d'un portier.

Si le portier a un identificateur conforme à la norme IETF RFC 822, il est facile d'extraire un nom de domaine d'un tel identificateur. En réalité, il peut être pratique d'attribuer des identificateurs conformes au document IETF RFC 822 aux extrémités puis de stipuler que la partie nom de domaine de l'identificateur renvoie au domaine du portier.

IV.1.1.2.2.1 Interrogation relative à des enregistrements de ressources SRV

La première solution consiste à utiliser le fait que le portier est fondamentalement un service de système et que l'adresse de transport d'un service de système nommé peut être extraite du système DNS grâce à une interrogation relative à un nouveau type d'enregistrement de ressources du système DNS, appelé SRV ("*service location record*", enregistrement de localisation de service). Etant donné un nom de domaine, une interrogation relative aux enregistrements SRV doit être faite pour déterminer l'adresse de transport du service RAS pour ce domaine. Le nom de domaine proprement dit, ou un nom de domaine renvoyé dans la réponse à l'interrogation, est utilisé comme identificateur de portier. L'enregistrement SRV et son usage sont définis dans le document RFC 2872 du groupe IETF.

IV.1.1.2.2.2 Interrogation relative à des enregistrements TXT

Toutes les mises en œuvre actuelles de système DNS prennent en charge l'enregistrement de ressources TXT. A la base, il s'agit de texte libre qui peut être renvoyé pour chaque nom de domaine. Il est possible de stocker de nombreuses ressources TXT pour un même domaine. La norme stipule que tous les enregistrements TXT doivent être renvoyés lorsqu'une interrogation doit être faite les concernant.

Il est possible d'utiliser des interrogations TXT si les interrogations SRV échouent. Prenons comme hypothèse la même convention concernant l'extraction d'un nom de domaine que celle qui est proposée ci-dessus. Des chaînes conformes au document IETF RFC 822 (noms de type adresse électronique) ou conformes à la norme IETF RFC 1768 (adresses URL) peuvent être utilisées comme identificateurs gatekeeperIdentifiers. Dans l'un ou l'autre cas, le nom de domaine sert à faire

une interrogation TXT dans le système DNS relative au nom de domaine. Les enregistrements de ressources renvoyés sont des lignes de texte libre et le terminal recherchera alors dans la réponse, les lignes de la forme:

ras [<gk id>@]<domain name >[:<portno>] [<priority>]

Le champ <gk id> est un identificateur de portier facultatif qui est distinct du nom de domaine. Si ce champ est absent, le nom de domaine proprement dit est supposé être l'identificateur du portier.

Le champ <domain name> peut être soit le nom de l'enregistrement A qui contient l'adresse IP du portier soit une adresse IP brute sous forme pointée. Il n'est pas nécessaire que le nom de domaine soit complet; s'il ne l'est pas, le sous-domaine dans lequel l'enregistrement TXT a été trouvé doit lui être rattaché afin de constituer le nom d'enregistrement A complet.

Le champ [:<portno>] facultatif peut servir à spécifier un numéro d'accès autre que l'accès RAS normalisé.

Le champ [<priority>] facultatif spécifie l'ordre dans lequel il convient d'accéder aux portiers énumérés pour une recherche ou pour des interrogations LRQ s'il existe plusieurs enregistrements TXT RAS. Plus le numéro est petit, meilleur est le rang de priorité.

A noter qu'avec ce format, si le champ <gk id> est absent, les identificateurs de portier sont en réalité des noms de domaines juridiques. Toutefois, s'il est nécessaire qu'un même serveur prenne en charge plusieurs portiers logiques, chacun avec un identificateur distinct, le format le permettra. Cela est dû au fait que des enregistrements A distincts peuvent contenir la même adresse IP.

Des blancs servent de délimiteurs entre **ras** et **gk id** – S'il est présent – ou **domain name** ainsi qu'entre **portno** et **priority**. Les blancs sont composés d'un nombre quelconque d'espaces et de tabulations.

Exemples d'enregistrements TXT de portiers valides:

- ras gk1
- ras gk1.company.com
- ras gk1:1500 3
- ras 172.11.22.33:1500 2

Le client analyse les lignes renvoyées et à partir de ces lignes, il obtient l'adresse de transport du portier à l'intérieur du domaine considéré à laquelle il peut envoyer des messages RAS.

