Union internationale des télécommunications

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

H.225.0

(05/2006)

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Multiplexage et synchronisation en transmission

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

Recommandation UIT-T H.225.0

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

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

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

Recommandation UIT-T H.225.0

Protocoles de signalisation d'appel et paquétisation 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 et F.720, lorsque sur le trajet de transmission se trouvent un ou plusieurs réseaux à commutation de 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 qu'exigés par la Rec. UIT-T H.320 doivent être assurés par les terminaux. On remarque que la Rec. 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 Recommandations 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 de paquets afin d'assurer des services conversationnels dans des équipements de type H.323.

Les produits revendiquant la conformité à la version 6 de la Rec. UIT-T H.225.0 (présente version) doivent en satisfaire toutes les exigences obligatoires. Les produits de la version 6 peuvent être identifiés par des messages H.225.0 contenant une valeur de champ **protocolIdentifier** égale à {itu-t (0) recommendation (0) h (8) 2250 version (0) 6}.

La présente révision apporte les modifications suivantes:

1) extension de la structure H.225.0 AliasAddress aux fins de prise en charge des codes de chiffres 10 à 14;
2) adjonction de la capacité d'un portier à attribuer un alias E.164 à une extrémité n'enregistrant aucun alias par elle-même;
3) adjonction du code d'erreur "pas de largeur de bande" dans l'élément H.225.0 AdmissionRejectReason;
4) modifications de la notation ASN.1 et du texte requises pour les procédures relatives au portier attribué;
5) modification du § 7.5 pour ajouter la prescription visant à relancer la temporisation T310 en cas de réception d'une valeur Indicateur de progression égale à 1 ou 8;
6) modifications de la norme ASN H.225.0 pour prendre en charge les modifications H.361;
7) modifications de la définition et du texte ASN.1 pour prendre en charge l'adjonction du champ 'language' dans les structures LRQ et RRQ pour la nouvelle Rec. H.460.21 (ex H.460.MB);
8) correction d'une faute d'orthographe dans les commentaires relatifs à l'élément unallocatedNumber dans la spécification ASN.1.

La présente révision apporte également des clarifications ou corrige des erreurs précédemment recensées dans les guides d'implémentation: adjonction de tableaux de mappage relatifs aux éléments LocationRejectReason/AdmissionRejectReason et AccessRejectionReason/AdmissionRejectReason, clarification de la description de l'insertion d'éléments additionalSourceAddresses par un portier, clarification du texte sur l'utilisation du message Facility pour acheminer le champ h245Address et correction d'un texte décrivant la longueur du champ UUIE.

Source

La Recommandation UIT-T H.225.0 a été approuvée le 29 mai 2006 par la Commission d'études 16 (2005-2008) de l'UIT-T selon la procédure définie dans la Recommandation UIT-T A.8.

AVANT-PROPOS

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

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

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

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

NOTE

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

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

A la date d'approbation de la présente Recommandation, l'UIT avait été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux développeurs de consulter la base de données des brevets du TSB sous http://www.itu.int/ITU-T/ipr/.

TABLE DES MATIÈRES

Page

Recommandation UIT-T H.225.0

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

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 de 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. La Rec. 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 de 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 de paquets, utilisant le même protocole de transport***. Ce réseau à commutation de 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 de 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 de 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 de 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 de 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.

La Rec. UIT-T H.323 et la présente Recommandation étendent les conférences de la Rec. UIT-T H.320 et les connexions de la Rec. UIT-T H.221 aux réseaux à commutation de paquets à qualité de service non garantie. En tant que tel, le modèle de conférence principal[1] 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 média pour tous les réseaux à commutation de 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

[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.

transport non RTP pour l'établissement de l'appel et la négociation de capacité (voir les Recommandations UIT-T H.323 et 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 la Rec. 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 de paquets 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 de 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 de 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 de 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 de 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 la Rec. UIT-T H.323 et de la 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 de la Rec. UIT-T 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 la Rec. 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).

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 de 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 la Rec. 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;

---

[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.

- 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 de paquets;
- les messages de signalisation d'appel H.245 et H.225.0 peuvent être envoyés sur les liaisons fiables offertes par le réseau à commutation de paquets;
- la souplesse et la puissance offertes par le protocole H.245 comparativement à celles offertes par le protocole H.242.

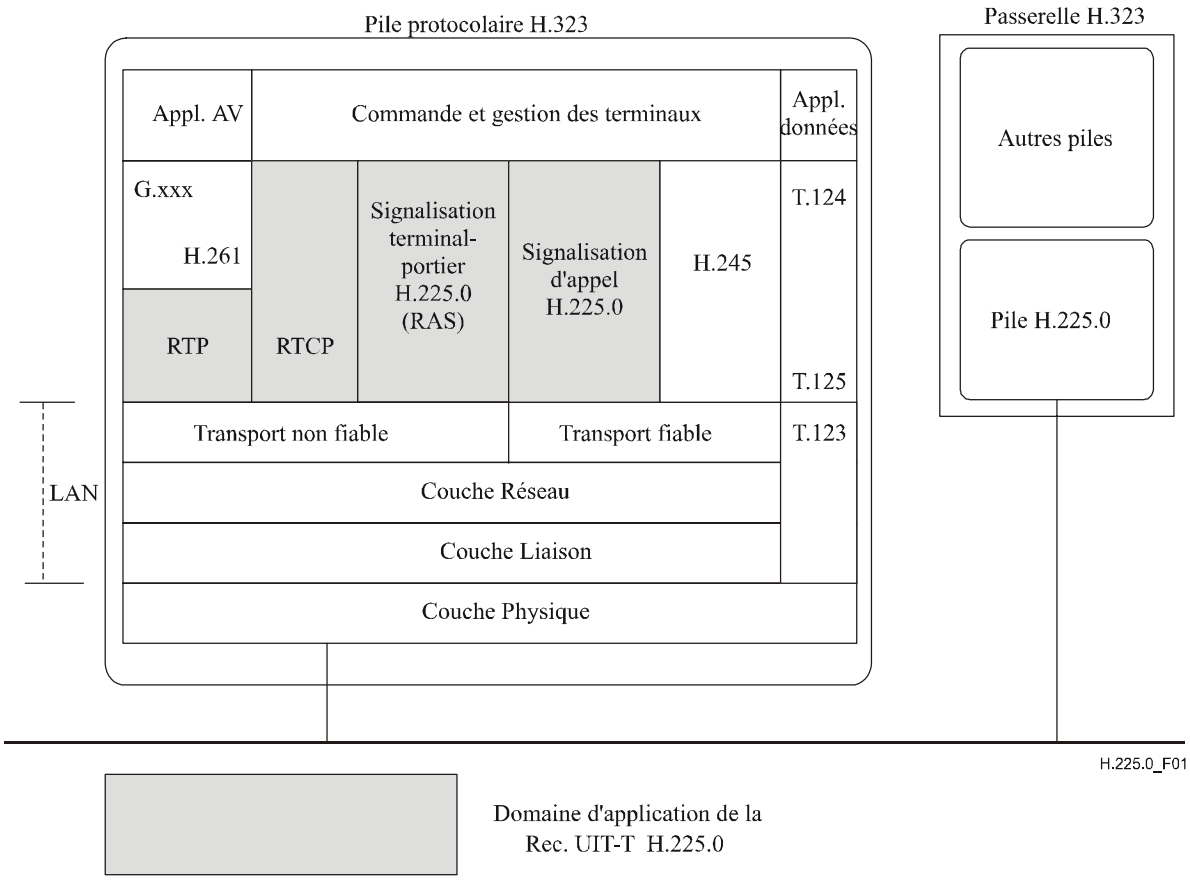


Figure 1/H.225.0 – Domaine d'application H.225.0

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. La référence à un document figurant dans la présente Recommandation ne donne pas à ce document, en tant que tel, le statut d'une Recommandation.

[1] Recommandation UIT-T G.711 (1988), *Modulation par impulsions et codage (MIC) des fréquences vocales*.

[2] Recommandation UIT-T G.722 (1988), *Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s*.

[3] Recommandation UIT-T G.728 (1992), *Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code*.

[4] Recommandation UIT-T G.723.1 (1996), *Codeurs vocaux: codeur vocal à double débit pour communications multimédias acheminées à 5,3 kbit/s et à 6,3 kbit/s*.

[5] Recommandation UIT-T G.729 (1996), *Codage de la parole à 8 kbit/s par prédiction linéaire avec excitation par séquences codées à structure algébrique conjuguée*.

[6] Recommandation UIT-T H.221 (2004), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels*.

[7] Recommandation UIT-T H.230 (2004), *Signaux de commande et d'indication synchrones de la trame pour les systèmes audiovisuels*.

[8] Recommandation UIT-T H.233 (2002), *Système de confidentialité pour les services audiovisuels*.

[9] Recommandation UIT-T H.242 (2004), *Procédures pour l'établissement de communications entre terminaux audiovisuels sur des canaux numériques d'un débit allant jusqu'à 2 Mbit/s*.

[10] Recommandation UIT-T H.243 (2005), *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] Recommandation UIT-T H.245 version 13 (2006), *Protocole de commande pour communications multimédias*.

[12] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à* $p \times 64$ *kbit/s*.

[13] Recommandation UIT-T H.263 (2005), *Codage vidéo pour communications à faible débit*.

[14] Recommandation UIT-T H.320 (2004), *Systèmes et équipements terminaux visiophoniques à bande étroite*.

[15] Recommandation UIT-T T.122 (1998), *Service de communication multipoint – Définition du service*.

[16] Recommandation UIT-T T.123 (1999), *Piles de protocoles de données propres au réseau pour conférences multimédias*.

[17] Recommandation UIT-T T.125 (1998), *Spécification du protocole du service de communication multipoint*.

[18] Recommandation UIT-T H.321 (1998), *Adaptation des terminaux visiophoniques H.320 aux environnements RNIS à large bande*.

[19] Recommandation UIT-T H.322 (1996), *Systèmes et équipements terminaux visiophoniques pour réseaux locaux offrant une qualité de service garantie*.

[20] Recommandation UIT-T H.324 (2005), *Terminal pour communications multimédias à faible débit*.

[21] Recommandation UIT-T H.310 (1998), *Systèmes et terminaux de communication audiovisuels à large bande*.

[22] Recommandation 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] Recommandation UIT-T Q.932 (1998), *Système de signalisation d'abonné numérique n° 1 – Procédures génériques pour la commande des services complémentaires RNIS*.

[24] Recommandation UIT-T X.680 (2002) | ISO/CEI 8824-1:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

[25] Recommandation UIT-T X.681 (2002) | ISO/CEI 8824-2:2002, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels*.

[26] Recommandation UIT-T X.691 (2002) | ISO/CEI 8825-2:2002, *Technologies de l'information – Règles de codage ASN.1 – Spécification des règles de codage compact*.

[27] Recommandation UIT-T E.164 (2005), *Plan de numérotage des télécommunications publiques internationales*.

[28] ISO/CEI 10646:2003, *Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC)*.

[29] Recommandation 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] Recommandation UIT-T Q.950 (2000), *Protocoles pour services complémentaires, structure et principes généraux*.

[31] Recommandation UIT-T H.235.0 (2005), *Cadre de sécurité H.323: cadre de sécurité pour les systèmes multimédias de la série H (systèmes H.323 et autres systèmes de type H.245)*.

[32] ISO/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] Recommandation UIT-T H.248.1 (2005), *Protocole de commande de passerelle version 3*.

[37] IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.

[38] IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control*.

[39] IETF RFC 2032 (1996), *RTP Payload Format for H.261 Video Streams*.

[40] Recommandation UIT-T X.690 (2002) | ISO/CEI 8825-1:2002, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives*.

3 Définitions

Voir les définitions de la Rec. 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 (*bit rate allocation signal*)

CIF format intermédiaire commun (*common intermediate format*)

CRV valeur de référence d'appel (*call reference value*)

ECS signal de commande de chiffrement (*encryption control signal*)

FFS à étudier

GOB groupe de blocs (*group of blocks*)

H-MLP protocole multicouche à grande vitesse (*high speed multi-layer protocol*)

HSD données à grande vitesse (*high speed data*)

IA5 alphabet international n° 5 (*international alphabet n° 5*)

IE élément d'information (*information element*)

IETF groupe de travail d'ingénierie Internet (*Internet engineering task force*)

IP protocole Internet (*Internet protocol*)

LAN réseau local (*local area network*)

LD-CELP prédiction linéaire à faible délai à excitation par code (*low delay – code excited linear prediction*)

LSB bit de plus faible poids (*least significant bit*)

LSD données à faible vitesse (*low speed data*)

MB macrobloc (voir la Rec. UIT-T H.261)

MBE extension multioctet (*multi-byte extension*)

MCC conférence à commande multipoint (*multipoint command conference*)

MCN négation à commande multipoint (*multipoint command negating*)

MCS service de communication multipoint (*multipoint communication service*)

MCS transmission de données symétriques de commande multipoint (*multipoint command symmetrical data transmission*)

MCU pont de conférence; unité de commande multipoint (*multipoint control unit*)

MF multitrames (*multiframe*)

MIC modulation par impulsions et codage

MLP protocole multicouche (*multi-layer protocol*)

MPI intervalle d'image minimal (*minimum picture interval*)

MSB bit de plus fort poids (*most significant bit*)

NA non applicable

NS non normalisé (*non-standard*)

NSAP point d'accès au service de réseau (*network service access point*)

PDU unité de données protocolaire (*protocol data unit*)

QCIF quart de format intermédiaire commun (*quarter common intermediate format*)

QS qualité de service

RAS enregistrement, admission et statut (*registration, admission and status*)

RCC réseau à commutation de circuits

RTCP protocole de commande de transport en temps réel (*real-time transport control protocol*)

RTP protocole de transport en temps réel (*real-time transport protocol*)

SBE extension à un octet (*single byte extension*)

SC canal de service (*service channel*)

SCM mode de communication sélectionné (*selected communication mode*)

TCP protocole de commande de transport (*transport control protocol*)

TSAP point d'accès au service de transport (*transport service access point*)

UDP protocole datagramme d'utilisateur (*user datagram protocol*)

URL localisateur de ressource normalisé (*uniform resource locator*)

VCF commande vidéo "demande d'arrêt sur image" (*video command "freeze picture request"*)

VCU commande vidéo "demande d'actualisation rapide" (*video command "fast update request"*)

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

ACF confirmation d'admission (*admissions confirm*)

ARJ refus d'admission (*admissions reject*)

ARQ demande d'admission (*admissions request*)

BCF confirmation de largeur de bande (*bandwidth confirm*)

BRJ refus de largeur de bande (*bandwidth reject*)

BRQ demande de largeur de bande (*bandwidth request*)

DCF confirmation de désengagement (*disengage confirm)*

DRJ refus de désengagement (*disengage reject*)

DRQ demande de désengagement (*disengage request*)

GCF confirmation de portier (*gatekeeper confirm*)

GRJ refus de portier (*gatekeeper reject*)

GRQ demande de portier (*gatekeeper request*)

IACK accusé de réception de demande d'information (*information request acknowledgement*)

INAK accusé de réception négatif de demande d'information (*information request negative acknowledgement*)

IRQ demande d'information (*information request*)

IRR réponse à une demande d'information (*information request response*)

LCF confirmation de localisation (*location confirm*)

LRJ refus de localisation (*location reject*)

LRQ demande de localisation (*location request*)

RAC confirmation de disponibilité de ressources (*resource availability confirmation*)

RAI indication de disponibilité de ressources (*resource availability indication*)

RCF confirmation d'enregistrement (*registration confirm*)

RIP demande en cours (*request in progress*)

RRJ refus d'enregistrement (*registration reject*)

RRQ demande d'enregistrement (*registration request*)

SCI indication de commande de service (*service control indication*)

SCR réponse de commande de service (*service control response*)

UCF confirmation de non-enregistrement (*unregistration confirm*)

URJ refus d'annulation d'enregistrement (*unregistration reject*)

URQ demande d'annulation d'enregistrement (*unregistration request*)

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 la Rec. 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 de signalisation d'appel H.225 ont un champ pour un message non standard facultatif dans l'élément d'information Utilisateur à 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 d'enregistrement, admission et état (RAS) est appelé *canal RAS*. La méthode générale pour lancer un appel consiste à émettre d'abord une demande d'admission obligatoire sur le canal RAS[3], 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 fondée sur les opérations de signalisation d'appel H.225.0 avec les améliorations décrites ci-après. La séquence se termine lorsque le terminal reçoit, dans un message de connexion, une adresse de transport fiable à laquelle il envoie le message de commande H.245[4].

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 implémentations 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 de paquets (centralisée ou distribuée/multidiffusion) est négociée pour chaque connexion[5]. 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 réémis pour ne pas allonger les délais de manière excessive dans un environnement de réseau à commutation de paquets[6]. 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

---

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 Setup.

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 la Rec. 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.

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.

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 de paquets de la Rec. UIT-T T.123 selon le cas. Les capacités T.120 doivent être acheminées sur le réseau à commutation de 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 TRUE 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 d'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 de paquets et différer uniquement par l'identificateur TSAP, certains fabricants peuvent choisir d'utiliser différentes adresses de réseau à commutation de 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 RTP[7].

Le Tableau 1 décrit le cas élémentaire d'un mode d'exploitation unidiffusion 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 accès 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, ainsi que dans le cas d'une communication avec intervention du portier. La connexion sémaphore d'appel fiable doit être maintenue active jusqu'à la réception d'un message de fin de libération pour tous les appels actifs qui sont signalés dans le canal de signalisation d'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 de 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; la Rec. 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 de 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 la Rec. 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 de 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 implémentations devront correspondre aux implémentations 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 de 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 contribuante) 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 de 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 la Rec. UIT-T H.245 et/ou la Rec. UIT-T 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 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 la Rec 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 les Recommandations UIT-T H.245 et 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 de paquets ne doit pas correspondre à celle de la Rec. UIT-T H.221;
- la largeur de bande audio sur le réseau à commutation de paquets peut correspondre à celle de la Rec. 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 de 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 la Rec. UIT-T G.723.1, qui utilisent plusieurs tailles de trame audio, les limites de taille audio dans chaque paquet doivent êtres signalées au canal audio par signalisation dans la bande.

Pour les algorithmes audio à taille de trame fixe (voir les Recommandations UIT-T G.728 et 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 la Rec. 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 de 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 de paquets pour adapter le débit G.711 utilisé en H.320. Ainsi, du côté réseau à commutation de 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 de 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 la Rec. 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 de paquets pour que le débit corresponde au débit G.711 utilisé dans H.320. Ainsi, du côté réseau à commutation de 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 des Recommandations UIT-T H.261 ou H.263 doivent être utilisés dans le cas des codecs définis par l'UIT signalés dans la Rec. 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 la Rec. 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 dans la section 5 de RFC 2032 [39] 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 de 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 de 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 de 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 de paquets, conformément aux indications de la Rec. UIT-T T.123. Une comparaison entre les conférences de données centralisées ou décentralisées sur le réseau à commutation de paquets est faite dans la Rec. UIT-T H.323, et est négociée via le protocole H.245.

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

Voir la Rec. 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 de 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 la Rec. UIT-T Q.931, comme cela est spécifié dans la présente Recommandation. Les terminaux peuvent également prendre en charge les unités APDU facultatives H.450 contenues dans l'élément d'information Utilisateur à utilisateur. Les messages contiendront tous les éléments d'information obligatoires et pourront contenir tout élément d'information facultatif défini dans la Rec. 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 la Rec. 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 les messages de signalisation d'appel H.225.0, y compris ceux qui contiennent une unité APDU H.450 dans l'élément d'information Utilisateur à utilisateur. Elle doit avoir la capacité de traiter les messages de signalisation d'appel H.225.0 obligatoires et peut avoir celle de traiter les messages de signalisation d'appel H.225.0 facultatifs. Dans tous les cas, chaque extrémité H.225.0 devra pouvoir ignorer les messages qui lui sont inconnus 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 de signalisation d'appel H.225.0 et pour les unités APDU H.450 d'un élément d'information Utilisateur à utilisateur, selon le cas. 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 de signalisation d'appel H.225.0 ou dans une unité APDU 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 la Rec. UIT-T Q.931, sauf indication contraire dans la présente Recommandation. La Rec. 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 de signalisation d'appel H.225.0 et les éléments d'information facultatifs:

1) la passerelle devrait et le portier doit retransmettre, après modification convenable, tous les éléments d'information (facultatifs ou obligatoires) associés aux messages de signalisation d'appel H.225.0 obligatoires, soit du terminal vers la passerelle/le portier et en sens inverse. Cela inclut des éléments d'information telles les informations d'utilisateur à utilisateur et les informations d'affichage;

2) une passerelle devrait retransmettre dans les deux sens tous les messages de signalisation d'appel H.225.0, y compris ceux qui contiennent des unités APDU et des éléments d'information H.450;

3) un portier doit retransmettre dans les deux sens, après modification appropriée, tous les messages de signalisation d'appel H.225, y compris ceux qui contiennent des unités APDU et des éléments d'information H.450. 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 de signalisation d'appel H.225.0.

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 la Rec. 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 tunnélisation de la signalisation non H.323 dans les signaux H.225.0. Les détails sont dans l'Annexe M/H.323 (voir les § 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 la Rec. 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 la implémentation que si la signalisation indique un support, il n'existe pas de canaux B réels du RNIS du côté réseau à commutation de 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 de 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 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 de paquets.

Tableau 4/H.225.0 – Utilisation de messages Q.931/Q.932 par la Rec. UIT-T H.225.0

	Emission (M, F, O, CM) (Note 1)	**Réception et action (M, F, O (Note 2), CM)**
Messages d'établissement d'appel		
Alerting	M	M
Call Proceeding	O	CM (Notes 3 et 6)
Connect	M	M
Connect Acknowledge	F	F
Progress	O	CM (Note 6)
Setup	M	M
Setup Acknowledge	O	O
Messages de libération d'appel		
Disconnect	F	F
Release	F	F
Release Complete	M (Note 4)	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 (Note 6)
Notify	O	O
Status	M (Note 5)	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

Tableau 4/H.225.0 – Utilisation de messages Q.931/Q.932 par la Rec. UIT-T H.225.0

	Emission (M, F, O, CM) (Note 1)	**Réception et action (M, F, O (Note 2), CM)**
Retrieve Acknowledge	F	F
Retrieve Reject	F	F

NOTE 1 – M: obligatoire (*mandatory*), F: interdit (*forbidden*), O: facultatif (*optional*), CM: obligatoire sous condition (*conditionally mandatory*). Un message est obligatoire sous condition s'il est obligatoire lorsqu'une option est prise en charge.

NOTE 2 – 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.

NOTE 3 – Les terminaux utilisant des passerelles pourront recevoir le message Call Proceeding et y réagir.

NOTE 4 – Le message Release Complete est exigé afin de fermer le canal de signalisation d'appel fiable H.225.0, qui doit toutefois rester ouvert si d'autres communications l'utilisant sont encore en cours. Par ailleurs, le portier peut donner la valeur TRUE au fanion **maintainConnection** afin d'empêcher la fermeture du canal de signalisation d'appel.

NOTE 5 – 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.

NOTE 6 – Les extrémités qui prennent en charge des éléments de service facultatifs utilisant ces messages (tels que la tunnélisation H.245, les services complémentaires H.450, la tunnélisation des protocoles, ou les éléments de service qui utilisent la structure **genericData**) doivent traiter ces messages.

7.2 Eléments d'information Q.931 communs

7.2.1 Eléments d'information d'en-tête

Pour tous les messages de signalisation d'appel H.225.0, il y a trois champs communs qui sont obligatoires outre le type de message; ce type est décrit dans le présent 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 la Rec. UIT-T Q.931.

Noter que les valeurs de référence d'appel transmises dans les messages RAS doivent être conformes à la structure spécifiée dans la Rec. 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 Rec. UIT-T H.225.0 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 de 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 la Rec. 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 la Rec. 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 la Rec. UIT-T Q.2931 à la Rec. 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^{os} 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 la Rec. 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 la Rec. 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 de 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 à 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 la Rec. 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" et "9".

7.2.2.5 Sous-adresse de l'appelé

A utiliser comme indiqué dans la Rec. 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 la Rec. 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 de 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 à 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" et "9".

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. Les portiers acheminant des messages Setup H.225.0 envoyés par une extrémité H.323 peuvent insérer un numéro dans le champ **additionalSourceAddresses** avant de l'envoyer à son prochain destinataire.

7.2.2.7 Sous-adresse de l'appelant

Conformément à la Rec. UIT-T Q.931.

7.2.2.8 Cause

Lorsque cet élément d'information est reçu, les règles définies dans la Rec. 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 de paquets (voir Tableau 5). (Le mappage inverse n'est pas nécessaire car les entités du réseau à commutation de paquets sont tenues de décoder l'élément d'information Cause.)

Les passerelles doivent également effectuer le mappage des éléments **AdmissionRejectReason** et **LocationRejectReason** sur l'élément d'information Cause lors de l'envoi d'un message Release Complete au côté en commutation de circuits après réception d'un élément **AdmissionReject** ou **LocationReject** (Tableau 6).

Tableau 5/H.225.0 – Mappage de l'élément ReleaseCompleteReason sur 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	127 – Interfonctionnement, non spécifié
unreachableGatekeeper	38 – Réseau en dérangement
gatewayResources	42 – Encombrement de l'équipement de commutation
badFormatAddress	28 – Format de numéro non valide

Tableau 5/H.225.0 – Mappage de l'élément ReleaseCompleteReason sur l'élément d'information Cause

Code de l'élément ReleaseCompleteReason	Valeur de cause Q.931/Q.850 correspondante
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
securityWrongSyncTime	31 – Normal, non spécifié
securityReplay	31 – Normal, non spécifié
securityWrongGeneralID	31 – Normal, non spécifié
securityWrongSendersID	31 – Normal, non spécifié
securityMessageIntegrityFailed	31 – Normal, non spécifié
securityWrongOID	31 – Normal, non spécifié
securityDHmismatch	31 – Normal, non spécifié
securityCertificateExpired	31 – Normal, non spécifié
securityCertificateDateInvalid	31 – Normal, non spécifié
securityCertificateRevoked	31 – Normal, non spécifié
securityCertificateNotReadable	31 – Normal, non spécifié
securityCertificateSignatureInvalid	31 – Normal, non spécifié
securityCertificateMissing	31 – Normal, non spécifié
securityCertificateIncomplete	31 – Normal, non spécifié
securityUnsupportedCertificateAlgOID	31 – Normal, non spécifié
securityUnknownCA	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
InvalidCID	3 – Pas d'acheminement vers la destination
hopCountExceeded	3 – Pas d'acheminement vers la destination

Tableau 6/H.225.0 – Mappage des éléments AdmissionRejectReason/LocationRejectReason sur l'élément d'information Cause

Code de l'élément AdmissionRejectReason ou LocationRejectReason	Valeur de cause Q.931/Q.850 correspondante
calledPartyNotRegistered	20 – Abonné absent
invalidPermission	127 – Interfonctionnement, non spécifié
requestDenied	31 – Normal, non spécifié
undefinedReason	31 – Normal, non spécifié
callerNotRegistered	31 – Normal, non spécifié
routeCallToGatekeeper	Non applicable
invalidEndpointIdentifier	127 – Interfonctionnement, non spécifié
resourceUnavailable	47 – Ressource indisponible, non spécifiée
securityDenial	31 – Normal, non spécifié
qosControlNotSupported	63 – Service ou option indisponible, non spécifié
incompleteAddress	28 – Format de numéro non valide
aliasesInconsistent	31 – Normal, non spécifié
routeCallToSCN	3 – Pas d'acheminement vers la destination
exceedsCallCapacity	41 – Dérangement temporaire
collectDestination	31 – Normal, non spécifié
collectPIN	31 – Normal, non spécifié
genericDataReason	31 – Normal, non spécifié
neededFeatureNotSupported	31 – Normal, non spécifié
securityWrongSyncTime	31 – Normal, non spécifié
securityReplay	31 – Normal, non spécifié
securityWrongGeneralID	31 – Normal, non spécifié
securityWrongSendersID	31 – Normal, non spécifié
securityIntegrityFailed	31 – Normal, non spécifié
securityWrongOID	31 – Normal, non spécifié
secuirtyDHMismatch	31 – Normal, non spécifié
noRouteToDestination	3 – Pas d'acheminement vers la destination
unallocatedNumber	1 – Numéro non attribué
noBandwidthAvailable	34 – Pas de circuit/canal disponible

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 Elément 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.95.x, il doit être codé conformément au § 8.2.4/Q.932. Dans ce cas, les unités ADU de service doivent être constituées en fonction de l'élément ROSE (qui utilise la Rec. UIT-T X.680 (Spécification de l'ASN.1) et la Rec. UIT-T X.690 (Spécification des règles de codage de base pour l'ASN.1)) comme cela est défini dans la Rec. 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 la Rec. UIT-T H.450, l'élément d'information Utilisateur à utilisateur du message Facility 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.95.x, il doit être codé conformément au § 8.2.3/Q.932. Dans ce cas, les unités ADU de service doivent être constituées en fonction de l'élément ROSE (qui utilise la Rec. UIT-T X.680 (Spécification de l'ASN.1) et la Rec. UIT-T X.690 (Spécification des règles de codage de base pour l'ASN.1)) comme cela est défini dans la Rec. 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 la Rec. 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 la Rec. 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 à 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 à utilisateur doit être utilisé par toutes les entités H.323 pour acheminer les informations H.323 associées. L'information réelle Utilisateur à utilisateur à échanger entre les terminaux intervenant est emboîtée dans le champ **user-data** de l'unité PDU **H323-UserInformation** (qui ne fait l'objet d'aucune restriction).

Les restrictions suivantes sont applicables:

Longueur des contenus utilisateur à 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 aux Recommandations UIT-T X.680 et X.690 (ASN.1) ("00000101").

  NOTE – Ce codage est conforme à la révision de 1998 de la Rec. UIT-T Q.931 qui renvoie aux révisions précédentes de l'ASN.1. Les références correctes à l'ASN.1 sont les Recommandations UIT-T X.680 (syntaxe) et 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 codée en utilisant la variation "aligned" des règles de codage compactes spécifiées dans la Rec. 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 de signalisation H.225.0 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 à 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 à utilisateur. Cela est nécessaire afin de fournir le champ **callIdentifier**.

– **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.

– **h245Tunnelling** – Cet élément est mis à la valeur TRUE si la tunnélisation des messages H.245 est activée. Les systèmes conformes au moins à la version 4 de la Rec. UIT-T H.225.0 doivent mettre cet élément à la valeur TRUE si la procédure de connexion rapide est utilisée pour établir l'appel.

– **h245Control** – Ce champ contient une séquence d'unités PDU H.245 tunnélisées. 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 la Rec. UIT-T H.323.

– **tunnelledSignallingMessage** – Ce champ désigne un message de signalisation entièrement tunnelisé 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 tunnélisés, dans leur format binaire d'origine; cela permet l'agrégation de messages canalisés en un seul message H.225.0. Si le champ **tunnellingRequired** est présent, l'appel ne peut être établi que si la tunnélisation est prise en charge.

– **provisionalRespToH245Tunnelling** – Ce fanion est utilisé pour signaler que l'entité appelée n'a pas encore déterminé si la tunnélisation H.245 est applicable pour l'appel considéré. S'il est présent, le fanion **h245Tunnelling** 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 tunnélisation 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 de signalisation d'appel H.225.0 de type 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, non explicitement indiquée, de l'élément d'information Utilisateur à utilisateur correspond à la taille de la **séquence H323-UserInformation codée par les règles PER** . 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 7.

Tableau 7/H.225.0 – Contenu du message Alerting (Alerte)

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
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 à utilisateur	M	*

L'élément d'information Utilisateur à 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 la Rec. 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, Connect ou Facility.

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 d'appel rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'une voie logique. Il utilise la structure **OpenLogicalChannel** définie dans la Rec. 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 médias.

multipleCalls – S'il est à TRUE, 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 TRUE, 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 **alertingAddress** 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 8.

Tableau 8/H.225.0 – Contenu du message Call Proceeding (Appel en cours)

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
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 à utilisateur	M	*

L'élément d'information Utilisateur à 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 la Rec. 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 la Rec. 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 médias.

multipleCalls – S'il est à TRUE, 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 TRUE, 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 9 ci-dessous.

Tableau 9/H.225.0 – Contenu du message Connect (Connexion)

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
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
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 à utilisateur	M	*

L'élément d'information Utilisateur à 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 la Rec. 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, Call Proceeding ou Facility.

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 la Rec. 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 médias.

multipleCalls – S'il est à TRUE, 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 TRUE, 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 Connect 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 indiquées au Tableau 10.

Tableau 10/H.225.0 – Contenu du message Information

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
Fin de numérotation	O	1
Affichage	O	2-82
Fonctionnalité clavier	O	2-34
Signal	O	2-3
Numéro appelé	O (Note)	2-35
Utilisateur à utilisateur	M	*
NOTE – L'élément d'information Numéro appelé sera utilisé pour acheminer des numéros issus d'un plan de numérotage privé lors d'une émission avec chevauchement conformément au § 8.1.12/H.323.		

L'élément d'information Utilisateur à 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 la Rec. 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 11 ci-dessous.

Tableau 11/H.225.0 – Contenu du message Progress (Progression)

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
Capacité support	O	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 à utilisateur	M	*

L'élément d'information Utilisateur à 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 Rec. 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 doit être envoyée si elle a été envoyée antérieurement dans le message Call Proceeding, Alerting, Connect ou Facility.

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 la Rec. 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 médias.

multipleCalls – S'il est à TRUE, 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 TRUE, 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. 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 son principal objet est d'indiquer la fin de libération de ressources à commutation de circuits. Comme elle ne s'applique pas à un environnement de réseau à commutation de paquets, l'on utilise la méthode à une seule étape d'envoi du message de fin de libération uniquement.

Se conformer au Tableau 3-11/Q.931. Les modifications du Tableau 12 s'y appliquent.

Tableau 12/H.225.0 – Contenu du message Release Complete (Fin de libération)

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
Cause	CM (Note)	2-32
Fonctionnalité	O	8-*
Indicateur de notification	O	2-*
Affichage	O	2-82
Signal	O	2-3
Utilisateur à utilisateur	M	*
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 la Rec. UIT-T Q.931.

L'élément d'information Utilisateur à 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 la Rec. 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 tunnélisation est requise afin que l'appel soit établi. Une valeur de cause **hopCountExceeded** indique que l'appel a été rejeté parce que la valeur **hopCount** a atteint 0 et que l'appel ne peut donc plus progresser.

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 3-15/Q.931 tel que modifié par le Tableau 13.

Tableau 13/H.225.0 – Contenu du message Setup (Etablissement)

Elément 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
Fonctionnalité	O	8-*
Indicateur de progression	O	2-4
Fonctionnalités propres au réseau	F	NA

Tableau 13/H.225.0 – Contenu du message Setup (Etablissement)

Elément d'information	Statut H.225.0 (M/F/O/CM)	Longueur H.225.0
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
Compatibilité de couche inférieure	A étudier	NA
Compatibilité de couche supérieure	A étudier	NA
Utilisateur à utilisateur	M	*
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 de 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 à 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 la Rec. 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 contenu dans le message de signalisation d'appel H.225.0. 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 la Rec. 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 médias.

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

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

endpointIdentifier – 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 Setup est envoyé au portier auprès duquel l'extrémité est enregistrée; il ne doit pas être présent quand le message d'établissement est envoyé à une autre entité.

multipleCalls – S'il est à TRUE, 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 TRUE, 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 présent, 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.

hopCount – Ce champ spécifie une valeur d'entier afin d'indiquer le nombre de bonds que la signalisation d'appel peut encore effectuer.

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 Acquittement 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 14.

Tableau 14/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 à utilisateur	M	*

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4 de la Rec. UIT-T H.225.0, l'expéditeur de ce message ne doit pas inclure le champ **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information Utilisateur à utilisateur.

L'élément d'information Utilisateur à 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 Rec. UIT-T H.225.0 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 15.

Tableau 15/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 à utilisateur	M	*
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 de la Rec. UIT-T H.225.0, l'expéditeur de ce message ne doit pas inclure le champ **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information Utilisateur à utilisateur.

L'élément d'information Utilisateur à 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 la Rec. 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 16.

Tableau 16/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 à utilisateur	M	*
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 de la Rec. UIT-T H.225.0, l'expéditeur de ce message ne doit pas inclure le champ **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information Utilisateur à utilisateur.

L'élément d'information Utilisateur à 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 la Rec. 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 de signalisation d'appel H.225.0 de type Q.932

Les messages définis ci-dessous dérivent des Recommandations UIT-T Q.932 et H.450, auxquelles on se reportera 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 Utilisateur à utilisateur de message Facility 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 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.x. C'est pourquoi une ou plusieurs unités APDU de services complémentaires H.450 doivent être transportées à l'intérieur de l'élément d'information Utilisateur à utilisateur de message Facility. Les unités APDU de services complémentaires H.450 doivent être codées conformément au § 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 Utilisateur à utilisateur de message Facility, 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 à 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 la Rec. UIT-T Q.932 et codé conformément aux dispositions des Recommandations UIT-T Q.932 et Q.95.x 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 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 17.

Tableau 17/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 à utilisateur	M	*
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. NOTE 2 – Si le message Facility est utilisé pour transporter la signalisation de services complémentaires Q.95.x, 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.		

Codage de l'élément d'information Type de message

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

L'élément d'information Utilisateur à 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 la Rec. 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 appelée 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 la Rec. 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 TRUE, 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 TRUE, 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 Rec. UIT-T H.225.0.

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 la Rec. 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 18.

Tableau 18/H.225.0 – Notify (notification)

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
Capacité support	O (Note)	5-6
Indicateur de notification	M	3
Affichage	O	2-82
Utilisateur à utilisateur	M	*
NOTE – Inclus pour indiquer un changement de la capacité support.		

Afin d'assurer la rétrocompatibilité avec les systèmes antérieurs à la version 4 de la Rec. UIT-T H.225.0, l'expéditeur de ce message ne doit pas inclure les champs **h4501SupplementaryService** ou **h245Control** dans le champ **h323-message-body** de l'élément d'information Utilisateur à utilisateur.

L'élément Utilisateur à 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 la Rec. 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 de signalisation d'appel H.225.0

Les temporisations Q.931 suivantes doivent être prises en charge:

- la temporisation "d'appel présent" T303 (voir Tableaux 9-1/Q.931 et 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 envoyé par l'extrémité appelée après qu'elle a envoyé un message Setup. Cette temporisation doit être d'au moins 4 secondes. 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);
- la temporisation "d'appel reçu" T301 (voir Tableaux 9-1/Q.931 et 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 commence 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 min);
- la temporisation "d'émission avec chevauchement" T302 (voir Tableaux 9-1/Q.931 et 9-2/Q.931) qui définit le temps après lequel l'extrémité appelée doit arrêter d'attendre les chiffres composés par l'extrémité appelante lors de l'émission avec chevauchement. Cette temporisation commence lorsque le message SETUP ACK est émis ou que le message INFORMATION est reçu. Elle se termine normalement lors de la réception de l'indication d'envoi complet. Cette valeur de temporisation doit être de 10 à 15 secondes;
- la temporisation "de réception avec chevauchement" T304 (voir Tableaux 9-1/Q.931 et 9-2/Q.931) qui définit le temps après lequel l'extrémité appelante doit arrêter d'attendre les chiffres composés par l'extrémité appelée lors de la réception avec chevauchement. Cette temporisation commence lorsque le message SETUP ACK est émis, reprend lorsque le message INFORMATION est reçu et se termine normalement lors de la réception du message CALL PROCEEDING, ALERTING ou CONNECT. Cette valeur de temporisation doit être d'au moins 20 secondes;
- la temporisation "appel entrant en cours" T310 (voir Tableaux 9-1/Q.931 et 9-2/Q.931) qui définit le temps après lequel l'extrémité appelée doit arrêter d'attendre les chiffres composés par l'extrémité appelante lors de l'émission avec chevauchement. Cette temporisation commence lorsque le message CALL PROCEEDING est reçu et se termine normalement lors de la réception du message ALERTING ou CONNECT, ou lorsque l'appelant met fin à la tentative d'appel et envoie le message Release Complete. La temporisation T310 doit être arrêtée et la temporisation T301 doit être lancée après réception d'un élément d'information Indicateur de progression ayant pour valeur 1 ou 8. Cette valeur de temporisation doit être d'au moins 10 secondes;
- la temporisation "état d'appel" T322 (voir Tableaux 9-1/Q.931 et 9-2/Q.931) qui définit le temps après lequel l'extrémité appelée doit arrêter d'attendre le message STATUS en réponse au message STATUS ENQUIRY qu'elle a envoyé. Cette temporisation commence lorsque le message STATUS ENQUIRY est envoyé. Elle se termine normalement lors de la réception du message STATUS. Cette valeur de temporisation doit être d'au moins 4 secondes.

Il convient de noter que les valeurs de ces temporisations côté réseau à commutation de 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.x 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 à utilisateur des messages de signalisation d'appel.

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 OCTET STRING (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 OCTET STRING (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 OCTET STRING (chaîne d'octets).

Il convient de noter que ces structures n'utilisent pas d'adresse de transport = langage "adresse de réseau à commutation de paquets plus identificateur TSAP" de la Rec. 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 TRUE. 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 type d'extrémité simple (SET, *simple endpoint type*) 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 **subIdentifier** permet la spécification d'une version particulière d'un protocole normalisé.

La structure **TunnelledProtocolAlternateIdentifier** 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. Le champ **enterpriseNumber** désigne le fabricant et est attribué par l'Autorité chargée de l'assignation des numéros Internet (IANA, *Internet Assigned Numbers Authority*).

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 de 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. Lorsqu'elle enregistre chez un portier un code de signal d'adresse ISUP, une extrémité doit utiliser le champ **isupNumber**.

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 19. 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 de signalisation d'appel H.225.0 et comme une adresse pseudonyme contenant des chiffres composés manuellement **dialedDigits AliasAddress** lorsqu'elle le code dans des messages de signalisation RAS.

Tableau 19/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 UUIE 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** de **PartyNumber AliasAddress**, **TypeOf Number** = **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** de **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** de **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") (Note 1)	**privateNumber** de **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** de **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** de **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** de **PartyNumber AliasAddress**, **TypeOfNumber** = **nationalNumber**

Tableau 19/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 UUIE 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** de **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** de **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** de **PartyNumber AliasAddress**, **TypeOfNumber** = **abbreviatedNumber**
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. NOTE 2 – Une structure **privateTypeOfNumber** = **Unknown PartyNumber AliasAddress** doit être traitée comme une structure **AliasAddress** = **dialedDigits**.			

La structure **MobileUIM** représente un module d'identification pour les systèmes compatibles avec les réseaux sans fil de 2e et de 3e 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 (*international mobile station identification*).
- **min** – Cette option concerne les numéros d'identification de station mobile (*mobile identification number*).
- **mdn** – Cette option concerne les numéros d'annuaire du service mobile (*mobile directory number*).
- **msisdn** – Cette option concerne les numéros de station mobile RNIS (*mobile station ISDN number*).
- **esn** – Cette option concerne les numéros de série électronique (*electronic serial number*).
- **mscid** – Cette option concerne les numéros de commutateur du service mobile plus les numéros d'identification de marché ou de système (*mobile switching centre number plus market identification or system identification number*).
- **sid** – Cette option concerne les numéros d'identification de système (*system identification number*).

- **mid** – Cette option concerne les numéros d'identification de marché (*market identification number*).
- **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 (*SIM electronic serial number*).
- **soc** – Cette option concerne les codes d'opérateur de système (*system operator code*).

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 (*international mobile station identification*).
- **tmsi** – Cette option concerne l'identification temporaire de station mobile (*temporary mobile station identification*).
- **msisdn** – Cette option concerne les numéros de station mobile RNIS (*mobile station ISDN*).
- **imei** – Cette option concerne les numéros d'identification internationale d'équipements mobiles (*international mobile equipment identification*).
- **hplmn** – Cette option concerne les numéros publics d'abonné d'un réseau mobile terrestre (*home public land mobile network*).
- **vplmn** – Cette option concerne les numéros publics d'étranger d'un réseau mobile terrestre (*visiting public land mobile network*).

La structure **IsupNumber** représente un code de signal d'adresse ISUP défini dans la Rec. UIT-T Q.763. Les options offertes sont les suivantes:

- **e164Number –** Cas d'un code de signal d'adresse utilisant un plan de numérotage conforme aux Recommandations UIT-T E.163 ou E.164.
- **dataPartyNumber –** Cette option n'est pas utilisée pour l'instant.
- **telexPartyNumber –** Cette option n'est pas utilisée pour l'instant.
- **privateNumber –** Cas d'un code de signal d'adresse utilisant un plan de numérotage conforme à l'ISO/CEI 11571.
- **nationalStandardPartyNumber** – Cette option n'est pas utilisée pour l'instant.

La structure **IsupPublicPartyNumber** représente un code de signal d'adresse ISUP conforme à un plan de numérotage public:

- **natureOfAddress –** Type de numérotage utilisé pour le numéro considéré.
- **address –** Champ acheminant les chiffres du numéro.

La structure **IsupPrivatePartyNumber** représente un code de signal d'adresse ISUP conforme à un plan de numérotage privé:

- **privateTypeOfNumber –** Type de numérotage utilisé pour le numéro considéré.
- **address –** Champ acheminant les chiffres du numéro.

La structure **NatureOfAddress** représente un type de numérotage utilisé pour un numéro ISUP. Les options correspondent à l'indicateur de nature d'adresse (NOA, *nature of address*) défini dans la Rec. UIT-T Q.762. Les options offertes sont les suivantes:

- **unknown –** Il n'y a pas de type de numérotage spécifique ou le type de numérotage n'est pas connu.

- **subscriberNumber –** Cas d'un numéro d'abonné tel que défini dans la Rec. UIT-T Q.763.
- **nationalNumber –** Cas d'un numéro national tel que défini dans la Rec. UIT-T Q.763.
- **internationalNumber –** Cas d'un numéro international tel que défini dans la Rec. UIT-T Q.763.
- **networkSpecificNumber –** Cas d'un numéro spécifique de réseau tel que défini dans la Rec. UIT-T Q.763.
- **routingNumberNationalFormat –** Cas d'un numéro d'acheminement de réseau au format de numéro (significatif) national tel que défini dans la Rec. UIT-T Q.769.1.
- **routingNumberNetworkSpecificFormat –** Cas d'un numéro d'acheminement de réseau au format de numéro spécifique de réseau tel que défini dans la Rec. UIT-T Q.769.1.
- **routingNumberWithCalledDirectoryNumber –** Cas d'un numéro d'acheminement de réseau concaténé au numéro d'annuaire de l'appelé tel que défini dans la Rec. UIT-T Q.769.1.

Le mappage entre les codes d'adresse ISUP et les chiffres d'adresse H.225.0 dans la structure **IsupNumber** est le suivant (voir le Tableau 20):

Tableau 20/H.225.0 – Mappage entre les types de représentation de numéro

Code ISUP	Signal d'adresse ISUP	Chiffres IsupNumber H.225.0
0 0 0 0	chiffre 0	0
0 0 0 1	chiffre 1	1
0 0 1 0	chiffre 2	2
0 0 1 1	chiffre 3	3
0 1 0 0	chiffre 4	4
0 1 0 1	chiffre 5	5
0 1 1 0	chiffre 6	6
0 1 1 1	chiffre 6	7
1 0 0 0	chiffre 8	8
1 0 0 1	chiffre 9	9
1 0 1 0	réservé	A
1 0 1 1	chiffre 11	B
1 1 0 0	chiffre 12	C
1 1 0 1	réservé	D
1 1 1 0	réservé	E
NOTE – La valeur '1111' est mappée vers l'élément d'information Fin de numérotation Q.931 dans l'Annexe C/H.246.		

La structure **ExtendedAliasAddress** permet d'associer des informations communes à des adresses pseudonymes. La structure **presentationIndicator** indique si la présentation de l'**address** doit être autorisée ou interdite. La structure **screeningIndicator** indique si **address** 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** – **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** – Elément 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 et la valeur **sctp** désigne l'usage du protocole de transport des commandes de flux (SCTP, *stream control transmission protocol*).

La structure **AlternateGK** sert à indiquer une liste de portiers de remplacement ou de secours, ou le portier attribué:

- **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 TRUE 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 TRUE 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 FALSE si seul le message qui a causé le rejet doit être renvoyé. Ce fanion doit être mis à la valeur TRUE si un fanion **needToRegister** est mis à TRUE 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 GRQ. L'utilisation de la structure QSeriesOptions n'est pas encore définie dans l'attente d'un complément d'étude.

Les identificateurs **GloballyUniqueID** et **ConferenceIdentifier** sont supposés être uniques sur le plan mondial (**GloballyUniqueID**); leur utilisation est décrite dans la Rec. UIT-T H.323. L'identificateur **GloballyUniqueID** est codé avec l'octet zéro codé en premier. Il est constitué conformément au Tableau 21.

Tableau 21/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 Tableau 22).

Tableau 22/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 de 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 22;
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 une chaîne d'octets qui contiendra le descripteur **PackagesDescriptor** H.248, codé par les règles PER de notation ASN.1.

La structure **H248SignalsDescriptor** est est une chaîne d'octets qui contiendra le descripteur **SignalsDescriptor** H.248, codé par les règles PER de notation ASN.1.

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 de sortie.

La structure **CircuitIdentifier** désigne un dispositif permettant un compte rendu par une passerelle ou une sélection par un portier. La structure **CircuitIdentifier** 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 la Rec. 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 la Rec. 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 le champ **group**.

La structure **CarrierInfo** contient des renseignements sur la sélection de l'exploitant. La valeur **carrierIdentificationCode** désigne l'exploitant (par exemple le code d'identification d'exploitant contenu dans le message IAM de l'ISUP) choisi par l'abonné ou déterminé par les applications de routage, comme une chaîne binaire de chiffres. Le champ **carrierName** est un autre moyen d'identifier l'exploitant, sous la forme d'une chaîne de caractères ASCII.

La structure **carrier** est un code d'identification ou de sélection d'exploitant pour le routage des appels, déterminé par les applications de routage ou préféré par l'abonné.

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 la Rec. UIT-T H.248.1, 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'usager 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 TRUE si le message est expédié par l'émetteur du flux, ou à la valeur FALSE 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'usager. 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'usager, p. ex. "10.00$". La chaîne doit incure 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 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 **GenericIdentifier** 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 TRUE 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 FALSE.

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 § 6 et dans l'Annexe A.
- **ssrc** – Ce champ sert à identifier la source d'un flux RTP, comme décrit dans le § 6 et dans l'Annexe A.
- **sessionId** – Ce champ fournit l'identificateur de la session RTP en cours, comme décrit dans la Rec. UIT-T H.245.
- **associatedSessionIds** – Ce champ fournit les identificateurs des sessions RTP associées, comme décrit dans la Rec. 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.

La structure **RehomingModel** sert à indiquer le modèle utilisé par une extrémité pour l'identification et le réengistrement auprès de son portier attribué. Les options suivantes sont possibles:

- **gatekeeperBased** – L'extrémité se réengistrera auprès du portier attribué lorsque celui-ci lui en donnera l'instruction.
- **endpointBased** – L'extrémité interrogera le portier attribué et se réengistrera lorsque celui-ci lui répondra.

La structure **TransportQoS** sert à indiquer les capacités de réservation de ressources prises en charge par une extrémité. Les options suivantes sont possibles:

- **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.
- **qOSCapabilities** – Les capacités de qualité de service de l'extrémité sont décrites par les champs d'une structure **QoSCapability**.

7.7 Prise en charge requise des messages RAS

Le Tableau 23 montre les différents messages RAS qui sont pris en charge par différents types d'extrémité.

Tableau 23/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

Tableau 23/H.225.0 – Statut des messages RAS

Message RAS	Extrémité (émission)	Extrémité (réception)	Portier (émission)	Portier (réception)
RAC		O	M	
SCI	O	O	O	O
SCR	O	O	O	O

M obligatoire (*mandatory*)

O facultatif (*optional*)

F interdit (*forbidden*)

CM obligatoire sous condition (*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 (RAS). Le portier doit envoyer les messages RAS à cette adresse et non pas à l'adresse à partir de laquelle le message a été envoyé, à moins que la structure **rasAddress** ne puisse pas être décodée.

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.

algorithmOID – 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.

supportsAssignedGK – Indique si l'extrémité prend en charge le mécanisme de portier attribué.

assignedGatekeeper – Ce champ indique le portier attribué actuel de l'extrémité.

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.

assignedGatekeeper – Portier attribué de l'extrémité.

rehomingModel – Indique le mécanisme à utiliser par l'extrémité pour le réengistrement auprès du portier attribué.

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 TRUE si l'extrémité demandeur a fait précéder ce message de la procédure de recherche de portier; mis à FALSE 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 TRUE.

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. Le portier doit envoyer les messages RAS à cette adresse et non pas à l'adresse à partir de laquelle le message a été envoyé, à moins que la structure **rasAddress** ne puisse pas être décodée.

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 **terminalAlias** peut être attribuée par le portier et incluse dans le message RCF. Si le champ **terminalAlias** ne contient pas d'adresse **dialedDigits**, **partyNumber** ou **isupNumber**, une adresse **dialedDigits**.**partyNumber** ou

isupNumber peut être attribuée 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 TRUE, 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 TRUE 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.

willSupplyUUIEs – Si ce champ est mis à TRUE, il indique que l'extrémité fournira des informations de message de signalisation d'appel H.225.0 dans les messages IRR si le portier le demande.

maintainConnection – Si ce champ est mis à TRUE, 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 TRUE, 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 TRUE, 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.

restart – Ce champ indique, s'il est activé, qu'il s'agit de la première demande RRQ envoyée par l'extrémité après son redémarrage ou après un événement anormal ayant entraîné la perte de ses communications. Il permet au portier d'effectuer tout nettoyage ou toutes autres fonctions qui seraient nécessaires.

supportsACFSequences – Ce champ indique, s'il est activé, que l'extrémité est en mesure de recevoir et de traiter une séquence de messages ACF en réponse à un unique message de demande ARQ.

supportsAssignedGatekeeper – Indique si l'extrémité prend en charge le mécanisme de portier attribué.

assignedGatekeeper – Ce champ indique le portier attribué actuel de l'extrémité.

transportQoS – Une extrémité peut utiliser ce champ pour indiquer sa capacité à réserver des ressources de transport.

language – Indique le ou les langues dans lesquelles l'utilisateur préférerait recevoir les annonces et les invites.

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. Si ce champ n'est pas inclus et que des adresses pseudonymes aient été proposées dans le message RRQ, cela indique que le portier a accepté toutes les adresses pseudonymes proposées. Si ce champ est inclus et spécifie un sous-ensemble des adresses pseudonymes proposées dans le message RRQ, cela indique que le portier n'a accepté que ces adresses.

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

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. Noter que même si ces champs sont mis à TRUE, 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 TRUE, 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 FALSE, 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 TRUE, 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 TRUE, 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 FALSE, 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 TRUE, 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 chargé 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 TRUE, 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 présent, 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. Si ce champ n'est pas inclus et que des préfixes pseudonymes aient été proposés dans le message RRQ, cela indique que le portier a accepté tous les préfixes pseudonymes proposés. Si ce champ est inclus et spécifie un sous-ensemble des préfixes pseudonymes proposés dans le message RRQ, cela indique que le portier n'a accepté que ces préfixes.

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. Si ce champ n'est pas inclus et que les préfixes d'adresse ont été proposés dans le message RRQ, cela indique que le portier a accepté tous les préfixes proposés. Si ce champ est inclus et spécifie un sous-ensemble de préfixes d'adresses proposé dans le message RRQ, cela signifie que le portier n'a accepté que ces préfixes.

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.

assignedGatekeeper – Portier attribué de l'extrémité.

rehomingModel – Indique le mécanisme à utiliser par l'extrémité pour le réengistrement auprès du portier attribué.

transportQoS – Le portier peut utiliser ce champ pour indiquer le mécanisme de réservation de ressources à utiliser par l'extrémité.

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.

protocolIdentifier – 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. De toute façon, tous les pseudonymes, toutes les adresses et tous les préfixes pris en charge dans le message RRQ associé sont rejetés avec ceux qui sont spécifiés dans le champ **invalidTerminalAliases**. Une cause ayant 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**. Une cause ayant pour valeur

registerWithAssignedGK indique que la demande a été rejetée parce que le portier attribué est devenu disponible; l'extrémité s'enregistre auprès de son portier attribué lorsqu'il reçoit cette cause.

gatekeeperIdentifier – 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.

assignedGatekeeper – Portier attribué de l'extrémité.

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. Une cause ayant pour valeur **registerWithAssignedGK** indique que l'extrémité n'est pas enregistrée parce que le portier attribué est devenu disponible; l'extrémité s'enregistre auprès de son portier attribué lorsqu'il reçoit cette cause.

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.

assignedGatekeeper – Portier attribué de l'extrémité.

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.

assignedGatekeeper – Portier attribué de l'extrémité.

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 de 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.

endpointIdentifier – 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**. Si le portier n'assure pas la validation, il considèrera la première adresse enregistrée comme étant celle de la destination.

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 extraite des messages de signalisation d'appel H.225.0 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 TRUE, l'appelant incorpore un contrôleur multipoint activé; dans le cas contraire, sa valeur est FALSE.

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

canMapAlias – Si sa valeur est à TRUE, 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 FALSE, le portier devrait refuser le message ARQ. Les systèmes conformes à la version 4 de la Rec. UIT-T H.225.0 et aux versions ultérieures doivent mettre ce champ à TRUE.

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.

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**.

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 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.

willSupplyUUIEs – Si sa valeur est à TRUE, indique que l'extrémité fournira des informations de message de signalisation d'appel H.225.0 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 § 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 tunnélisation 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 tunnéliser en transparence des informations à travers le service RAS.

canMapSrcAlias – Ce champ, s'il est mis à la valeur TRUE, indique que si le message ACF résultant contient le champ **modifiedSrcInfo**, l'extrémité doit copier ces informations dans le champ **sourceInfo** du message Setup et/ou dans l'élément d'information Numéro de l'appelant, selon le cas. Si le portier remplace les informations d'adressage dans le message ARQ et que la valeur du champ **canMapSrcAlias** soit FALSE, alors le portier doit rejeter le message ARQ.

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 H.225.0, 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 – TRUE 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 **willSupplyUUIEs** sur TRUE. **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 TRUE, 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 FALSE, 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.

modifiedSrcInfo – Ce champ contient l'adresse pseudonyme qui devrait être utilisée par l'extrémité, comme **dialedDigits**, **PartyNumber** (**e164Number** ou **privateNumber**) ou **h323-ID**. Ce champ devrait être utilisé lors d'une traduction/modification de l'adresse pseudonyme de l'extrémité appelante et d'une tentative d'acheminement de l'appel vers la destination primaire ou vers l'une des extrémités de remplacement. Ces adresses ne devraient être utilisées par l'extrémité que pour l'appel considéré.

assignedGatekeeper – Portier attribué de l'extrémité.

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 FALSE). 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**. Il y a lieu que l'extrémité se réinscrive auprès du portier si elle reçoit une erreur de type **invalidEndpointIdentifier**. Une cause ayant pour valeur **registerWithAssignedGK** indique que la demande a été rejetée parce que le portier attribué est devenu disponible; l'extrémité s'enregistre auprès de son portier attribué lorsqu'il reçoit cette cause.

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.

assignedGatekeeper – Portier attribué de l'extrémité.

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 de 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.

endpointIdentifier – 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 contenue dans les messages de signalisation d'appel H.225.0 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).

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.

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 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 TRUE 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 § 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.

transportQoS – Une extrémité peut utiliser ce champ pour indiquer sa capacité à réserver des ressources de transport.

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 bits.

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.

transportQoS – Le portier peut utiliser ce champ pour indiquer le mécanisme de réservation de ressources à utiliser par l'extrémité.

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 bits 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.

endpointIdentifier – 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 TRUE, 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 FALSE, le portier devrait refuser le message LRQ. Les systèmes conformes à la version 4 de la Rec. UIT-T H.225.0 et aux versions supérieures doivent mettre ce champ à TRUE.

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 tunnélisation 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 réexpédité 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.

callIdentifier – Ce champ est un identificateur d'appel mondialement unique qui est fixé par l'extrémité d'origine et qui peut servir à associer la signalisation RAS à la signalisation de commande d'appel utilisée dans la présente Recommandation. Lors de l'envoi d'une demande LRQ à l'appui d'un message ARQ ou SETUP, le portier doit transférer l'identificateur d'appel contenu dans le message ARQ ou SETUP vers la demande LRQ. Une extrémité qui envoie un message LRQ afin de préparer l'établissement d'une communication doit remplir ce champ avec l'identificateur d'appel de cette communication. Les messages LRQ envoyés hors du contexte d'un appel ne doivent pas contenir le champ d'identificateur d'appel.

bandWidth – Ce champ indique la largeur de bande requise dans les deux sens pour l'appel, en unités de 100 bits. Par exemple, un appel à 128 kbit/s sera signalé comme étant une demande de 256 kbit/s. La valeur ne se rapporte qu'au débit audio et vidéo, à l'exclusion des en-têtes et du surdébit de service.

sourceEndpointInfo – Ce champ indique la séquence d'adresses pseudonymes pour l'extrémité d'origine, comme **dialedDigits**, **PartyNumber** (**e164Number** ou **privateNumber**) ou **h323-ID**. Il y a lieu que le portier copie ces informations pour l'extrémité au titre de laquelle il doit envoyer le message LRQ considéré ou, s'il réexpédie un message LRQ reçu, que le portier copie les informations **sourceEndpointInfo** à partir du message LRQ reçu.

canMapSrcAlias – S'il est mis à la valeur TRUE et si le message LCF résultant contient **modifiedSrcInfo**, ce champ indique que l'extrémité peut transférer ces renseignements dans le champ **sourceInfo** du message Setup. Si le message LRQ doit être envoyé par le portier à la suite de la réception d'une demande ARQ, le portier doit copier ce champ à partir de cette demande ARQ. Si le portier doit remplacer les informations d'adressage extraites du message LRQ et que le champ **canMapSrcAlias** ait la valeur FALSE, alors le portier doit rejeter ce message LRQ.

language – Indique le ou les langues dans lesquelles l'utilisateur préférerait recevoir les annonces et les invites.

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 H.225.0; 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 TRUE, 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 FALSE, 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.

modifiedSrcInfo – Ce champ contient l'adresse pseudonyme qui devrait être utilisée par l'extrémité, comme **dialedDigits**, **PartyNumber** (**e164Number** ou **privateNumber**) ou **h323-ID**. Ce champ devrait être utilisé lors d'une traduction/modification de l'adresse pseudonyme de l'extrémité appelante et d'une tentative d'acheminement de l'appel vers la destination primaire ou vers l'une des extrémités de remplacement. Si le message LCF se traduit par une réponse de confirmation ACF vers l'extrémité, ce champ doit être copié dans le message ACF.

bandWidth – Ce champ indique la largeur de bande maximale qui est permise pour l'appel. Sa valeur peut être inférieure à celle qui est demandée.

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.

endpointIdentifier – 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 contenue dans les messages de signalisation d'appel H.225.0 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).

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.

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 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 TRUE 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 § 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.

assignedGatekeeper – Portier attribué de l'extrémité.

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 de signalisation d'appel H.225.0 Status.

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 (acquittement positif) ou un message INAK (acquittement 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 **callIdentifier** – en pareil cas, le portier doit attribuer à l'identificateur **callIdentifier** 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.

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.

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 **willSupplyUUIEs** sur TRUE. **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 § 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.

assignedGatekeeper – Portier attribué de l'extrémité.

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 TRUE).

endpointType – Fournit des informations sur l'extrémité.

endpointIdentifier – 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 contenue dans un message de signalisation H.225.0 de l'appel sur lequel porte la réponse;
- **conferenceID** – Identificateur univoque de la conférence;
- **originator** – Lorsqu'il est égal à TRUE, l'extrémité qui fait l'objet de la demande est l'appelant, s'il est égal à FALSE, 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;
- **callSignalling** – Adresse de transport du canal de signalisation d'appel H.225.0;
- **callType** – Renseigne sur la topologie de l'appel;
- **bandwith** – 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 TRUE pour indiquer que l'extrémité a envoyé l'élément **h323pdu**; mis à FALSE 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 § 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 TRUE 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 TRUE), 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 FALSE), 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 IRQ 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 de la Rec. UIT-T H.323 et ultérieures doivent mettre ce champ à la valeur TRUE dans les messages de demande IRR non sollicitée, comme décrit dans le § 8.4.2/H.323 et doivent le mettre à la valeur FALSE 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 (acquittement 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 (acquittement 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 acquittement 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 la Rec. UIT-T H.323 et aux versions antérieures. Les extrémités conformes à la version 4 de la Rec. UIT-T H.323 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 TRUE, 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 FALSE.

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 BOOLEEN du champ **almostOutOfResources** car le champ **capacity** fournit des informations plus détaillées; cependant, l'opérateur BOOLEEN 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 24 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 24/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.

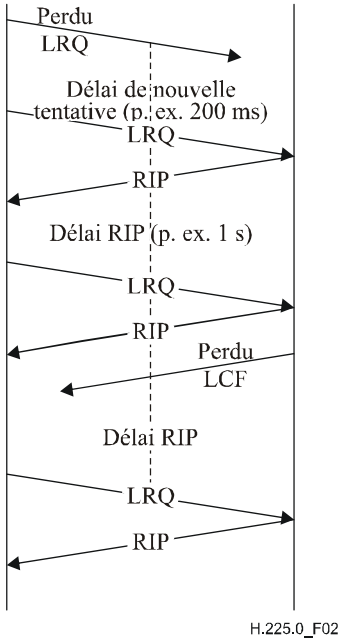


Figure 2/H.225.0 – Exemple d'utilisation du message RIP

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.

endpointIdentifier – 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 **callIdentifier, 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.

7.21 AdmissionConfirmSequence

La séquence AdmissionConfirmSequence contient un ou plusieurs messages ACF du service RAS. Elle peut être utilisée par le portier afin de répondre à un message ARQ unique au lieu du message ACF unique lorsqu'il y a différents jetons de sécurité, différentes informations d'origine traduites, etc., que l'on ne peut pas exprimer facilement dans un seul message ACF. Les extrémités indiquent leur prise en charge de la réception de la séquence AdmissionConfirmSequence en réglant le fanion **supportsACFSequences** dans le message RRQ.

7.22 Mappage du code d'erreur

Un portier devant renvoyer un message **AdmissionReject** en réponse à un message **AdmissionRequest** d'une extrémité, après réception d'un message **LocationReject** ou d'un message H.501 **AccessRejection** en réponse à son envoi d'un message **LocationRequest** ou **AccessRequest**, devrait utiliser les Tableaux 25 et 26 pour mapper le code d'erreur qu'il renvoie dans le message **AdmissionReject**.

Tableau 25/H.225.0 – Mappage de LocationRejectReason vers AdmissionRejectReason

LocationRejectReason	Elément AdmissionRejectReason correspondant
notRegistered	calledPartyNotRegistered
invalidPermission	invalidPermission
requestDenied	requestDenied
undefinedReason	undefinedReason
securityDenial	securityDenial
aliasInconsistent	aliasesInconsistent
routeCallToSCN	routeCallToSCN

Tableau 25/H.225.0 – Mappage de LocationRejectReason vers AdmissionRejectReason

LocationRejectReason	Elément AdmissionRejectReason correspondant
resourceUnavailable	resourceUnavailable
genericDataReason	genericDataReason
neededFeatureNotSupported	neededFeatureNotSupported
hopCountExceeded	noRouteToDestination
incompleteAddress	incompleteAddress
securityWrongSyncTime	securityWrongSyncTime
securityReplay	securityReplay
securityWrongGeneralID	securityWrongGeneralID
securityWrongSendersID	securityWrongSendersID
securityMessageIntegrityFailed	securityMessageIntegrityFailed
securityWrongOID	securityWrongOID
securityDHmismatch	securityDHmismatch
noRouteToDestination	noRouteToDestination
unallocatedNumber	unallocatedNumber

Tableau 26/H.225.0 – Mappage de AccessRejectionReason vers AdmissionRejectReason

AccessRejectionReason	Elément AdmissionRejectReason correspondant
noMatch	noRouteToDestination
packetSizeExceeded	undefinedReason
Security	securityDenial
hopCountExceeded	noRouteToDestination
needCallInformation	undefinedReason
noServiceRelationship	noRouteToDestination
undefined	undefinedReason
neededFeature	neededFeatureNotSupported
genericDataReason	genericDataReason
destinationUnavailable	resourceUnavailable
aliasesInconsistent	aliasesInconsistent
resourceUnavailable	resourceUnavailable
incompleteAddress	incompleteAddress
unknownServiceID	noRouteToDestination
usageUnavailable	undefinedReason
cannotSupportUsageSpec	undefiedReason
unknownUsageSendTo	undefinedReason

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 de 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 de 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 de 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 de 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 de paquets. Voir le paragraphe A.6.3.4 pour 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 de paquets conformément aux procédures décrites au § 8.4. 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 de 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

La Rec. 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 de 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 de 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 de 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 la Rec. 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

Les protocoles RTP et RTCP sont définis dans la référence IETF RFC 3550 [37], à laquelle il est également fait référence dans l'Appendice I. Aussi bien la présente annexe que l'Appendice I sont conservés dans la présente Recommandation afin d'assurer l'équivalence avec les versions antérieures de celle-ci.

On notera que toutes les références contenues en [37] visent une bibliographie et ne sont pas normatives, à l'exception de la référence à l'ISO/CEI 10646, qui apparaît également dans le paragraphe relatif aux références de la présente Recommandation.

On notera également que la terminologie utilisée en [37] diffère quelque peu de celle qui est utilisée dans la Rec. UIT-T H.323 et dans la présente Recommandation, comme indiqué 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 la référence [37] (RTP/RTCP)
Flux multimédia	Données
Adresse de transport	Adresse de transport
Adresse de réseau à commutation de 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 paragraphes ne doivent pas être considérés comme normatifs.

Annexe B

Profil RTP

Le profil RTP est défini dans la référence IETF RFC 3551 [38], à laquelle il est également fait référence dans l'Appendice II. Aussi bien la présente annexe que l'Appendice II sont conservés dans la présente Recommandation afin d'assurer l'équivalence avec les versions antérieures de celle-ci.

Voir l'introduction de l'Annexe A, dont toutes les mises en garde sont également applicables à la présente annexe.

Annexe C

Format de charge utile RTP pour les flux vidéo H.261

Le format de charge utile RTP pour les flux vidéo H.261 est défini dans la référence IETF RFC 2032 [39], à laquelle il est également fait référence dans l'Appendice III. Aussi bien la présente annexe que l'Appendice III sont conservés dans la présente Recommandation afin d'assurer l'équivalence avec les versions antérieures de celle-ci.

Voir l'introduction de l'Annexe A dont toutes les mises en garde sont également applicables à la présente annexe.

Annexe D

Format de charge utile RTP pour les flux vidéo H.261A

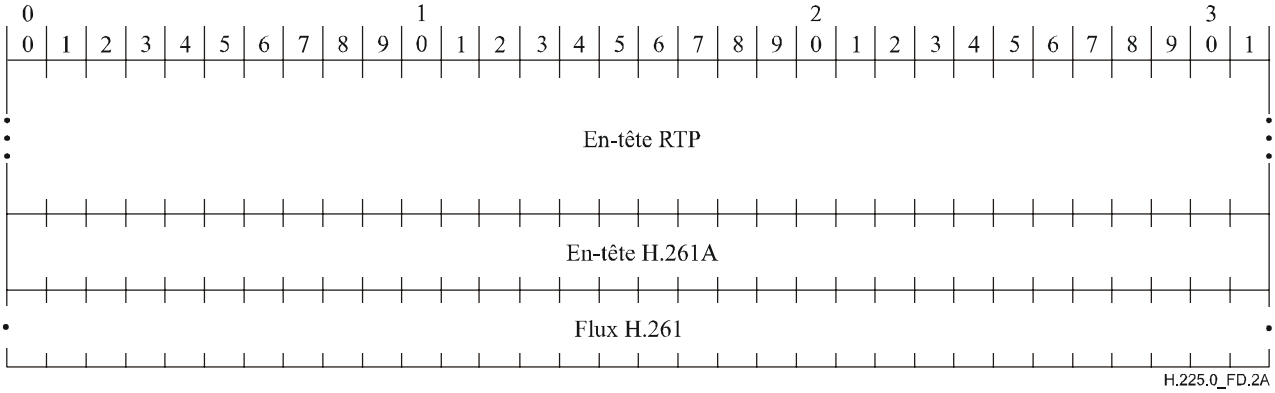
D.1 Introduction

Pour améliorer l'interfaçage des flux vidéo H.323 vers le RCC via des passerelles, la Rec. 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.

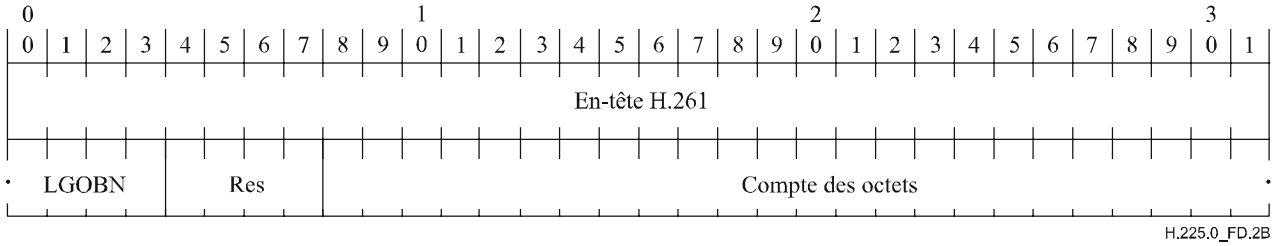
D.2 Mise en paquets RTP H.261A

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 la Rec. 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, *RTP Payload Format for MPEG-4 Audio/Visual Streams* (Format de charge utile des flux MPEG-4 en protocole RTP)

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 1998) de la Rec. 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 la Rec. 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.

La Norme RFC 3551 [38] section 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, *RTP Payload Format for MPEG-4 Audio/Visual Streams* (Format de charge utile des flux MPEG-4 en protocole RTP)

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, *RTP Payload Format for MPEG1/MPEG2 Video* (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 aux Tableaux 5/G.723.1 et 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.

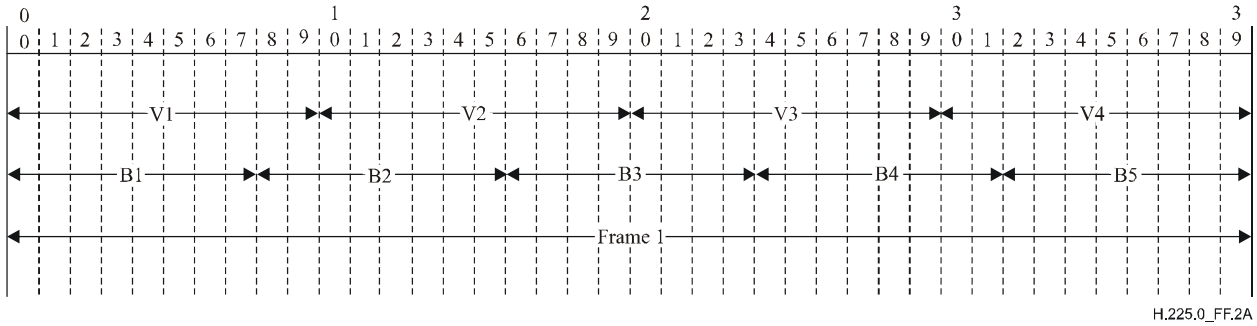
La mise en paquets G.723.1 est conforme à l'Annexe B sauf pour l'intervalle de mise en paquets (30 ms et non 20 ms, qui est la valeur par défaut):

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 8000 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).



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 rafales 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 la Rec. UIT-T G.729 et dans les Annexes A/G.729 et 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 les Annexes B/G.729, F/G.729 et G/G.729 occupe 2 octets, comme indiqué sur la Figure F.1.

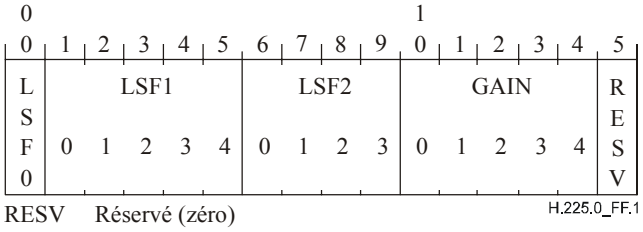


Figure F.1/H.225.0 – Format de mise en paquets de générateur CNG selon les Annexes B/G.729, F/G.729 et G/G.729

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.

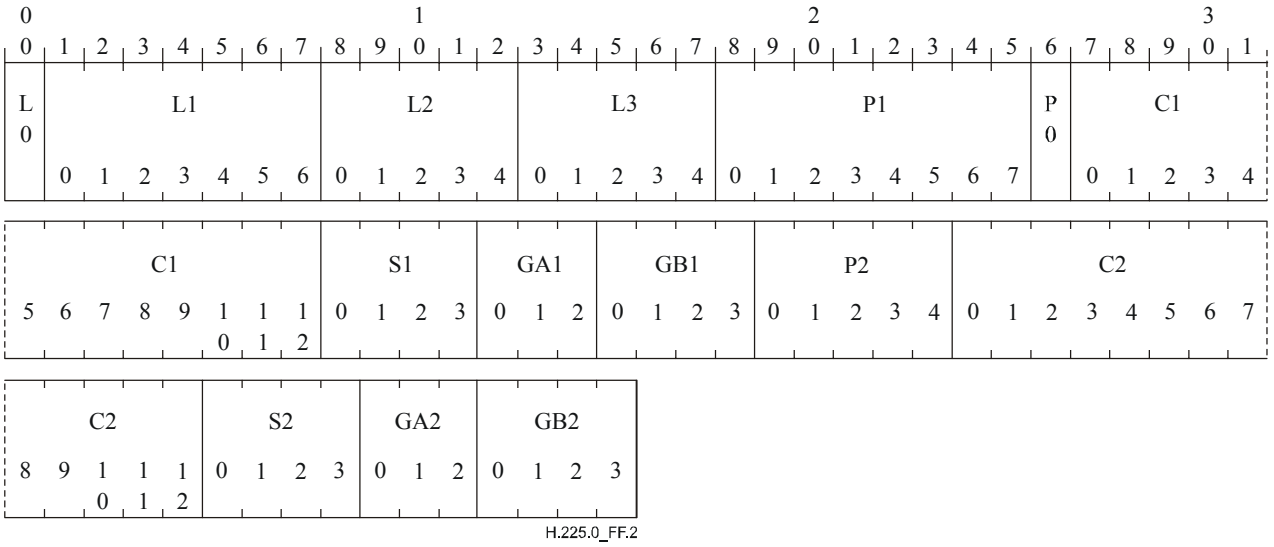


Figure F.2/H.225.0 – Format de mise en paquets selon G.729, Annexe A/G.729 et Annexe C/G.729

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.

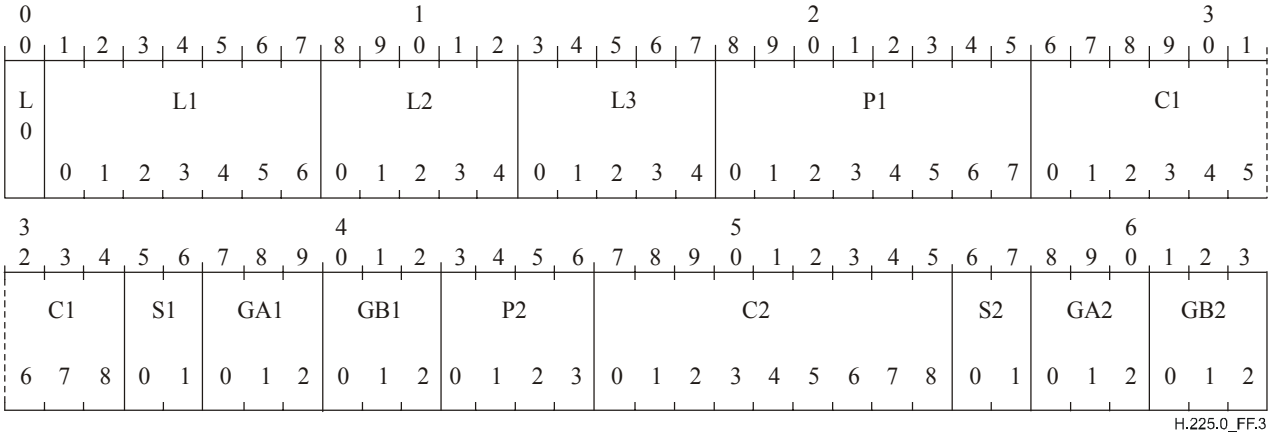


Figure F.3/H.225.0 – Format de mise en paquets G.729-6,4

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.

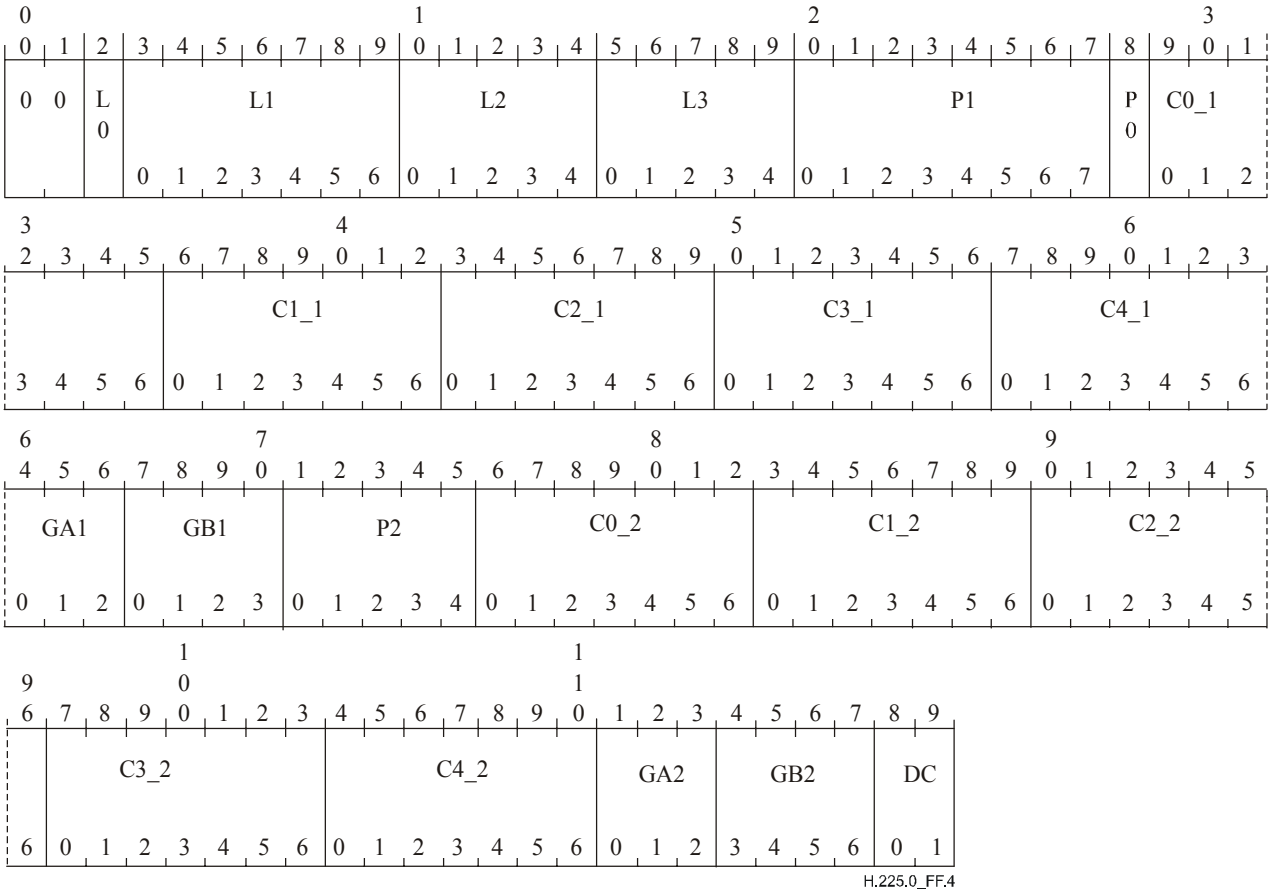


Figure F.4/H.225.0 – Format de mise en paquets G.729-12 pour le mode adaptatif anticipé

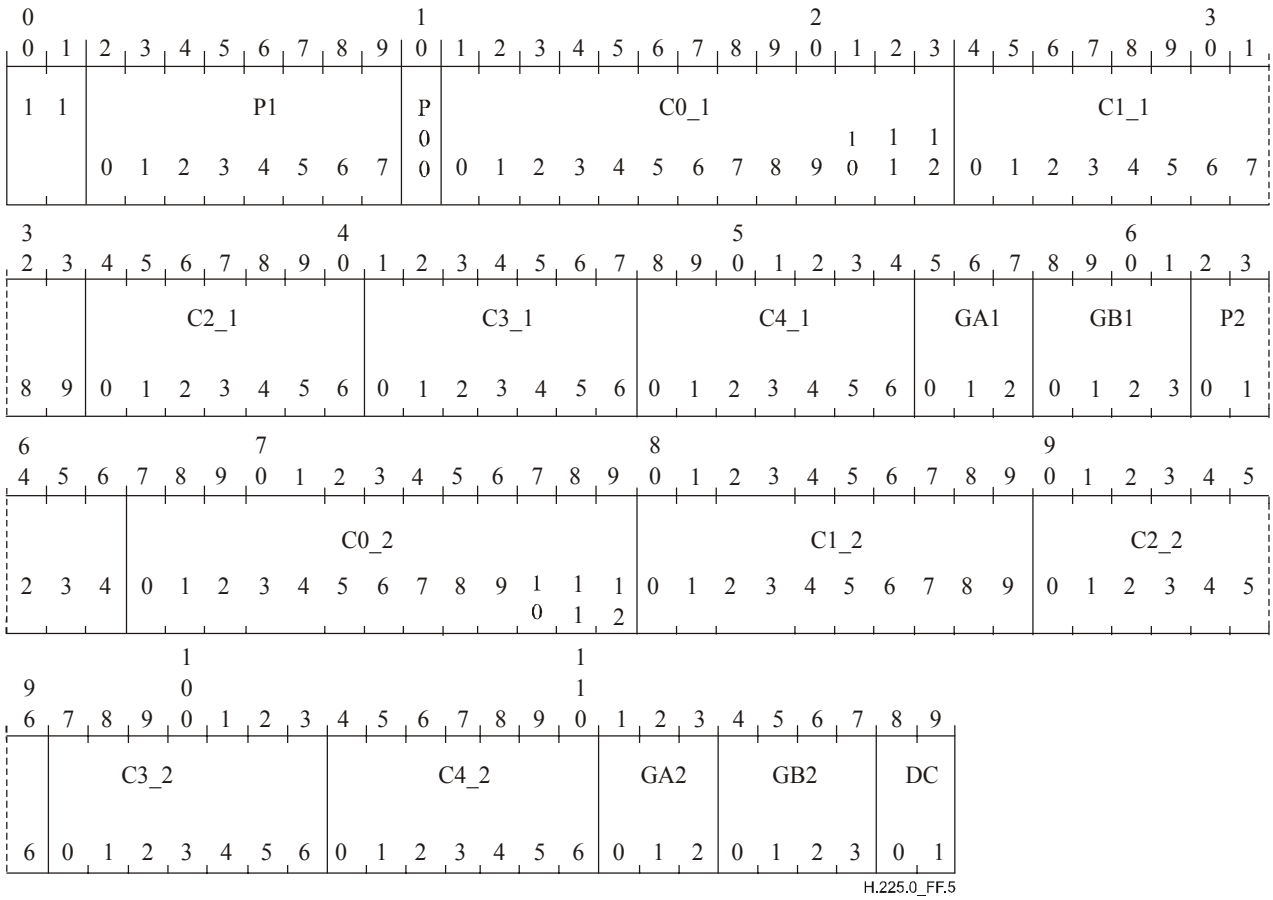


Figure F.5/H.225.0 – Format de mise en paquets G.729-12 pour le mode adaptatif différé

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 la présente Recommandation, 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 1 binaire.

Si une extrémité met le paramètre `comfortNoise` à `TRUE`, 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 bit/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 bit/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 bit/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 300 964), *Digital cellular telecommunications system; Full rate speech; Discontinuous Transmission (DTX) 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), *Digital cellular telecommunications system; 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 la Rec. 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, la Rec. 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.

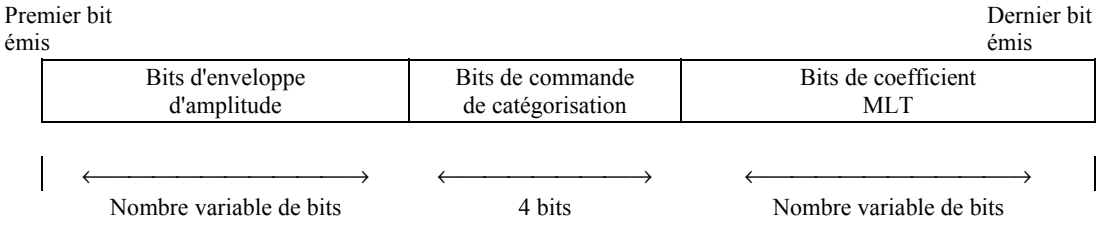


Figure F.6/H.225.0 – Ordre de transmission des principaux champs du flux binaire G.722.1

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.

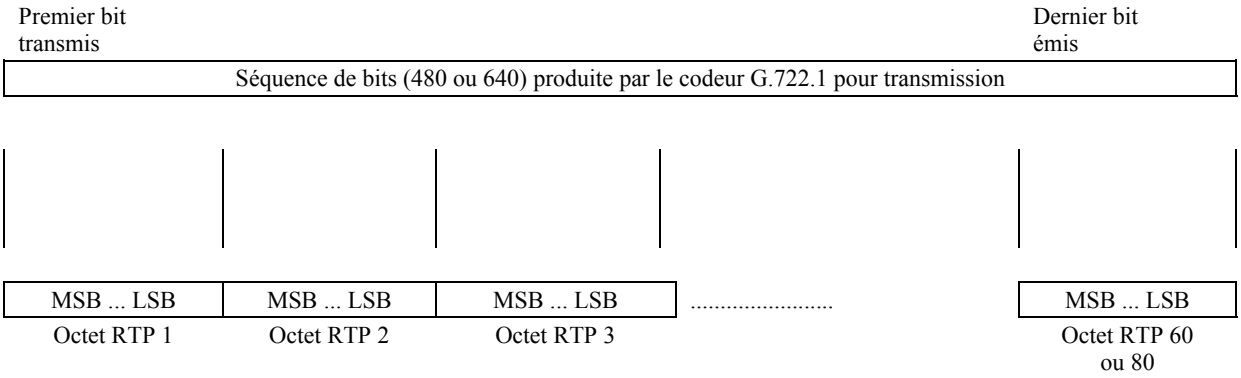


Figure F.7/H.225.0 – Mappage sur le protocole RTP du flux codé G.722.1

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 TDMA. 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 indicator*) 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. 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 trois fanions et sept bits de bourrage (formant une séquence de zéros) afin de constituer une trame de six octets.

La trame CN de 48 bits (6 octets) est décrite par la Figure F.9. 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 Généralités

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-95 de l'association TIA/EIA. La fréquence d'échantillonnage est de 8000 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 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 (kbit/s)	Débit résultant (kbit/s)
1	Plein	171	8,55	9,6
2	Demi	80	4,0	4,8
3 (Note)	Quart (compatibilité avec option de service 1)	40		
4	Huitième	16	0,8	1,2
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. 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. 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. 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 la Rec. 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, la Rec. 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.

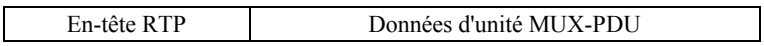
L'utilisation des champs de l'en-tête des paquets RTP doit être la suivante:

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

Communications administratives interdomaniales et intradomaniales

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'éléments 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.

C'est pourquoi il convient également de spécifier un protocole efficace entre les éléments H.323 d'un même domaine administratif.

La présente annexe décrit des méthodes permettant de traiter la résolution d'adresse, l'autorisation d'accès, et la communication de rapports d'utilisation intra et inter domaines administratifs de systèmes H.323 en vue de l'établissement d'appels. Les éléments H.323 mis en communication selon les procédures décrites dans la présente annexe sont appelés éléments homologues. Un domaine administratif se présente vis-à-vis d'autres domaines administratifs sous la forme d'un élément logique appelé élément frontière. Les éléments frontière sont des cas particuliers d'éléments homologues, dont un au moins des homologues relève d'un autre domaine administratif. Un élément homologue peut occuper le même emplacement que toute autre entité (par exemple un portier). La présente annexe n'impose pas qu'un domaine administratif divulgue les détails de son organisation ou de son architecture. La présente annexe ne prescrit aucune architecture spécifique au sein d'un domaine administratif. La présente annexe 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 homologues, à échanger des informations portant sur les adresses que chacun d'eux est en mesure de résoudre. Les éléments frontière échangent des informations portant sur les adresses que leurs domaines administratifs sont 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 la Figure G.1 appartiennent à un réseau global par paquets sans bords. Il convient de noter que la Figure G.1 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.

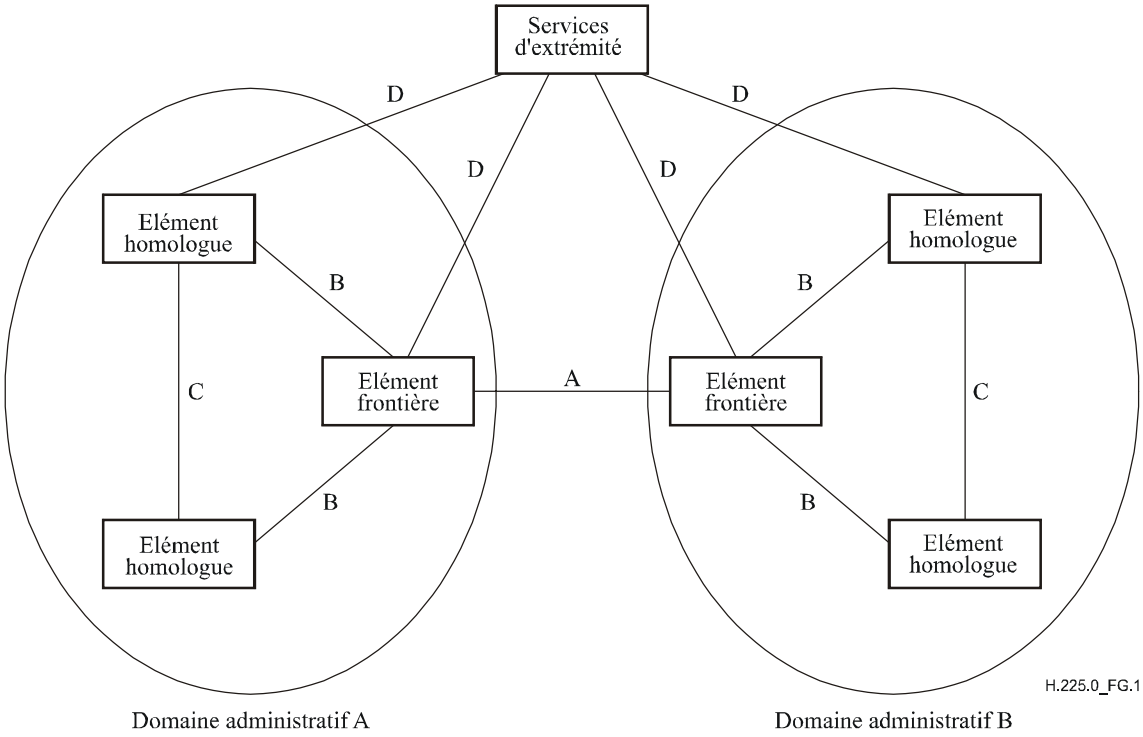


Figure G.1/H.225.0 – Points de référence système

La Figure G.1 met en évidence les points de référence suivants:

A – entre éléments frontière de différents domaines administratifs;

B – entre éléments frontière et éléments homologues d'un même domaine;

C – entre éléments homologues d'un même domaine;

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 aux points de référence A, B et C. Comme indiqué précédemment, un élément homologue peut être situé au même emplacement qu'un autre élément H.323.

Le paragraphe G.7 "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 supports d'extrémité: les services supports 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 homologue: comme le définit la Rec. UIT-T H.501, élément logique à l'origine ou destinataire des messages de signalisation, définis dans cette Recommandation. Un élément

homologue peut être combiné avec d'autres éléments H.323, tels qu'un portier et une passerelle. Un domaine administratif peut contenir un nombre quelconque d'éléments homologues.

G.2.4 élément frontière: cas particulier de l'élément homologue, l'élément frontière est un élément fonctionnel dont au moins un homologue est extérieur à son domaine administratif. Il prend en charge l'accès public à un domaine administratif, à des fins d'établissement d'appel ou de tout autre service impliquant une communication multimédia avec d'autres éléments situés au sein du domaine administratif. L'élément frontière gère la vue externe du domaine administratif.

G.2.5 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 (*gatekeeper*)

GW passerelle (*gateway*)

PE élément homologue (*peer element*)

RCC réseau à commutation de circuits

T terminal

G.4 Références normatives

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

– Recommandation UIT-T H.323 (2006), *Systèmes de communication multimédia en mode paquet.*

– Recommandation UIT-T H.501 (2002), *Protocole de gestion de la mobilité et communications intra et interdomainiales dans les systèmes multimédias*.

– Recommandation UIT-T H.460.2 (2001), *Procédures d'interfonctionnement pour la portabilité des numéros entre les réseaux H.323 et les réseaux à commutation de circuits*.

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 paragraphes qui suivent fournissent quelques exemples d'architecture qui sont considérés comme des illustrations sans aucun caractère exhaustif.

Rappelons qu'un élément homologue est un élément fonctionnel qui peut coexister avec tout autre élément H.323. La Figure G.2 présente quelques exemples d'implémentation d'élément homologue en combinaison avec d'autres éléments.

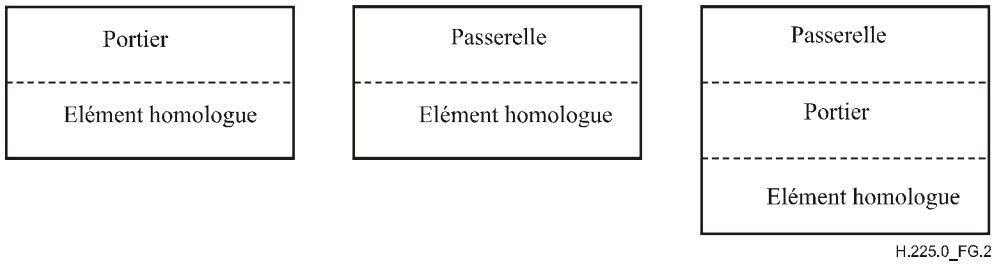


Figure G.2/H.225.0 – Exemples de localisation d'élément homologue

On considère en général qu'un domaine administratif se constitue d'un nombre quelconque de zones et d'un nombre quelconque d'éléments homologues. Les relations entre les domaines administratifs et entre les éléments homologues d'un domaine administratif peuvent être organisées de diverses manières. Les paragraphes qui suivent en fournissent des exemples. S'ils sont censés être appliqués entre domaines administratifs, les exemples de structure hiérarchique, de structure répartie/maillée ou de point d'agrégation peuvent aussi s'appliquer à la structure des éléments homologues au sein d'un domaine administratif.

A noter une nouvelle fois que les exemples de structures présentées ci-dessous sont donnés à titre purement indicatif et qu'ils n'excluent pas d'autres structures possibles.

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.

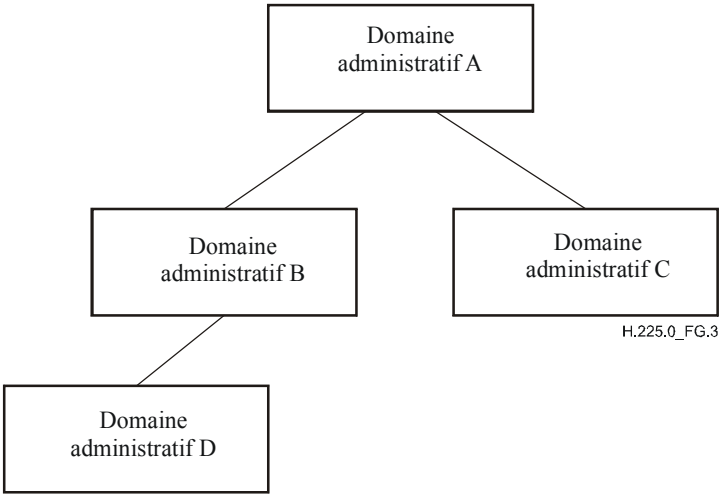


Figure G.3/H.225.0 – Structure hiérarchique simple

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.

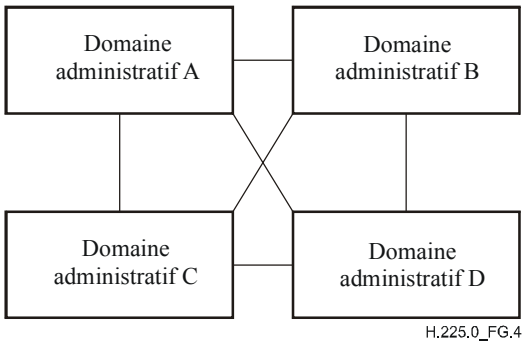


Figure G.4/H.225.0 – Exemple de structure répartie

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. A noter que du fait qu'un résolveur d'adressage constitue une entité extérieure à un domaine administratif, les éléments homologues qui communiquent avec cette entité sont par définition des éléments frontière.

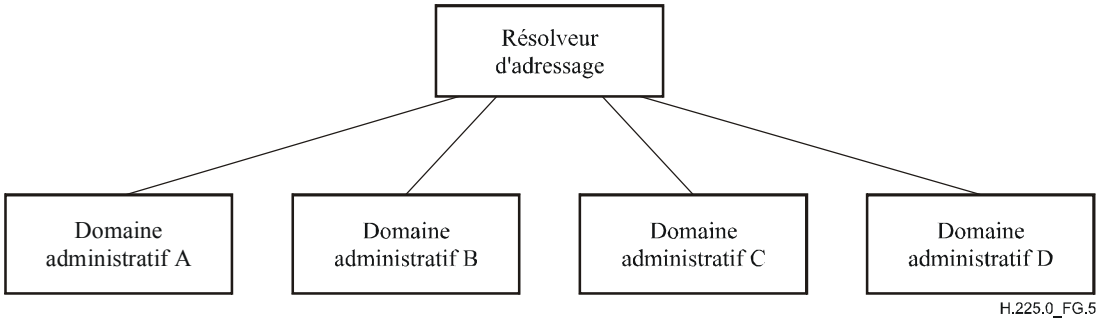


Figure G.5/H.225.0 – Exemple de structure de résolveur d'adressage

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.

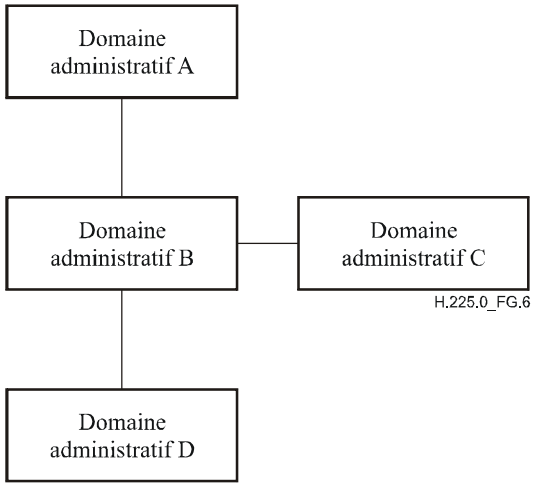


Figure G.6/H.225.0 – Exemple de point d'agrégation

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 Fonctionnement

G.6.1 Utilisation des messages H.501

Les implémentations définies dans l'Annexe G/H.225.0 doivent utiliser les messages définis dans la Rec. UIT-T H.501. Les entités qui échangent des messages H.501 sont appelées dans cette Recommandation éléments homologues.

Les messages H.501 utilisés dans la présente annexe sont énumérés ci-dessous:

demande de service (*ServiceRequest*)

confirmation de service (*ServiceConfirmation*)

rejet de service (*ServiceRejection*)

libération du service (*ServiceRelease*)

demande de descripteur (*DescriptorRequest*)

confirmation de descripteur (*DescriptorConfirmation*)

rejet de descripteur (*DescriptorRejection*)

demande d'identificateur de descripteur (*DescriptorIDRequest*)

confirmation d'identificateur de descripteur (*DescriptorIDConfirmation*)

rejet d'identificateur de descripteur (*DescriptorIDRejection*)

mise à jour de descripteur (*DescriptorUpdate*)

accusé de réception de mise à jour de descripteur (*DescriptorUpdateAck*)

demande d'accès (*AccessRequest*)

confirmation d'accès (*AccessConfirmation*)

rejet d'accès (*AccessRejection*)

demande en cours (*RequestInProgress*)

demande non normalisée (*NonStandardRequest*)

confirmation non normalisée (*NonStandardConfirmation*)

rejet non normalisé (*NonStandardRejection*)

réponse à un message non reconnu (*UnknownMessageResponse*)

demande d'utilisation (*UsageRequest*)

confirmation d'utilisation (*UsageConfirmation*)

rejet d'utilisation (*UsageRejection*)

indication d'utilisation (*UsageIndication*)

confirmation d'indication d'utilisation (*UsageIndicationConfirmation*)

rejet d'indication d'utilisation (*UsageIndicationRejection*)

demande de validation (*ValidationRequest*)

confirmation de validation (*ValidationConfirmation*)

rejet de validation (*ValidationRejection*)

Un élément homologue de l'Annexe G/H.225.0 qui reçoit un message de demande H.501 ne figurant pas dans la liste ci-dessus doit répondre à ce message en envoyant un message de réponse à un message non reconnu (UnknownMessageResponse).

Les messages doivent comporter tous les champs obligatoires définis dans la Rec. UIT-T H.501 et peuvent, si besoin est, comporter des champs facultatifs.

G.6.2 Canevas et descripteurs d'adresse

Un élément homologue obtient des canevas par l'un des mécanismes suivants:

- configuration statique;
- réception de descripteur émis par d'autres éléments homologues en réponse à des demandes générales;
- réception de réponses à des demandes spécifiques.

G.6.2.1 Configuration statique

Un élément homologue gérera des canevas pour toutes les zones dont il est responsable. Ces canevas peuvent être fournis de manière explicite dans l'élément homologue ou, dans le cas où celui-ci coexiste avec les portiers, obtenus en résumant des informations fournies par chaque portier avec lequel l'élément homologue communique. L'élément homologue peut mettre ces informations à la disposition d'autres éléments homologues 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 AccessRequest (demande d'accès)) qui représente la totalité de sa zone et fait référence à un portier pour le traitement des appels d'arrivée;
- 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 Setup (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 **sendAccessRequest** (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 **sendAccessRequest** (émission de demande d'accès) pour chaque domaine administratif au sein de 44.

Les éléments homologues n'ont pas l'obligation de gérer une copie complète de la base de données. S'il ne dispose pas d'une telle copie, l'élément homologue doit alors utiliser des canevas de configuration statique dont les marques **sendAccessRequest** (émission de demande d'accès) indiquent un élément frontière résolveur d'adressage qui sera utilisé pour résoudre les autres demandes.

G.6.2.2 Réception de descripteurs

Un élément homologue peut demander à un autre élément homologue de lui fournir des canevas de configuration statique. Ce dernier décide de la réponse donnée à la demande. L'élément homologue demandeur émet un message DescriptorRequest (demande de descripteur) indiquant les descripteurs qu'il souhaite recevoir. S'il est en mesure de transférer les descripteurs, l'élément homologue propriétaire répond alors au moyen d'un message DescriptorConfirmation (confirmation de descripteur) qui contient tous les canevas.

L'élément homologue 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 homologue propriétaire doit alors émettre un message DescriptorUpdate (mise à jour de descripteur) à destination des éléments homologues dont il a connaissance. Lorsqu'il reçoit un message "mise à jour de descripteur", un élément homologue 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 homologue 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'il est lui-même le contact pour une AccessRequest (demande d'accès), même s'il a reçu d'un autre élément frontière des descripteurs indiquant ce dernier comme contact.

Un élément homologue 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 AccessRequest (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 homologue reçoit un message de demande d'accès ne contenant pas d'informations propres à l'appel, et si la stratégie adoptée consiste à exiger des informations pour chaque appel, l'élément homologue répondra par un message AccessRejection (rejet d'accès) portant la mention explicative **needCallInformation**.

Un élément homologue peut émettre un message DescriptorUpdate (mise à jour de descripteur) à destination d'autres éléments homologues connus individuellement ou émettre ce message en multidiffusion. L'élément homologue doit déterminer le domaine de diffusion dans le cas de multidiffusion d'un tel message. Le message DescriptorUpdate (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 DescriptorUpdate (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.6.2.3 Réception de réponses à des demandes spécifiques

Un élément homologue peut émettre un message AccessRequest (demande d'accès) à destination d'un autre élément homologue 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 homologue 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 homologue répondra alors en général avec un canevas indiquant qu'un message AccessRequest (demande d'accès) ou un message Setup (établissement) doit être émis. S'il joue le rôle d'un résolveur d'adressage, l'élément homologue de destination répondra alors normalement avec un canevas indiquant que le message AccessRequest (demande d'accès) doit être émis.

L'élément homologue 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 AccessRequest (demande d'accès): un canevas de courte durée de vie (de quelques minutes ou secondes) spécifiant l'emplacement vers lequel un message Setup (établissement) doit être émis et des canevas supplémentaires spécifiant que des messages AccessRequest (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 homologue peut mémoriser dans un cache, et jusqu'à l'expiration de sa durée de vie, un canevas reçu dans un message AccessConfirmation (confirmation d'adresse).

G.6.3 Découverte d'un élément homologue ou d'un ensemble d'éléments homologues

G.6.3.1 Statique

Un élément homologue peut disposer d'un ensemble administré d'autres éléments homologues 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, par exemple entre un domaine administratif et d'autres domaines administratifs. Les domaines administratifs peuvent utiliser de manière facultative les services d'un résolveur d'adressage.

G.6.3.2 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 **sendAccessRequest** (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.6.3.3 Autres méthodes

L'utilisation d'autres méthodes de localisation d'un élément homologue appelle une étude ultérieure.

G.6.4 Procédures de résolution

G.6.4.1 Procédure de résolution au sein d'un domaine administratif

L'élément homologue 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 **sendSetup** ((l'émission d'une demande d'établissement) est préférable à **sendAccessRequest** (l'émission d'une demande d'accès)).

L'élément homologue 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 **sendSetup** (émission d'établissement), l'élément homologue émettra alors, à destination de l'adresse spécifiée dans le canevas, un message AccessRequest (demande d'accès) contenant une adresse de destination spécifique. Il peut mémoriser dans son cache la réponse reçue de l'élément homologue et renvoyer au demandeur l'adresse vers laquelle doit être émis le message Setup (établissement).

G.6.4.2 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 reçoit un message de demande d'accès en provenance d'un élément frontière d'un autre domaine administratif.

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 **sendSetup** ((l'émission d'une demande d'établissement) est préférable à **sendAccessRequest** (l'émission d'une demande d'accès)). On rejette lors de chaque tri les canevas qui ne fournissent pas la correspondance la plus précise.

Si les canevas conformes sont marqués **sendAccessRequest** (émission de demande d'accès), l'élément frontière peut alors choisir de transmettre le message AccessRequest (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 AccessRequest (demande d'accès) vers un autre élément homologue si le compteur de bonds du message AccessRequest (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 AccessRejection (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 AccessConfirmation (confirmation d'accès).

A ce stade, l'élément frontière peut utiliser un autre élément frontière (par exemple un résolveur d'adressage) afin d'autoriser la demande d'accès. A cet effet, il envoie un message ValidationRequest (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 ValidationConfirmation (confirmation de validation).

L'élément frontière renvoie ensuite un message AccessConfirmation (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 qu'il jugera 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.6.5 Echange d'information d'utilisation

Des éléments homologues peuvent demander à d'autres éléments homologues de leur communiquer des informations d'utilisation lors de l'établissement d'appels particuliers. Des messages UsageIndication (d'indication d'utilisation) peuvent être communiqués à n'importe quel stade de l'établissement de l'appel. De plus, plusieurs messages d'indication d'utilisation peuvent être transmis pour le même appel, chacun contenant le cas échéant, des informations plus récemment mises à jour, ou signalant des segments d'appel consécutifs ou l'utilisation de différents types de médias. Voir le § G.6.5.1 pour plus de précisions.

Les échanges de messages UsageIndication (d'indication d'utilisation) sont admis qu'il existe ou non une relation de service entre les deux éléments homologues. Toutefois, un élément homologue peut avoir pour politique de ne pas admettre de tels échanges en l'absence d'une relation de service. En pareil cas, l'élément homologue peut rejeter le message d'indication d'utilisation, en indiquant comme motif du rejet **`noServiceRelationship`** (absence de relation de service).

Des demandes d'indication d'utilisation seront envoyées, à chaque fois qu'un élément homologue en fera la demande, dans les canevas pour lesquels il fait office d'élément à contacter ou s'il l'indique dans le message de demande de service qu'il envoie pendant l'établissement d'une relation de service avec un élément homologue distant ou 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'information d'utilisation.

G.6.5.1 Indications d'utilisation multiples pour le même appel

Les indications d'utilisation multiples pour le même appel fournissent toujours plus d'informations à jour sur les mêmes types de médias, ou des informations d'utilisation sur les nouveaux types de médias créés au cours d'un même appel. En outre, du fait que des éléments homologues peuvent prendre en charge des appels en cours d'établissement, toutes les indications d'utilisation ne proviennent pas nécessairement du même élément homologue. Les règles définissant la sémantique sont les suivantes:

1) la réception d'un message UsageIndication (d'indication d'utilisation) assorti d'une indication de l'état d'appel d'utilisation (**`usageCallStatus`**) du message appel en cours (**`callInProgress`**) suppose la réception d'un message UsageIndication (d'indication d'utilisation) ultérieur assorti du même identificateur d'appel (**`callIdentifier`**) et du même message rôle de l'expéditeur (**`senderRole`**). Si sa configuration permet la reprise après défaillance, le destinataire peut arriver à la conclusion au bout d'un intervalle de temps de configuration sans nouveaux messages UsageIndication (d'indication d'utilisation) qu'une défaillance s'est produite, et sera autorisé à récupérer toutes les données des messages UsageIndication (d'indication d'utilisation) reçus qu'il pourra;

2) les messages UsageIndication (d'indication d'utilisation) ultérieurs assortis des mêmes identificateurs de champ d'utilisation (**usageField**) devraient signaler tout instant de début (**startTime**) correspondant à l'instant de fin (**endTime**) du message précédent (bien que cela puisse être impossible pour un autre élément homologue). Les destinataires admettront que chaque rapport correspond à une période différente. Les autres informations figurant dans le champ d'utilisation (**usageField**) l'emportent sur les informations reçues dans les messages précédents assortis du même identificateur de champ d'utilisation (**usageField**);

3) un élément homologue devrait envoyer un nouveau message UsageIndication (d'indication d'utilisation) à chaque fois que le type de média est modifié durant l'appel: arrêt du son et démarrage de la télécopie, ou changement de codec. Si plusieurs types de médias sont utilisés en même temps (audio et vidéo, par exemple) ils devraient être signalés dans le même message UsageIndication (d'indication d'utilisation).

G.6.5.2 Demande et négociation d'une information d'utilisation pendant l'établissement d'une relation de service

Un élément homologue PE_A peut inclure un élément "spécification d'utilisation" (**UsageSpecification**) dans un message ServiceRequest (de demande de service) adressé à un deuxième élément homologue PE_B. Cet élément **UsageSpecification** sera utilisé pour définir l'information d'utilisation par défaut à communiquer pour tous les appels établis au cours de la relation de service entre les deux éléments homologues PE_A et PE_B. Cet élément **UsageSpecification** doit être utilisé pour tous les appels pour lesquels PE_B envoie des messages UsageIndication (d'indication d'utilisation) à PE_A.

Si un élément **UsageSpecification** arrive à PE_B dans un autre message de PE_A (AccessConfirmation un message de confirmation d'accès, par exemple), alors le nouvel élément **UsageSpecification** l'emporte sur l'élément **UsageSpecification** par défaut pour tous les appels se rapportant au nouveau message.

Un élément homologue qui reçoit un message de demande de service contenant un élément **UsageSpecification** doit procéder comme suit:

i) s'il est prêt à accepter le message de demande de service et l'élément **UsageSpecification** contenu dans celui-ci, ledit élément homologue (élément homologue destinataire) doit envoyer un message ServiceConfirmation (de confirmation de service) contenant le même élément **UsageSpecification** que celui qui a été reçu dans le message ServiceRequest (de demande de service). L'élément **UsageSpecification** doit s'appliquer à la fois aux appels entrants à destination de l'élément homologue destinataire en provenance de l'élément homologue demandeur ainsi qu'aux appels sortants en provenance de l'élément homologue destinataire à destination de l'élément homologue demandeur;

ii) s'il est prêt à accepter le message ServiceRequest (de demande de service) sans toutefois être prêt à accepter l'élément **UsageSpecification** contenu dans celui-ci, l'élément homologue destinataire doit envoyer soit un message ServiceConfirmation (de confirmation de service) contenant un élément **UsageSpecification** différent qui précise l'information d'utilisation qu'il est en mesure de communiquer à l'élément homologue demandeur, soit un message ServiceRejection (de rejet de service) indiquant pour motif "spécification d'utilisation ne pouvant être prise en charge" (**cannotSupportUsageSpec**);

iii) s'il ne prend pas en charge le compte rendu d'utilisation, l'élément homologue destinataire doit renvoyer un message ServiceRejection (de rejet de service) avec pour motif "utilisation indisponible" (**usageUnavailable**).

Un élément homologue qui reçoit un message ServiceConfirmation (de confirmation de service) doit procéder comme suit:

i) si l'élément **`UsageSpecification`** contenu dans le message ServiceConfirmation (confirmation de service) est identique à celui qui a été envoyé dans le message ServiceRequest (de demande de service), cela indique que l'élément homologue émetteur et l'élément homologue destinataire ont établi entre eux une relation de service;

ii) si l'élément **`UsageSpecification`** contenu dans le message ServiceConfirmation (de confirmation de service) est différent de celui qui a été envoyé dans le message ServiceRequest (de demande de service), alors, si l'élément homologue émetteur est prêt à utiliser le nouvel élément **`UsageSpecification`**, la relation de service est établie. S'il n'est pas prêt à utiliser le nouvel élément **`UsageSpecification`**, l'élément homologue émetteur doit envoyer un message ServiceRelease (de libération de service) avec pour motif "terminé" (**`terminated`**). L'élément homologue émetteur pourra ensuite analyser l'élément **`UsageSpecification`** renvoyé dans le message ServiceConfirmation (de confirmation de service) afin d'élaborer un nouveau message ServiceRequest (de demande de service) dont l'élément **`UsageSpecification`** modifié puisse être accepté par les deux éléments homologues;

iii) si le message ServiceConfirmation (de confirmation de service) ne contient pas d'élément **`UsageSpecification`** (alors que le message ServiceRequest (de demande de service) en contenait un), l'élément homologue qui a envoyé le message ServiceConfirmation (de confirmation de service) sera dans l'impossibilité de faire usage du rapport d'utilisation au niveau de la relation de service, et n'en fera pas usage. Tel est le cas lorsque, par exemple, l'élément homologue destinataire applique la version 1 de la présente annexe. Dans ce cas, l'élément homologue émetteur peut soit mettre fin à la relation de service (en envoyant un message ServiceRelease (de libération de service) avec pour code de motif "terminé" (**`terminated`**), soit ne pas mettre fin à la relation de service. Dans un cas comme dans l'autre, s'il souhaite recevoir des informations d'utilisation relatives aux appels, l'élément homologue émetteur doit en faire la demande selon les mécanismes décrits dans la version 1 de la présente annexe (c'est-à-dire en envoyant les éléments **`UsageSpecification`** dans les messages AccessRequest (de demande d'accès), AccessConfirmation (de demande de confirmation) (dans les canevas d'adresse renvoyés), UsageRequest (de demande d'utilisation), ValidationRequest (de demande de validation) ou ValidationConfirmation (de confirmation de validation)).

G.6.6 Transmission de l'information de portabilité des numéros

La Rec. UIT-T H.460.2 décrit les mécanismes applicables à la portabilité des numéros dans les réseaux H.323. La prise en charge des procédures H.460.2 exige que la présente annexe autorise le transport de l'information de portabilité des numéros moyennant l'échange de messages de résolution d'adresse. L'interface entre l'élément frontière de l'Annexe G et les autres éléments de réseaux H.323 avec lesquels cet élément communique n'est pas traitée dans la présente annexe; on pose en principe que cette interface permet de transporter l'information de portabilité des numéros H.460.2 à destination et en provenance de l'élément frontière de l'Annexe G.

Tout message AccessRequest (de demande d'accès) envoyé véhiculera l'information de portabilité des numéros H.460.2, si elle est présente, dans le champ **`genericData`** de la portion d'information commune du message.

Les messages AccessConfirmation (de confirmation d'accès) et AccessRejection (de rejet d'accès) véhiculeront également l'information de réponse de portabilité des numéros correspondante dans le champ **`genericData`**. Dans le cas d'un message AccessRejection (de rejet d'accès), le motif de rejet indiqué doit être **`genericDataReason`**.

G.7 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 présentent des messages RAS LRQ/LCF échangés entre un portier et un élément frontière à l'intérieur du même domaine administratif. Ces exemples sont présentés à titre purement indicatif, et pourraient être remplacés par des messages analogues de l'Annexe G échangés entre l'élément frontière et un élément homologue résidant à l'intérieur du portier.

G.7.1 Répartition ou maillage total

La Figure G.7 donne un exemple de réseau réparti.

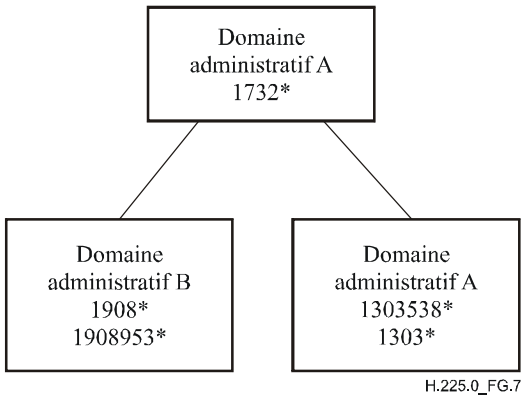


Figure G.7/H.225.0 – Réseau réparti pour les exemples de signalisation

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 "d2": Modèle = 1908* Adresse de transport = adresse Annexe g BE_B Type de message = **sendAccessRequest** Descripteur "d3": Modèle = 1908953* Adresse de transport = adresse de signalisation d'appel GW_{B1} Type de message = **sendSetup**	Un message AccessRequest (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 Setup (d'établissement) peut être émis directement à destination de la passerelle considérée ici dans le cas d'appels concernant les numéros 1908953*.
C	Descripteur "d4": Modèle = 1303538* Adresse de transport = adresse de signal d'appel GK_{C1} Type de message = **sendSetup** Descripteur "d5": 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 du portier considéré ici. 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.7.1.1 Echange d'informations de zone

Dans la structure répartie ou totalement maillée, chaque domaine administratif 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.

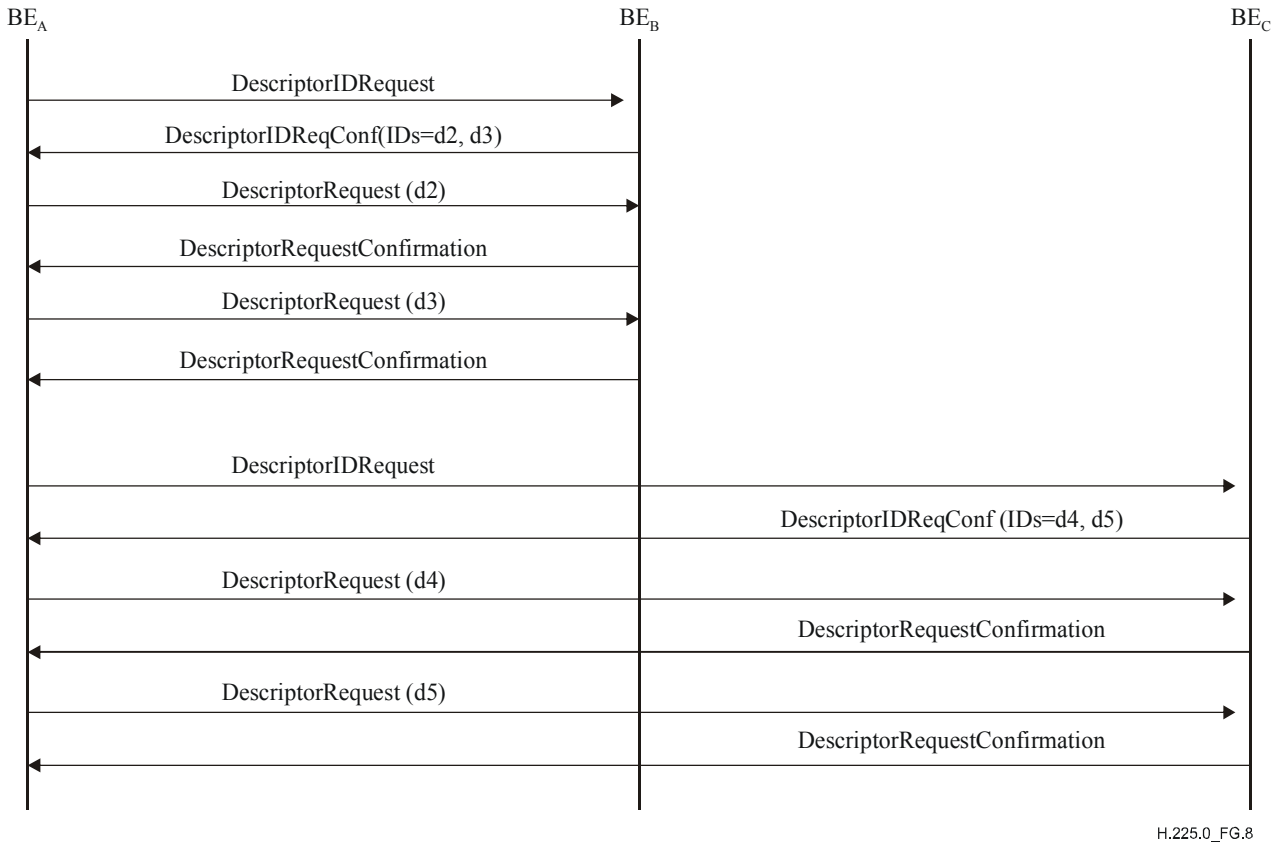


Figure G.8/H.225.0 – Exemple d'échange de descripteurs

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.7.1.2 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 dans la Rec. UIT-T H.323 et ses annexes.

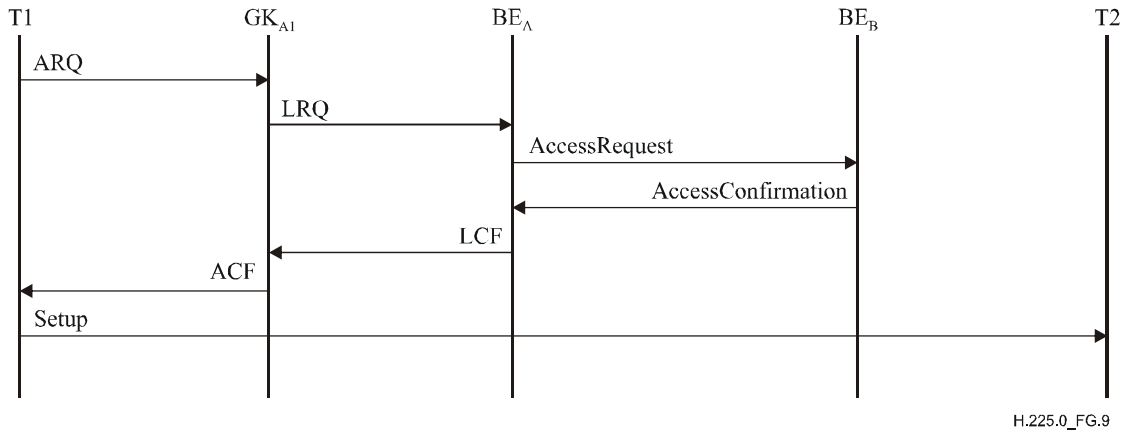


Figure G.9/H.225.0 – Exemple d'un appel simple

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.

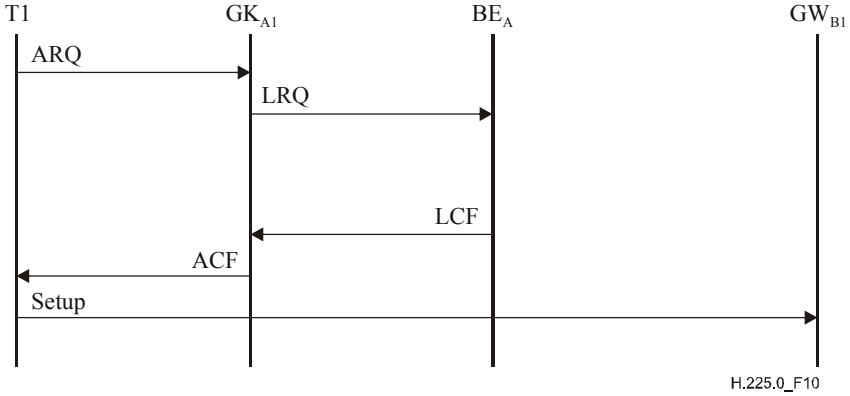


Figure G.10/H.225.0 – Exemple d'un appel avec adresse cache

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 d'acheminement 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.

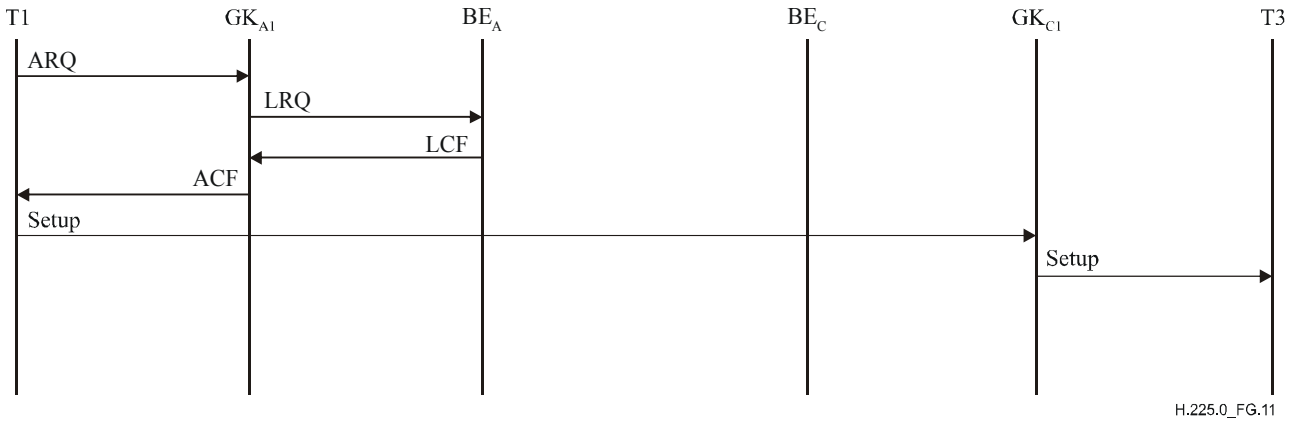


Figure G.11/H.225.0 – Exemple d'un appel acheminé par portier distant

Le portier du terminal T1 peut, en variante, implémenter le modèle avec acheminement par portier, comme indiqué dans la Figure G.12.

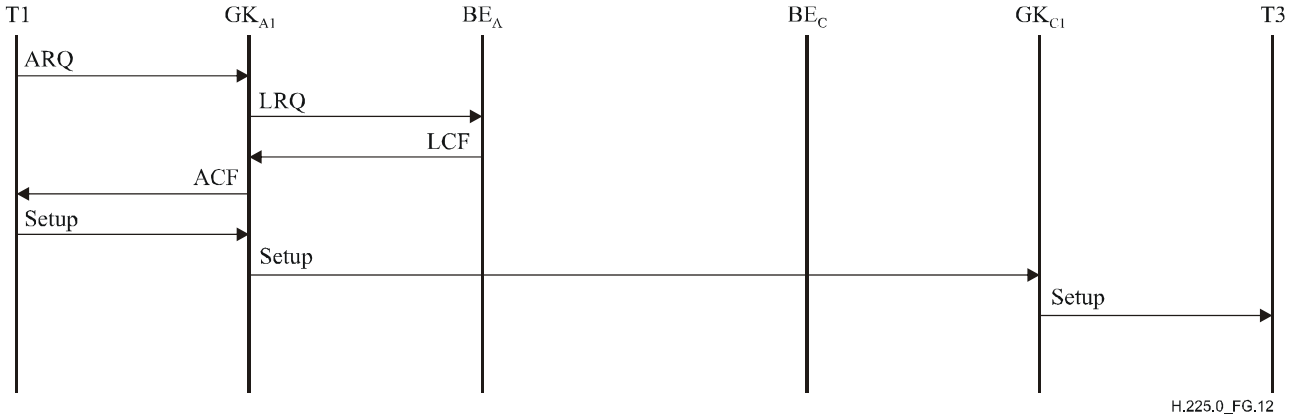


Figure G.12/H.225.0 – Exemple d'un appel acheminé par portier local

G.7.2 Résolveur d'adressage

La Figure G.13 présente un exemple de configuration qui utilise un résolveur d'adressage. La Figure G.13 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.

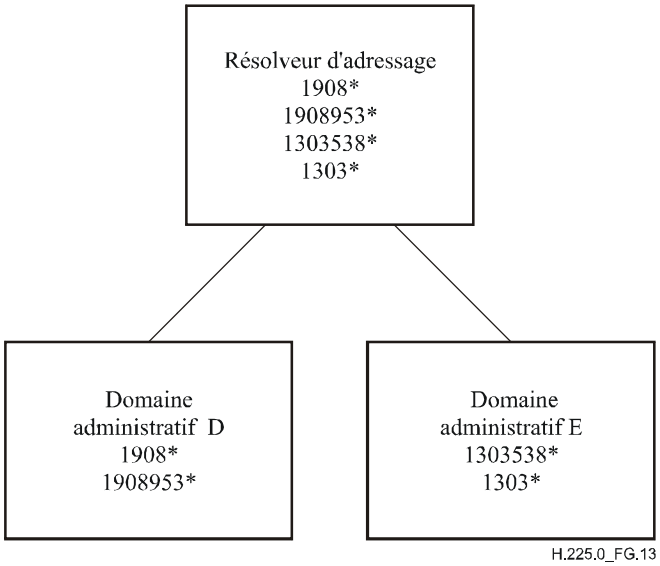


Figure G.13/H.225.0 – Exemple de configuration avec résolveur d'adressage

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 "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**	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 = 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** 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.7.2.1 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'il reçoit de chaque domaine administratif et les relaye à destination d'autres domaines administratifs. Le résolveur d'adressage est vu 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.

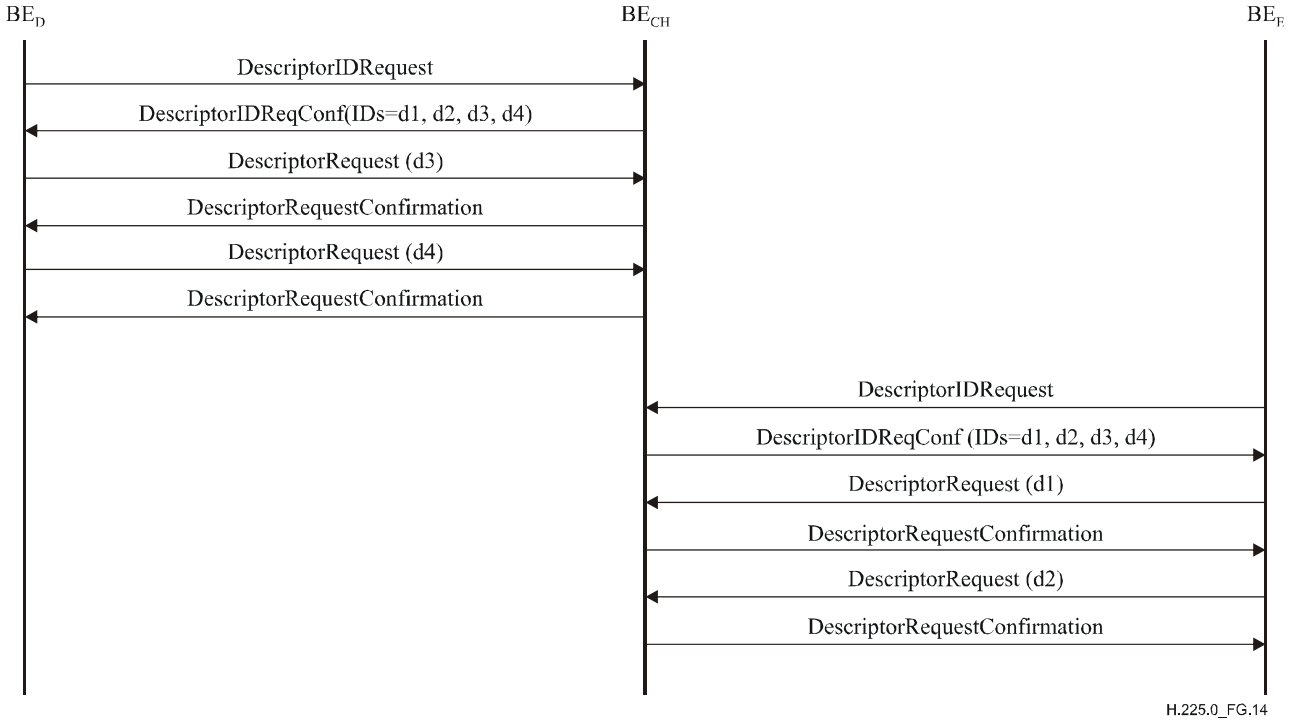


Figure G.14/H.225.0 – Exemple d'échange de descripteur avec un résolveur d'adressage

G.7.2.2 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 ce dernier doit être consulté 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 reçus 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.

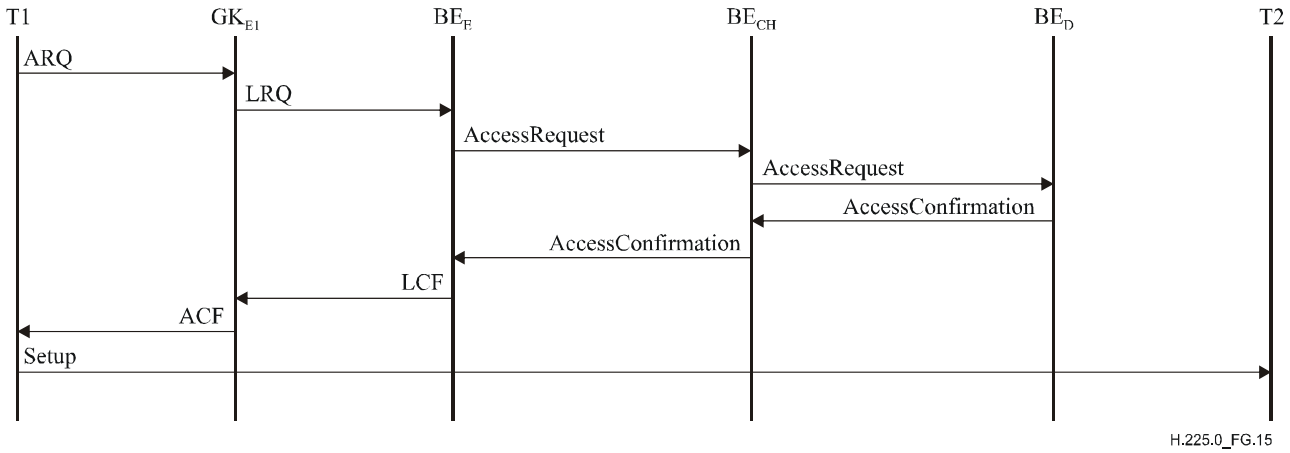


Figure G.15/H.225.0 – Exemple d'un appel avec un résolveur d'adressage

Le portier du terminal T1 peut, en variante, acheminer la signalisation d'appel comme indiqué dans la Figure G.16.

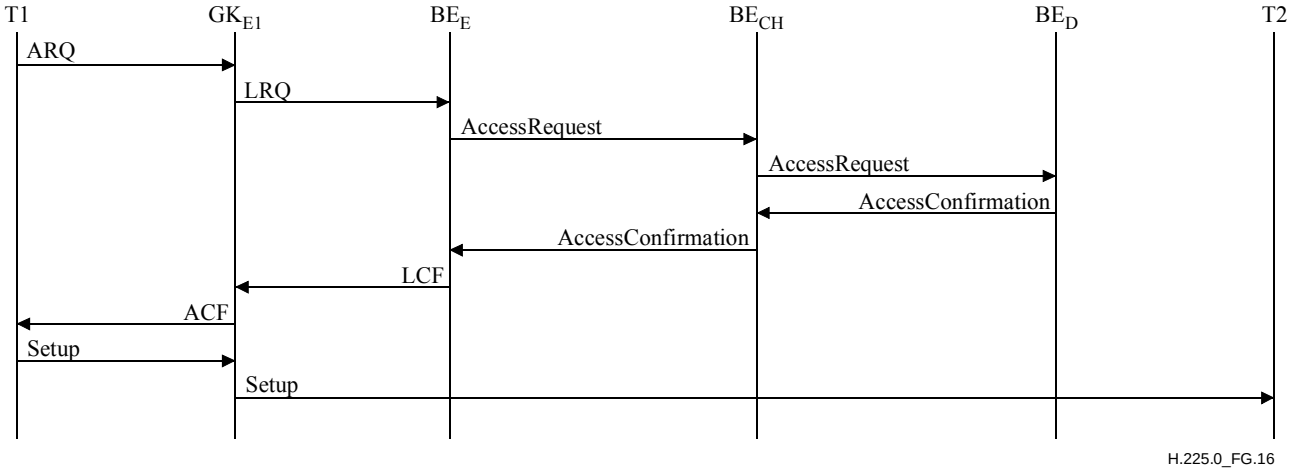


Figure G.16/H.225.0 – Exemple d'un appel acheminé par portier local avec résolveur d'adressage

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.

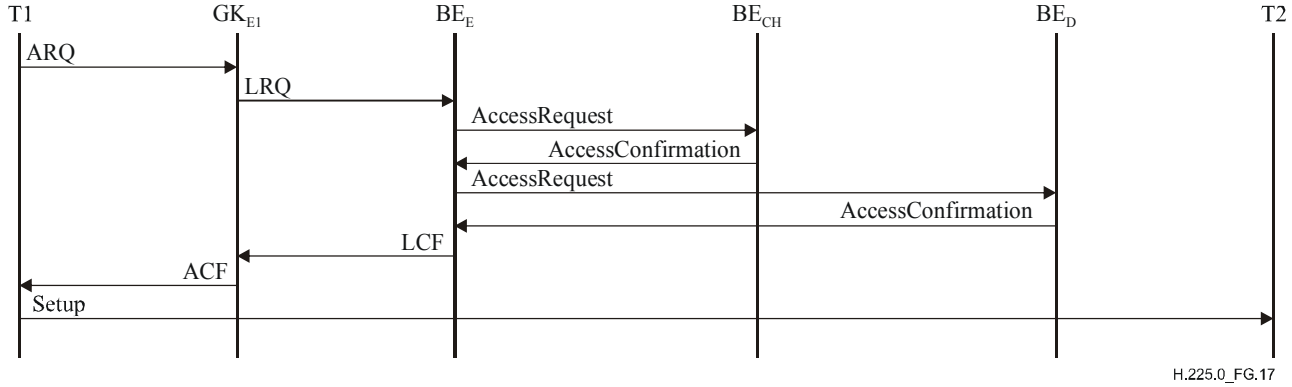


Figure G.17/H.225.0 – Exemple d'acheminement par résolveur d'adressage utilisant l'information de contact pour l'élément frontière distant

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.

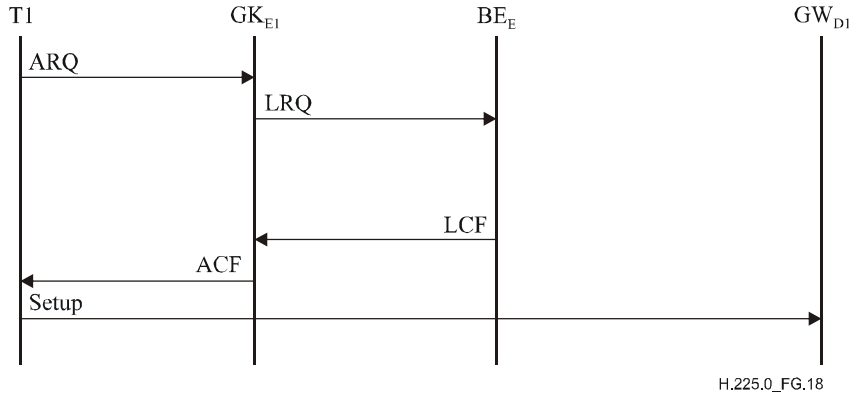


Figure G.18/H.225.0 – Exemple d'un appel utilisant un descripteur en cache dans l'élément frontière local

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 acheminés directement vers un portier dans le domaine administratif E sans utiliser un message AccessRequest (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 ce dernier n'a pas l'obligation, dans cet exemple, de fournir la résolution d'adresse pour ces appels.

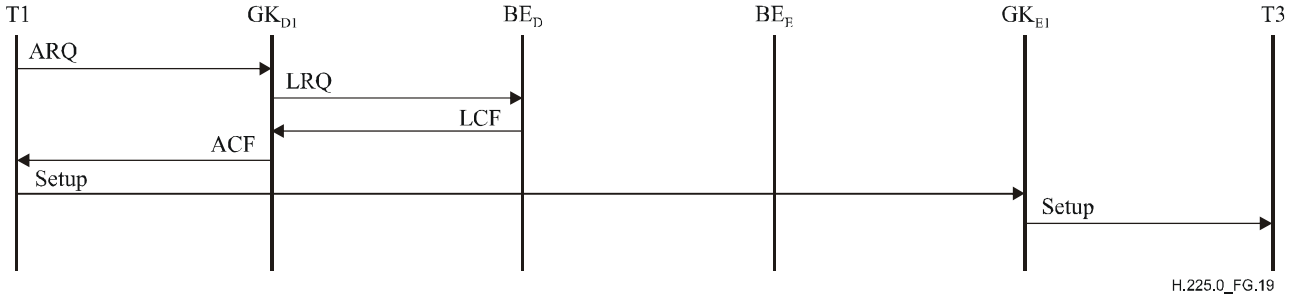


Figure G.19/H.225.0 – Exemple d'un appel acheminé par portier utilisant un descripteur en cache

Rappelons qu'un élément frontière peut être combiné avec un portier et peut également acheminer des appels dans le modèle avec acheminement 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 l'acheminement dans un domaine administratif si les descripteurs sont configurés en conséquence.

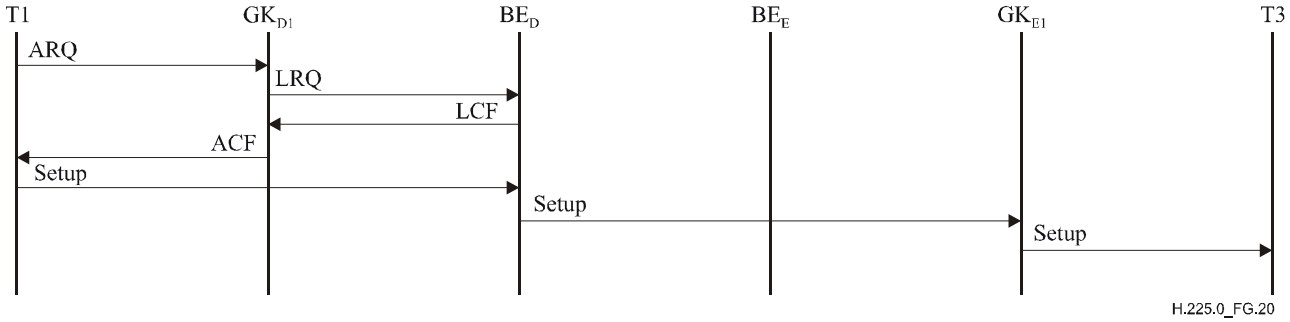


Figure G.20/H.225.0 – Exemple d'un appel avec élément frontière combiné avec portier d'acheminement

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.

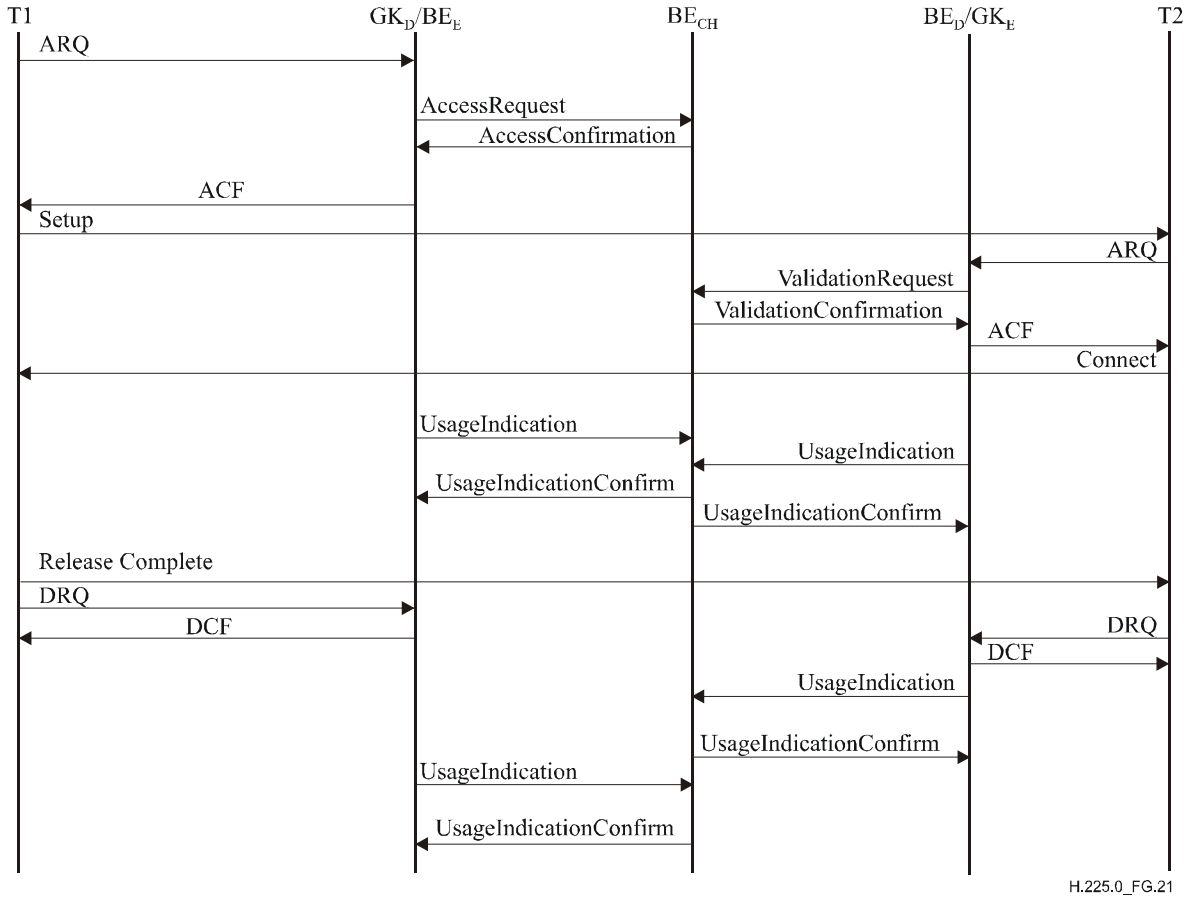


Figure G.21/H.225.0 – Exemple de validation d'appel et de rapport d'utilisation avec résolveur d'adressage

G.8 Profils de l'Annexe G

G.8.1 Introduction

La Rec. UIT-T H.501 offre un riche ensemble de messages et de champs qui peuvent être utilisés dans le cadre de la présente annexe aux fins de l'instauration d'un dialogue entre domaines administratifs et entre éléments homologues d'un même domaine administratif. Nombre de ces messages et de ces champs sont facultatifs et peuvent être utilisés de diverses manières aux fins de l'implémentation de services différents ou d'options de service différentes. Le présent paragraphe spécifie les profils d'implémentation qui définissent les messages, les champs et les procédures à utiliser pour assurer la mise en conformité avec un profil donné.

G.8.1.1 Indication et négociation de profils

Le cadre générique d'extensibilité H.323 peut être utilisé par un élément homologue pour indiquer à un autre élément homologue l'ensemble de profils qui lui sont nécessaires pour mener à bien une transaction, l'ensemble de profils qu'il souhaite utiliser et l'ensemble de profils qu'il prend en charge. Cette indication/négociation de profils peut être effectuée à l'occasion d'un échange ponctuel de messages (échange d'un message de demande d'accès/confirmation d'accès, par exemple), ou pendant l'établissement d'une relation de service. A noter que l'établissement d'une relation de service entre deux éléments homologues peut ne pas être exigé par un profil.

G.8.1.1.1 Traitement par l'entité émettrice de la demande

Une entité émettrice d'une demande (un élément homologue) utilise les éléments contenus dans la structure **`FeatureSet`** pour spécifier les divers profils dont elle a besoin. Elle spécifie l'ensemble des profils dont elle a besoin au moyen du champ **`neededFeatures`**, l'ensemble des profils qu'elle recherche au moyen du champ **`desiredFeatures`** et l'ensemble des profils qu'elle prend en charge au moyen du champ **`supportedFeatures`**. Ces trois champs font partie de la structure **`FeatureSet`**.

En réponse à la demande qu'elle a émise, une entité doit recevoir un message de confirmation ou de rejet.

Si la demande est rejetée, l'entité émettrice de la réponse peut avoir inclus un ensemble de fonctions **`neededFeatures`** dont elle a besoin, que l'entité émettrice de la demande doit pouvoir prendre en charge afin que sa demande soit satisfaite. Si tel est le cas et que l'entité émettrice de la demande prenne en charge les fonctions requises (un profil déterminé, par exemple), cette entité peut réémettre une demande spécifiant la prise en charge du profil dont l'entité émettrice de la réponse a besoin.

Si la demande est acceptée, des procédures spéciales doivent être appliquées pour garantir que la négociation fonctionne de manière compatible avec l'amont. A cette fin, l'entité émettrice de la demande vérifie que les profils qu'elle a spécifiés comme étant nécessaires figurent bien dans la réponse parmi les fonctions prises en charge. Si une entité émettrice de demande ne trouve pas les profils dont elle a besoin dans le champ **`supportedFeatures`** du message de réponse, elle supposera que l'entité qui répond ne prend pas en charge les profils dont elle a besoin. Si l'entité émettrice de la demande détermine qu'elle ne peut pas continuer dans ces conditions, elle doit annuler l'opération qu'elle tente d'effectuer (c'est-à-dire envoyer un message de libération de service si elle a initialement émis un message de demande de service), de façon que l'état de l'entité émettrice de la réponse soit "invalidé".

G.8.1.1.2 Traitement par l'entité émettrice de la réponse

L'entité émettrice de la réponse examine les profils spécifiés dans le champ **`neededFeatures`** de la demande afin de déterminer si elle peut accepter celle-ci. Elle examine également les champs **`neededFeatures`**, **`desiredFeatures`** et **`supportedFeatures`** afin de déterminer si les profils dont elle a besoin sont pris en charge par l'entité émettrice de la demande.

Si l'entité émettrice de la réponse détermine que les ensembles nécessaires de profils sont pris en charge par les deux entités, elle peut acquitter la demande. L'entité émettrice de la réponse énumère l'ensemble des profils qu'elle choisit de prendre en charge dans le champ **`supportedFeatures`** de sa réponse. Si la demande est acceptée, toutes les fonctions inscrites dans le champ **`neededFeatures`** de la demande doivent être incluses dans le champ **`supportedFeatures`** de la réponse. L'entité émettrice de la réponse peut également inclure des fonctions dans le champ **`desiredFeatures`**.

Si l'entité émettrice de la demande a besoin que des profils additionnels soient pris en charge par l'entité émettrice de la demande, elle doit rejeter cette demande. Si elle cherche à déclarer les profils qui doivent être pris en charge pour que la demande soit satisfaite, cette intention doit être spécifiée dans le champ **`neededFeatures`** du message de rejet. L'entité qui répond peut également inclure d'autres fonctions dans les champs **`desiredFeatures`** et **`supportedFeatures`** du message de rejet.

G.8.1.1.3 Identificateurs

L'identificateur suivant est utilisé dans une structure FeatureDescriptor (descripteur de fonction) pour spécifier que cette structure s'applique aux profils de l'Annexe G.

Valeur	Description
idAnnexGProfiles	Cet identificateur est utilisé dans le champ "id" d'une structure FeatureDescriptor pour indiquer que cette structure décrit les profils de l'Annexe G nécessaires/souhaités/pris en charge.

Le tableau suivant énumère les identificateurs utilisés dans le cadre générique d'extensibilité relatifs à la présente annexe.

Valeur normale de INTEGER	Description
0	Identificateur d'une structure FeatureDescriptor indiquant que cette structure décrit les profils de l'Annexe G
1	Identificateur d'une structure EnumeratedParameter identifiant le profil "A" de l'Annexe G

G.8.2 Profil "A": routage d'appel entre zones à destination d'un portier de confiance

Ce profil spécifie un service intradomanial simple, avec interrogation pour chaque appel d'une autre zone de confiance en vue de déterminer le point de terminaison constituant la configuration statique de l'adresse de signalisation de la présente annexe des zones de confiance. C'est là une des utilisations de la présente annexe les plus simples, semblable à l'utilisation du message LRQ avec le protocole RAS pour interroger une autre zone au sujet d'un point de terminaison. Le même profil peut être utilisé pour interroger un élément homologue de confiance, dont la connaissance approfondie du domaine lui permet de communiquer les informations d'acheminement ou de les obtenir en procédant à de nouvelles interrogations dans le cadre de l'Annexe G.

G.8.2.1 Messages nécessaires

Les entités qui se conforment à ce profil doivent prendre en charge les messages indiqués comme étant "obligatoires" dans le tableau suivant:

Message	**Emission** (**O**bligatoire, **F**acultatif, **R**ecommandé)	**Réception et action consécutive** (**O**bligatoire, **F**acultatif, **R**ecommandé)
Demande de service (*ServiceRequest*)	F	O (Note 1)
Confirmation de service (*ServiceConfirmation*)	F	F
Rejet de service (*ServiceRejection*)	O	F
Libération du service (*ServiceRelease*)	F	F
Demande de descripteur (*DescriptorRequest*)	F	O (Note 1)
Confirmation de descripteur (*DescriptorConfirmation*)	R (Note 2)	F
Rejet de descripteur (*DescriptorRejection*)	O	F
Demande d'identificateur de descripteur (*DescriptorIdRequest*)	F	O (Note 1)
Confirmation d'identificateur de descripteur (*DescriptorIdConfirmation*)	R (Note 2)	F
Rejet d'identificateur de descripteur (*DescriptorIdRejection*)	O	F
Mise à jour de descripteur (*DescriptorUpdate*)	F	O (Note 3)
Accusé de réception de mise à jour de descripteur (*DescriptorUpdateAck*)	O	F
Demande d'accès (*AccessRequest*)	O	O
Confirmation d'accès (*AccessConfirmation*)	O	O
Rejet d'accès (*AccessRejection*)	O	O
Demande en cours (*RequestInProgress*)	O	O
Demande non normalisée (*NonStandardRequest*)	F	O
Confirmation non normalisée (*NonStandardConfirmation*)	F	F
Rejet non normalisé (*NonStandardRejection*)	O	F
Réponse à un message non reconnu (*UnknownMessageResponse*)	O	O
Demande d'utilisation (*UsageRequest)*	F	O (Note 1)
Confirmation d'utilisation (*UsageConfirmation*)	F	F
Rejet d'utilisation (*UsageRejection*)	O	F
Indication d'utilisation (*UsageIndication*)	F	O (Note 1)
Confirmation d'indication d'utilisation (*UsageIndicationConfirmation*)	F	F
Rejet d'indication d'utilisation (*UsageIndicationRejection*)	O	F

Message	Emission (**O**bligatoire, **F**acultatif, **R**ecommandé)	Réception et action consécutive (**O**bligatoire, **F**acultatif, **R**ecommandé)
Demande de validation (*ValidationRequest*)	F	O (Note 1)
Confirmation de validation (*ValidationConfirmation*)	F	F
Rejet de validation (*ValidationRejection*)	O	F
NOTE 1 – Doit être reçu et au minimum rejeté. NOTE 2 – Il est recommandé qu'une entité renvoie au minimum un descripteur pour un canevas spécifiant que l'élément SendAccessRequest est dirigé vers elle. NOTE 3 – Doit être reçu et acquitté, mais n'a pas besoin d'être traité.		

G.8.2.2 Champs nécessaires

Tous les champs définis comme étant obligatoires dans la Rec. UIT-T H.501 sont également obligatoires dans le cadre de ce profil.

Les entités qui sont conformes à ce profil doivent également prendre en charge les champs spécifiés dans le tableau suivant.

Les autres champs définis comme étant facultatifs dans la Rec. UIT-T H.501 peuvent éventuellement être présents.

Message ou Structure	Champ nécessaire	Commentaire
AccessRequest message	`destinationInfo`	Une adresse contenant l'adresse E.164 requise de la destination
	`sourceInfo`	Inclut les éléments domainInfo et endpointType
	`callInfo`	
AccessConfirmation message	`templates`	Si des canevas sont présents, chacun d'entre eux correspond à une passerelle ou un portier de terminaison
	`partialResponse`	Mis à FALSE
AddressTemplate structure	`pattern`	Un modèle précis contenant le numéro E.164 est présent
	`routeInfo`	Une instance est présente
	`timeToLive`	
RouteInformation structure	`messageType`	Présent
	`callSpecific`	Mis à FALSE
	`contacts`	Une instance est présente
	`type`	Doit être présent si: messageType = **sendSetup**
ContactInformation structure	`transportAddress`	L'adresse IP de la passerelle ou du portier
	`priority`	

G.8.2.3 Procédures requises

Dans ce profil, les entités peuvent utiliser les procédures de découverte statique de la présente annexe (§ G.6.3.1) et disposeront ainsi d'une liste configurée d'éléments homologues ou de portiers auxquels elles pourront envoyer des demandes. Cette liste peut indiquer d'autres éléments à utiliser uniquement lorsque l'élément principal ne peut pas être atteint, ou faire simplement mention de ces autres éléments (éventuels).

Les entités peuvent également utiliser les procédures de découverte dynamique de la présente annexe (§ G.6.3.2).

Les entités doivent envoyer un message AccessRequest (de demande d'accès) à un élément homologue ou un portier choisi pour chaque appel. Si plusieurs éléments homologues ou portiers sont accessibles pour être interrogés au sujet d'un appel donné, il n'est pas précisé s'ils doivent être interrogés l'un après l'autre ou s'ils peuvent être interrogés simultanément. Ce choix est laissé à l'appréciation de l'entité émettrice de la demande.

La réponse pourra ou non comporter des canevas. **`timeToLive`** peut être fixé à 60 secondes ou moins pour indiquer qu'il ne peut pas être utilisé pour un autre appel.

Pour améliorer l'interfonctionnement avec des homologues moins spécialisés, il est proposé que dans le cas où l'élément homologue n'assure pas la prise en charge du descripteur, il devrait procéder comme suit:

- s'il reçoit un message DescriptorIDRequest, l'élément homologue devrait renvoyer un message DescriptorIDConfirmation contenant un seul élément **`DescriptorInfo`**. Cet élément **`DescriptorInfo`** désigne un descripteur contenant un seul canevas spécifiant que l'élément **`sendAccessRequest`** est dirigé vers l'élément homologue;
- s'il reçoit un message DescriptorRequest, l'élément homologue devrait renvoyer un message DescriptorConfirmation contenant un seul descripteur. Ce descripteur doit contenir un seul canevas spécifiant que l'élément **`sendAccessRequest`** est dirigé vers l'élément homologue.

G.8.2.4 Identificateurs pour le profil "A"

L'identificateur suivant est utilisé dans une structure **`EnumeratedParameter`** pour spécifier que cette structure spécifie le profil A de l'Annexe G.

Valeur	Description
idAnnexGProfileA	Cet identificateur est utilisé dans le champ "id" d'une structure **`EnumeratedParameter`** pour indiquer que le profil A de l'Annexe G est nécessaire/souhaité/pris en charge. A noter que le champ "content" de la structure **`EnumeratedParameter`** n'est pas présent.

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 Utilisateur à 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,
          QOSCapability
     FROM MULTIMEDIA-SYSTEM-CONTROL;
H323-UserInformation ::= SEQUENCE     -- racine pour tous les messages
                                      -- de signalisation d'appel H.225.0
{
     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
                                    -- version 4 de la Rec. UIT-T 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
     h245Tunnelling             BOOLEAN,
                                -- si la valeur est TRUE, la tunnélisation 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 tunnélisé
          messageContent          SEQUENCE OF OCTET STRING,  -- sequence de
                                                             -- message(s)
                                                             -- entier (s)
          tunnellingRequired      NULL OPTIONAL,
          nonStandardData         NonStandardParameter OPTIONAL,
          ...
     } OPTIONAL,
     provisionalRespToH245Tunnelling  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 RFC 1766
     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, -- l'appelé est occupé
    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,
    invalidCID                  NULL,
    securityError               SecurityErrors,
    hopCountExceeded            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,
    destExtraCRV            SEQUENCE OF CallReferenceValue OPTIONAL,
    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 RFC 1766
    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,
    hopCount                    INTEGER (1..31) 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 mise 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,
     sip                     SIPCaps
}

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,
     ...
}

SIPCaps ::= SEQUENCE
{
     nonStandardData         NonStandardParameter OPTIONAL,
     dataRatesSupported      SEQUENCE OF DataRate OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix OPTIONAL,
     ...
}

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
     ...,
     enterpriseNumber   OBJECT IDENTIFIER OPTIONAL
}

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
{
    dialledDigits IA5String (SIZE (1..128)) (FROM ("0123456789#*,")),
    h323-ID       BMPString (SIZE (1..256)), -- ISO/CEI 10646 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,
    isupNumber    IsupNumber
}

AddressPattern ::= CHOICE
{
wildcard AliasAddress,
range         SEQUENCE
{
startOfRange  PartyNumber,
endOfRange    PartyNumber
},
...
}

PartyNumber ::= CHOICE
{
    e164Number                  PublicPartyNumber,
                                -- le plan de numérotage est selon
                                -- les Recommandations UIT-T E.163 et 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,
                                -- 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'accés 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,
    mscid               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'usager
    -- 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
    -- RMTPE ou RMTPN correspond à Visiting ou Home Public Land Mobile
    -- Network number
    ...
}

IsupNumber ::= CHOICE
{
    e164Number                  IsupPublicPartyNumber,
                                    -- le plan de numérotage est conforme
                                    -- aux Recommandations UIT-T E.163 et
                                    -- E.164.
    dataPartyNumber             IsupDigits, -- champ inutilisé, valeur
                                            -- réservée.
    telexPartyNumber            IsupDigits, -- champ inutilisé, valeur
                                            -- réservée.
    privateNumber               IsupPrivatePartyNumber,
                                    -- le plan de numérotage est conforme
                                    -- à l'ISO/CEI 11571.
    nationalStandardPartyNumber IsupDigits, -- champ inutilisé, valeur
                                            -- réservée.

}
```

```
IsupPublicPartyNumber ::= SEQUENCE
{
     natureOfAddress    NatureOfAddress,
     address            IsupDigits,
     ...
}

IsupPrivatePartyNumber ::= SEQUENCE
{
     privateTypeOfNumber     PrivateTypeOfNumber,
     address                 IsupDigits,
     ...
}

NatureOfAddress ::= CHOICE
{
     unknown                                    NULL,
     subscriberNumber                           NULL,
     nationalNumber                             NULL,
     internationalNumber                        NULL,
     networkSpecificNumber                      NULL,
     routingNumberNationalFormat                NULL,
     routingNumberNetworkSpecificFormat         NULL,
     routingNumberWithCalledDirectoryNumber     NULL,
     ...
}

IsupDigits ::= IA5String (SIZE (1..128)) (FROM ("0123456789ABCDE"))
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,
     circuitInfo                  CircuitInfo   OPTIONAL,
     featureSet                   FeatureSet    OPTIONAL
}

AlternateTransportAddresses ::= SEQUENCE
{
     annexE              SEQUENCE OF TransportAddress OPTIONAL,
     ...,
     sctp                SEQUENCE OF TransportAddress OPTIONAL
}

UseSpecifiedTransport ::= CHOICE
{
     tcp                 NULL,
```

```
    annexE              NULL,
    ...,
    sctp                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,
    authenticaton           SecurityServiceMode,
    integrity               SecurityServiceMode,
    ...
}

SecurityErrors     ::=  CHOICE
{
    securityWrongSyncTime       NULL,     -- problème de serveur temporel ou
                                          -- retard du réseau
    securityReplay              NULL,     -- attaque par réexécution en cours
    securityWrongGeneralID      NULL,     -- ID général erroné
    securityWrongSendersID      NULL,     -- ID d'expéditeurs erronés
    securityIntegrityFailed     NULL,     -- échec de contrôle d'intégrité
    securityWrongOID            NULL,     -- OID de jeton erroné ou OID
                                          -- d'algorithme cryptographique
    securityDHmismatch          NULL,     -- discordance de paramètres DH
    securityCertificateExpired  NULL,     -- expiration du certificat
    securityCertificateDateInvalid   NULL,     -- certificat pas encore valide
    securityCertificateRevoked  NULL,     -- certificat trouvé révoqué
    securityCertificateNotReadable   NULL,     -- erreur de décodage
    securityCertificateSignatureInvalid   NULL,  -- signature erronée dans le
                                                 -- certificat
    securityCertificateMissing  NULL,     -- aucun certificat disponible
    securityCertificateIncomplete NULL,   -- absence du certificat attendu
                                          -- extensions
    securityUnsupportedCertificateAlgOID  NULL,  -- algorithmes
                                                 -- cryptographiques non
                                                 -- compris
    securityUnknownCA           NULL,     -- impossibilité de trouver
                                          -- certificat CA/racine
...
```

```
}

SecurityErrors2    ::=  CHOICE
{
     securityWrongSyncTime   NULL, -- problème de serveur temporel ou
                                   -- retard du réseau
     securityReplay          NULL, -- attaque par réexécution en cours
     securityWrongGeneralID  NULL, -- ID général erroné
     securityWrongSendersID  NULL, -- ID d'expéditeurs erronés
     securityIntegrityFailed NULL, -- échec de contrôle d'intégrité
     securityWrongOID        NULL, -- OID de jeton erroné ou OID
                                   -- d'algorithme cryptographique
     ...
}

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 centaine de bits
CallReferenceValue      ::=  INTEGER (0..65535)
EndpointIdentifier      ::=  BMPString (SIZE(1..128))
ProtocolIdentifier      ::=  OBJECT IDENTIFIER
TimeToLive              ::=  INTEGER (1..4294967295)   -- en secondes
H248PackagesDescriptor  ::=  OCTET STRING  -- cette chaîne d'octets contient un
                                           -- descripteur de paquetage H.248 à
                                           -- codage ASN.1 selon règles PER

H248SignalsDescriptor   ::=  OCTET STRING  -- cette chaîne d'octets contient un
                                           -- descripteur de signaux H.248 à
                                           -- codage ASN.1 selon règles PER

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-iso10118-2-s  EncryptIntAlg,     -- selon l'ISO/CEI 10118-2 par
                                           -- algorithme de chiffrement
                                           -- de base par blocs EncryptIntAlg
                                           -- (mécanisme MAC court)
     hMAC-iso10118-2-l  EncryptIntAlg,     -- selon l'ISO/CEI 10118-2 par
                                           -- algorithme de chiffrement
                                           -- de base par blocs EncryptIntAlg
                                           -- (mécanisme 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,      -- numérique selon l'ISO/CEI 9797
                                           -- avec OID comme algorithme de
                                           -- chiffrement base (X-CBC MAC)
     nonIsoIM      NonIsoIntegrityMechanism,
     ...
}

ICV ::= SEQUENCE
{
     algorithmOID  OBJECT IDENTIFIER,   -- algorithme utilisé pour calculer
                                        -- la signature,
     icv           BIT STRING           -- la valeur calculée d'intégrité
                                        -- 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'entité produisant le
                                         -- hachage
```

```
        timeStamp     TimeStamp,       -- pointeur temporel utilisé dans le
                                       -- hachage
        token         HASHED    { EncodedPwdCertToken -- generalID mis à
                                                      -- "alias" -- }
    },
    cryptoGKPwdHash  SEQUENCE
    {
        gatekeeperId  GatekeeperIdentifier,   -- identité du portier produisant
                                              -- le hachage
        timeStamp     TimeStamp,              -- pointeur temporel utilisé 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,
     ...,
     sipGwCallsAvailable                   SEQUENCE OF CallsAvailable OPTIONAL
}

CallsAvailable ::= SEQUENCE
{
     calls          INTEGER (0..4294967295),
     group          IA5String (SIZE (1..128)) OPTIONAL,
     ...,
     carrier        CarrierInfo OPTIONAL
}

CircuitInfo ::= SEQUENCE
{
     sourceCircuitID              CircuitIdentifier OPTIONAL,
     destinationCircuitID         CircuitIdentifier OPTIONAL,
     genericData                  SEQUENCE OF GenericData OPTIONAL,
     ...
}

CircuitIdentifier ::= SEQUENCE
{
     cic            CicInfo OPTIONAL,  group     GroupID OPTIONAL,
     ...,
     carrier        CarrierInfo OPTIONAL
}

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)),
     ...
}

CarrierInfo ::= SEQUENCE
{
```

```
    carrierIdentificationCode   OCTET STRING (SIZE (3..4)) OPTIONAL,
    carrierName                 IA5String (SIZE (1..128)) OPTIONAL,
    ...
}

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
}

RehomingModel ::= CHOICE
{
     gatekeeperBased             NULL,
     endpointBased               NULL
}

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,
    admissionConfirmSequence     SEQUENCE OF AdmissionConfirm
}

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,
    supportsAssignedGK           BOOLEAN,
    assignedGatekeeper           AlternateGK 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,
    assignedGatekeeper          AlternateGK OPTIONAL,
    rehomingModel               RehomingModel 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,
    securityError               SecurityErrors
}

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,
    restart                     NULL     OPTIONAL,
    supportsACFSequences        NULL     OPTIONAL,
    supportsAssignedGK          BOOLEAN,
    assignedGatekeeper          AlternateGK OPTIONAL,
    transportQOS                TransportQOS OPTIONAL,
    language                    SEQUENCE OF IA5String(SIZE (1..32)) 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,
     assignedGatekeeper                 AlternateGK OPTIONAL,
     rehomingModel                      RehomingModel OPTIONAL,
     transportQOS                       TransportQOS 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,
     assignedGatekeeper            AlternateGK 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,
     securityError                 SecurityErrors,
     registerWithAssignedGK        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     ICV OPTIONAL,
     reason                  UnregRequestReason OPTIONAL,
     endpointAliasPattern    SEQUENCE OF AddressPattern OPTIONAL,
     supportedPrefixes       SEQUENCE OF SupportedPrefix OPTIONAL,
     alternateGatekeeper     SEQUENCE OF AlternateGK OPTIONAL,
     genericData             SEQUENCE OF GenericData OPTIONAL,
     assignedGatekeeper      AlternateGK OPTIONAL
}

UnregRequestReason ::= CHOICE
{
     reregistrationRequired  NULL,
     ttlExpired              NULL,
     securityDenial          NULL,
     undefinedReason         NULL,
     ...,
     maintenance             NULL,
     securityError           SecurityErrors2,
     registerWithAssignedGK  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,
     assignedGatekeeper      AlternateGK 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'usager demandeur n'est pas autorisé
```

```
                                        -- à annuler l'enregistrement de l'usager
                                        -- spécifié
     securityDenial          NULL,
     securityError           SecurityErrors2
}

AdmissionRequest ::= SEQUENCE --(ARQ)
{
     requestSeqNum               RequestSeqNum,
     callType                    CallType,
     callModel                   CallModel OPTIONAL,
     endpointIdentifier          EndpointIdentifier,
     destinationInfo             SEQUENCE OF AliasAddress OPTIONAL,
     destCallSignalAddress       TransportAddress OPTIONAL,
     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,
     circuitInfo                 CircuitInfo OPTIONAL,
     desiredProtocols            SEQUENCE OF SupportedProtocols OPTIONAL,
     desiredTunnelledProtocol    TunnelledProtocol OPTIONAL,
     featureSet                  FeatureSet OPTIONAL,
     genericData                 SEQUENCE OF GenericData OPTIONAL,
     canMapSrcAlias              BOOLEAN
}

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,
     ...,
     qOSCapabilities         SEQUENCE SIZE(1..256) OF QOSCapability
}

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,
     modifiedSrcInfo             SEQUENCE OF AliasAddress OPTIONAL,
     assignedGatekeeper          AlternateGK 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,
    assignedGatekeeper      AlternateGK OPTIONAL
}

AdmissionRejectReason ::= CHOICE
{
    calledPartyNotRegistered    NULL,     -- adresse intraduisible
    invalidPermission           NULL,     -- expiration de permission
    requestDenied               NULL,
    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,
    securityError               SecurityErrors2,
    securityDHmismatch          NULL,     -- discordance de paramètres DH
    noRouteToDestination        NULL,     -- destination inatteignable
    unallocatedNumber           NULL,     -- numéro de destination
                                          -- non attribué
    registerWithAssignedGK      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,
    transportQOS            TransportQOS 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,
    transportQOS                TransportQOS 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,
securityError               SecurityErrors2}

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,
    callIdentifier              CallIdentifier OPTIONAL,
    bandWidth                   BandWidth OPTIONAL,
    sourceEndpointInfo          SEQUENCE OF AliasAddress OPTIONAL,
    canMapSrcAlias              BOOLEAN,
    language                    SEQUENCE OF IA5String(SIZE (1..32)) 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 AlternateTransportAddresses 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,
    modifiedSrcInfo             SEQUENCE OF AliasAddress OPTIONAL,
    bandWidth                   BandWidth 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 l'administrateur
                                      -- ou le dispositif
    requestDenied           NULL,
    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,
    hopCountExceeded             NULL,
    incompleteAddress            NULL,
    securityError                SecurityErrors2,
    securityDHmismatch           NULL,     -- discordance de paramètres DH
    noRouteToDestination         NULL,     -- destination inatteignable
    unallocatedNumber            NULL      -- numéro de destination non attribué
}

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,
    assignedGatekeeper      AlternateGK 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
                                      -- pour d'autres usagers
    ...,
    securityDenial          NULL,
    securityError           SecurityErrors2
}

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,
    assignedGatekeeper          AlternateGK 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,
        callSignalling          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,
     ...,
     securityError       SecurityErrors2

}
```

```
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 la Rec. 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 la Rec. 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 3550, RTP: A Transport Protocol for Real-Time Applications (RTP: un protocole de transport pour les applications en temps réel), *Internet Engineering Task Force*, 2003.

Appendice II

Profil RTP

Les renseignements auxquels il est fait référence peuvent être trouvés dans la proposition de norme Internet suivante:

– SCHULZRINNE (H.), CASNER (S.): RFC 3551, RTP Profile for Audio and Video Conferences with Minimal Control (profil RTP pour les conférences audio et vidéo avec commande minimale), *Internet Engineering Task Force*, 2003.

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 de paquets. Les réseaux à commutation de 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.

<table>
<tr><th>Applications avec remise non fiable</th><th>Canal H.245 et canal de signalisation d'appel</th></tr>
<tr><td rowspan="2">UDP</td><td>TPKT
— —
TCP</td></tr>
<tr><td colspan="2">IP</td></tr>
<tr><td colspan="2">Couche Liaison de données</td></tr>
<tr><td colspan="2">Couche Physique</td></tr>
</table>

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 § 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.

Noter que les termes "autre accord" peuvent désigner l'enregistrement d'une extrémité auprès d'un portier.

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 2782 du groupe IETF.

IV.1.1.2.2.2 Interrogation relative à des enregistrements TXT

Toutes les implémentations 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 la Rec. UIT-T H.323.

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.

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 de données	
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 la Rec. 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.

IV.3 SCTP

La pile de protocoles H.323 sur SCTP se présente comme suit:

Applications d'acheminement non fiables	Signalisation d'appel avec commande d'appel tunnelisée
UDP	SCTP
IP	
Couche Liaison de données	
Couche Physique	

Chaque message de signalisation d'appel H.225.0 doit être transféré dans un fragment distinct de données SCTP. Aucun en-tête ne doit être ajouté (c'est-à-dire aucun paquet TPKT). L'acheminement ordonné doit être spécifié.

IV.3.1 Flux

Tous les messages du même appel doivent utiliser le même flux SCTP. L'implémentation peut faire appel à différents flux pour différents appels.

IV.3.2 Identificateurs de protocole de charge utile

Le protocole SCTP peut être utilisé avec un identificateur indéfini (0) de protocole de charge utile ou avec le nombre 13, qui est l'indicatif attribué au protocole H.323 par l'autorité IANA.

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 TAGS.

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 tunnélisés

La présente Recommandation prend en charge la tunnélisation 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 tunnélisation pour des protocoles spécifiques. Dans la présente Recommandation, un protocole tunnélisé 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 tunnélisés qui sont définis dans la présente Recommandation sont décrits dans les Tableaux VI.1 et VI.2. Noter que la tunnélisation n'est pas limitée aux protocoles énumérés dans ces tableaux.

Tableau VI.1/H.225.0 – Protocoles tunnélisés dentifiés par tunnelledProtocolObjectID

Spécification de tunnélisation	Spécification de protocole	tunnelledProtocolObjectID	subIdentifier
M.1/H.323	ISO/CEI 11572 et 11582	{iso (1) identified-organization (3) icd-ecma (0012) private-isdn-signalling-domain (9)}	(Néant)
M.2/H.323	Rec. UIT-T Q.763 (1988)	{itu-t (0) recommendation (0) q (17) 763}	"1988"
M.2/H.323	Rec. UIT-T Q.763 (1993)	{itu-t (0) recommendation (0) q (17) 763}	"1993"

Tableau VI.2/H.225.0 – Protocoles tunnélisés identifiés par TunnelledProtocolAlternateIdentifier

Spécification de tunnélisation	Spécification de protocole	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 D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	**Systèmes audiovisuels et multimédias**
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	Gestion des télécommunications y compris le RGT et maintenance des réseaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données, communication entre systèmes ouverts et sécurité
Série Y	Infrastructure mondiale de l'information, protocole Internet et réseaux de prochaine génération
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

Imprimé en Suisse
Genève, 2007