Etant donné que le système DNS a besoin d'un serveur pour renvoyer tous les enregistrements TXT associés à un nom de domaine, le client peut filtrer les enregistrements et ne traiter que ceux qui lui sont utiles. Le système DNS peut aussi renvoyer une liste ordonnée de portiers qui peuvent servir de portiers de remplacement ou de secours, tels que définis dans l'UIT-T H.323.

A noter que ce que le serveur renvoie dans une telle interrogation pourrait être une vraie adresse de transport en notation décimale pointée, ou un nom FQDN qui, lui-même, nécessite une interrogation relative aux enregistrements A dans le système DNS pour déterminer l'adresse de transport. L'avantage lié à l'utilisation d'un nom FQDN tient à la dissimulation habituelle des numéros IP effectifs. L'avantage lié à l'utilisation de numéros IP tient à ce qu'une seconde interrogation dans le système DNS est évitée, d'où une accélération du temps de préétablissement d'appel.

IV.1.1.2.3 Traitement par le portier des identificateurs email-ID pendant les demandes ARQ et LRQ

Lorsque le champ **destinationInfo** d'un message ARQ ou LRQ contient une adresse pseudonyme **email-ID**, le portier doit d'abord vérifier si l'alias figure dans sa base de données d'enregistrement. Si l'adresse ne peut être résolue, le portier doit analyser l'alias pour récupérer sa partie domaine. Si aucun domaine n'est donné, le portier peut produire un domaine par défaut. Le domaine sert alors à

localiser un ou plusieurs portiers, au moyen des procédures du IV.1.1.2.2. Le portier peut alors interroger tous les portiers trouvés avec un échange de messages LRQ/LCF/LRJ.

A noter que plusieurs portiers peuvent avoir des enregistrements TXT correspondants dans un même domaine du système DNS. En conséquence, un même domaine du système DNS peut "contenir" plusieurs zones H.323. Ainsi, même si un portier ne peut pas résoudre un identificateur de courrier électronique dont la partie domaine est un de ses domaines par défaut, il peut toujours interroger d'autres zones du même domaine du système DNS.

Si le portier est présenté avec un alias dont l'enregistrement est annulé et qui est un identificateur **h323-id** et si l'identificateur peut être interprété comme une partie utilisateur juridique de nom IETF RFC 822, le portier peut interpréter l'alias comme s'il s'agissait d'un identificateur de courrier électronique dans son domaine par défaut et tenter de localiser l'alias chez un autre portier. De même, le portier peut enlever le nom de domaine d'un identificateur de courrier électronique extrait d'une demande LRQ entrante de sorte que cet identificateur puisse être localisé comme s'il s'agissait d'un identificateur h323-ID.

IV.1.2 Communications extrémité à extrémité

Les extrémités qui souhaitent recevoir des appels provenant de points d'extrémité en dehors de la zone relevant de leur portier doivent utiliser l'accès suivant pour la voie de signalisation d'appel:

- accès de signalisation d'appel TCP d'extrémité 1720

On peut utiliser des valeurs dynamiques pour ces accès afin de pouvoir placer plusieurs points d'extrémité sur un seul dispositif, mais il faut savoir que cela empêchera l'interfonctionnement avec les extrémités en dehors de la zone relevant du portier, sauf via une passerelle dans la zone.

IV.2 SPX/IPX

Il convient de noter que compte tenu de l'absence de réassemblage dans le réseau des grands paquets, l'utilisation de la fragmentation des macroblocs est essentielle.

Applications avec remise non fiable	Canal H.245 et canal de signalisation d'appel
PXP	SPX
IPX	
Couche Liaison	
Couche Physique	

IV.2.1 Découverte du portier

Dans la terminologie IPX, une "prise" ("socket") est équivalente à un accès dans IP et un "identificateur TSAP" dans la présente Recommandation et dans l'UIT-T H.323.

Sur les réseaux de type IPX, les portiers doivent faire connaître le "type de service portier" défini ci-après pour permettre aux extrémités de les localiser dans un réseau. De même, les extrémités doivent demander à connaître le "type de service portier" pour localiser le portier le plus proche.

- Type de service portier A étudier.

NOTE – Le type de service est appelé prise SAP dans certains documents IPX.

IV.2.2 Communication d'extrémité à extrémité

Les extrémités qui souhaitent recevoir des appels en provenance de points d'extrémité en dehors de la zone de leur portier doivent utiliser les "prises" suivantes pour la signalisation d'appel.

- Accès de signalisation d'appel IPX d'extrémité A étudier.

On peut utiliser des valeurs dynamiques pour ces "prises" pour pouvoir placer plusieurs points d'extrémité dans un seul dispositif, mais on doit comprendre que cela gênera l'interfonctionnement avec les extrémités en dehors de la zone relevant du portier, sauf lorsque cela s'effectue via une passerelle située dans la zone.

APPENDICE V

Utilisation de la notation ASN.1 dans la présente Recommandation

Le présent appendice énumère les conventions de notation ASN.1 qui ont été utilisées dans la présente Recommandation, dont les futures révisions ne devront utiliser que ces créations syntaxiques. D'autres créations ASN.1 ne seront prises en considération que dans des circonstances exceptionnelles.

V.1 Balisage

Toutes les balises de la présente Recommandation sont du type AUTOMATIC TAG.

V.2 Types

Les types suivants peuvent apparaître 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

V.3 Contraintes et étendues

La présente Recommandation utilise des contraintes de longueur ("SIZE") pour les chaînes, les ensembles de type SET OF et SEQUENCE OF, pour les étendues des valeurs d'entiers et pour les alphabets permis ("FROM").

V.4 Extensibilité

La présente Recommandation utilise le marqueur d'extension (signe de troncation comme: "...").

APPENDICE VI

Identificateurs H.225.0 des protocoles de signalisation canalisés en tunnel

La présente Recommandation prend en charge la canalisation en tunnel des protocoles de signalisation d'appel autres que H.323, comme décrit au 10.4/H.323. La série des Annexes M/H.323 (M.1/H.323, M.2/H.323, etc.) définit la canalisation en tunnel pour des protocoles spécifiques. Dans la présente Recommandation, un protocole canalisé en tunnel est identifié par des informations insérées dans la structure ASN.1 **TunnelledProtocol** qui est définie au 7.6 et dans l'Annexe H. Le

présent appendice énumère les identificateurs de la structure **TunnelledProtocol** qui ont été attribués à des protocoles en tunnel spécifiques.

Les protocoles en tunnel qui sont définis dans la présente Recommandation sont décrits dans les Tableaux VI.1 et VI.2. Noter que la canalisation en tunnel n'est pas limitée aux protocoles énumérés dans ces tableaux.

Tableau VI.1/H.225.0 – Protocoles en tunnel identifiés par tunnelledProtocolObjectID

Tunnelling specification	Protocol specification	tunnelledProtocolObjectID	subIdentifier
M.1/H.323	ISO/IEC 11572 and 11582	{iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9)}	(None)
M.2/H.323	ITU-T Q.763 (1988)	{itu-t (0) recommendation (0) q (17) 763}	"1988"
M.2/H.323	ITU-T Q.763 (1993)	{itu-t (0) recommendation (0) q (17) 763}	"1993"

Tableau VI.2/H.225.0 – Protocoles en tunnel identifiés par TunnelledProtocolAlternateIdentifier

Tunnelling Specification	Protocol specification	protocolType	protocolVariant	subIdentifier
M.2/H.323	ANSI T1.113-1988	"isup"	"ANSI T1.113-1988"	"1988"
M.2/H.323	ETS 300 121	"isup"	"ETS 300 121"	"121"
M.2/H.323	ETS 300 356	"isup"	"ETS 300 356"	"356"
M.2/H.323	BELLCORE GR-317	"isup"	"BELLCORE GR-317"	"317"
M.2/H.323	JT-Q761-4(1987-1992)	"isup"	"JT-Q761-4(1987-1992)"	"87"
M.2/H.323	JT-Q761-4(1993)	"isup"	"JT-Q761-4(1993)"	"93"

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	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	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication