UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T H.225.0

(09/99)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE H: SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Infrastructure des services audiovisuels – Multiplexage et synchronisation en transmission

---

Protocoles de signalisation d'appel et mise en paquets des trains multimédias dans les systèmes de communication multimédias en mode paquet

Recommandation UIT-T H.225.0

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE H

SYSTÈMES AUDIOVISUELS ET MULTIMÉDIAS

Caractéristiques des canaux de transmission pour des usages autres que téléphoniques	H.10–H.19
Emploi de circuits de type téléphonique pour la télégraphie à fréquence vocale	H.20–H.29
Circuits et câbles téléphoniques utilisés pour les divers types de transmission télégraphique et de transmissions simultanées	H.30–H.39
Circuits de type téléphonique utilisés en bélinographie	H.40–H.49
Caractéristiques des signaux de données	H.50–H.99
CARACTÉRISTIQUES DES SYSTÈMES VISIOPHONIQUES	H.100–H.199
INFRASTRUCTURE DES SERVICES AUDIOVISUELS	
Généralités	H.200–H.219
Multiplexage et synchronisation en transmission	**H.220–H.229**
Aspects système	H.230–H.239
Procédures de communication	H.240–H.259
Codage des images vidéo animées	H.260–H.279
Aspects liés aux systèmes	H.280–H.299
Systèmes et équipements terminaux pour les services audiovisuels	H.300–H.399
Services complémentaires en multimédia	H.450–H.499

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

RECOMMANDATION UIT-T H.225.0

PROTOCOLES DE SIGNALISATION D'APPEL ET MISE EN PAQUETS DES TRAINS MULTIMEDIAS DANS LES SYSTEMES DE COMMUNICATION MULTIMEDIAS EN MODE PAQUET

Résumé

La présente Recommandation traite des spécifications techniques relatives aux services visiophoniques à bande étroite définis dans les Recommandations des séries H.200/AV.120, lorsque sur le trajet de transmission se trouvent un ou plusieurs réseaux à commutation par paquets configurés et gérés de manière à offrir une qualité de service (QS) non garantie et non équivalente à celle qui est offerte par le RNIS-BE, de sorte que les mécanismes additionnels de protection et de rétablissement autres que ceux exigés par la Recommandation H.320 doivent être assurés par les terminaux. On remarque que la Recommandation 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 H.323 et H.225.0.

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

Source

La Recommandation UIT-T H.225.0, révisée par la Commission d'études 16 de l'UIT-T (1997-2000), a été approuvée le 30 septembre 1999 selon la procédure définie dans la Résolution n° 1 de la CMNT.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

TABLE DES MATIÈRES

Recommandation H.225.0

PROTOCOLES DE SIGNALISATION D'APPEL ET MISE EN PAQUETS DES TRAINS MULTIMEDIAS DANS LES SYSTEMES DE COMMUNICATION MULTIMEDIAS EN MODE PAQUET

(révisée en 1999)

L'UIT,

considérant

que la Recommandation H.320 est largement adoptée et de plus en plus utilisée dans le monde pour les services de visiophonie et de visioconférence sur des réseaux conformes aux caractéristiques du RNIS-BE spécifiées dans les Recommandations de la série I,

reconnaissant

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

et notant

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

recommande

d'utiliser pour ces services, des systèmes et des équipements conformes aux spécifications des Recommandations H.322 ou H.323.

1 Domaine d'application

La présente Recommandation décrit la façon dont les signaux audio, vidéo, de données et de commande sont associés, codés puis mis en paquets pour être acheminés entre des équipements H.323 sur un réseau à commutation par paquets. Pour cela, une passerelle H.323 est utilisée, cette passerelle pouvant être connectée à des terminaux H.320, H.324 ou H.310/H.321 sur le RNIS-BE, le RTGC ou le RNIS-LB respectivement. La Recommandation 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 points d'extrémité H.322 est également possible.

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

ou sur Internet n'entrent pas dans le domaine d'application de la présente Recommandation. Cependant, la présente Recommandation permet à l'utilisateur d'un équipement H.323 de savoir que les problèmes de qualité qu'il rencontre tiennent à un encombrement sur le réseau à commutation par paquets et de disposer de procédures permettant d'exécuter des actions correctives. Il convient également de noter que l'utilisation de plusieurs passerelles H.323 connectées sur le RNIS public est une méthode directe qui permet d'améliorer la qualité de service.

La Recommandation H.323 et la présente Recommandation étendent les conférences de la Recommandation H.320 et les connexions de la Recommandation H.221 aux réseaux à commutation par 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 multimédia pour tous les réseaux à commutation par paquets sous-jacents (voir Annexes A, B et C). Il convient de noter que l'utilisation du protocole RTP/RTCP telle que spécifiée dans la présente Recommandation n'est liée en aucune façon à l'utilisation du protocole TCP/IP/UDP. La présente Recommandation prend pour hypothèse un modèle d'appel dans lequel on utilise la signalisation initiale sur une adresse de transport non RTP pour l'établissement de l'appel et la négociation de capacité (voir Recommandations 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 Recommandation H.221, des signaux audio, vidéo, de données et de commande sont multiplexés en une ou plusieurs connexions RCC physiques synchronisées. Du côté réseau à commutation par 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∗64 kbit/s, par exemple 2×64 kbit/s, 3×64 kbit/s, etc. Ainsi, du côté du réseau à commutation par paquets, par exemple, il n'y a que des appels à connexion unique avec un débit maximal de 128 kbit/s et non pas des appels à débit fixe 2∗64 kbit/s. Dans un autre exemple, les appels de réseau à commutation par paquets à connexion unique avec un débit maximal limité à 384 kbit/s sont en interfonctionnement avec un appel de 6∗64 kbit/s du côté WAN[2]. La raison essentielle de cette méthode est de concentrer la complexité dans la passerelle et non pas dans le terminal et ainsi d'éviter l'intégration dans le réseau à commutation par paquets de fonctions H.320 qui sont étroitement liées au RNIS sauf si cela est nécessaire.

En général, les terminaux H.323 en interfonctionnement via une passerelle H.323 ne connaissent pas directement le débit de transfert H.320; en effet, la passerelle utilise les messages **FlowControlCommand** H.245 pour limiter le débit du média sur chaque canal logique utilisé à celui autorisé par le multiplex H.221. La passerelle peut permettre l'utilisation de débits vidéo côté

---

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

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

réseau à commutation par paquets très inférieurs à ceux utilisés côté WAN (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 Recommandation 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 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 Recommandation H.320. Il se peut également que la qualité de service côté H.320 dépende des caractéristiques et des capacités de la passerelle H.323 (voir Figure 1).

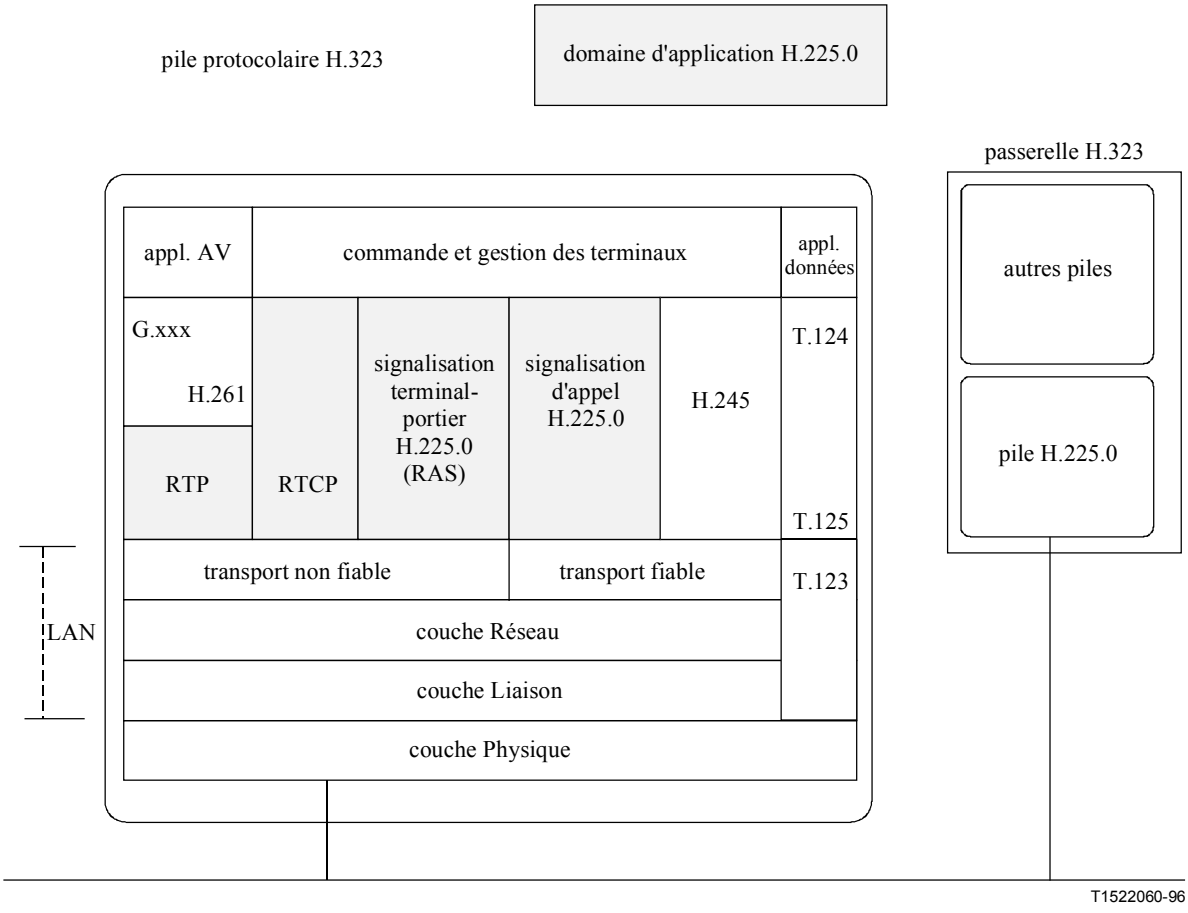


Figure 1/H.225.0 – Domaine d'application H.225.0

D'une manière générale, la présente Recommandation vise à décrire un moyen permettant de synchroniser les paquets qui utilisent les facilités sous-jacentes réseau à commutation par paquets/transport. La présente Recommandation 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 Recommandation H.221 ne sont pas utilisés pour les raisons suivantes:

- la non-utilisation de ces mécanismes permet l'utilisation de types de traitement des erreurs adaptés à chaque média;
- ces mécanismes sont relativement sensibles à une perte de groupes aléatoires de bits; la mise en paquets offre une plus grande fiabilité dans un environnement de réseau à commutation par paquets;

- les messages H.245 et Q.931 peuvent être envoyés sur les liaisons fiables offertes par le réseau à commutation par paquets;
- la souplesse et la puissance offertes par le protocole H.245 comparativement à celles offertes par le protocole H.242.

2 Références normatives

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

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

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

[3] Recommandation CCITT 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 de signaux vocaux 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 (1997), *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 (1997), *Signaux de commande et d'indication synchrones de la trame pour les systèmes audiovisuels*.

[8] Recommandation UIT-T H.233 (1995), *Système de confidentialité pour les services audiovisuels*.

[9] Recommandation UIT-T H.242 (1997), *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 (1997), *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 (1998), *Protocole de commande pour communications multimédias*.

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

[13] Recommandation UIT-T H.263 (1998), *Codage vidéo pour communications à faible débit*.

[14] Recommandation UIT-T H.320 (1997), *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 (1996), *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 (1998), *Terminal pour communications multimédias à faible débit*.

[21] Recommandation UIT-T H.310 (1996), *Systèmes et terminaux de communication audiovisuels à large bande*.

[22] Recommandation UIT-T Q.931 (1998), *Spécification de la couche 3 de l'interface usager-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 (1994), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

[25] Recommandation UIT-T X.680/Amd.1 (1995), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base – Amendement 1: règles d'extensibilité*.

[26] Recommandation UIT-T X.681/Amd.1 (1995), *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels – Amendement 1: règles d'extensibilité*.

[27] Recommandation UIT-T X.691 (1995), *Technologies de l'information – Règles de codage ASN.1 – Spécification des règles de codage compact*.

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

[29] ISO/CEI 10646-1:1993, *Technologies de l'information – Jeu universel de caractères codés à plusieurs octets – Partie 1: Architecture et table multilingue*.

[30] 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*.

[31] Recommandation UIT-T Q.950 (1997), *Protocoles pour services complémentaires, structure et principes généraux*.

[32] Recommandation UIT-T H.235 (1998), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)*.

3 Définitions

Voir les définitions de la Recommandation H.323. Selon celle-ci un "point d'extrémité" est un terminal, une passerelle ou un pont de conférence et 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 il 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 Abréviations

La présente Recommandation utilise les abréviations suivantes.

4.1 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*)

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 No. 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 avec excitation par code (*low delay – code excited linear prediction*)

LSB bit de plus faible poids; bit le moins significatif (*least significant bit*)

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

MB macrobloc (voir Recommandation 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 multitrame (*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; bit le plus significatif (*most significant bit*)

NS 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 état (*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 communications sélectionné (*selected communications 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*)

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"*)

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

ACF confirmation d'admission (*admission confirm*)

ARJ refus d'admission (*admission reject*)

ARQ demande d'admission (*admission 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*)

UCF confirmation d'annulation d'enregistrement (*unregistration confirm*)

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

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

5 Conventions

Dans la présente Recommandation le "futur" correspond à des prescriptions obligatoires et le conditionnel à des procédures optionnelles. La modale "pouvoir" correspond à des possibilités de développement sans qu'il y ait de préférences exprimées à ce sujet.

Lorsqu'un terme tel "MCU" est utilisé, il s'agit d'une unité MCU H.323. S'il s'agit d'une unité MCU H.231, cela sera explicitement mentionné.

Dans la présente Recommandation, le terme kbit désigne 1000 bits. 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 sera utilisée pour toutes les déclarations ASN.1 dans la présente Recommandation.

Les messages Q.931 sont en MAJUSCULES; L'ASN.1 est en **gras**.

6 Mécanisme de mise en paquets et de synchronisation

6.1 Méthode générale

Avant le lancement d'un appel, un point d'extrémité peut repérer un portier et s'enregistrer auprès de celui-ci. Si tel est le cas, il est souhaitable pour le point d'extrémité de connaître le "millésime" du portier auprès duquel il s'enregistre et inversement pour le portier de connaître le "millésime" du point d'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 Recommandation H.323 implémentée. Cette séquence peut aussi contenir des parties de message non standards facultatives pour permettre aux points d'extrémité d'établir des relations non standards. A la fin de cette séquence, les portiers et les points d'extrémité connaissent les numéros des versions et la situation non standard réciproque.

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

Le canal non fiable destiné à la messagerie enregistrement, admission et état est appelé le canal RAS. La méthode générale pour lancer un appel consiste à émettre d'abord une demande d'admission obligatoire sur le canal 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

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

une séquence d'établissement d'appel Q.931 avec les améliorations décrites ci-après. La séquence se termine lorsque le terminal reçoit dans un message de connexion, une adresse de transport fiable sur laquelle il envoie le message de commande H.245[4].

Lorsque les messages sont envoyés sur le canal 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 par 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 par 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 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 par paquets de la Recommandation T.123 selon le cas. Les capacités T.120 doivent être acheminées sur le réseau à commutation par paquets entre les points d'extrémité sur une autre adresse de

---

[4] Il convient de noter que l'adresse H.245 peut être envoyée dans le message d'alerte ALERTING ou d'appel en cours CALL PROCEEDING au message de connexion pour diminuer le temps d'établissement de l'appel. On notera que le canal H.245 peut être ouvert immédiatement après la réception de l'adresse H.245 dans le message d'établissement 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 Recommandation 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.

transport. Le Tableau 1 montre le nombre d'identificateurs TSAP utilisés pour chaque média dans un appel point à point. Il est également vrai qu'un terminal H.323 donné peut être en mesure de participer simultanément à plusieurs conférences, ce qui se traduit par l'utilisation d'identificateurs TSAP supplémentaires. Tous les canaux H.245 utilisés sont unidirectionnels sauf ceux qui sont associés au protocole T.120 qui sont bidirectionnels.

Tableau 1/H.225.0 – Identificateurs TSAP utilisés dans le cadre de la présente Recommandation pour chaque appel de monodiffusion 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'un seul point d'extrémité par adresse réseau. Aussi, dans le modèle d'appel direct, l'appelant doit disposer d'un identificateur TSAP connu pour le canal de signalisation d'appel pour pouvoir lancer l'appel.		

Bien que l'adresse de transport pour, par exemple, les informations audio et vidéo puisse partager la même adresse de réseau à commutation par paquets et différer uniquement par l'identificateur TSAP, certains fabricants peuvent choisir d'utiliser différentes adresses de réseau à commutation par paquets pour les données audio et pour les données vidéo. La seule condition à satisfaire est de respecter la convention des Annexes A et B pour la numérotation des identificateurs TSAP dans la session RTP[7].

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

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 chacun des points d'extrémité prenant en charge le modèle d'appel avec intervention du portier décrit à la Figure 11/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 limitée à un seul point d'extrémité par dispositif; par conséquent, il convient d'utiliser des identificateurs TSAP dynamiques. NOTE 2 – 0 pour le modèle d'appel direct; 2 pour le modèle d'appel avec intervention d'un portier.			

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

1) pour une signalisation d'appel de terminal à terminal (voir Figure 9/H.323), chaque terminal peut décider de fermer le canal de signalisation d'appel fiable ou de le laisser ouvert;

2) pour le cas de signalisation d'appel avec intervention du portier (voir Figure 8/H.323), les terminaux doivent laisser actif l'accès fiable pendant l'appel. Cependant, le portier peut choisir de fermer ce canal de signalisation, mais il doit maintenir le canal ouvert pour tous les appels qui font intervenir des passerelles. Cela permet une transmission de bout en bout des éléments d'information Q.931 (l'information d'affichage, par exemple);

3) si pour certaines raisons, la liaison fiable devient inactive suite à une anomalie au niveau transport ou à un autre problème, la liaison doit être réouverte et l'appel ne doit pas être interrompu. L'état de l'appel et l'utilisation de la valeur de référence d'appel Q.931 (CRV, *call reference value*) ne sont pas affectés par la fermeture de la liaison fiable sauf si le canal H.245 est fermé, indiquant ainsi la fin de l'appel.

Il convient de noter qu'il est possible d'ouvrir simultanément plusieurs canaux H.245 c'est-à-dire qu'un point d'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'accusé de réception d'ouverture de canal logique, le point d'extrémité utilise la séquence ARQ/ACF ou BRQ/BCF pour faire en sorte qu'une largeur de bande suffisante soit disponible pour le nouveau canal (à moins qu'une largeur de bande suffisante soit disponible d'une précédente séquence ARQ/ACF ou BRQ/BCF). Dans certains cas, la passerelle peut s'apercevoir que la modification de mode RCC est plus rapide que la modification de mode côté réseau à commutation par paquets, ce qui peut induire une perte de l'information audio. La passerelle peut adopter plusieurs approches au choix du constructeur:

a) la passerelle peut transcoder l'information audio, dissimulant ainsi les modifications de mode RCC;

b) la passerelle peut simplement rejeter l'information audio;

c) la passerelle peut fonctionner comme une unité 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 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 Recommandation H.245 fournit cette association qui peut être utilisée pour la synchronisation audio/vidéo.

En général, deux modes de fonctionnement conférence sont possibles du côté réseau à commutation par paquets: le mode décentralisé et le mode centralisé. Il est également possible de choisir différents modes pour les différents médias, par exemple décentralisé pour l'audio/vidéo et centralisé pour les données. Les procédures permettant de déterminer le type de conférence à mettre en place sont décrites dans la Recommandation 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

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

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

Sauf mention explicite, les 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 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, devra être considéré comme étant donné pour information. Il convient de noter que les passerelles et les unités MCU disposent de

certaines fonctions de traduction et de mélange et que les informations données dans les Annexes A et B peuvent être utiles pour l'implémentation de passerelles et d'unités MCU. Toutefois, ces unités ne sont pas des mélangeurs et inversement. Il convient aussi de noter que par exemple, dans un appel entre réseaux à commutation par paquets via une passerelle, celle-ci peut se comporter comme un traducteur.

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

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

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

Lorsque le protocole RTCP est utilisé, les paquets RR ou les paquets SR seront 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 Recommandation H.245 ou T.120 devront être considérées comme des informations correctes.

Il ne faut pas compter sur le message RTCP BYE pour la fin de session RTP. Le terminal H.323 détecte la déconnexion d'un appel au moyen des procédures 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 Recommandation 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 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 par paquets ne doit pas correspondre à celle de la Recommandation H.221;
- la largeur de bande audio sur le réseau à commutation par paquets peut correspondre à celle de la Recommandation H.221 sur le WAN, mais avec transcodage de passerelle, une correspondance n'est pas requise;
- dans le cas où la passerelle utilise un réducteur de débit; le terminal H.323 côté réseau à commutation par paquets ne doit pas dépasser le débit H.245 signalé, qui est probablement inférieur au débit émis sur le WAN.

Le chiffrement pour les points d'extrémité 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 Recommandation G.723, 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 Recommandations 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 seront 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-T signalés dans la Recommandation 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 le plus significatif 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 le plus significatif de l'octet tel que défini par le format du serveur).

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

Lorsqu'elle envoie un signal G.722 à 48/56 kbit/s en direction du réseau à commutation par paquets, la passerelle H.323 doit remplir d'un ou de deux bits supplémentaires chaque octet, et utiliser les types de charge utile RTP dynamique signalés dans la Recommandation H.245 pour distinguer les signaux à 64 kbit/s (qui utilisent PT = 9) des signaux à débit réduit. Dans le sens inverse, la passerelle H.323 tronquera le signal G.722 à 64 kbit/s du côté réseau à commutation par paquets pour que le débit corresponde au débit G.711 utilisé dans H.320. Ainsi, du côté réseau à commutation par paquets seuls des signaux G.722 à 64 kbit/s seront utilisés.

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

6.2.2 Messages vidéo

Type de charge utile (PT): seuls les types de charge utile définis par l'UIT-T tels ceux de la Recommandation H.261 ou H.263 seront utilisés dans le cas des codecs définis par l'UIT-T signalés dans la Recommandation 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 Recommandation 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 perte de paquets vidéo, les messages H.245 **VideoFastUpdatePicture**, **VideoFastUpdateMB** et **VideoFastUpdateGOB** doivent être pris en charge. L'utilisation des paquets de commande RTCP (FIR, *full intra request*) [Envoyez-moi une trame complète] et accusé de réception négatif (NACK, *negative acknowledgment*) [Envoyez-moi certains paquets] est optionnelle; elle est signalée dans les capacités H.245.

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

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

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

SBIT est le nombre de bits le plus significatif qu'il faut ignorer dans le premier octet de données. EBIT est le nombre de bits le moins significatif qu'il faut ignorer dans le dernier octet de données.

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

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

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

6.2.3 Messages de données

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

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

On se reportera à la Recommandation H.323 pour les procédures utilisées pour connecter une conférence T.120 en cours à une conférence H.323, ou ajouter un appel H.323 à une conférence T.120.

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

7 Définition des messages H.225.0

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

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

7.1 Utilisation des messages Q.931

Les implémentations se conformeront à la Recommandation Q.931, comme cela est spécifié dans la présente Recommandation. Les terminaux peuvent également prendre en charge les messages facultatifs Q.931 et H.450. Les messages contiendront tous les éléments d'information obligatoires et pourront contenir tout élément d'information facultatif défini dans la Recommandation Q.931, comme cela est décrit dans la présente Recommandation. Il convient de noter que le point d'extrémité H.225.0 peut, d'après la Recommandation 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 STATUS.

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

Chaque point d'extrémité H.225.0 doit être en mesure d'interpréter et de produire des éléments d'information rendus obligatoires dans ce qui suit pour les messages respectifs Q.931 et H.450. Il peut aussi interpréter et produire les éléments d'information facultatifs définis ci-dessous. Il peut aussi interpréter tout autre élément d'information du protocole H.450 ou Q.931 ou d'autres protocoles de la série Q. Les points d'extrémité doivent être en mesure d'ignorer les éléments d'information inconnus contenus dans un message Q.931 ou H.450 sans perturber le fonctionnement. Les procédures applicables à la réception d'éléments d'information "nécessaires à la compréhension" non reconnus doivent s'appliquer conformément au 5.8.7.1/Q.931.

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

1) la passerelle devrait et le portier devra retransmettre, après modification convenable, tous les éléments d'information (facultatifs ou obligatoires) associés aux messages Q.931 obligatoires, soit du terminal vers la passerelle/le portier et en sens inverse. Cela inclut des éléments d'information telles les informations d'utilisateur à utilisateur et les informations d'affichage;

2) une passerelle devrait retransmettre tous les éléments d'information et messages facultatifs Q.931 ou H.450 dans les deux sens. Si le canal de signalisation d'appel n'est pas conservé par le portier, cette retransmission n'est pas possible;

3) aussi longtemps que le canal de signalisation d'appel Q.931 est en fonctionnement, un portier retransmettra tous les éléments d'information et messages facultatifs Q.931 ou H.450 dans les deux sens après modification convenable. Si le canal de signalisation d'appel n'est pas conservé par le portier, cette retransmission n'est pas possible. A noter que le portier peut agir comme un élément de signalisation pouvant offrir des fonctions (fonctions de services complémentaires par exemple) et qu'il peut donc modifier, terminer ou envoyer des messages Q.931.

Les passerelles H.323 peuvent être capables de convertir des services complémentaires de la série H.450 en services complémentaires ISO/CEI 11582 correspondants et inversement. Elles peuvent aussi être capables de transmettre sans modification la signalisation d'un service complémentaire ISO/CEI 11582 dans l'environnement H.323 à l'intérieur de l'élément d'information d'utilisateur à utilisateur. Les détails appellent un complément d'étude.

Dans la présente version de la Recommandation, toutes les références concernent la version de 1993 de la Recommandation Q.931. Les procédures décrites au 3.1/Q.931 pour l'établissement d'une connexion en mode circuit sont respectées. Cependant, il est rappelé à la personne chargée de l'implémentation que si la signalisation indique un support, il n'existe pas de canaux B réels du RNIS du côté réseau à commutation par paquets. L'aboutissement de l'appel se traduit par la mise en place d'un canal fiable de bout en bout prenant en charge les messages H.245. L'établissement d'un "support" réel est effectué au moyen des procédures H.245. Cependant, l'utilisation du mode Q.931 du côté réseau à commutation par paquets permet l'interfonctionnement avec le mode Q.931 du côté WAN, 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. Ceci 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 points d'extrémité qui ne prennent pas en charge les jeux de code avec verrouillage Q.931 ignoreront tous les messages Q.931 utilisant ces méthodes.

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

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

	Emission (M, F, O, CM)[a)]	**Réception et action (M, F, O[b)], CM)**
Messages d'établissement d'appel		
Alerting	M	M
Call Proceeding	O	CM[c)]
Connect	M	M
Connect Acknowledge	F	F
Progress	O	O
Setup	M	M
Setup Acknowledge	O	O
Messages de libération d'appel		
Disconnect	F	F
Release	F	F
Release Complete	M[d)]	M
Messages de la phase information de l'appel		
Resume	F	F
Resume Acknowledge	F	F
Resume Reject	F	F
Suspend	F	F
Suspend Acknowledge	F	F
Suspend Reject	F	F
User Information	O	O
Messages divers		
Congestion Control	F	F
Information	O	O
Notify	O	O
Status	M[e)]	M
Status Inquiry	O	M
Messages Q.932/H.450		
Facility	M	M
Hold	F	F
Hold Acknowledge	F	F
Hold Reject	F	F

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

	Emission (M, F, O, CM)[a)]	Réception et action (M, F, O[b)], CM)
Retrieve	F	F
Retrieve Acknowledge	F	F
Retrieve Reject	F	F

a) M: obligatoire (*mandatory*), F: interdit (*forbidden*), O: optionnel, CM: obligatoire conditionnel (*conditionally mandatory*). Un message est obligatoire conditionnel s'il est obligatoire lorsqu'une option est prise en charge.

b) Il convient de noter qu'il ne faut pas envoyer de message STATUS en réponse à un message classé ici "O"; le récepteur devra simplement ignorer le message s'il ne le prend pas en charge.

c) Les terminaux utilisant des passerelles pourront recevoir le message appel en cours CALL PROCEEDING et y réagir.

d) Le message Release Complete est exigé pour toute situation dans laquelle un canal de signalisation d'appel fiable H.225.0 est ouvert. Si ce canal n'est pas ouvert, la fin de la session H.245 peut être utilisée pour terminer la conférence.

e) Le point d'extrémité réagira à un message inconnu avec un message STATUS; la réaction à un message demande d'état STATUS INQUIRY est également obligatoire. Cependant, un point d'extrémité n'est pas tenu d'envoyer un message STATUS INQUIRY. Dans la pratique, le point d'extrémité doit être capable de comprendre un message STATUS reçu en réaction à un message envoyé qui n'a pas été connu du récepteur.

7.2 Eléments d'information Q.931 communs

7.2.1 Eléments d'information d'en-tête

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

7.2.1.1 Discriminateur de protocole

Tel que défini au 4.2/Q.931.

Sa valeur est 08H – Cette valeur identifie le message comme un message Utilisateur-réseau 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 tout point d'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 le plus significatif 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 (CRV), 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 devra être choisi conformément aux procédures décrites dans la Recommandation Q.931.

A noter que les valeurs de référence d'appel transmises dans les messages RAS doivent être conformes à la structure spécifiée dans la Recommandation Q.931. En particulier, le fanion de référence d'appel doit correspondre au bit le moins significatif 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.

7.2.1.3 Type de message

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

7.2.2 Eléments d'information propres au message

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

7.2.2.1 Capacité support

Cet élément d'information est codé conformément à la Figure 4-11/Q.931 et au Tableau 4-6/Q.931. Lorsque cet élément d'information est reçu dans un appel entre réseaux à commutation par paquets, il peut être ignoré par le récepteur. Lorsque cet élément d'information figure dans un message Setup pour une connexion de signalisation indépendante de l'appel telle que définie dans la Recommandation H.450.1, le codage doit être conforme au 7.2/H.450.1. Dans tous les autres cas, les informations suivantes s'appliquent à l'utilisation des différents champs de cet élément d'information (les références de numéro d'octet renvoient à la Figure 4-11/Q.931):

Capacité de transfert de l'information (octet n° 3)

– le bit d'extension (bit 8) doit être mis à '1';
– le champ norme de codage (bits 6 et 7) doit être mise à '00' indiquant 'UIT-T';
– capacité de transfert d'information (bits 0-5):
  - pour les appels provenant d'un point d'extrémité de RNIS, l'information indiquée à la passerelle doit être retransmise;

    NOTE – Cela est destiné à permettre la retransmission vers le point d'extrémité H.323 d'information à l'avance sur la nature de la connexion, par exemple voix uniquement par opposition à données par opposition à vidéo; cette information aurait des conséquences sur la largeur de bande requise ainsi que sur la possibilité ou la volonté d'accepter un appel ou de le refuser.
  - les appels qui proviennent d'un point d'extrémité H.323 doivent utiliser ce champ pour indiquer qu'il désire lancer une appel audiovisuel. Par conséquent, ce champ doit indiquer "information numérique sans restriction" (c'est-à-dire '01000') ou "information numérique avec restriction" (c'est-à-dire '01001'). Si l'on veut lancer uniquement un appel vocal, le terminal H.323 doit indiquer pour la capacité de transfert d'information "parole" (c'est-à-dire '00000') ou "audio à 3,1 kHz" (c'est-à-dire '10000').

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

– Doit être mis à '0' si le débit de transfert d'information est "multidébit", ou à '1' dans les autres cas.

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

– Doit spécifier "mode circuit", sa valeur est '00'.

Débit de transfert d'information

– Peut être codé conformément au Tableau 4-6/Q.931 sauf que la valeur '00000' (pour le mode paquet) n'est pas autorisée sauf si la passerelle ne se connecte à un réseau de transmission par paquets.

Multiplicateur de débit – octet n° 4.1

– Doit être présent si le débit de transfert de l'information est fixé à "multidébit".
– Le bit d'extension (bit 8) doit être mis à '1'.
– Les bits 1-7 doivent indiquer la largeur de bande nécessaire pour l'appel tel que défini dans ce qui suit (il convient de noter que par rapport à la Recommandation Q.931, la valeur '0000001' est autorisée).
– Pour un appel provenant d'un point d'extrémité de RNIS, la passerelle doit simplement faire transférer l'information reçue du RNIS.
– Pour un appel provenant d'un point d'extrémité H.324, la passerelle doit fixer le multiplicateur de débit à 01H.
– Pour un appel entrant provenant d'un RNIS-LB, une certaine traduction Q.2931 à Q.931 doit être effectuée. Ce sujet appelle un complément d'étude.
– Pour un appel provenant d'un point d'extrémité H.323, cet octet sera utilisé pour indiquer la largeur de bande à utiliser pour cet appel. Si le système appelé est un autre point d'extrémité H.323, cette valeur peut indiquer la largeur de bande à utiliser sur le réseau à commutation par paquets, mais le terminal de réception n'est pas tenu de se conformer à cette information. S'il y a intervention d'une passerelle, la valeur doit indiquer le nombre de connexions externes à établir. La largeur de bande nécessaire pour l'appel correspond à la largeur de bande nécessaire du côté RCC, elle ne correspond pas nécessairement à la largeur de bande autorisée sur le réseau à commutation par paquets par les messages ACF/BCF.

Protocole de couche 1 – octet n° 5

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

Les octets n° 5a, n° 5b, n° 5c et n° 5d ne seront pas présents.

Identificateur de protocole de couche 2 – Octet n° 6

– Ne sera pas présent.

Identificateur de protocole de couche 3 – Octet n° 7

– Ne sera pas présent.

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 →portier → terminal et terminal → passerelle → terminal et l'acheminement source libre.

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 seront pas 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. Lorsque le code est '1001' (plan de numérotage privé) dans un appel au départ d'un réseau à commutation par paquets, cela indique que:

  1) l'adresse E.164 n'est pas présente dans le message SETUP;

  2) l'appel sera acheminé via une adresse alias dans l'information utilisateur-utilisateur.

Chiffres du numéro

– Nombre quelconque de caractères IA5, selon les formats spécifiés dans le plan de numérotage/de numérotation.

7.2.2.5 Sous-adresse de l'appelé

A utiliser comme indiqué dans la Recommandation 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.

Octet n° 3 Extension (bit 8)

– Mis à '1'.

Type de numéro (octet n° 3, bits 5-7)

– Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-9/Q.931.

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

– Codé conformément aux valeurs et aux règles indiquées dans le Tableau 4-9/Q.931. Si sa valeur est '1001' (plan de numérotage privé) dans un appel au départ d'un réseau à commutation par paquets, cela indique que:

  1) l'adresse E.164 n'est pas présente dans le message SETUP;

  2) que l'appel sera acheminé via une adresse alias dans l'information 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.

7.2.2.7 Sous-adresse de l'appelant

Conformément à la Recommandation Q.931.

7.2.2.8 Cause

Lorsque cet élément d'information est reçu, les règles définies dans la Recommandation Q.850 sont applicables. Il convient de noter que l'un ou l'autre de l'élément d'information cause et de l'élément **RelCompReason** est obligatoire pour le message libération terminée 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 circuits depuis le côté réseau à commutation par paquets (voir Tableau 5). (Le mappage inverse n'est pas nécessaire car les entités du réseau à commutation par paquets sont tenues de décoder l'élément d'information cause.)

Tableau 5/H.225.0 – Mappage entre le motif de fin de libération et l'élément d'information "cause"

Code de l'élément ReleaseCompleteReason	Valeur de cause Q.931/Q.850 correspondante
noBandwidth	34 – pas de circuit/canal disponible
gatekeeperResources	47 – ressource non disponible
unreachableDestination	3 – pas de route vers la destination
destinationRejection	16 – libération normale d'appel
invalidRevision	88 – destination incompatible
noPermission	111 – erreur de protocole, non spécifié
unreachableGatekeeper	38 – réseau hors service
gatewayResources	42 – encombrement de l'équipement de commutation
badFormatAddress	28 – format de numéro non valide
adaptiveBusy	41 – échec temporaire
inConf	17 – utilisateur occupé

Tableau 5/H.225.0 – Mappage entre le motif de fin de libération et l'élément d'information "cause" *(fin)*

Code de l'élément ReleaseCompleteReason	Valeur de cause Q.931/Q.850 correspondante
undefinedReason	31 – normal, non spécifié
facilityCallDeflection	16 – libération normal d'appel
securityDenied	31 – normal, non spécifié
calledPartyNotRegistered	20 – abonné absent
callerNotRegistered	31 – normal, non spécifié

7.2.2.9 Identification de canal

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

7.2.2.10 Numéro connecté

Codé conformément au 5.4.1/Q.951.

7.2.2.11 Sous-adresse du numéro connecté

Codé conformément au 5.4.2/Q.951.

7.2.2.12 Niveau d'encombrement

Ne sera pas 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 Fonctionnalité étendue

Lorsque cet élément d'information est utilisé pour indiquer une sémantique non modifiée telle que définie dans les Recommandations de la série Q.95x, il doit être codé conformément au 8.2.4/Q.932. Dans ce cas, les unités PDU de service doivent être constituées en fonction de l'élément ROSE [qui utilise la Recommandation X.208 (spécification de la syntaxe abstraite numéro un (ASN.1)) et la Recommandation X.209 (spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1))] comme cela est défini dans la Recommandation 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 acheminé au portier) ou en cas de signalisation d'un service complémentaire conformément aux Recommandations de la série H.450x, 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 de facilité de longueur zéro, c'est-à-dire que l'élément d'information de facilité 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 de facilité;

- 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 de facilité 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 un point d'extrémité est amené à appeler un autre point d'extrémité car le point d'extrémité appelant souhaite participer à une conférence et le point d'extrémité appelé n'incorpore pas le contrôleur multipoint, l'élément d'information de facilité doit aussi être vide. L'élément **conferenceID** devra indiquer la conférence à laquelle le point d'extrémité souhaite participer et le motif figurant dans l'élément **Facility-UUIE** devra être **routeCallToMC**.

De même, dans le cas où le point d'extrémité appelant est amené à signaler le point d'extrémité appelé au portier de ce dernier, l'élément d'information de facilité est vide. Le champ **conferenceID** de l'élément d'information **Facility-UUIE** devra indiquer la conférence à laquelle le point d'extrémité souhaite participer et le motif figurant dans l'élément **Facility-UUIE** devra être **routeCallToGatekeeper**.

Lorsque l'élément d'information de facilité est utilisé pour indiquer une sémantique non modifiée telle que définie dans les Recommandations de la série Q.95x, il doit être codé conformément au 8.2.3/Q.932. Dans ce cas, les unités PDU de service doivent être constituées en fonction de l'élément ROSE [qui utilise la Recommandation X.208 (Spécification de la syntaxe abstraite numéro un (ASN.1)) et la Recommandation X.209 (Spécification des règles de codage de base pour la notation de syntaxe abstraite numéro un (ASN.1))] comme cela est défini dans la Recommandation X.229.

7.2.2.17 Compatibilité de couche supérieure

A étudier.

7.2.2.18 Clavier

Codé conformément à la Figure 4-24/Q.931.

7.2.2.19 Compatibilité de couche inférieure

A étudier.

7.2.2.20 Données à suivre

Ne sera pas utilisé.

7.2.2.21 Facilités spécifiques au réseau

Ne sera pas 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)

– Indiquera "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 Indicateur de répétition

Ne sera pas utilisé.

7.2.2.25 Indicateur de reprise

Ne sera pas utilisé.

7.2.2.26 Message fractionné

Ne sera pas utilisé. Il convient de noter qu'il n'y a pas de limite supérieure critique pour la taille d'un message dans la Recommandation H.323 et la présente Recommandation.

7.2.2.27 Numérotation complète

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

Il n'y a pas de restriction.

7.2.2.28 Signal

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

Il n'y a pas de restriction.

7.2.2.29 Sélection du réseau de transit

Ne sera pas utilisé.

7.2.2.30 Utilisateur-utilisateur

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

L'élément d'information utilisateur-utilisateur sera utilisé par toutes les entités H.323 pour acheminer les informations H.323 associées. L'information réelle utilisateur-utilisateur à échanger entre les terminaux intervenant est emboîtée avec l'unité H.323-UserInformation PDU (qui ne fait l'objet d'aucune restriction).

Les restrictions suivantes sont applicables:

Longueur des contenus utilisateur-utilisateur

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

Discriminateur de protocole

– Indiquera une information d'utilisateur codée à X.208 et X.209 (ASN.1) ('00000101').

NOTE – Ce codage est conforme à la révision de 1993 de la Recommandation Q.931 qui renvoie aux révisions précédentes de l'ASN.1. Les références correctes à l'ASN.1 sont les Recommandations X.680 (syntaxe) et X.691 (règles de codage compactes PER).

Information d'utilisateur

– Contiendra une structure ASN.1 qui parallèlement aux informations pertinentes H.323 inclut des données d'utilisateur réelles, par exemple comme suit. L'ASN.1 est codé en utilisant la variation "basic aligned" des règles de codage compactes spécifiées dans la Recommandation X.691. Il convient de noter que la structure ASN.1 commence par **H323-UserInformation**;

– On utilisera pour le champ information d'utilisateur les règles spécifiées au 4.5.30/Q.931.

7.3 Informations complémentaires concernant les messages Q.931

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

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

7.3.1 Alerte (Alerting)

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

Se conformer au Tableau 3-2/Q.931 (version 1993) tel que modifié ci-après dans le Tableau 6.

Tableau 6/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
Facilité étendue	O	8-*
Identification du canal	A étudier	Sans objet
Facilité	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	Sans objet
Utilisateur à utilisateur	M	2-131

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 Recommandation H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type de point 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 le point d'extrémité appelé ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse peut également être envoyée dans le message Call Proceeding, Progress ou Connect.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message Setup avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CallProceeding, Alerting, Progress ou Connect.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion d'appel rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'un canal logique. Il utilise la structure OpenLogicalChannel définie dans la Recommandation 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 trains multimédias.

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

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

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

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

7.3.2 Appel en cours (Call Proceeding)

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

Tableau 7/H.225.0 – Contenu du message Call Proceeding (Appel en cours)

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
Facilité étendue	O	8-*
Identification du canal	A étudier	Sans objet

Tableau 7/H.225.0 – Contenu du message Call Proceeding (Appel en cours) *(fin)*

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Facilité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Compatibilité de couche supérieure	A étudier	Sans objet
Utilisateur à utilisateur	M	2-131

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 Recommandation H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type de point 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 le point d'extrémité appelé ou le portier qui traite l'appel aimerait établir la signalisation H.245.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message Setup avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CallProceeding, Alerting, Progress ou Connect.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'un canal logique. Il utilise la structure OpenLogicalChannel définie dans la Recommandation 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 trains multimédias.

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

7.3.3 Connexion (Connect)

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

Tableau 8/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 (Note)	5-6
Facilité étendue	O	8-*
Identification du canal	A étudier	Sans objet
Facilité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*
Affichage	O	2-82
Date/heure	O	8
Compatibilité de couche supérieure	A étudier	Sans objet
Compatibilité de couche inférieure	A étudier	Sans objet
Utilisateur à utilisateur	M	2-131
Numéro connecté	O	2-*
Sous-adresse du numéro connecté	O	2-23
NOTE – L'élément d'information capacité support est obligatoire si le message est échangé entre un terminal et une passerelle.		

L'élément d'information utilisateur 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 Recommandation H.225.0 acceptée.

h245Address – Adresse de transport spécifique sur laquelle le point d'extrémité appelé ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse doit être envoyée si elle a été envoyée antérieurement dans le message Alerting, Progress ou Call Proceeding.

destinationInfo – Contient un **EndpointType** (type de point 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 global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message Setup avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CallProceeding, Alerting, Progress ou Connect.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'un canal logique. Il utilise la structure OpenLogicalChannel définie dans la Recommandation 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 trains multimédias.

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'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 à la norme RFC 1766.

connectedAddress – Contient les adresses alias pour le correspondant connecté (qui répond); le numéro E.164 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 le point d'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.

7.3.4 Accusé de réception de connexion (Connect Acknowledge)

Ce message ne devra pas être envoyé.

7.3.5 Déconnexion (Disconnect)

Ce message ne sera pas 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; son traitement à la réception est facultatif.

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

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

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
Facilité clavier	O	2-34
Signal	O	2-3
Numéro demandé	O	2-35
Utilisateur à utilisateur	M	2-131

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 Recommandation H.225.0 acceptée.

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

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 un point d'extrémité H.323 avant le message Connect, en fonction de l'interaction avec des services complémentaires.

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

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

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
Cause	O	2-32
Facilité étendue	O	8-*
Identification du canal	A étudier	Sans objet
Facilité	O	8-*
Indicateur de progression	O	2-4
Indicateur de notification	O	2-*

Tableau 10/H.225.0 – Contenu du message Progress (Progression) *(fin)*

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Affichage	O	2-82
Compatibilité de couche supérieure	A étudier	Sans objet
Utilisateur à utilisateur	M	2-131
NOTE – L'élément d'information capacité support est obligatoire si le message est échangé entre un terminal et une passerelle.		

L'élément d'information utilisateur 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 Recommandation H.225.0 acceptée.

destinationInfo – Contient un **EndpointType** (type de point 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 le point d'extrémité appelé ou le portier qui traite l'appel aimerait établir la signalisation H.245. Cette adresse doit être envoyée si elle a été envoyée antérieurement dans le message Call Proceeding, Alerting ou Connect.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

h245SecurityMode – Une entité H.323 qui reçoit un message Setup avec la capacité **h245SecurityCapability** positionnée doit répondre avec le mode **h245SecurityMode** correspondant, acceptable dans le message CallProceeding, Alerting, Progress ou Connect.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion d'appel rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'un canal logique. Il utilise la structure OpenLogicalChannel définie dans la Recommandation 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 trains multimédias.

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

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 Tableau 3-10/Q.931 et au 10.5 de l'ISO/CEI 11582.

7.3.9 Fin de libération (Release Complete)

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

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

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

Tableau 11/H.225.0 – Contenu du message Release Complete (Fin de libération)

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)	1
Facilité	O	8-*
Indicateur de notification	O	2-*
Affichage	O	2-82
Signal	O	2-3
Utilisateur à utilisateur	M	2-131
NOTE – L'un ou l'autre de l'élément d'information Cause et de l'élément **ReleaseCompleteReason** devra être présent.		

Si ce message est envoyé en réponse à un message Facility où l'élément d'information de facilité 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 Recommandation 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 Recommandation H.225.0 acceptée.

reason – Plus de renseignements sur le motif de la libération de l'appel.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

busyAddress – Contient les adresses alias 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 le point d'extrémité ou le réseau (portier) et si elle a été filtrée par un portier.

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 3-16/Q.931 modifié comme au Tableau 12.

Tableau 12/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	Sans objet
Capacité support	M	5-6
Facilité étendue	O	8-*
Identification du canal	A étudier	Sans objet
Facilité	O	8-*
Indicateur de progression	F	Sans objet
Facilités propres au réseau	F	Sans objet
Indicateur de notification	O	2-*
Affichage	O	2-82
Facilité clavier	O	2-34
Signal	O	2-3
Numéro du demandeur	O	2-131
Sous-adresse du demandeur	CM (Note 1)	Sans objet
Numéro du demandé	O	2-131
Sous-adresse du demandé	CM (Note 1)	Sans objet
Sélection du réseau de transit	F	Sans objet
Indicateur de répétition	F	Sans objet
Compatibilité de couche inférieure	A étudier	Sans objet
Compatibilité de couche supérieure	A étudier	Sans objet
Utilisateur à utilisateur	M	2-131
NOTE 1 – Les sous-adresses sont nécessaires pour certains scénarios d'appel RCC; elles ne doivent pas être utilisées pour des appels côté réseau à commutation par paquets seulement. NOTE 2 – Si un message ARQ a été précédemment envoyé, la valeur de référence d'appel utilisée ici devra 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 Recommandation H.225.0 acceptée.

h245Address – Adresse de transport spécifique sur laquelle le point d'extrémité appelé 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 le canal de signalisation d'appel.

sourceAddress – Contient les adresses alias pour la source; le numéro E.164 de la source est dans l'élément d'information numéro de l'appelant Q.931. L'adresse primaire doit se trouver en premier.

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 le point d'extrémité souhaite être connecté. L'adresse primaire devra se trouver en premier. Pour appeler un point d'extrémité en utilisant uniquement l'adresse E.164, cette adresse devra être placée dans l'élément d'information numéro de l'appelé Q.931. Si le champ destinationAddress est disponible, il devra être inclus dans le message Setup par les terminaux de version 2.

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 WAN. Doit seulement contenir les adresses E.164 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 utilisée dans la présente Recommandation.

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

conferenceID – Identificateur univoque de la conférence.

conferenceGoal:

create – Démarrer 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 devra 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 devra être rempli. La valeur du champ sourceCallSignalAddress devra être identique à la valeur qui a été utilisée dans le message ARQ par l'expéditeur du message Setup et devra être utilisée par le point d'extrémité recevant le message Setup dans son message ARQ.

remoteExtensionAddress – Contient l'adresse alias d'un point d'extrémité appelé 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 devra être rempli.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

fastStart – Utilisé uniquement dans la procédure de connexion rapide, **fastStart** prend en charge la signalisation nécessaire à l'ouverture d'un canal logique. Il utilise la structure OpenLogicalChannel définie dans la Recommandation 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 trains multimédias.

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

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

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

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'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 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 points d'extrémité 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 points d'extrémité 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 points d'extrémité 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 à la norme 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 le point d'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.

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 seront 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 sera traité comme ayant échoué.

7.3.11 Accusé de réception 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.

7.3.12 Etat (Status)

Le message STATUS sera 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 avec la seule modification suivante: la valeur de l'élément d'information CRV est de 2 octets.

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 avec la seule modification suivante: la longueur de l'élément d'information référence d'appel est de 3 octets.

7.4 Détails des message Q.932

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

7.4.1 Facilité (Facility)

Le message Facility sera utilisé pour fournir des informations sur l'endroit où un appel doit être dirigé (FacilityReason = routeCallToMC) ou sera utilisé par un point d'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 du message Facility est utilisé. Ce cas particulier doit être indiqué par le codage d'un élément d'information de facilité de longueur zéro. Dans ce cas, l'élément d'information de facilité doit comporter exactement 2 octets. Une entité H.323 doit traiter correctement l'élément d'information de facilité (propre aux procédures H.323) vide et doit être capable de sauter les autres éléments d'information de facilité qu'elle ne comprend pas.

Le message Facility peut être utilisé pour demander un service complémentaire ou en accuser réception conformément aux Recommandations de la série H.450. C'est pourquoi une ou plusieurs unités APDU de services complémentaires H.450 doivent être transportées à l'intérieur de l'élément d'information d'utilisateur à utilisateur du message Facility. Les unités APDU de services complémentaires H.450 doivent être codées conformément au paragraphe 8/H.450.1. L'élément d'information de facilité contenu doit être de longueur zéro. A noter qu'un message Facility qui ne transporte que des unités APDU de services complémentaires H.450 n'utiliserait pas l'élément d'information Facility-UUIE, mais utiliserait à la place l'élément **h323-message-body** "vide".

Si un élément d'information de facilité transportant la sémantique de la Recommandation Q.932 et codé conformément aux dispositions des Recommandations Q.932 et Q.95x est présent, il devra être composé d'au moins 8 octets, comme cela est exigé dans le Tableau 7-2/Q.932. L'utilisation d'éléments d'information de facilité de ce type appelle un complément d'étude.

Le message de facilité peut être utilisé par un point d'extrémité ou par un portier pour demander au destinataire d'établir une voie H.245 entre les deux entités (FacilityReason = starth245).

Se conformer au 7.1.1/Q.932 et au 10.8 de l'ISO/CEI 11582, tels que modifiés au Tableau 13.

Tableau 13/H.225.0 – Contenu du message Facility (Facilité)

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
Facilité étendue	O (Note)	8-*
Facilité	O (Note)	2 ou 8-*
Indicateur de notification	O	2-*
Affichage	O	2-82
Numéro de l'appelant	F	Sans objet

Tableau 13/H.225.0 – Contenu du message Facility (Facilité) *(fin)*

Elément d'information	Statut H.225.0 (M/F/O)	Longueur H.225.0
Numéro de l'appelé	F	Sans objet
Utilisateur-utilisateur	M	2-131
NOTE – Si le message Facility est utilisé pour transporter la signalisation de services complémentaires Q.95x, l'un des deux éléments d'information facilité et facilité 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 réacheminement vers les fonctions du contrôleur multipoint/portier, alors l'élément d'information de facilité de longueur zéro est nécessaire.		

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

L'élément d'information de type de message du message Facility devra 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 Recommandation 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 alias 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.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité de départ qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée 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 WAN. Doit seulement contenir les adresses E.164 et ne doit pas contenir le numéro du canal initial.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 de facilité souhaite que le destinataire établisse la voie de signalisation H.245.

multipleCalls – Si mis à TRUE, indique que l'expéditeur du message est en mesure de signaler plusieurs appels sur une seule connexion de signalisation d'appel.

maintainConnection – Si mis à TRUE, indique que l'expéditeur du message est en mesure de prendre en charge une connexion de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

7.4.2 Notification (Notify)

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

Le contenu et la sémantique d'un message NOTIFY sont définis au 3.1.7/Q.931.

7.4.3 Autres messages

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

7.5 Temporisations Q.931

Deux temporisations Q.931 doivent être prises en charge:

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

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

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

7.6 Eléments communs des messages H.225.0

Le présent sous-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 Q.931.

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

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

nonStandardParameter: ce paramètre est facultatif dans les séquences recherche, enregistrement/connexion pour permettre aux correspondants concernés de déterminer le statut non standard des points d'extrémité. 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 de plus fort poids des adresses IPv4 et IPv6 sera le premier octet de la chaîne d'octets, par exemple le '130' de l'adresse IPv4 de classe B 130.1.2.97 sera codé dans le premier octet de la chaîne d'octets, suivi du '1' et ainsi de suite.

Le 'a1' de l'adresse IPv6 a148:2:3:4:a:b:c:d sera 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 de plus fort poids de chaque champ des adresses IPX **node, netnum** et **port** sera le premier octet de la chaîne d'octets.

Il convient de noter que ces structures n'utilisent pas d'adresse de transport = langage "adresse de réseau à commutation par paquets plus identificateur TSAP" de la Recommandation H.323. On utilise seulement les termes communs à chaque domaine de transport.

La structure **EndpointType** achemine les informations concernant l'élément H.323 à l'extrémité de la liaison de signalisation. L'élément H.323 remplirait un ou plusieurs des éléments de message **gatekeeper**, **gateway**, **mcu** ou **terminal**. Si l'élément H.323 comporte un contrôleur multipoint, le booléen **mc** serait "Vrai". La présence de l'élément **set** indique que l'entité est un dispositif type de point d'extrémité simple (SET, *simple endpoint type*) tel que défini dans l'Annexe F de la Recommandation H.323, notamment. Les positions binaires à l'intérieur de cet élément indiquent le type de dispositif SET; leur signification est définie dans l'Annexe F/H.223 et dans d'autres Recommandations qui spécifient les types de dispositifs SET.

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

dataRatesSupported indique les débits supportés pour chaque protocole que le dispositif prend en charge. supportedPrefixes indique les préfixes associés à un protocole pris en charge et, dans certains cas aussi, aux débits.

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

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

La structure **AliasAddress** est destinée à saisir les différents formats d'adresse extérieurs qui référencent un emplacement de transport particulier sur le réseau à commutation par paquets. Lorsqu'un point d'extrémité enregistrera une adresse E.164 auprès d'un portier, il n'utilisera que les chiffres 0-9 dans le champ **e164**.

Les mécanismes qu'un point d'extrémité utilise pour déterminer le type d'adresse seront choisis au moment de l'implémentation. La représentation des divers types de numéro dans les messages est donnée dans le Tableau 14. A noter que si un point d'extrémité ne connaît pas le type ou la portée d'un numéro E.164, il doit le représenter comme "privé inconnu" lorsqu'il le code dans un message Q.931 et comme "e164 AliasAddress" lorsqu'il le code dans des messages RAS.

Tableau 14/H.225.0 – Mappage des représentations des types de numéros

Type de numéro	Représentation Q.931	Représentation RAS H.225
Inconnu (mode d'interfonctionnement pour la version 1 et par défaut)	Plan de numérotage privé – Inconnu	e164 AliasAddress
Privé inconnu	Plan de numérotage privé – Inconnu	e164 AliasAddress
Privé, abonné/abrégé, etc.	Plan de numérotage privé, type de numéro fonction de la portée	privateNumber de PartyNumber AliasAddress
Numéro public, tout type	Plan de numérotage RNIS/téléphonie	publicNumber de PartyNumber AliasAddress

La structure **Endpoint** sert à donner des informations en double, redondantes ou d'autres informations concernant un point d'extrémité:

– **nonStandardData**: transporte des informations non définies dans la présente Recommandation (par exemple, données privées).

– **aliasAddress**: liste des adresses alias, au moyen desquelles d'autres points d'extrémité peuvent identifier le point d'extrémité considéré.

– **callSignalAddress**: adresse de transport utilisée par le point d'extrémité considéré pour la signalisation d'appel.

– **rasAddress**: adresse de transport utilisée par le point d'extrémité considéré pour les messages d'enregistrement et d'état.

– **endpointType**: spécifie le type de point d'extrémité.

– **tokens**: jetons associés à cette extrémité (c'est-à-dire au point d'extrémité décrit dans la structure Endpoint).

– **cryptoTokens**: cryptojetons associés à cette extrémité (c'est-à-dire au point décrit dans la structure Endpoint).

– **priority**: utilisé lorsqu'une SEQUENCE de points d'extrémité est présentée. Les points d'extrémité ayant un petit numéro de priorité sont prioritaires par rapport à ceux qui ont un grand numéro de priorité. Les points d'extrémité sans numéro de priorité sont équivalents à ceux qui ont la priorité 0 (premier rang de priorité).

– **remoteExtensionAddress**: élément contenant l'adresse alias d'une extrémité lorsque cette information est nécessaire pour traverser des passerelles multiples.

– **destExtraCallInfo**: contient des adresses externes pour les appels multiples.

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

– **rasAddress**: adresse de transport utilisée pour la signalisation des messages RAS.

– **gatekeeperIdentifier**: inclus facultativement pour identifier le portier de secours ou de remplacement. Si ce champ est fourni, il devra figurer dans les futurs messages RAS envoyés au portier de secours.

– **needToRegister**: mis à TRUE pour indiquer que le point d'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 pour permettre au client de renouveler sa demande au moyen de l'identificateur de portier et de l'adresse de signalisation RAS.

– **altGKisPermanent**: ce paramètre a la valeur TRUE si tous les futurs signaux RAS doivent être renvoyés vers une adresse à partir d'un portier de remplacement. Il a la valeur FALSE si seul le message qui a causé le rejet doit être renvoyé.

Un portier peut envoyer à un point d'extrémité une liste de portiers de remplacement dans divers messages. Lorsqu'il se mettra en rapport avec son portier, un point d'extrémité qui implémente le mécanisme de portiers de remplacement devra remplacer toute liste de portiers de remplacement reçue antérieurement par la dernière liste de portiers de remplacement reçue. Un portier de remplacement peut lui-même envoyer une liste de portiers de remplacement. Si un point d'extrémité envoie une demande à un portier de remplacement qui est susceptible de devenir son portier permanent, ce point d'extrémité doit accepter la nouvelle liste de portiers de remplacement. Dans le cas contraire, si le portier de remplacement n'est pas susceptible de devenir le portier permanent de ce point d'extrémité, toute liste de portiers de remplacement reçue doit être ignorée. Un portier peut éventuellement devenir le portier permanent d'un point d'extrémité si le portier actuel ne réagit pas ou si le fanion "altGKisPermanent" est mis à TRUE dans la structure "AltGKInfo".

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

Les identificateurs GloballyUniqueID et ConferenceIdentifier sont supposés être uniques sur le plan global (GloballyUniqueID); leur utilisation est décrite dans la Recommandation H.323. L'identificateur GloballyUniqueID est codé avec l'octet zéro codé en premier. Il est constitué conformément au Tableau 15.

Tableau 15/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 représentant le temps universel coordonné (UTC, *coordinated universal time*) sous forme du nombre d'intervalles de 100 nanosecondes écoulés 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 bits 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 16).

Tableau 16/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 monodiffusion/multidiffusion et c'est l'octet de l'adresse qui est transmis en premier dans un réseau à commutation par paquets 802.3.

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

- le générateur d'identificateur GloballyUniqueID détecte que la valeur locale du temps UTC a fait marche arrière; ceci 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 généré 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 15;
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 monodiffusion/multidiffusion, qui ne sera 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 générer 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 généré. 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 valable.

7.7 Prise en charge requise des messages RAS

Le Tableau 17 montre les différents messages RAS qui sont pris en charge par différents types de point d'extrémité:

Tableau 17/H.225.0 – Statut des messages RAS

Message RAS	Point d'extrémité (émission)	Point d'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

Tableau 17/H.225.0 – Statut des messages RAS *(fin)*

Message RAS	Point d'extrémité (émission)	Point d'extrémité (réception)	Portier (émission)	Portier (réception)
URJ	O	O	M	O
ARQ	M			M
ACF		M	M	
ARJ		M	M	
BRQ	M	M	O	M
BCF	M (Note 1)	M	M	O
BRJ	M	M	M	O
IRQ		M	M	
IRR	M			M
IACK		O	CM	
INAK		O	CM	
DRQ	M	M	O	M
DCF	M	M	M	M
DRJ	M (Note 2)	M	M	M
LRQ	O		O	M
LCF		O	M	O
LRJ		O	M	O
NSM	O	O	O	O
XRS	M	M	M	M
RIP	CM	M	CM	M
RAI	O			M
RAC		O	M	

M obligatoire (*mandatory*), O optionnel, F interdit (*forbidden*), CM obligatoire conditionnel (*conditionally mandatory*), un blanc "non applicable".

NOTE 1 – Il convient de noter que si un portier envoie un message BRQ demandant un débit plus faible, le point d'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, le point d'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 un 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 le point d'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 point d'extrémité logique; ainsi une unité MCU ou une passerelle peut en envoyer plusieurs.

Le message GRQ comprend ce qui suit:

requestSeqNum – Numéro strictement croissant unique du point de vue de 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 du point d'extrémité expéditeur.

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

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

endpointType – Spécifie le ou les types du point d'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 alias 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 – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

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

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

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 H.225.0 du portier qui accepte la demande.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple, 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. Le client doit utiliser ces autres portiers dans le futur si, le portier ne répond pas à une demande.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

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, données privées).

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

rejectReason – Codes indiquant pourquoi le message GRQ a été refusé par ce portier.

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande à l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.9 Messages d'enregistrement de terminal et de portier

Le message RRQ est une demande d'enregistrement formulée 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 RRQ (demande d'enregistrement)

Le message RRQ comprend ce qui suit:

requestSeqNum – Numéro strictement croissant unique du point de vue de l'expéditeur. Il doit être renvoyé par le destinataire dans toute réponse associée à ce message spécifique.

protocolIdentifier – Identifie le millésime H.225.0 du point d'extrémité expéditeur.

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

discoveryComplete – Mis à TRUE si le point d'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 le point d'extrémité ne réussira pas à envoyer de message RRQ ou ARQ, le code de motif étant respectivement **discoveryRequired** et **notRegistered**. Cela indique que le point d'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 le point d'extrémité considéré 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 le point d'extrémité considéré pour les messages d'enregistrement et d'état.

terminalType – Spécifie le ou les types du point d'extrémité qui sont enregistrés; il convient de noter que le bit MC ne sera pas fixé par lui-même; le bit de terminal, de MCU, de passerelle ou de portier sera également fixé. Si des informations relatives au vendeur sont fournies, elles doivent être identiques à celles qui sont contenues dans **endpointVendor**.

terminalAlias – Cette valeur facultative est une liste d'adresses alias, au moyen desquelles d'autres terminaux peuvent identifier le terminal considéré. Si la séquence **terminalAlias** est nulle, ou une adresse E.164 n'est pas présente, une adresse E.164 peut être assignée par le portier et incluse dans le message RCF. Si un identificateur email-ID est disponible pour le point d'extrémité, il doit être enregistré. Il convient de noter que plusieurs adresses alias peuvent renvoyer aux mêmes adresses de transport. Tous les alias du point d'extrémité doivent figurer dans chaque message RRQ.

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

endpointVendor – Information sur le vendeur du point d'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 – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 le point d'extrémité a envoyé ce message RRQ comme un "maintien d'enregistrement". Un point d'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 le point d'extrémité n'est pas enregistré.

endpointIdentifier – Identificateur de point d'extrémité fourni par le portier dans le message RCF d'origine.

willSupplyUUIEs – Si ce champ est mis à TRUE, il indique que le point d'extrémité fournira des informations de message Q.931 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 de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

supportAnnexECallSignalling – Si ce champ est mis à TRUE, il indique que l'expéditeur de ce message est en mesure d'assurer la signalisation d'appel sur un canal de transport non fiable tel que défini dans l'Annexe E de la Recommandation H.323.

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, 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. Ces adresses incluent l'identificateur TSAP.

terminalAlias – Cette valeur facultative est une liste d'adresses alias, au moyen desquelles d'autres terminaux peuvent identifier le terminal en question.

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. Le client doit utiliser ces autres éléments dans le futur si le portier ne répond pas à une demande.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons chiffrés.

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

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

preGrantedARQ – Indique les événements pour lesquels le portier a préaccordé l'admission. Ceci permet d'établir plus rapidement les appels dans les environnements où l'admission est garantie par des moyens autres que l'échange des messages ARQ/ACF. A noter que même si ces champs sont mis à TRUE, un point d'extrémité peut toujours envoyer un message ARQ au portier pour certains motifs (traduction d'une adresse, non-prise en charge par le point d'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 au point d'extrémité l'autorisation de lancer des appels sans avoir à envoyer d'abord un message ARQ. Si le fanion **makeCall** est à FALSE, le point d'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 le point d'extrémité n'envoie pas de message ARQ au portier pour lancer un appel, il doit envoyer toute la signalisation d'appel H.225.0 au canal de signalisation d'appel du portier.

– **answerCall** – Si le fanion **answerCall** est à TRUE, le portier a préaccordé au point d'extrémité l'autorisation de répondre aux appels sans avoir à envoyer d'abord un message ARQ. Si le fanion **answerCall** est à FALSE, le point d'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'un point d'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 un point d'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 le point d'extrémité participe à une ou plusieurs communications. S'il n'est pas présent, le portier ne veut pas de messages IRR non sollicités. Lorsque le point d'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", et 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** – Limite l'utilisation de la largeur de bande au seul point d'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.

– **useAnnexECallSignalling** – S'il est mis à TRUE, ce paramètre indique que le point d'extrémité qui reçoit ce message doit utiliser la signalisation d'appel exclusivement sur un canal de transport non fiable tel que défini dans l'Annexe E/H.323 lorsqu'il établit des appels. S'il est mis à FALSE, il ne doit pas utiliser la signalisation d'appel définie dans

l'Annexe E/H.323. S'il n'est pas présent (ou s'il provient d'un portier de la version 2 ou d'une version antérieure), le point d'extrémité peut essayer les deux méthodes de compatibilité en amont décrites dans l'Annexe E/H.323.

maintainConnection – Si mis à TRUE, indique que le portier (en cas d'acheminement par portier) est en mesure de prendre en charge une connexion de signalisation lorsqu'aucun appel n'est en cours de signalisation sur la connexion.

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, données privées).

rejectReason – Motif du refus d'enregistrement.

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

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande à l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

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

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

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 strictement croissant unique du point de vue de 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 le point d'extrémité considéré pour la signalisation de l'appel, dont l'enregistrement doit être annulé.

endpointAlias – Cette valeur facultative est une liste d'adresses alias au moyen desquelles d'autres terminaux peuvent identifier le terminal considéré. Si ce champ facultatif n'est pas présent, tous les alias font l'objet d'une annulation d'enregistrement dans un seul message. L'adresse E.164, 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 l'adresse E.164.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple, 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 que le client a reçu dans la liste alternateGatekeeper du message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message URJ. Utilisé comme secours si le portier d'origine n'a pas répondu ou a refusé la demande.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 du point d'extrémité est annulé.

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, données privées).

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.10.3 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, données privées).

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande à l'un des portiers de

remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.11 Messages d'admission du terminal au portier

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

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

Le message ARQ comprend ce qui suit:

requestSeqNum – Numéro strictement croissant unique du point de vue de 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**, le point d'extrémité demande le modèle d'appel direct terminal à terminal. Si sa valeur est **gatekeeperRouted**, le point d'extrémité demande le modèle avec intervention du portier. Le portier n'est pas tenu de se conformer à cette demande.

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

destinationInfo – Séquence d'adresses alias pour la destination, il peut s'agir d'adresses E.164 ou d'identificateurs H323_ID. Lors de l'envoi du message ARQ en réponse à un appel, destinationInfo indique la destination de l'appel (point d'extrémité qui répond). Si au moins une adresse alias est enregistrée auprès d'un portier et si le message ARQ ne comporte pas deux adresses alias enregistrées auprès de personnes différentes, le portier doit reconnaître le message ARQ comme se rapportant à l'identité enregistrée. Dans le cas d'adresses alias incompatibles, il convient de refuser la demande d'admission en indiquant comme motif AliasesInconsistent. S'il ne fournit pas cette validation, le portier doit considérer la première adresse enregistrée comme étant celle de 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 alias pour le point d'extrémité source, il peut s'agir d'adresses E.164 ou d'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 – Débit exprimé en centaines de bits par seconde demandé pour l'appel bidirectionnel, par exemple un appel à 128 kbit/s sera 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 préfixes.

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

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple, 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 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 TRUE, indique que si le message ACF résultant contient les champs **destinationInfo**, **destExtraCallInfo** ou **remoteExtension**, le point d'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 ARQ et que **canMapAlias** est à FALSE, le portier devrait refuser le message ARQ.

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée 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 que le client a reçu dans la liste alternateGatekeeper du message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message ARJ. Utilisé comme secours si le portier d'origine n'a pas répondu ou a refusé la demande.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 – Un point d'extrémité peut utiliser cet élément pour indiquer sa capacité à réserver des ressources de transport.

willSupplyUUIE – Si sa valeur est TRUE, indique que le point d'extrémité fournira des informations de message Q.931 dans les messages IRR si le portier le demande.

La structure TransportQOS comprend ce qui suit:

endpointControlled – Le point d'extrémité appliquera son propre mécanisme de réservation.

gatekeeperControlled – Le portier effectuera la réservation de ressources au nom du point d'extrémité.

noControl – Aucune réservation de ressources n'est nécessaire.

NOTE – La présence simultanée des séquences **destinationInfo** et **destCallSignalAddress** n'est pas impérative, mais une de ces séquences au moins doit être présente sauf si le point d'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 E.164 doit être fournie si elle est disponible. Les meilleurs résultats seront 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 point d'extrémité à point d'extrémité est utilisé.

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

irrFrequency – Fréquence, en secondes, à laquelle le point d'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, le point d'extrémité n'envoie pas de message IRR pendant la phase active d'un appel et on s'attend à ce que le portier interroge le point d'extrémité.

nonStandardData – Transporte des informations non définies dans la présente Recommandation (par exemple, 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 WAN. Doit seulement contenir les adresses E.164 et ne doit pas contenir le numéro du canal initial.

destinationType – Spécifie le type du point d'extrémité de destination.

remoteExtensionAddress – Contient l'adresse alias d'un point d'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 destCallSignalAddress ou destinationInfo.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 au point d'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 TRUE.

uuiesRequested – Sur demande du portier, le point d'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 le point d'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 à la norme RFC 1766.

useAnnexECallSignalling – S'il est mis à TRUE, ce paramètre indique que le point d'extrémité qui reçoit ce message doit utiliser la signalisation d'appel exclusivement sur un canal de transport non fiable tel que défini dans l'Annexe E/H.323 pour la signalisation d'appel à destination de l'adresse de signalisation d'appel indiquée ci-dessus. S'il est mis à FALSE, il ne doit pas utiliser la signalisation d'appel définie dans l'Annexe E/H.323. S'il n'est pas présent (ou s'il provient d'un portier de la version 2 ou d'une version antérieure), le point d'extrémité peut essayer les deux méthodes de compatibilité en amont décrites dans l'Annexe E/H.323.

NOTE – Si la confirmation d'admission se rapporte à un appel entrant, ce champ doit être ignoré.

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 du motif 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 le motif du refus **rejectReason** est routeCallToSCN, le motif 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.

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

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande de l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.12 Demandes de modification de largeur de bande émises par le terminal à l'intention du portier

Le message BRQ est une demande adressée au portier de modification de la largeur de bande du réseau à commutation par paquets, le portier accède à la demande par un message BCF ou la refuse par un message BRJ.

Le portier peut, au moyen d'un message BRQ, demander à ce qu'un point d'extrémité augmente ou diminue la largeur de bande utilisée. S'il s'agit d'une augmentation, le point d'extrémité peut répondre au moyen d'un message BRJ ou BCF, s'il s'agit d'abaisser le débit, le point d'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 strictement croissant unique du point de vue de l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentifier – Identificateur de point d'extrémité qui a été attribué au terminal par un message RCF.

conferenceID – Identificateur de l'appel pour lequel la largeur de bande doit être modifiée.

callReferenceValue – Valeur CRV Q.931 pour cet appel; sa validité est uniquement locale. Elle est utilisée par un portier pour associer le message BRQ à un appel donné.

callType – Lorsqu'il utilise cette valeur, le portier peut essayer de déterminer la largeur de bande réellement utilisée.

bandWidth – Le "nouveau" nombre de pas de 100 bit/s demandé pour l'appel. 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 des préfixes.

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

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

gatekeeperIdentifier – Identificateur de portier que le client a reçu dans la liste alternateGatekeeper du message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message BRJ. Utilisé comme secours si le portier d'origine n'a pas répondu ou a refusé la demande.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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).

7.12.2 Message BCF (confirmation de largeur de bande)

Le message BCF comprend ce qui suit:

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

bandWidth – Valeur maximale autorisée à l'instant considéré par pas de 100 bit/s.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.12.3 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 – Motif pour lequel la modification a été refusée par le portier.

allowedBandWidth – Maximum autorisé à l'instant considéré par pas de 100 bit/s y compris l'attribution courante.

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

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande de l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.13 Messages de demande de localisation

Un message LRQ est une demande formulée 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 strictement croissant unique du point de vue de l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentifier – Identificateur de point d'extrémité qui a été attribué par le terminal au moyen d'un message RCF.

destinationInfo – Séquence d'adresses alias pour la destination, il peut s'agir d'adresses E.164 ou d'identificateurs H323_ID. Si au moins une adresse alias est enregistrée auprès d'un portier et si le message LRQ ne compte pas deux adresses alias enregistrées auprès de personnes différentes, le portier doit reconnaître le message LRQ comme se rapportant à l'identité enregistrée. Dans le cas d'adresses alias incompatibles, il convient de refuser la demande de localisation en indiquant comme motif AliasesInconsistent. S'il ne fournit pas cette validation, le portier doit considérer la première adresse enregistrée comme étant celle de destination.

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

replyAddress – Adresse de transport à laquelle doivent être envoyés les messages LCF/LRQ.

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 TRUE, indique que si le message LCF résultant contient les champs **destinationInfo**, **destExtraCallInfo** ou **remoteExtension**, le point d'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** est à FALSE, le portier devrait refuser le message LRQ.

gatekeeperIdentifier – Identificateur de portier que le client a reçu dans la liste alternateGatekeeper du message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message LRJ. Utilisé comme secours si le portier d'origine n'a pas répondu ou a refusé la demande.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.13.2 Message LCF (confirmation de localisation)

Le message LCF comprend ce qui suit:

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

callSignalAddress – Adresse de transport à laquelle doit être envoyée la signalisation d'appel Q.931; utilise l'accès connu fiable ou l'accès dynamique, mais peut être une adresse de point d'extrémité ou de portier selon le modèle d'appel utilisé.

rasAddress – Adresse utilisée par le point d'extrémité localisé pour les messages d'enregistrement, d'admissions et d'état.

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

destinationInfo – Séquence d'adresses alias pour la destination, il peut s'agir d'adresses E.164 ou d'identificateurs H323_ID.

destExtraCallInfo – Contient des adresses externes pour les appels multiples.

destinationType – Spécifie le type du point d'extrémité de destination.

remoteExtensionAddress – Contient l'adresse alias d'un point d'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 – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

supportsAnnexECallSignalling – Si mis à TRUE, indique que le point d'extrémité indiqué dans l'adresse callSignallingAddress est en mesure de recevoir la signalisation d'appel sur un canal de transport non fiable tel que défini dans l'Annexe E/H.323. Si mis à FALSE, indique que le point d'extrémité n'est pas en mesure de recevoir la signalisation d'appel définie dans l'Annexe E/H.323 et que par conséquent cette signalisation ne doit pas être utilisée. Si ce champ n'est pas présent, le point d'extrémité ou le portier appelant peut essayer les deux méthodes de compatibilité en amont décrites sans l'Annexe E/H.323.

7.13.3 Message LRJ (refus de localisation)

Le message LRJ comprend ce qui suit:

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

rejectReason – Il s'agit du motif pour lequel la demande de localisation a été refusée. Si le motif du refus **rejectReason** est routeCallToSCN, le motif 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.

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

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande de l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.14 Messages de désengagement

7.14.1 Message DRQ (demande de désengagement)

Lorsqu'il est envoyé à partir d'un point d'extrémité vers un portier, le message DRQ informe le portier qu'un point d'extrémité est abandonné. S'il est envoyé par un portier à un point d'extrémité, le message DRQ oblige d'abandonner un appel, une telle demande ne sera pas refusée. Le message DRQ n'est pas envoyé entre point d'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 le canal de signalisation d'appel.

Le message DRQ comprend ce qui suit:

requestSeqNum – Numéro strictement croissant unique du point de vue de l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

endpointIdentifier – Identificateur de point d'extrémité qui a été assigné au terminal par un message RCF.

conferenceID – Identificateur de l'appel pour lequel la largeur de bande doit être libérée.

callReferenceValue – Valeur CRV Q.931 pour cet appel; sa validité est uniquement locale. Elle est utilisée par un portier pour associer le message à un appel particulier.

disengageReason – Motif pour lequel la modification a été demandée par le portier ou le terminal.

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

callIdentifier – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

gatekeeperIdentifier – Identificateur de portier que le client a reçu dans la liste alternateGatekeeper du message RCF envoyé par le portier lorsqu'il s'est enregistré ou dans un précédent message DRJ. Utilisé comme secours si le portier d'origine n'a pas répondu ou a refusé la demande.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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).

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, données privées).

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.14.3 Message DRJ (refus de désengagement)

Le message DRJ est envoyé par le portier si l'enregistrement du point d'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 – Motif du refus de la demande.

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

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande de l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 Messages de demande d'état

Le message IRQ est envoyé par un portier à un terminal pour demander les informations d'état sous la forme d'un message IRR. Le message IRR peut également être envoyé par le terminal à un intervalle spécifié dans le message ACF sans réception d'un message IRQ émis par le portier. Il ne faut pas confondre ce message avec le message STATUS Q.931.

Lorsqu'un point d'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 (accusé de réception positif) ou un message INAK (accusé de réception 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 strictement croissant unique du point de vue de 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 devra être envoyé en réponse à une valeur nulle de CallReferenceValue. Si la valeur callReferenceValue est égale à 0, le point d'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, 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 global fixé par le point d'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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, le point d'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 le point d'extrémité doit signaler au portier.

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, données privées).

requestSeqNum – Doit contenir le numéro de séquence extrait du message IRR ou la valeur un pour un rapport non demandé à un portier de version 1. Il contient un numéro strictement croissant (que le portier doit renvoyer dans sa réponse) si **needResponse** est à TRUE.

endpointType – Fournit des informations sur le point d'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 alias associés au point d'extrémité.

perCallInfo – Informations associées à un appel donné:

- **nonStandardData** – Transporte des informations non définies dans la présente Recommandation (par exemple, données privées);
- **callReferenceValue** – Valeur CRV Q.931 de l'appel sur lequel porte la réponse;
- **conferenceID** – Identificateur univoque de la conférence;
- **originator** – Lorsqu'il est égal à TRUE, le point d'extrémité qui fait l'objet de la demande est l'appelant, s'il est égal à FALSE, le point d'extrémité est l'appelé;
- **audio –** Informations concernant le ou les canaux audio;
- **video** – Informations concernant le ou les canaux vidéo;
- **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 le point d'extrémité;
- **callIdentifier** – Identificateur d'appel unique sur le plan global fixé par le point d'extrémité expéditeur qui peut être utilisé pour associer la signalisation RAS à la signalisation Q.931 modifiée utilisée dans la présente Recommandation;
- **tokens** – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message;
- **cryptoTokens** – Jetons 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 le point d'extrémité a envoyé l'élément **h323pdu**; mis à FALSE pour indiquer que le point d'extrémité a reçu l'élément **h323pdu**.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 TRUE et que le portier a 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**.

7.15.3 Message IACK (accusé de réception 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, données privées).

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 (accusé de réception 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, données privées).

nakReason – Motif pour lequel le message IRR a fait l'objet d'un accusé de réception négatif.

altGKInfo – Informations facultatives sur des portiers de remplacement. Si cette information est communiquée, un point d'extrémité doit retransmettre la demande de l'un des portiers de remplacement figurant sur la liste. Si un portier de remplacement refuse la demande, le point d'extrémité doit accepter ce refus. Si un portier de remplacement ne répond pas, le point d'extrémité peut envoyer la demande à un autre portier de remplacement figurant sur la liste.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 standard

La structure de **NonStandardMessage** est la suivante:

requestSeqNum – Numéro strictement croissant unique du point de vue de l'expéditeur.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.17 Message incompris

Ce message est envoyé à chaque fois qu'un point d'extrémité H.323 reçoit un message RAS qu'il ne comprend pas.

requestSeqNum – Sera le **requestSeqNum** du message incompris, si celui-ci peut être décodé, autrement zéro.

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.18 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 strictement croissant unique du point de vue de l'expéditeur. Il doit être renvoyé par le destinataire dans tous les messages associés à ce message spécifique.

protocolIdentifier – Identifie le millésime du point d'extrémité expéditeur.

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

endpointIdentifier – Chaîne d'identité du point d'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 – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.18.2 Message RAC (confirmation de disponibilité de ressources)

Le message RAC comprend ce qui suit:

requestSeqNum – La valeur doit être la même que celle 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, données privées).

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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.

7.19 Temporisations RAS et message demande en cours (RIP, *request in progress*)

On trouvera au Tableau 18 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 18/H.225.0 – Valeurs de temporisations 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
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 supérieure) à laquelle une réponse ne peut pas être générée 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 générée. 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.

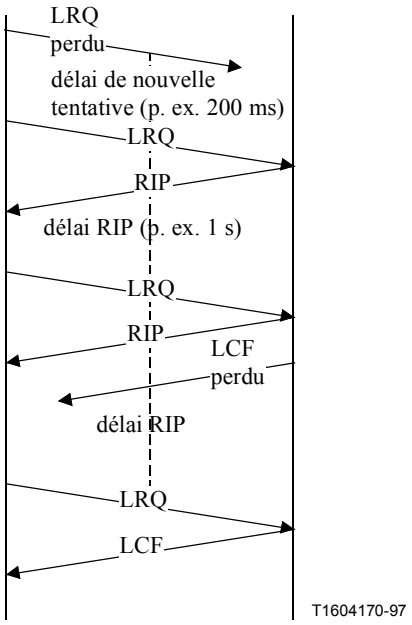


Figure 2/H.225.0 – Exemple d'utilisation du message RIP

Le message RIP comprend ce qui suit:

requestSeqNum – requestSeqNum de la demande en cours de traitement.

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

tokens – Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.

cryptoTokens – Jetons 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 un point d'extrémité peut faire une nouvelle tentative. Le point d'extrémité qui répond peut répondre avant l'expiration de cette période.

8 Mécanismes permettant de conserver la qualité de service (QS)

8.1 Méthode générale et hypothèses

La qualité de service du transport sur un réseau à commutation par paquets inclut certaines caractéristiques comme:

- le taux d'erreur;
- le taux de perte de paquets;
- les temps de propagation.

Toute signalisation associée à la qualité de service transport (par exemple une demande de réservation à l'intention d'un routeur) est faite par le terminal dès que possible, ou par le portier au nom du terminal. Le terminal peut souhaiter faire certaines réservations puisque le portier peut ne

pas se trouver logiquement proche du terminal, ou être en mesure de formuler des demandes relatives à la qualité de service au nom du terminal. Les moyens par lesquels le terminal ou le portier fait des réservations de qualité de service ou de largeur de bande n'entrent pas dans le domaine d'application de la présente Recommandation.

Les rapports d'émetteur et de récepteur du protocole RTCP sont des moyens par lesquels la qualité de service sera évaluée.

Il y a deux types d'encombrement associés au temps de propagation qui peuvent être mesurés:

- les augmentations à court terme des temps de propagation qui se traduiront par une diminution du débit de trame perceptible mais non gênante;
- une augmentation générale des temps de propagation due à un encombrement du réseau à commutation par paquets avec le temps de sorte qu'un mécanisme de rétroaction est utile.

Essentiellement, les rafales d'erreurs à court terme peuvent être compensées par une dissimulation des erreurs, et l'encombrement à plus long terme peut être compensé en réduisant la charge multimédia. L'hypothèse retenue est que tous les terminaux multimédias du réseau à commutation par paquets sont des terminaux H.323, et tous tenteront de diminuer l'utilisation du réseau en question en cas d'encombrement et ne tenteront pas de "voler" de la largeur de bande aux autres.

Les erreurs binaires sur un réseau à commutation par paquets sont en général corrigées dans les couches inférieures ou se traduisent par des pertes de paquets de sorte qu'elles ne seront 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 l'heure 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 interarrivée.

Les paramètres 2) et 3) sont utilisés pour calculer le nombre de paquets perdus depuis le précédent rapport de récepteur. Cette valeur peut être considérée comme une mesure à long terme de l'encombrement du réseau à commutation par paquets. Le A.6.3.4 donne un exemple de calcul. Si le taux de perte dépasse la valeur fixée par le constructeur, le terminal H.225.0 devra réduire les débits de transmission des médias côté réseau à commutation par paquets conformément aux procédures décrites au 8.4 ci-après. Si le paramètre 1) dépasse la valeur fixée par le constructeur, il peut être aussi souhaitable de prendre des mesures correctives.

Si l'intervalle entre les rapports de récepteur dépasse une valeur fixée par le constructeur, les terminaux H.323 devront utiliser le paramètre 1) comme indicateur d'un encombrement grave nécessitant une réduction du débit des médias côté réseau à commutation par paquets.

Le paramètre 4) doit être utilisé comme une indication d'encombrement imminent. Si la gigue interarrivée 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 Recommandation 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 du point d'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 interarrivée dans le récepteur distant. Ces méthodes peuvent être groupées en méthodes de réaction rapide à un problème à court terme tels la perte d'un paquet ou le retard d'un paquet et celles qui conviennent à une réponse à un problème à plus long terme tel un encombrement croissant sur le réseau à commutation par paquets. Il convient de noter que ces méthodes ne cherchent pas à maintenir la qualité de service actuelle, mais plutôt à obtenir une dégradation ordonnée du service. Les priorités ci-dessous seront observées de sorte que la dégradation affectera les médias dans l'ordre suivant, par ordre décroissant d'importance: vidéo, données, audio, commande.

Réactions à court terme:

- réduire le débit de trame pendant une courte période de temps. Cela peut se traduire dans la passerelle H.323 par l'envoi de trames de remplissage supplémentaires H.261 dans le sens réseau à commutation par paquets → WAN pour compenser le sous-débit de paquets;
- diminuer le débit de paquets en passant au mode facultatif dans lequel les flux audio/vidéo sont mélangés en un paquet (appelle un complément d'étude);

- le débit de paquets peut être réduit en utilisant la fragmentation de macroblocs du flux vidéo.

Réactions à plus long terme:

- la diminution du débit média (par exemple en passant de 384 kbit/s à 256 kbit/s). Cette réaction peut être déclenchée par une simple instruction donnée au codeur dans un terminal ou faire intervenir une fonction de réduction de débit dans la passerelle H.323. Ces modifications sont signalées par des commandes **FlowControl** H.245 ou par une signalisation de canal logique selon le cas;
- arrêter le média d'importance moindre (par exemple stopper le flux vidéo pour permettre un fort volume de trafic T.120);
- renvoi d'un signal d'occupation (occupation adaptative) au récepteur pour indiquer l'encombrement du réseau à commutation par paquets. Cette action peut être associée avec l'arrêt d'un média, ou même de tous les médias autres que l'accès transport de commande. L'occupation adaptative est signalée par une valeur de cause Q.931 dans le message **ReleaseComplete**.

Il convient de noter que la réaction à une gigue interarrivée 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 H.323. 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 utilisé pour appeler un poste téléphonique du RTGC, on se trouve dans le cas général où l'on 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.

La limitation de l'écho dû au transformateur différentiel (conversion 2 à 4 fils) relève de la responsabilité de la passerelle H.323.

ANNEXE A

Protocoles RTP/RTCP

On notera que toutes les références de la présente annexe, qui sont des références à une bibliographie, ne sont données qu'à titre d'information, à l'exception de la référence [A-10] (norme ISO/CEI 10646-1), qui apparaît également dans le paragraphe références de la présente Recommandation. Dans certains cas, il apparaîtra une référence à l'Appendice I; ces références ne sont données qu'à titre d'information. Tous les détails nécessaires à l'implémentation de la Recommandation H.323 et de la présente Recommandation figurent dans la présente annexe et dans d'autres annexes et Recommandations ou Normes internationales publiées par l'UIT-T ou l'ISO.

On notera que la présente annexe ne contient pas la spécification primaire et complète des protocoles RTP/RTCP; on trouvera la référence à ces informations à l'Appendice I. La présente annexe ne se rapporte qu'à la Recommandation H.323 et à la présente Recommandation.

On notera également que la terminologie utilisée dans la présente annexe et celle qui est utilisée dans la Recommandation H.323 et la présente Recommandation présentent les différences indiquées dans le Tableau A.1.

Tableau A.1/H.225.0 – Correspondance terminologique

Terme utilisé dans H.323 et H.225.0	Terme utilisé dans l'Annexe A (Protocoles RTP/RTCP)
Flux multimédia	Données
Adresse de transport	Adresse de transport
Adresse de réseau à commutation par paquets	Adresse de réseau
Identificateur TSAP	Accès
Annexe A	Spécification ou document

On notera en outre que les "traducteurs" et les "mélangeurs" ne font pas partie du système H.323. Mais les extrémités de système H.323 – passerelles et unités MCU par exemple – ayant quelques-unes des caractéristiques des traducteurs et des mélangeurs, ce texte a été retenu pour servir de guide aux personnes chargées de l'implémentation. Toutefois, les traducteurs et les mélangeurs n'étant pas pris en charge dans le système H.323, ces sous-paragraphes ne doivent pas être considérés comme normatifs.

Enfin, il est rappelé aux personnes implémentant le protocole RTP qu'elles doivent impérativement se conformer aux dispositions de la présente Recommandation, y compris des Annexes A, B et C, qui donnent des précisions sur les systèmes H.323/H.225.0. Dans tous les cas, le texte de la présente Recommandation l'emportera sur le texte des Annexes A, B ou C.

A.1 Introduction

La présente spécification spécifie le protocole de transport en temps réel (RTP, *real-time transport protocol*), qui permet d'assurer des services de remise de bout en bout pour les données ayant des caractéristiques de temps réel, comme les données audio et vidéo interactives. Ces services comprennent l'identification du type de charge utile, la numérotation des séquences, l'horodatage et le contrôle de la remise. Les applications emploient généralement le protocole RTP au-dessus du protocole UDP pour utiliser ses services de multiplexage et de contrôle de somme; les deux protocoles contribuent à la fonctionnalité du protocole de transport. Le protocole RTP peut toutefois être utilisé avec d'autres protocoles de transport ou de réseau sous-jacents (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport). Le protocole RTP prend en charge le transfert de données vers plusieurs destinations au moyen de la distribution en mode multidiffusion lorsqu'elle est fournie par le réseau sous-jacent.

On notera que le protocole RTP ne prévoit aucun mécanisme fournissant une garantie de remise en temps utile ou d'autres garanties de qualité de service; ce sont les services de couche inférieure qui doivent fournir ce mécanisme. Le protocole RTP ne garantit pas la remise, n'empêche pas les remises désordonnées, il ne suppose pas non plus que le réseau sous-jacent est fiable et effectue une remise des paquets en séquence. Les numéros de séquence figurant dans l'en-tête des paquets RTP permettent au récepteur de reconstituer la séquence de paquets de l'émetteur; ces numéros peuvent également servir à déterminer l'emplacement exact d'un paquet, dans un décodage vidéo par exemple, sans avoir à décoder les paquets de la séquence.

Le protocole RTP est d'abord conçu pour répondre aux besoins des conférences multimédia multi-participants, mais il peut servir pour d'autres applications: enregistrement de données continues, simulation répartie interactive, badge actif, applications de commande et de mesure.

La présente Recommandation définit le protocole RTP, qui est constitué de deux parties étroitement liées:

- le protocole de transport en temps réel (RTP), pour transporter des données ayant des caractéristiques de temps réel;
- le protocole de commande RTP (RTCP, *real-time transport control protocol*), pour contrôler la qualité de service et pour acheminer des renseignements sur les participants au cours d'une session. Cette deuxième finalité du protocole RTCP peut suffire pour les sessions "à commande souple", c'est-à-dire pour lesquelles il n'existe pas de procédures explicites de contrôle et de raccordement des membres, mais le but n'est pas nécessairement de prendre en charge toutes les prescriptions de communication de commande d'une application. Cette fonctionnalité peut être entièrement ou partiellement incluse dans un protocole de commande de session distinct, mais cela sort du cadre de la présente Recommandation.

Le protocole RTP représente un nouveau style de protocole qui suit les principes de tramage au niveau application et de traitement de couche intégré proposés par Clark et Tennenhouse [A-1]. Cela signifie que le protocole RTP doit être malléable pour pouvoir fournir les informations requises par une application donnée et qu'il sera souvent intégré dans le traitement au niveau application et non pas implémenté sous forme de couche distincte. Le protocole RTP est un cadre de protocole qui est délibérément incomplet. La présente Recommandation spécifie les fonctions qui seront communes pour toutes les applications utilisant le protocole RTP. A la différence des protocoles classiques pour lesquels on peut accepter des fonctions supplémentaires qui rendent le protocole plus général ou qui ajoutent un mécanisme optionnel nécessitant une analyse, le protocole RTP peut être adapté par des modifications ou des ajouts aux en-têtes, lorsque c'est utile. Des exemples sont donnés au A.5.3, Modifications de l'en-tête RTP propres au profil.

Pour spécifier entièrement le protocole RTP pour une application donnée, il faudra donc compléter la présente Recommandation par un ou plusieurs documents associés (voir les Annexes B et C):

- un document de spécification de profil, qui définit un ensemble de codes de type de charge utile et leur mappage avec les formats de charge utile (codages multimédia par exemple). Un profil peut également définir des extensions ou des modifications du protocole RTP qui sont propres à une classe d'applications. Une application ne fonctionnera généralement que conformément à un seul profil. On trouvera à l'Annexe B un profil pour les données audio et vidéo;
- des documents de spécification de format de charge utile, qui définissent la manière dont une charge utile donnée, comme un codage audio ou vidéo, doit être transportée dans le protocole RTP. Voir l'Annexe C.

Plusieurs applications RTP, à la fois expérimentales et commerciales, ont déjà été implémentées à partir des premières spécifications. Ces applications comprennent des outils audio et vidéo ainsi que des outils de diagnostic comme des contrôleurs de trafic. Les utilisateurs de ces outils se comptent par milliers. Toutefois, le réseau Internet actuel ne peut pas encore prendre en charge toute la demande potentielle pour des services en temps réel. Les services à large bande utilisant le protocole RTP, comme les services vidéo, peuvent dégrader gravement la qualité de service des autres services de réseau. Les personnes chargées de l'implémentation doivent donc prendre les précautions nécessaires pour limiter l'utilisation accidentelle de largeur de bande. La documentation sur les applications doit indiquer clairement les limites et l'influence possible de l'exploitation de services en temps réel à large bande sur l'exploitation des services Internet et des services d'autres réseaux.

A.2 Scénarios d'utilisation du protocole RTP

Les sous-paragraphes suivants décrivent certaines caractéristiques d'utilisation du protocole RTP. Les exemples ont été choisis de façon à illustrer l'exploitation de base des applications utilisant le protocole RTP, mais ils ne visent pas à limiter les possibilités d'utilisation du protocole RTP. Dans

ces exemples, le protocole RTP, utilisé au-dessus des protocoles IP et UDP, suit les conventions établies dans le profil pour les données audio et vidéo spécifié dans l'Annexe B.

A.2.1 Audioconférence simple en mode multidiffusion

Un groupe de travail de l'IETF se réunit pour discuter du projet de protocole sous sa dernière forme, qui utilise les services de multidiffusion IP d'Internet pour les communications vocales. Par l'intermédiaire d'un certain mécanisme d'attribution, la présidence du groupe de travail obtient une adresse de groupe de multidiffusion et deux accès, l'un pour les données audio et l'autre pour les paquets (RTCP) de commande. Ces informations d'adresse et d'accès sont transmises aux participants voulus. Si l'on souhaite que les informations soient confidentielles, les paquets de données et de commande peuvent être chiffrés selon les spécifications de la Recommandation H.323. Par l'intermédiaire de l'application d'audioconférence, chaque participant à l'audioconférence envoie des données audio par petites bribes de 20 ms de durée par exemple. Chaque bribe de données audio est précédée d'un en-tête RTP; cet en-tête et les données figurent à leur tour dans un paquet UDP. L'en-tête RTP indique le type de codage audio (MIC, MICDA ou LPC) de chaque paquet; les émetteurs peuvent ainsi modifier le codage au cours d'une conférence, par exemple pour pouvoir communiquer avec un nouveau participant qui est raccordé via une liaison à faible largeur de bande ou pour réagir à des indications d'encombrement de réseau.

Comme pour les autres réseaux par paquets, il arrive à Internet de perdre des paquets, de les redemander et de les retransmettre avec des retards variables. Pour faire face à ces dégradations, l'en-tête RTP contient des informations de rythme et un numéro de séquence qui permettent aux récepteurs de reconstituer le rythme généré par la source de sorte que, dans cet exemple, le locuteur produit de façon continue des bribes de données audio de 20 ms. Le rythme est reconstitué séparément pour chaque source de paquets RTP de la conférence. Le numéro de séquence peut également servir au récepteur à évaluer le nombre de paquets perdus.

Etant donné que les membres du groupe de travail peuvent arriver et partir au cours de la conférence, il est utile de connaître les participants à un moment donné et de savoir avec quelle qualité ils reçoivent les données audio. Pour cela, chaque instance de l'application audio dans la conférence diffuse périodiquement un rapport de réception ainsi que le nom de l'utilisateur se trouvant à l'accès (de commande) RTCP. Le rapport de réception, qui indique avec quelle qualité le locuteur du moment est reçu, peut servir à commander des codages adaptatifs. Le nom de l'utilisateur peut être accompagné d'autres informations d'identification sous réserve de satisfaire aux limites de largeur de bande de commande. Un site envoie le paquet RTCP BYE (voir A.6.5, BYE: paquet RTCP au revoir) lorsqu'il quitte la conférence.

A.2.2 Conférence audio et vidéo

Si les médias audio et vidéo sont tous deux utilisés dans une conférence, ils sont transmis dans des sessions RTP séparées; pour chacun de ces supports, les paquets RTCP sont transmis au moyen de deux couples d'accès UDP différents ou de deux adresses de multidiffusion différentes. Il n'existe pas de couplage direct au niveau RTP entre les sessions audio et vidéo, mis à part le fait qu'un utilisateur participant aux deux sessions doit utiliser le même nom (canonique) distinctif dans les paquets RTCP afin de pouvoir associer ces deux sessions.

L'une des raisons de cette séparation est de permettre aux participants à la conférence qui le souhaitent de ne recevoir qu'un seul support d'information. On trouvera plus de détails au A.5.2, Sessions RTP avec multiplexage des données. Malgré la séparation, il est possible de synchroniser les données audio et vidéo d'une source grâce aux informations de rythme figurant dans les paquets RTCP pour les deux sessions.

A.2.3 Mélangeurs et traducteurs

Jusqu'ici, nous avons supposé que tous les sites souhaitent recevoir les données de média dans le même format. Toutefois, cette hypothèse ne convient pas dans tous les cas. Considérons le cas où les

participants d'une zone sont raccordés via une liaison à faible vitesse à la majorité des participants à la conférence qui bénéficient d'un accès au réseau à vitesse élevée. Au lieu de forcer tout le monde à utiliser une largeur de bande moins élevée et un codage audio de qualité réduite, un relais au niveau RTP, appelé mélangeur, peut être placé près de la zone à faible largeur de bande. Ce mélangeur resynchronise les paquets audio entrants pour reconstituer l'espacement constant de 20 ms généré par l'émetteur, mélange ces flux audio reconstitués pour ne former qu'un seul flux, traduit le codage audio en un codage à largeur de bande plus faible et transmet le flux de paquets à largeur de bande plus faible sur la liaison à faible vitesse. Ces paquets peuvent être monodiffusés vers un seul destinataire ou multidiffusés à une adresse différente vers plusieurs destinataires. L'en-tête RTP doit permettre aux mélangeurs d'identifier les sources qui ont contribué à un paquet mélangé de façon à pouvoir fournir une indication correcte de locuteur aux récepteurs.

Certains participants prévus à l'audioconférence peuvent être raccordés par l'intermédiaire de liaisons à large bande mais il est possible qu'on ne puisse pas les joindre directement via la multidiffusion IP. Ils peuvent par exemple se trouver derrière une barrière au niveau application qui ne laisse passer aucun paquet IP. Pour ces sites, on n'utilise pas de mélangeur, mais on a recours à un autre type de relais au niveau RTP, appelé traducteur. Deux traducteurs sont installés, un de chaque côté de la barrière; celui qui se trouve à l'extérieur envoie tous les paquets de multidiffusion reçus sur une connexion sûre en forme d'entonnoir vers le traducteur se trouvant à l'intérieur. Ce traducteur-ci renvoie ces paquets sous forme de paquets de multidiffusion à un groupe de multidiffusion restreint au réseau interne du site.

Les mélangeurs et les traducteurs peuvent être conçus pour diverses applications. Citons par exemple le cas d'un mélangeur vidéo qui met les images d'une personne donnée dans différents flux vidéo après leur avoir fait subir un changement d'échelle et qui compose ces flux dans un seul flux vidéo pour simuler une scène de groupe. Parmi les exemples de traduction, citons le raccordement d'un groupe de serveurs n'ayant que les protocoles IP/UDP à un groupe de serveurs ne comprenant que le protocole ST-II, ou encore la traduction de flux vidéo provenant de sources distinctes en codage paquet par paquet sans resynchronisation ni mélange. On trouvera des détails sur le fonctionnement des mélangeurs et des traducteurs au A.7, Traducteurs et mélangeurs RTP.

A.3 Définitions

A.3.1 charge utile RTP: données transportées par le protocole RTP dans un paquet, par exemple des échantillons audio ou des données vidéo compressées. Le format de charge utile et son interprétation sortent du cadre de la présente Recommandation.

A.3.2 paquet RTP: paquet de données comprenant: l'en-tête RTP fixe, une liste de sources contributives éventuellement vide (voir plus loin) et des données de charge utile. Certains protocoles sous-jacents peuvent nécessiter une encapsulation du paquet RTP à définir. Un paquet du protocole sous-jacent contient généralement un seul paquet RTP, mais il peut en contenir plusieurs si la méthode d'encapsulation le permet (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport).

A.3.3 paquet RTCP: paquet de commande comportant une partie d'en-tête fixe semblable à celle des paquets de données RTP, suivie d'éléments structurés qui varient selon le type de paquet RTCP. Les formats sont définis au A.6, protocole de commande RTP – protocole RTCP. Généralement, plusieurs paquets RTCP sont envoyés ensemble sous la forme d'un paquet RTCP composé dans un unique paquet du protocole sous-jacent; cela est permis par le champ de longueur se trouvant dans l'en-tête fixe de chaque paquet RTCP.

A.3.4 accès: "abstraction que les protocoles de transport utilisent pour faire la distinction entre plusieurs destinations à l'intérieur d'un serveur donné. Les protocoles TCP/IP identifient les accès au moyen de petits entiers positifs". (NdT: traduction d'une citation de la référence [A-2].) Les sélecteurs de transport (TSEL, *transport selectors*) utilisés par la couche Transport OSI sont équivalents à des accès. Le protocole RTP s'appuie sur le protocole de couche inférieure pour fournir

un mécanisme (des accès par exemple) permettant de multiplexer les paquets RTP et RTCP d'une session.

A.3.5 adresse de transport: combinaison d'une adresse de réseau et d'un accès qui identifie une extrémité au niveau transport, par exemple une adresse IP et un accès UDP. Les paquets sont transmis d'une adresse de transport de source vers une adresse de transport de destination.

A.3.6 session RTP: association de participants communicant grâce au protocole RTP. Pour chaque participant, la session est définie par un couple d'adresses de transport de destination (une adresse de réseau plus un couple d'accès pour les protocoles RTP et RTCP). Le couple d'adresses de transport de destination peut être commun à tous les participants, comme dans le cas de la multidiffusion IP, ou chaque participant peut avoir un couple d'adresses différent, comme dans le cas des accès et des adresses de réseau de monodiffusion individuels. Dans une session multimédia, chaque média est transporté dans une session RTP distincte avec ses propres paquets RTCP. On peut faire la distinction entre les diverses sessions RTP grâce à des couples de numéros d'accès différents ou à des adresses de multidiffusion différentes.

A.3.7 source de synchronisation (SSRC, *synchronization source*): source d'un flux de paquets RTP, caractérisée par un identificateur SSRC numérique à 32 bits figurant dans l'en-tête RTP de sorte qu'il ne dépende pas de l'adresse de réseau. Tous les paquets provenant d'une source de synchronisation appartiennent au même espace de rythme et de numéro de séquence, les récepteurs peuvent donc regrouper les paquets par source de synchronisation avant de les lire. Parmi les exemples de sources de synchronisation, citons un mélangeur RTP (voir plus loin) ou encore un émetteur qui envoie un flux de paquets découlant d'une source de signaux comme un microphone ou une caméra. Une source de synchronisation peut modifier son format de données, codage audio par exemple, au cours du temps. L'identificateur SSRC est une valeur choisie au hasard qui est globalement unique dans une session RTP donnée (voir A.8, Attribution et utilisation des identificateurs SSRC). Les participants n'ont pas besoin d'utiliser le même identificateur SSRC pour toutes les sessions RTP d'une session multimédia; le lien entre les différents identificateurs SSRC est assuré par le protocole RTCP (voir A.6.4.1, CNAME: élément SDES identificateur d'extrémité canonique). Si un participant génère plusieurs flux dans une même session RTP, à partir de diverses caméras vidéo par exemple, chaque flux doit être identifié comme provenant d'une source SSRC différente.

A.3.8 source contributive (CSRC, *contributing source*): source d'un flux de paquets RTP qui contribue au flux combiné produit par un mélangeur RTP (voir plus loin). Le mélangeur insère la liste des identificateurs SSRC des sources qui contribuent à la génération d'un paquet donné, dans l'en-tête RTP de ce paquet. Cette liste s'appelle la liste CSRC. Comme exemple d'application, citons une audioconférence dans laquelle un mélangeur indique tous les locuteurs dont les paroles ont été combinées pour produire le paquet sortant, ce qui permet au récepteur d'indiquer le locuteur du moment, même si tous les paquets audio contiennent le même identificateur SSRC (celui du mélangeur).

A.3.9 système d'extrémité: application qui génère le contenu à envoyer dans des paquets RTP ou qui consomme le contenu des paquets RTP reçus. Un système d'extrémité joue généralement le rôle d'une seule source de synchronisation dans une session RTP donnée, mais il peut aussi jouer le rôle de plusieurs sources.

A.3.10 mélangeur: système intermédiaire qui reçoit des paquets RTP en provenance d'une ou plusieurs sources, qui modifie éventuellement le format des données, qui combine les paquets d'une certaine manière et qui transmet ensuite un nouveau paquet RTP. Etant donné que le rythme parmi les différentes sources d'entrée ne sera généralement pas synchronisé, le mélangeur procédera à des ajustements de rythme entre les flux et générera son propre rythme pour le flux combiné. On considérera alors que la source de synchronisation de tous les paquets de données provenant d'un mélangeur est le mélangeur.

A.3.11 traducteur: système intermédiaire qui transmet des paquets RTP avec leur identificateur de source de synchronisation intact. Exemples de traducteurs: dispositifs convertissant les codages sans effectuer de mélange, dispositifs de passage de la multidiffusion à la monodiffusion, filtres au niveau application dans les barrières.

A.3.12 contrôleur: application qui reçoit les paquets RTCP envoyés par les participants à une session RTP, en particulier les rapports de perception, et qui évalue la qualité de service courante aux fins de contrôle de la distribution, de diagnostic des défauts et de statistiques à long terme. La fonction de contrôle devrait être intégrée dans la ou les applications participant à la session, mais elle pourra aussi constituer une application distincte qui ne participe pas à la session et qui n'envoie pas de paquets de données RTP et n'en reçoit pas. On parle alors de contrôleurs tiers.

A.3.13 moyens non RTP: protocoles et mécanismes qui peuvent être nécessaires en plus du protocole RTP pour fournir un service facile à utiliser. Pour les conférences multimédia en particulier, une application de commande de conférence peut distribuer des adresses de multidiffusion et des clés de chiffrement, négocier l'algorithme de chiffrement à utiliser et définir les mappages dynamiques entre les valeurs de type de charge utile RTP et les formats de charge utile pour lesquels aucune valeur de type de charge utile n'est prédéfinie. Pour les applications simples, on peut également utiliser le courrier électronique ou une base de données de conférence. La spécification de ces protocoles et mécanismes sort du cadre de la présente Recommandation.

A.4 Ordre des octets, alignement et format temporel

Tous les champs d'entiers sont ordonnés dans l'ordre réseau des octets, c'est-à-dire l'octet le plus significatif en premier. Cet ordre des octets est appelé en anglais "big-endian". L'ordre de transmission est décrit en détails dans la référence [A-3]. Sauf indication contraire, les constantes numériques sont décimales (base 10).

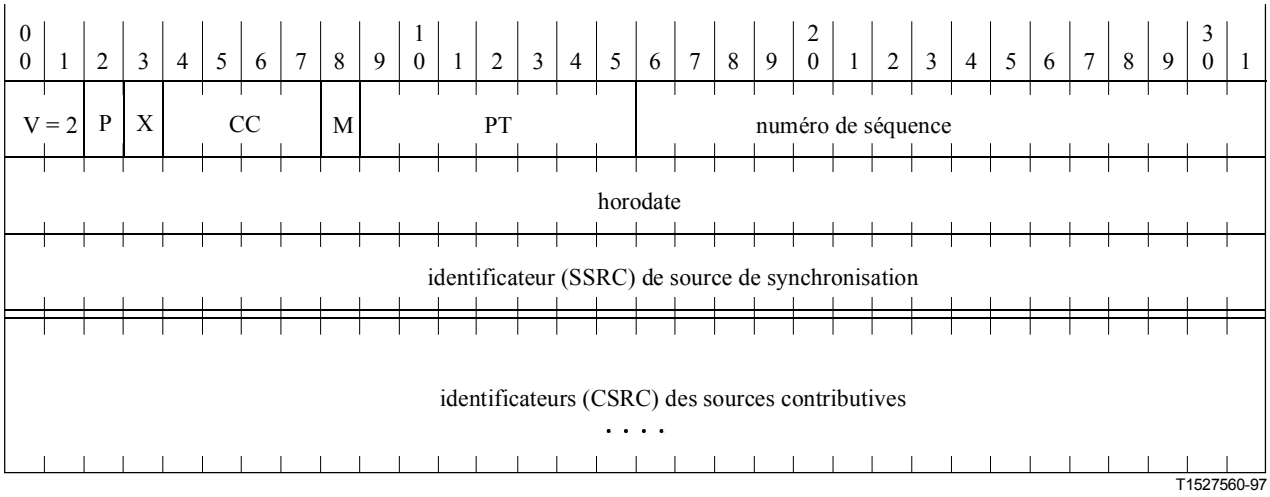
Toutes les données d'en-tête sont alignées sur leur longueur naturelle, c'est-à-dire que les champs à 16 bits sont alignés sur des décalages pairs, les champs à 32 bits sont alignés sur des décalages divisibles par quatre, etc. Les octets désignés comme étant des octets de remplissage ont la valeur zéro.

L'heure lue sur une horloge murale (heure absolue) est représentée à l'aide du format d'horodate du protocole temporel de réseau (NTP, *network time protocol*), qui est en secondes par rapport à 0h UTC du 1er janvier 1900 [A-4]. L'horodate NTP à résolution complète est un nombre à virgule fixe non signé sur 64 bits, la partie entière se trouvant dans les 32 premiers bits et la partie décimale dans les 32 derniers bits. Dans certains champs où il convient d'utiliser une représentation plus compacte, seuls les 32 bits du milieu sont utilisés, à savoir les 16 bits inférieurs de la partie entière et les 16 bits supérieurs de la partie décimale. Les 16 bits supérieurs de la partie entière doivent être déterminés de façon indépendante.

A.5 Protocole de transfert de données RTP

A.5.1 Champs de l'en-tête fixe RTP

L'en-tête RTP a le format suivant:



Les douze premiers octets figurent dans chaque paquet RTP, alors que la liste des identificateurs CSRC n'est présente que lorsqu'elle est insérée par un mélangeur. Les champs ont la signification suivante:

version (V): 2 bits. Ce champ identifie la version du protocole RTP. La version définie par la présente spécification correspond à 2. (La valeur 1 est utilisée par le projet de première version du protocole RTP et la valeur 0 est utilisée par le protocole qui a été implémenté au départ dans l'outil audio "vat".)

remplissage (P): 1 bit. Si le bit de remplissage est positionné, le paquet contient un ou plusieurs octets supplémentaires de remplissage à la fin qui ne font pas partie de la charge utile. Le dernier octet de remplissage indique le nombre d'octets de remplissage qu'il faut ignorer. Le remplissage peut être nécessaire pour certains algorithmes de chiffrement avec des tailles de bloc fixes ou pour le transport de plusieurs paquets RTP dans une unité de données protocolaires de couche inférieure.

extension (X): 1 bit. Si le bit d'extension est positionné, l'en-tête fixe est suivie d'une extension d'en-tête et d'une seule, avec le format défini au A.5.3, Modifications de l'en-tête RTP propres au profil.

compte CSRC (CC): 4 bits. Le compte CSRC contient le nombre d'identificateurs CSRC qui suivent l'en-tête fixe.

marqueur (M): 1 bit. L'interprétation du marqueur est définie dans les profils. Il sert à repérer des événements significatifs, comme les frontières de trame, dans le flux de paquets. Chaque profil peut définir des bits de marqueur supplémentaires ou spécifier qu'il n'y a pas de bit de marqueur en modifiant le nombre de bits dans le champ de type de charge utile (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

type de charge utile (PT): 7 bits. Ce champ identifie le format de la charge utile RTP et détermine la façon dont l'application doit l'interpréter. Chaque profil spécifie un mappage statique par défaut entre les codes de type de charge utile et les formats de charge utile. D'autres codes de type de charge utile peuvent être définis dynamiquement grâce aux moyens non RTP (voir A.3, Définitions). L'Annexe B spécifie un ensemble initial de mappages par défaut pour les données audio et vidéo. A un instant donné, un émetteur RTP transmet un type de charge utile RTP et un seul; ce champ n'est pas prévu pour le multiplexage de divers flux de média (voir A.5.2, Sessions RTP avec multiplexage des données).

numéro de séquence: 16 bits. Le numéro de séquence, incrémenté d'une unité à chaque paquet de données RTP envoyé, peut servir au récepteur à détecter les pertes de paquets et à restaurer la séquence de paquets. La valeur initiale du numéro de séquence est aléatoire (imprévisible) pour rendre plus difficiles les attaques en clair connues contre le chiffrement, même si la source n'effectue pas de chiffrement, car les paquets peuvent passer dans un traducteur qui effectue des chiffrements.

Les techniques permettant de choisir des nombres imprévisibles sont abordées dans la référence [A-5].

horodate: 32 bits. L'horodate correspond à l'instant d'échantillonnage du premier octet dans le paquet de données RTP. Cet instant doit être lu sur une horloge indiquant des dates croissantes et espacées linéairement dans le temps pour permettre la synchronisation et les calculs de gigue (voir A.6.3.1, SR: paquet RTCP rapport de l'émetteur). La résolution de l'horloge doit être suffisante pour la précision de synchronisation souhaitée et pour la mesure de la gigue d'arrivée des paquets (un top d'horloge par trame vidéo n'est généralement pas suffisant). La fréquence de l'horloge, fonction du format des données de la charge utile, est spécifiée statiquement dans le profil ou dans la spécification de format de charge utile qui définit le format, ou peut être spécifiée dynamiquement pour les formats de charge utile définis par des moyens non-RTP. Si les paquets RTP sont générés périodiquement, il faut utiliser l'instant d'échantillonnage nominal déterminé à partir de l'horloge d'échantillonnage, et non pas lire l'horloge du système. Pour des données audio à débit fixe par exemple, il devrait y avoir un nouveau top d'horloge à chaque période d'échantillonnage. Si une application audio lit des blocs couvrant 160 périodes d'échantillonnage et provenant du dispositif d'entrée, l'estampille horaire devra être incrémentée de 160 à chaque bloc, que le bloc soit transmis dans un paquet ou qu'il soit éliminé en tant que silence.

La valeur initiale de l'horodate est aléatoire, comme pour le numéro de séquence. Plusieurs paquets RTP consécutifs peuvent avoir des horodates égales s'ils sont générés (logiquement) en même temps, par exemple s'ils appartiennent à la même trame vidéo. L'horodate d'un paquet RTP peut être aussi bien postérieure qu'antérieure à l'horodate du paquet RTP consécutif si les données ne sont pas transmises dans leur ordre d'échantillonnage, comme c'est le cas pour les trames vidéo interpolées MPEG. (Mais le numéro de séquence des paquets transmis sera toujours monotone croissante.)

SSRC: 32 bits. Le champ SSRC identifie la source de synchronisation. Cet identificateur est choisi au hasard, avec comme intention que deux sources de synchronisation quelconques à l'intérieur d'une même session RTP aient des identificateurs SSRC différents. Le sous-paragraphe A.8.2 présente un exemple d'algorithme générant un identificateur aléatoire. La probabilité pour que plusieurs sources choisissent le même identificateur est faible, mais toutes les implémentations du protocole RTP doivent prévoir la détection et la résolution des collisions. Le sous-paragraphe A.8, Attribution et utilisation des identificateurs SSRC, donne la probabilité de collision et décrit un mécanisme permettant de résoudre les collisions et de détecter les boucles au niveau RTP sur la base de l'unicité des identificateurs SSRC. Si une source change d'adresse de transport, elle doit aussi choisir un nouvel identificateur SSRC pour éviter d'être interprétée comme une source bouclée.

Liste CSRC: 0 à 15 éléments, 32 bits chacun. La liste CSRC identifie les sources qui ont contribué à la charge utile du paquet. Le nombre d'identificateurs est donné par le champ CC. Si le nombre de sources contributives est supérieur à 15, seules 15 peuvent être identifiées. Les mélangeurs insèrent des identificateurs CSRC, en utilisant les identificateurs SSRC des sources contributives. Pour les paquets audio par exemple, on énumère les identificateurs SSRC de toutes les sources qui ont été mélangées ensemble pour créer un paquet, ce qui permet de fournir une indication correcte de locuteur au récepteur.

A.5.2 Sessions RTP avec multiplexage des données

Pour un fonctionnement efficace du protocole, le nombre de points de multiplexage doit être réduit au minimum, comme le décrit le principe de traitement de couche intégré [A-1]. Dans le protocole RTP, le multiplexage est permis par l'adresse de transport de destination (adresse de réseau et numéro d'accès) qui définit une session RTP. Par exemple, dans une téléconférence composée de média audio et vidéo codés séparément, chaque médium doit être transporté dans une session RTP distincte avec sa propre adresse de transport de destination. Il n'est pas prévu de transporter les données audio et vidéo dans une seule session RTP et de les démultiplexer en fonction du type de

charge utile ou des champs SSRC. L'entrelacement de paquets avec des types de charge utile différents mais utilisant le même identificateur SSRC serait à l'origine de plusieurs problèmes:

1) si l'un des types de charge utile changeait au cours d'une session, il n'y aurait pas de moyen général qui permettrait de déterminer l'ancienne valeur que la nouvelle valeur a remplacée;
2) par définition, un identificateur SSRC désigne un unique espace de rythme et de numéro de séquence. Si divers types de charge utile sont entrelacés, il faudrait alors différents espaces de rythme si les fréquences d'horloge diffèrent selon les médias et il faudrait différents espaces de numéro de séquence pour savoir quel type de charge utile a subi une perte de paquet;
3) les rapports d'émetteur et de récepteur RTCP (voir A.6.3, Rapports d'émetteur et de récepteur) ne peuvent décrire qu'un seul espace de rythme et de numéro de séquence par identificateur SSRC et ne comportent pas de champ de type de charge utile;
4) un mélangeur RTP n'est pas en mesure de combiner des flux entrelacés de média incompatibles en un seul flux;
5) le transport de plusieurs médias dans une même session RTP empêche: l'utilisation de conduits de réseau différents ou d'attributions de ressources de réseau différentes lorsque c'est possible; la réception d'une partie seulement des médias lorsqu'on le souhaite, par exemple la réception des données audio uniquement dans le cas où les données vidéo dépasseraient la largeur de bande disponible; et l'implémentation de récepteurs qui utilisent des processus distincts pour les différents médias, alors que le recours à des sessions RTP distinctes permet des implémentations avec au choix un seul ou plusieurs processus.

L'utilisation d'un identificateur SSRC différent pour chaque support d'information et l'envoi de tous les média dans la même session RTP permettraient d'éviter les trois premiers problèmes mais pas les deux derniers.

A.5.3 Modifications de l'en-tête RTP propres au profil

L'en-tête de paquet de données RTP existant passe pour être complet pour l'ensemble des fonctions communes requises pour toutes les classes d'application que le protocole RTP est susceptible de prendre en charge. Toutefois, conformément au principe de conception ALF, l'en-tête peut être adapté par des modifications ou des ajouts qui sont définis dans une spécification de profil et qui n'entravent pas le fonctionnement des outils d'enregistrement et de contrôle indépendants du profil:

- le bit de marqueur et le champ de type de charge utile contiennent des informations propres au profil, mais ces champs ont été affectés dans l'en-tête fixe car ils seront vraisemblablement nécessaires à de nombreuses applications et auraient nécessité sinon un autre mot de 32 bits uniquement pour les contenir. Les profils peuvent redéfinir l'octet comportant ces champs pour répondre à diverses prescriptions, avec par exemple un nombre plus grand ou plus petit de marqueurs. Si l'octet comporte des marqueurs, l'un de ceux-ci occupera la position du bit le plus significatif, étant donné qu'il est possible que les contrôleurs indépendants du profil puissent établir une corrélation entre les pertes de paquets et le bit de marqueur;
- les informations supplémentaires qui sont nécessaires pour un format de charge utile donné, comme un codage vidéo par exemple, doivent figurer dans la section de charge utile du paquet. Ces informations pourraient se trouver dans un en-tête qui est toujours présent au début de la section de charge utile, ou pourraient être indiquées par une valeur réservée dans la séquence de données;
- si une classe d'applications particulière a besoin d'une fonctionnalité supplémentaire indépendante du format de charge utile, le profil de ces applications doit définir des champs fixes supplémentaires qui viennent immédiatement à la suite du champ SSRC de l'en-tête fixe existant. Ces applications pourront accéder rapidement et directement à ces champs

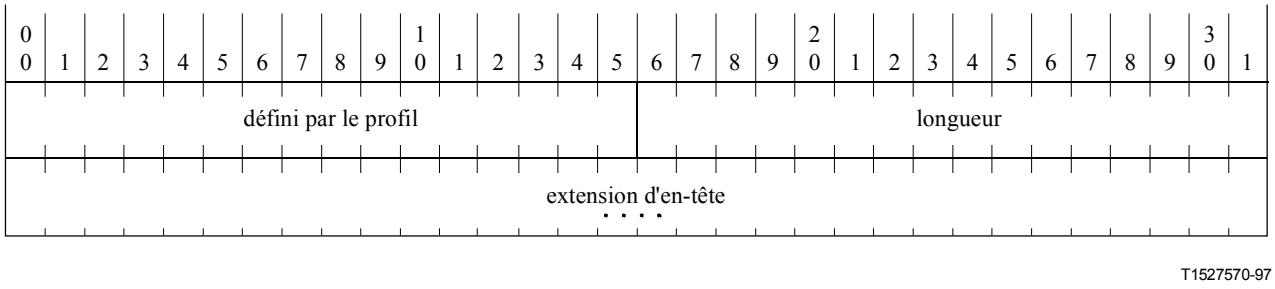
supplémentaires tandis que les enregistreurs ou les contrôleurs indépendants du profil pourront continuer à traiter les paquets RTP en n'interprétant que les douze premiers octets.

Si une fonctionnalité supplémentaire devient nécessaire pour tous les profils, une nouvelle version du protocole RTP doit alors être définie afin d'effectuer une modification permanente de l'en-tête fixe.

A.5.3.1 Extension de l'en-tête RTP

Un mécanisme d'extension est fourni pour permettre d'expérimenter des implémentations particulières dans lesquelles sont intégrées de nouvelles fonctions qui sont indépendantes du format de charge utile et qui nécessitent des informations supplémentaires dans l'en-tête de paquet de données RTP. Ce mécanisme est conçu de sorte que l'extension d'en-tête puisse être ignorée par les autres implémentations en interfonctionnement qui n'ont pas été étendues.

On notera que cette extension d'en-tête n'est prévue que pour une utilisation limitée. Pour la plupart des utilisations potentielles de ce mécanisme, il serait mieux de ne pas recourir à ce mécanisme et d'utiliser les méthodes décrites au sous-paragraphe précédent. Par exemple, la réalisation d'une extension de l'en-tête fixe qui soit propre au profil est moins coûteuse car cette extension n'est pas conditionnelle et elle n'est pas placée dans une position variable. Les informations supplémentaires requises pour un format de charge utile donné ne doivent pas figurer dans une telle extension d'en-tête, mais elles doivent au contraire figurer dans la section de charge utile du paquet:



T1527570-97

Si le bit X de l'en-tête RTP vaut un, une extension d'en-tête de longueur variable est rattachée à l'en-tête RTP, à la suite de la liste CSRC si elle est présente. L'extension d'en-tête contient un champ de 16 bits qui indique le nombre de mots de 32 bits figurant dans l'extension, non compris les quatre octets que compte l'en-tête d'extension (zéro est donc une longueur valable). Une seule extension peut être rattachée à l'en-tête RTP. Pour permettre à diverses implémentations en interfonctionnement d'être expérimentées indépendamment avec des extensions d'en-tête différentes, ou pour permettre à une implémentation donnée d'être expérimentée avec plusieurs types d'extension d'en-tête, les 16 premiers bits de l'extension d'en-tête sont laissés libres pour pouvoir indiquer s'il s'agit d'identificateurs ou de paramètres. Le format de ces 16 bits doit être défini par la spécification du profil des implémentations. La présente spécification RTP ne définit pas d'extension d'en-tête.

A.6 Protocole de commande RTP (RTCP)

Le protocole de commande RTP (RTCP), basé sur la transmission périodique de paquets de commande à tous les participants à une session, utilise le même mécanisme de distribution que celui utilisé pour les paquets de données. Le protocole sous-jacent doit assurer le multiplexage des paquets de données et de commande, par exemple en utilisant des numéros d'accès distincts et le protocole UDP. Le protocole RTCP exécute quatre fonctions:

1) la première fonction consiste à fournir un retour sur la qualité de distribution des données. Cette fonction fait partie intégrante du rôle du protocole RTP étant donné qu'il s'agit d'un protocole de transport, et cette fonction est liée aux fonctions de commande de flux et d'encombrement appartenant aux autres protocoles de transport. Le retour peut être

directement utile pour commander des codages adaptatifs [A-6] et [A-7], mais des expériences menées avec la multidiffusion IP ont montré qu'il est également très important d'obtenir un retour des récepteurs pour diagnostiquer les défauts dans la distribution. L'envoi de rapports de retour de réception à tous les participants permet à celui qui observe les problèmes de déterminer s'il s'agit de problèmes locaux ou globaux. Avec un mécanisme de distribution comme la multidiffusion IP, il est également possible à une entité qui ne participe pas à la session – un fournisseur de service de réseau par exemple – de recevoir les informations de retour et de jouer le rôle d'un contrôleur tiers pour diagnostiquer les problèmes de réseau. Cette fonction de retour est réalisée par les rapports d'émetteur et de récepteur RTCP, qui sont décrits au A.6.3, Rapports d'émetteur et de récepteur.

2) les paquets RTCP contiennent de façon permanente un identificateur au niveau transport caractérisant une source RTP, cet identificateur s'appelle nom canonique ou CNAME (voir A.6.4.1, CNAME: élément SDES identificateur d'extrémité canonique). Etant donné que l'identificateur SSRC peut changer en cas de découverte d'un conflit ou de redémarrage d'un programme, les récepteurs exigent que le CNAME ne perde pas la trace de chaque participant. Ils exigent aussi que le CNAME associe plusieurs flux de données provenant d'un participant donné en un ensemble de sessions RTP connexes, par exemple pour synchroniser les données audio et vidéo;

3) pour les deux premières fonctions, il est nécessaire que tous les participants envoient des paquets RTCP; il faut donc que la fréquence d'envoi soit commandée pour que le protocole RTP puisse s'étendre à un grand nombre de participants. Chaque participant envoyant ses paquets de commande à tous les autres, chaque participant peut observer de façon indépendante le nombre de participants. Ce nombre sert à calculer la fréquence d'envoi des paquets, comme l'explique A.6.2, Intervalle de transmission RTCP;

4) une quatrième fonction, optionnelle, consiste à acheminer les informations de commande de session minimales, par exemple l'identification du participant qui doit être affichée à l'interface utilisateur. Cette fonction devrait surtout trouver son utilité dans les sessions "à commande souple" où les participants rejoignent et quittent la session sans contrôle de membre et sans négociation des paramètres. Le protocole RTCP sert de canal pour atteindre tous les participants, mais il n'est pas prévu pour nécessairement prendre en charge toutes les prescriptions de communication de commande d'une application. Un protocole de commande de session de niveau supérieur, qui sort du cadre de la présente Recommandation, peut être nécessaire.

Les fonctions 1 à 3 sont obligatoires en cas d'utilisation du protocole RTP dans l'environnement de multidiffusion IP et elles sont recommandées pour tous les autres environnements. Il est conseillé aux concepteurs d'application RTP d'éviter les mécanismes qui ne peuvent fonctionner qu'en mode monodiffusion et qui ne pourront être étendus à de plus grands nombres de participants.

A.6.1 Format de paquet RTCP

La présente spécification définit plusieurs types de paquets RTCP qui comportent diverses informations de commande:

SR: rapport d'émetteur, pour les statistiques de transmission et de réception des participants qui sont des émetteurs actifs.

RR: rapport de récepteur, pour les statistiques de réception des participants qui ne sont pas des émetteurs actifs.

SDES: éléments de description de source, y compris le CNAME.

BYE: indication de fin de participation.

APP: fonctions propres à l'application.

Chaque paquet RTCP commence par une partie fixe semblable à celle des paquets de données RTP, suivie d'éléments structurés qui peuvent être de longueur variable selon le type de paquet mais la fin du paquet se trouve toujours sur une frontière à 32 bits. Le respect de la prescription d'alignement et l'inclusion d'un champ de longueur dans la partie fixe permettent "d'empiler" les paquets RTCP. Plusieurs paquets RTCP peuvent être concaténés sans séparateur pour former un paquet RTCP composé qui est envoyé dans un même paquet du protocole de couche inférieure, par exemple du protocole UDP. Les paquets RTCP contenus dans le paquet composé ne sont pas dénombrés explicitement puisqu'il est prévu que les protocoles de couche inférieure fournissent une longueur globale pour déterminer la fin du paquet composé.

Chaque paquet RTCP contenu dans le paquet composé peut être traité indépendamment, sans prescription ni sur l'ordre ni sur la combinaison des paquets. Toutefois, la réalisation des fonctions du protocole est soumise aux contraintes suivantes:

- les statistiques de réception (dans les paquets SR ou RR) doivent être transmises aussi souvent que le permettront les contraintes de largeur de bande afin de maximiser la résolution des statistiques; chaque paquet RTCP composé transmis périodiquement doit donc comporter un paquet de rapport;
- les nouveaux récepteurs ont besoin de recevoir le CNAME d'une source le plus tôt possible pour identifier la source et pour commencer à associer les médias à des fins de postsynchronisation par exemple, chaque paquet RTCP composé doit également comporter l'élément SDES CNAME;
- le nombre de types de paquet qui peuvent apparaître en premier dans le paquet composé doit être limité pour augmenter le nombre de bits constants dans le premier mot et la probabilité de valider avec succès les paquets RTCP et de ne pas valider les paquets de données RTP ou les autres paquets non connexes dont l'adresse est incorrecte.

Il faut donc que tous les paquets RTCP soient envoyés dans un paquet composé contenant au moins deux paquets distincts, avec le format recommandé suivant:

préfixe de chiffrement: si le paquet composé doit être chiffré et seulement dans ce cas, il doit avoir un préfixe aléatoire de 32 bits, différent pour chaque paquet composé transmis.

SR ou RR: le premier paquet RTCP contenu dans le paquet composé doit toujours être un paquet de rapport afin de faciliter la validation d'en-tête décrite au A.2. Ceci est vrai même si aucune donnée n'est envoyée ni reçue, auquel cas un paquet RR vide est envoyé, et même si le seul autre paquet RTCP contenu dans le paquet composé est un paquet BYE.

RR supplémentaires: si le nombre de sources faisant l'objet d'un rapport sur les statistiques de réception dépasse 31 – nombre maximal pour un seul paquet SR ou RR – alors des paquets RR supplémentaires doivent suivre le paquet de rapport initial.

SDES: un paquet SDES contenant un élément CNAME doit figurer dans chaque paquet RTCP composé. D'autres éléments de description de source peuvent optionnellement être inclus s'ils sont requis par une application donnée, sous réserve de satisfaire les contraintes de largeur de bande (voir A.6.2.2, Attribution de largeur de bande pour la description de source).

BYE ou APP: d'autres types de paquets RTCP, y compris ceux qu'il reste à définir, peuvent venir à la suite dans un ordre quelconque, à l'exception du paquet BYE qui doit être le dernier paquet envoyé avec un identificateur SSRC/CSRC donné. Chaque type de paquet peut apparaître plusieurs fois.

Il est recommandé, chaque fois que c'est possible, que les traducteurs et les mélangeurs combinent les paquets RTCP provenant de plusieurs sources et transmis sous forme d'un seul paquet composé afin de limiter le préfixe de paquet (voir A.7, traducteurs et mélangeurs RTP). La Figure A.1 montre un exemple de paquet composé RTCP pouvant être produit par un mélangeur. Dans le cas où la longueur totale d'un paquet composé dépasserait l'unité de transmission maximale (MTU, *maximum*

transmission unit) du conduit de réseau, ce paquet peut être segmenté en plusieurs paquets composés plus courts à transmettre dans des paquets distincts du protocole sous-jacent. On notera que chacun des paquets composés doit commencer par un paquet SR ou RR.

Une implémentation peut ignorer les paquets RTCP entrants dont elle ne connaît pas les types. Des types de paquets RTCP supplémentaires peuvent être enregistrés auprès de l'autorité chargée de l'assignation des numéros Internet (IANA, *Internet assigned numbers authority*).

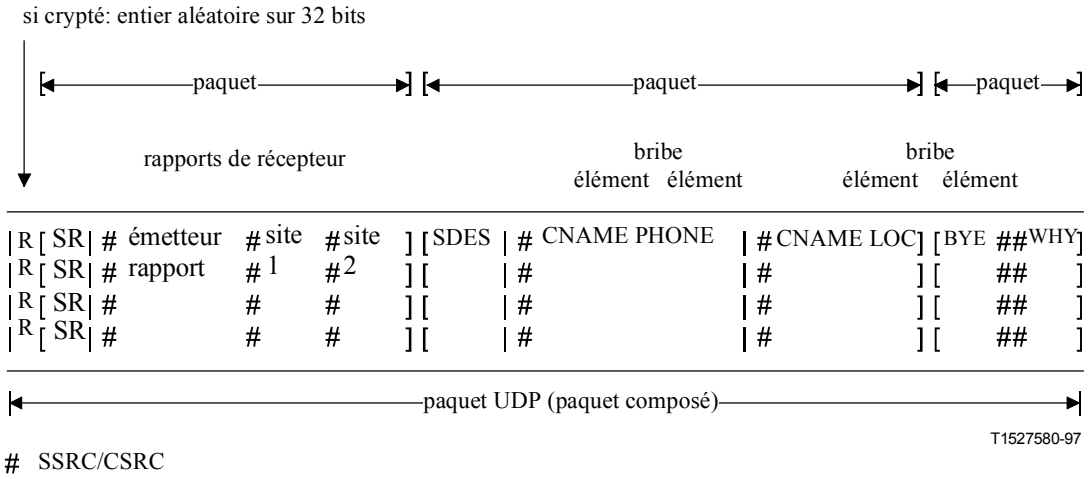


Figure A.1/H.225.0 – Exemple de paquet composé RTCP

A.6.2 Intervalle de transmission RTCP

Le protocole RTP est conçu de telle sorte qu'une application puisse s'adapter automatiquement à des tailles de session allant de quelques participants à des milliers de participants. Dans une audioconférence par exemple, le trafic des données est forcément autolimité car deux personnes au plus parleront en même temps; avec une distribution en mode multidiffusion, le débit des données sur une liaison donnée quelconque restera donc relativement constant quel que soit le nombre de participants. En revanche, le trafic de commande n'est pas autolimité. Si les rapports de réception de chaque participant sont envoyés à débit constant, le trafic de commande augmentera linéairement en fonction du nombre de participants. Il faut donc réduire ce débit.

Pour chaque session, on suppose que le trafic de données ne doit pas dépasser une limite globale appelée "largeur de bande de session" qui doit être répartie entre les participants. Cette largeur de bande peut être réservée et la limite imposée par le réseau, ou il peut simplement s'agir d'un partage raisonnable. La largeur de bande de session peut être choisie sur la base de certains coûts ou d'une connaissance *a priori* de la largeur de bande de réseau disponible pour la session. Elle est en quelque sorte indépendante du codage du média, mais le choix du codage peut être limité par la largeur de bande de la session. Une application de gestion de session devra fournir le paramètre de largeur de bande de session lorsqu'elle invoquera une application de média, mais les applications de média peuvent aussi fixer un paramètre par défaut en fonction de la largeur de bande correspondant aux données d'un seul émetteur et au codage choisi pour la session. L'application peut aussi imposer des limites de largeur de bande sur la base de règles s'appliquant à la multidiffusion ou sur la base d'autres critères.

Les calculs de largeur de bande pour le trafic de commande et de données tiennent compte des protocoles de réseau et de transport de couche inférieure (UDP et IP par exemple) étant donné qu'il s'agit des informations que le système de réservation des ressources a besoin de connaître. L'application devrait également savoir lequel de ces protocoles est utilisé. Dans le calcul, il n'est pas

tenu compte des en-têtes au niveau liaison car le paquet sera encapsulé avec différents en-têtes au niveau liaison tout au long de sa transmission.

Le trafic de commande doit être limité à un faible pourcentage connu de la largeur de bande de la session: faible de façon à ne pas dégrader la première fonction du protocole de transport qui est d'acheminer les données; connu de sorte que le trafic de commande puisse figurer dans la spécification de largeur de bande donnée à un protocole de réservation des ressources, et que chaque participant puisse calculer indépendamment sa part. Il est proposé que le pourcentage de la largeur de bande de la session attribué aux paquets RTCP soit fixe et égal à 5%. Cette valeur et la valeur d'autres constantes intervenant dans le calcul de l'intervalle ne sont pas des valeurs seuils, mais il faut que tous les participants à la session utilisent les mêmes valeurs afin de calculer le même intervalle. Ces constantes doivent donc être fixes pour un profil donné.

L'algorithme décrit au A.7 a été conçu pour répondre aux objectifs présentés ci-dessus. Il calcule l'intervalle entre les envois de paquets RTCP composés pour pouvoir répartir la largeur de bande de trafic de commande autorisée entre les participants. Grâce à cela, une application qui fournit des réponses rapides pour les petites sessions dans lesquelles, par exemple, l'identification de tous les participants est importante, peut néanmoins s'adapter automatiquement aux grandes sessions. L'algorithme présente les caractéristiques suivantes:

- l'ensemble des émetteurs se voit attribuer au moins 1/4 de la largeur de bande du trafic de commande de sorte que dans les sessions comportant beaucoup de récepteurs et peu d'émetteurs, les participants souhaitant rejoindre la session à titre d'émetteur recevront un CNAME plus rapidement;
- il est nécessaire que l'intervalle calculé entre les paquets RTCP soit supérieur à au moins cinq secondes afin d'éviter que des rafales de paquets RTCP dépassent la largeur de bande autorisée lorsque le nombre de participants est faible et que la courbe du trafic n'est pas lisse conformément à la loi des grands nombres;
- on fait varier de façon aléatoire l'intervalle entre les paquets RTCP d'une valeur valant entre 0,5 et 1,5 fois l'intervalle calculé de manière à éviter la synchronisation non voulue de tous les participants [A-8]. Le premier paquet RTCP envoyé après avoir rejoint une session est également retardé d'une valeur aléatoire autour de la moitié de l'intervalle RTCP minimal au cas où l'application soit démarrée simultanément dans plusieurs sites, par exemple à la suite d'une annonce de session;
- la taille moyenne de tous les paquets RTCP composés reçus et envoyés est évaluée dynamiquement, afin de permettre une adaptation automatique aux variations de quantité d'informations de commande transmises.

Cet algorithme peut être utilisé pour les sessions dans lesquelles tous les participants sont autorisés à envoyer des paquets. Dans ce cas, le paramètre de largeur de bande de session est le produit de la largeur de bande attribuée à un émetteur par le nombre de participants, et la largeur de bande RTCP vaut 5% de cette largeur.

A.6.2.1 Mise à jour du nombre de membres de la session

Le calcul de l'intervalle entre les paquets RTCP est fonction de l'évaluation du nombre de sites participant à la session. De nouveaux sites sont comptés lorsqu'ils sont entendus, et une rubrique est créée pour chacun dans un tableau indexé par l'identificateur SSRC ou CSRC (voir A.8.2, Résolution des collisions et détection des boucles) afin de garder la trace de ces sites. Les nouvelles rubriques ne peuvent être considérées comme valables qu'après la réception de plusieurs paquets comportant le nouvel identificateur SSRC (voir A.6.1). Une rubrique peut être supprimée du tableau après la réception d'un paquet RTCP BYE comportant l'identificateur SSRC correspondant.

Un participant peut marquer un autre site comme étant inactif, ou le supprimer s'il n'est plus valide, si aucun paquet RTP ou RTCP n'a été reçu au bout d'un petit nombre d'intervalles de rapport RTCP (5 est proposé). Cela fournit une certaine robustesse contre la perte de paquet. Il faut que tous les

sites calculent approximativement la même valeur pour l'intervalle de rapport RTCP pour que cette temporisation fonctionne correctement.

Une fois qu'un site a été validé, si par la suite il est marqué inactif, il faut continuer à conserver l'état de ce site et à compter ce site dans le nombre total des sites utilisant en partage la largeur de bande RTCP pendant une période suffisamment longue pour couvrir les partitions typiques du réseau. Le but est d'éviter l'excès de trafic, lorsque la partition se résorbe, dû à un intervalle de rapport RTCP trop petit. Une temporisation de 30 minutes est proposée. On notera que cette valeur est toujours supérieure à 5 fois la plus grande valeur prévue pour l'intervalle de rapport RTCP (environ 2 à 5 minutes).

A.6.2.2 Attribution de largeur de bande pour la description de source

La présente spécification définit plusieurs éléments de description de source (SDES, *source description*) qui viennent s'ajouter à l'élément CNAME obligatoire, par exemple NAME (nom personnel) et EMAIL (adresse électronique). Elle permet en outre de définir de nouveaux types de paquet propre à l'application. Les applications doivent être prudentes lorsqu'elles attribuent une largeur de bande de commande pour ces informations supplémentaires car cette nouvelle largeur de bande attribuée ralentira la fréquence d'envoi des rapports de réception et du CNAME, d'où une dégradation de la performance du protocole. Il est conseillé de ne pas utiliser plus de 20% de la largeur de bande RTCP attribuée à un seul participant pour transporter les informations supplémentaires. Le but n'est donc pas d'inclure tous les éléments SDES dans chaque application. Les éléments qui sont inclus doivent se voir attribuer une fraction de la largeur de bande qui est fonction de leur utilité. Plutôt que d'évaluer dynamiquement ces fractions, il est recommandé de traduire statiquement les pourcentages en nombres d'intervalles de rapport sur la base de la longueur type d'un élément.

Une application peut par exemple être conçue pour envoyer les éléments CNAME, NAME et EMAIL et seulement ceux-là. L'élément NAME peut se voir accorder un rang de priorité plus élevé que l'élément EMAIL car l'élément NAME est affiché en permanence au niveau de l'interface utilisateur de l'application, alors que l'élément EMAIL n'est affiché qu'à la demande. A chaque intervalle RTCP, un paquet RR et un paquet SDES comportant l'élément CNAME sont envoyés. Pour une petite session fonctionnant avec l'intervalle minimal, cet envoi se produit toutes les 5 secondes en moyenne. Un intervalle sur trois (toutes les 15 secondes), un élément supplémentaire est inclus dans le paquet SDES. Sept fois sur huit, il s'agit de l'élément NAME, et une fois sur huit (toutes les deux minutes), il s'agit de l'élément EMAIL.

Lorsque plusieurs applications fonctionnent simultanément grâce à la possession d'un seul élément CNAME par participant permettant d'assurer un lien entre ces applications, par exemple dans une conférence multimédia composée d'une session RTP pour chaque support, les informations SDES supplémentaires peuvent être transmises dans une seule session RTP. Dans les autres sessions, seul l'élément CNAME est transmis.

A.6.3 Rapports d'émetteur et de récepteur

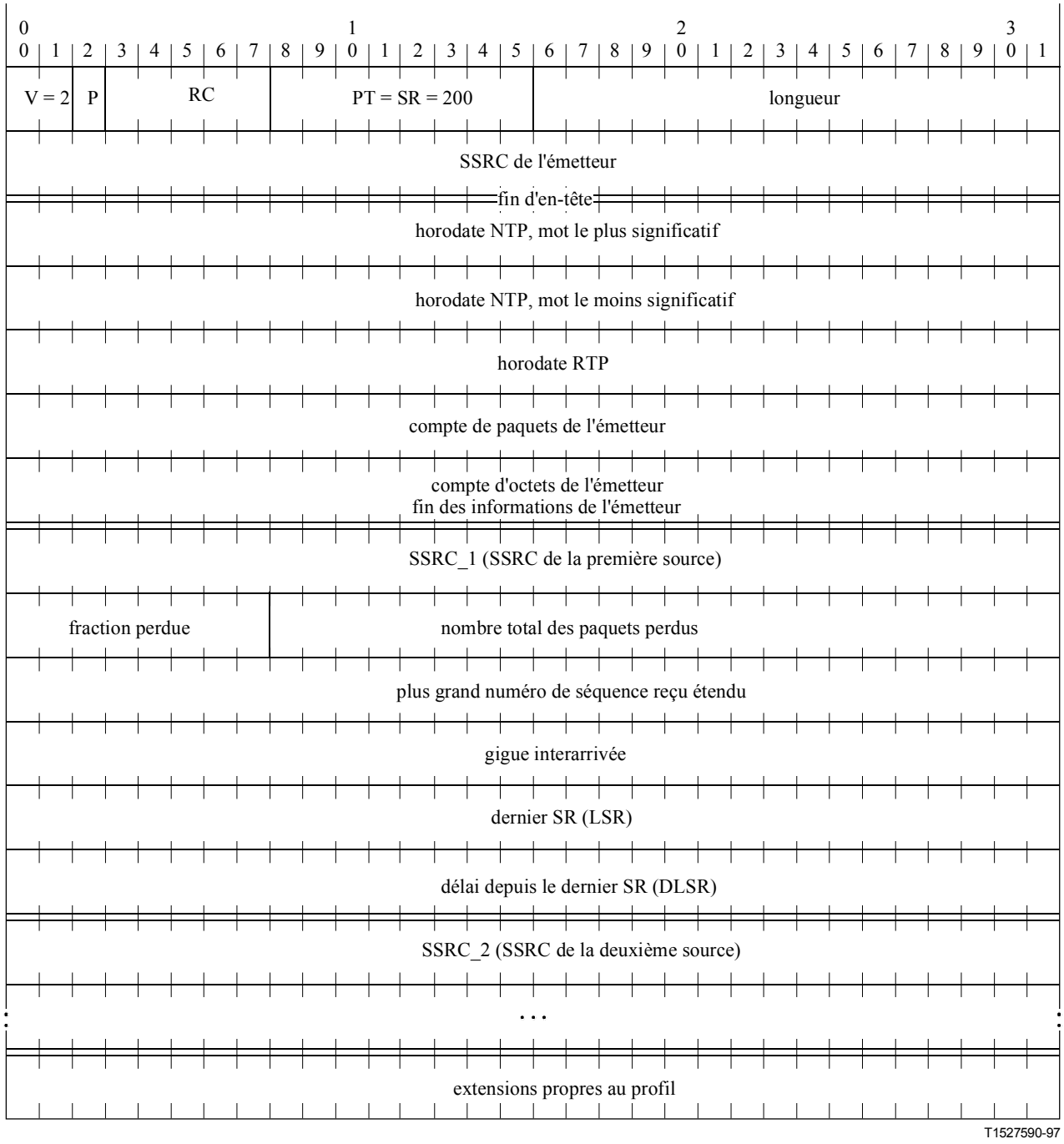
Les récepteurs RTP fournissent un retour de qualité de réception à l'aide de paquets de rapport RTCP; il existe deux types de rapport selon que le récepteur est aussi émetteur ou non. La seule différence entre le rapport d'émetteur (SR, *sender report*) et le rapport de récepteur (RR, *receiver report*), mis à part le code de type de paquet, réside dans le fait que le rapport d'émetteur contient une section d'informations d'émetteur sur 20 octets à l'intention des émetteurs actifs. Un site envoie un paquet SR s'il a envoyé des paquets de données depuis l'envoi du dernier ou de l'avant-dernier rapport, dans les autres cas, il envoie un paquet RR.

Les deux types de paquet SR et RR comportent zéro, un ou plusieurs blocs de rapport de réception, un pour chacune des sources de synchronisation en provenance desquelles ce récepteur a reçu des paquets de données RTP depuis le dernier rapport. Aucun rapport n'est transmis pour les sources contributives énumérées dans la liste CSRC. Chaque bloc de rapport de réception fournit des

statistiques concernant les données reçues en provenance de la source indiquée dans ce bloc. Etant donné qu'un paquet SR ou RR peut contenir au plus 31 blocs de rapport de réception, des paquets RR supplémentaires peuvent être empilés après le paquet SR ou RR initial, si nécessaire, pour faire figurer les rapports de réception de toutes les sources entendues depuis l'envoi du dernier rapport.

Les sous-paragraphes qui suivent définissent les formats des deux rapports, la façon dont ils peuvent être étendus d'une manière propre au profil si une application nécessite des informations de retour supplémentaires, et la façon dont les rapports peuvent être utilisés. L'établissement de rapports de réception par les traducteurs et les mélangeurs est détaillé au A.7, Traducteurs et mélangeurs RTP.

A.6.3.1 SR: paquet RTCP rapport d'émetteur



Le paquet de rapport d'émetteur est constitué de trois sections, éventuellement suivies d'une quatrième section d'extensions propre au profil, si elle est définie. La première section – l'en-tête – a une longueur de 8 octets. La signification des champs est la suivante:

version (V): 2 bits. Ce champ identifie la version du protocole RTP, qui est la même dans les paquets RTCP et les paquets RTP. La version définie par la présente spécification est deux (2).

remplissage (P, ***padding*)**: 1 bit. Si le bit de remplissage est positionné, ce paquet RTCP contient à la fin quelques octets supplémentaires de remplissage qui ne font pas partie des informations de commande. Le dernier octet de remplissage donne le nombre d'octets de remplissage qu'il faut ignorer. Le remplissage peut être requis par certains algorithmes de chiffrement qui nécessitent des tailles de bloc fixes. Dans un paquet RTCP composé, le remplissage ne doit être requis qu'au niveau du dernier paquet simple car le paquet composé est chiffré comme un tout.

compte de rapports de réception (RC): 5 bits. Ce champ donne le nombre de blocs de rapport de réception contenus dans ce paquet. La valeur zéro est valable.

type de paquet (PT): 8 bits. Ce champ contient la constante 200 qui indique qu'il s'agit d'un paquet RTCP SR.

longueur: 16 bits. Il s'agit de la longueur de ce paquet RTCP en nombre de mots de 32 bits moins un, y compris l'en-tête et le remplissage éventuel. (Grâce au décalage de un, zéro est une longueur valable et les éventuelles boucles infinies lors de l'analyse d'un paquet RTCP composé sont évitées; par ailleurs le fait de compter le nombre de mots de 32 bits permet d'éviter le contrôle de validité sur les multiples de 4.)

SSRC: 32 bits. Il s'agit de l'identificateur de source de synchronisation correspondant à l'émetteur de ce paquet SR.

La deuxième section, les informations de l'émetteur, a une longueur de 20 octets et figure dans chaque paquet de rapport d'émetteur. Elle récapitule les transmissions de données faites par cet émetteur. La signification des champs est la suivante:

horodate NTP: 64 bits. Ce champ indique l'heure – lue sur une horloge murale – à laquelle ce rapport a été envoyé; la combinaison de cette horodate et des horodates renvoyées dans les rapports de réception des autres récepteurs permet de mesurer le temps de transmission aller-retour vers ces récepteurs. Les récepteurs doivent savoir que la précision de la mesure de l'horodate peut être beaucoup moins bonne que la résolution de l'horodate NTP. L'incertitude sur la mesure de l'horodate n'est pas indiquée car elle peut ne pas être connue. Un émetteur qui peut garder la trace du temps écoulé mais qui ne connaît pas l'heure indiquée par l'horloge murale peut se servir du temps écoulé jusqu'à ce qu'il rejoigne la session. Ce temps est supposé être inférieur à 68 ans, le bit le plus élevé vaudra donc zéro. Il est permis d'utiliser l'horloge d'échantillonnage pour évaluer l'heure indiquée par l'horloge murale. Un émetteur qui ne connaît pas l'heure indiquée par l'horloge murale et qui n'a pas la notion du temps écoulé peut mettre l'horodate NTP à zéro.

horodate RTP: 32 bits. Cette horodate correspond à l'horodate NTP (ci-dessus), mais elle a la même unité et le même décalage aléatoire que les horodates RTP des paquets de données. Cette correspondance peut servir à la synchronisation intra- et intermédia des sources dont les horodates NTP sont synchronisées; elle peut aussi servir aux récepteurs indépendants du média à évaluer la fréquence nominale de l'horloge RTP. On notera que dans la plupart des cas, cette horodate sera différente de l'horodate RTP d'un paquet de données adjacent quelconque. Elle est plutôt calculée à partir de l'horodate NTP correspondante et en fonction de la relation qui existe entre l'horodate RTP et le temps réel et qui est mise à jour périodiquement par le contrôle de l'heure indiquée par l'horloge murale à un instant d'échantillonnage.

compte des paquets de l'émetteur: 32 bits. Il s'agit du nombre total de paquets de données RTP envoyés par l'émetteur depuis le début de la transmission et jusqu'au moment où ce paquet SR a été généré. Le compte est réinitialisé si l'émetteur change d'identificateur SSRC.

compte des octets de l'émetteur: 32 bits. Il s'agit du nombre total d'octets de charge utile (c'est-à-dire non compris l'en-tête et le remplissage) transmis par l'émetteur dans des paquets de données RTP depuis le début de la transmission et jusqu'au moment où ce paquet SR a été généré. Le compte est réinitialisé si l'émetteur change d'identificateur SSRC. Ce champ peut servir à évaluer le débit moyen de données de charge utile.

La troisième section contient zéro, un ou plusieurs blocs de rapport de réception selon le nombre d'autres sources que cet émetteur a entendues depuis le dernier rapport. Chaque bloc de rapport de réception comporte des statistiques sur la réception des paquets RTP provenant d'une même source de synchronisation. Les récepteurs n'indiquent pas de statistiques lorsqu'une source change d'identificateur SSRC en raison d'une collision. Ces statistiques sont les suivantes:

SSRC_n (identificateur de la source): 32 bits. Il s'agit de l'identificateur SSRC de la source sur laquelle portent les informations figurant dans ce bloc de rapport de réception.

fraction perdue: 8 bits. Il s'agit de la fraction des paquets de données RTP provenant de la source SSRC_n qui ont été perdus depuis l'envoi du paquet SR ou RR précédent, exprimée par un nombre à virgule fixe, la virgule binaire étant placée au bord gauche du champ. (Cela revient à prendre la partie entière après avoir multiplié la fraction perdue par 256.) Cette fraction est définie au nombre de paquets perdus divisé par le nombre de paquets prévus (voir la définition dans le prochain alinéa). Une implémentation est montrée au A.6.3. Si la perte est négative en raison de paquets en double, la fraction perdue est mise à zéro. On notera qu'un récepteur ne peut pas dire si des paquets ont été perdus après le dernier reçu, et qu'aucun bloc de rapport de réception ne sera transmis pour une source donnée si tous les paquets envoyés par cette source pendant le dernier intervalle de rapport ont été perdus.

nombre total des paquets perdus: 24 bits. Il s'agit du nombre total de paquets de données RTP provenant de la source SSRC_n qui ont été perdus depuis le début de la réception. Par définition, c'est le nombre de paquets prévus moins le nombre de paquets effectivement reçus, où le nombre de paquets reçus comprend les paquets en retard et les paquets en double. Par conséquent, les paquets qui arrivent en retard ne sont pas comptés comme des paquets perdus, et la perte peut être négative s'il y a des paquets en double. Le nombre de paquets prévus est défini comme le dernier numéro de séquence reçu étendu (défini plus loin) moins le numéro de séquence initial reçu. Ce nombre peut se calculer par la méthode donnée au A.6.3.

plus grand numéro de séquence reçu étendu: 32 bits. Les 16 bits inférieurs contiennent le plus grand numéro de séquence reçu dans un paquet de données RTP provenant de la source SSRC_n, et les 16 bits les plus significatifs étendent ce numéro de séquence par le nombre correspondant de cycles de numéros de séquence, qui peut être mis à jour au moyen de l'algorithme au A.13. On notera que des récepteurs distincts à l'intérieur d'une même session généreront des extensions du numéro de séquence différentes si les heures auxquelles ils ont démarré sont très différentes.

gigue interarrivée: 32 bits. Il s'agit d'une évaluation de la variance statistique du temps interarrivée des paquets de données RTP, mesurée en unités d'horodate et exprimée par un entier non signé. La gigue interarrivée J est définie comme l'écart moyen (valeur absolue lissée) de la différence D d'espacement de deux paquets au niveau du récepteur par rapport à l'émetteur. Comme le montre l'équation ci-dessous, la différence D est équivalente à la différence de "temps de transit relatif" pour les deux paquets; le temps de transit relatif est la différence entre l'horodate RTP d'un paquet et l'heure indiquée par l'horloge du récepteur au moment de l'arrivée du paquet, mesurées dans la même unité.

Si "Si" est l'horodate RTP du paquet i, et "Ri" l'heure d'arrivée en unités d'horodate RTP du paquet i, alors pour les deux paquets i et j, D peut s'exprimer comme suit:

$$D(i+j) = (Rj - Ri) - (Sj - Si) = (Rj - Sj) - (Rj - Si)$$

Chaque fois qu'un paquet de données i est reçu en provenance de la source SSRC_n, la gigue interarrivée est recalculée en fonction de la différence D s'appliquant à ce paquet et au paquet i – 1 le précédant dans l'ordre d'arrivée (pas nécessairement le paquet le précédant dans la séquence), selon l'équation:

$$J = J + \frac{\left|D(i-1,i)\right| - J}{16}$$

Chaque fois qu'un rapport de réception est transmis, la valeur courante de J est échantillonnée.

Le calcul de la gigue est prescrit ici pour permettre aux contrôleurs indépendants du profil de faire des interprétations valables des rapports venant de différentes implémentations. Cet algorithme est l'estimateur optimal du premier ordre et le paramètre de gain 1/16 donne un bon facteur de réduction du bruit tout en conservant une vitesse de convergence raisonnable [A-9]. Le sous-paragraphe A.8 donne un exemple d'implémentation.

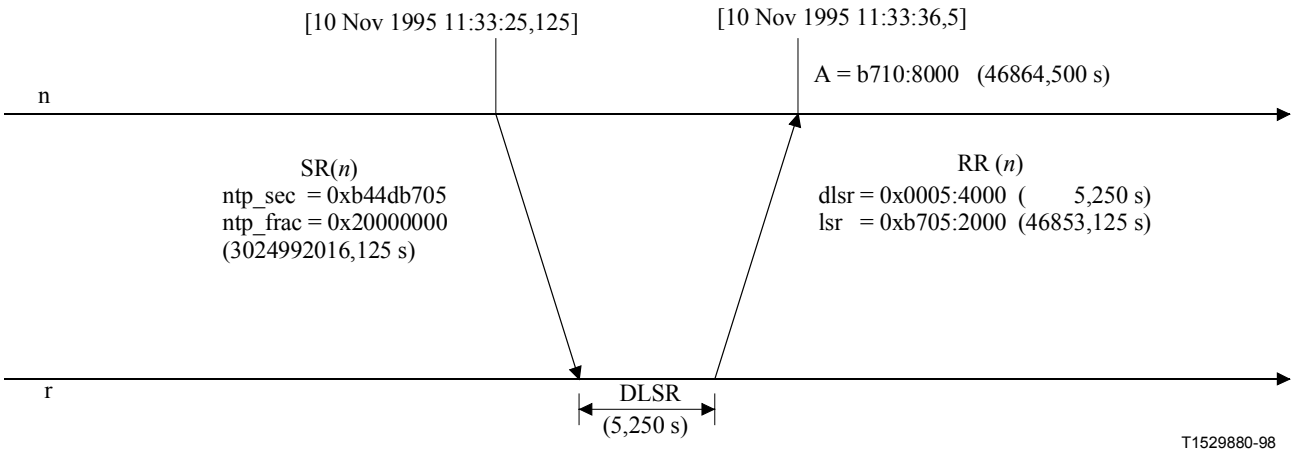
horodate du dernier paquet SR (LSR, *last SR timestamp*): 32 bits. Il s'agit des 32 bits du milieu parmi les 64 de l'horodate NTP (comme expliqué au A.4, ordre des octets, alignement et format temporel) qui figurent dans le dernier paquet RTCP SR (rapport d'émetteur) reçu en provenance de la source SSRC_n. Si aucun paquet SR n'a encore été reçu, le champ est mis à zéro.

délai depuis le dernier paquet SR (DLSR, *delay since last SR*): 32 bits. Il s'agit du délai, exprimé dans l'unité 1/65536 secondes, entre la réception du dernier paquet SR envoyé par la source SSRC_n et l'envoi de ce bloc de rapport de réception. Si aucun paquet SR n'a encore été reçu de la source SSRC_n, le champ DLSR est mis à zéro.

Désignons par SSRC_r le récepteur qui envoie ce rapport de réception. La source SSRC_n peut calculer le temps de transmission total vers le récepteur SSRC_r en enregistrant l'heure A à laquelle ce bloc de rapport de réception est reçu. Elle calcule le temps total aller-retour A – LSR en utilisant le champ LSR (horodate du dernier paquet SR), et en soustrayant ensuite le champ DLSR pour aboutir au temps de transmission aller-retour (A – LSR – DLSR). Ceci est illustré sur la Figure A.2.

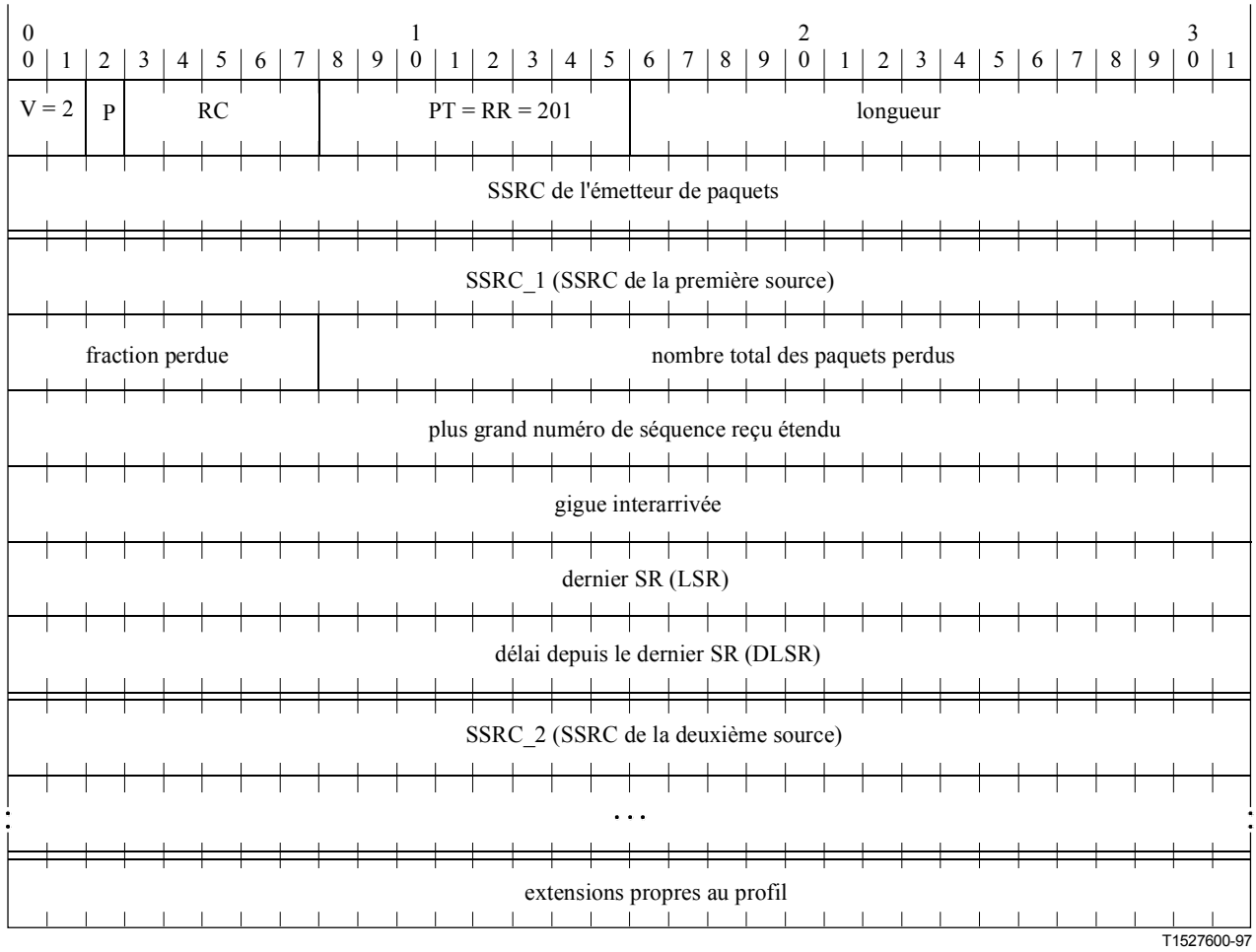
Ce délai peut servir de mesure approximative de la distance à un groupe de récepteurs, même si certaines liaisons présentent des délais très asymétriques.

A.6.3.2 RR – Paquet RTCP rapport de récepteur



A	0xb710:8000	(46864,500 s)
– DLSR	0x0005:4000	(5,250 s)
– LSR	0xb705:2000	(46853,125 s)
= délai	0x0006:2000	(6,125 s)

Figure A.2/H.225.0 – Exemple de calcul de temps de transmission aller-retour



Le format du paquet RR (rapport de récepteur) est identique à celui du paquet SR aux exceptions près suivantes: le champ de type de paquet contient la constante 201 et les cinq mots d'informations de l'émetteur sont omis (il s'agit des horodates NTP et RTP, du compte des paquets de l'émetteur et du compte des octets de l'émetteur). Les champs restants ont la même signification que pour le paquet SR.

Un paquet RR vide (RC = 0) est placé en tête d'un paquet RTCP composé lorsqu'il n'y a ni transmission ni réception de données à signaler.

A.6.3.3 Extension des rapports d'émetteur et de récepteur

Un profil doit définir des extensions aux rapports d'émetteur et de récepteur qui sont propres au profil ou à l'application si des informations supplémentaires concernant l'émetteur ou les récepteurs doivent faire l'objet de rapports réguliers. Il vaut mieux utiliser cette méthode plutôt que définir un autre type de paquet RTCP car cette méthode nécessite un préfixe moins long:

- moins d'octets dans le paquet (ni en-tête RTCP ni champ SSRC);
- analyse plus simple et plus rapide car les applications qui fonctionnent sous ce profil sont programmées pour toujours attendre les champs d'extension à une position directement accessible après les rapports de réception.

Si des informations d'émetteur supplémentaires sont requises, elles doivent d'abord être incluses dans l'extension aux rapports d'émetteur, mais elles ne figurent pas dans les rapports de récepteur. Si des informations concernant les récepteurs doivent être incluses, ces données doivent être structurées sous forme d'un tableau de blocs qui est parallèle au tableau des blocs de rapport de réception existant; c'est-à-dire que le nombre de blocs est indiqué par le champ RC.

A.6.3.4 Analyse des rapports d'émetteur et de récepteur

Normalement, le retour sur la qualité de réception sera utile non seulement à l'émetteur mais aussi aux autres récepteurs et aux contrôleurs tiers. L'émetteur peut modifier ses transmissions en fonction du retour; les récepteurs peuvent déterminer si les problèmes sont locaux, régionaux ou globaux; les gestionnaires de réseau peuvent utiliser des contrôleurs indépendants du profil qui ne reçoivent que les paquets RTCP et pas les paquets de données RTP correspondants pour évaluer la performance de leurs réseaux concernant la distribution en mode multidiffusion.

Les comptes totaux sont utilisés à la fois dans les informations d'émetteur et dans les blocs de rapport de récepteur pour calculer les différences entre deux rapports quelconques afin de pouvoir effectuer des mesures à la fois sur des périodes courtes et sur des périodes longues, et afin de fournir une certaine robustesse contre la perte d'un rapport. La différence entre les deux derniers rapports reçus peut servir à évaluer la qualité récente de la distribution. L'horodate NTP est incluse de façon à pouvoir calculer les taux à partir de ces différences sur l'intervalle séparant deux rapports. Cette horodate étant indépendante de la fréquence d'horloge pour le codage des données, il est possible d'implémenter des contrôleurs de la qualité indépendants du codage et du profil.

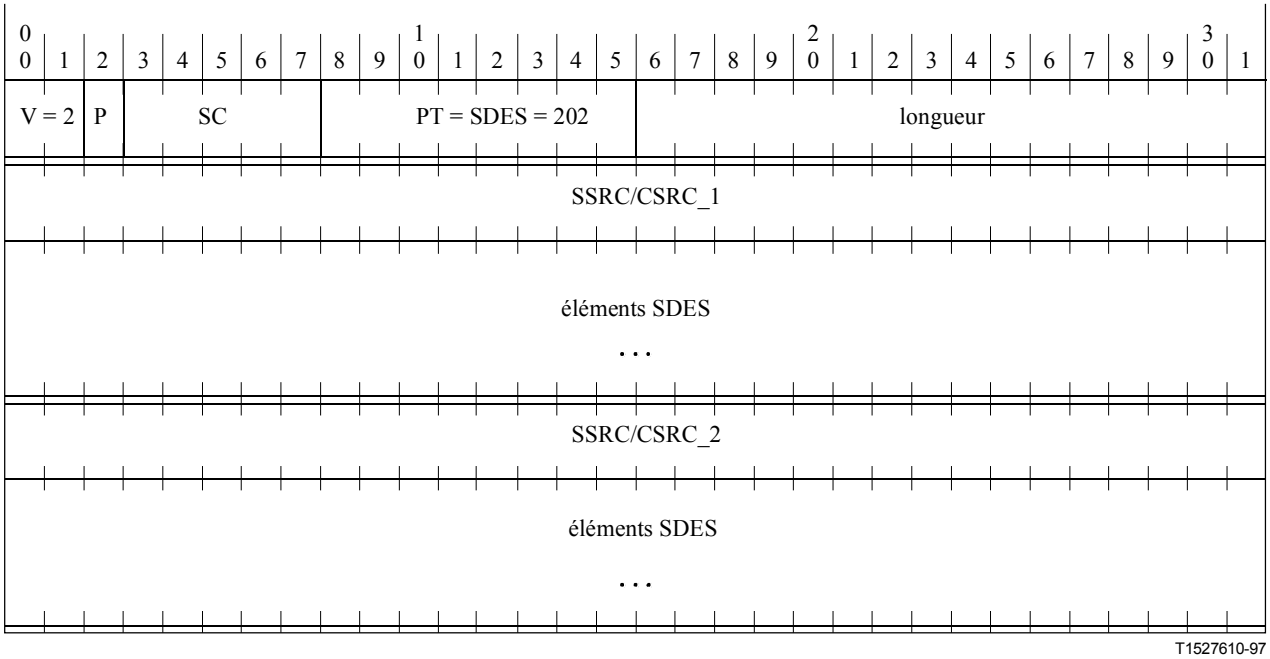
Comme exemple de calcul, citons le taux de paquets perdus sur l'intervalle séparant deux rapports de réception. La différence entre les nombres totaux de paquets perdus donne le nombre de paquets perdus pendant l'intervalle. La différence entre les derniers numéros de séquence étendus reçus donne le nombre de paquets prévus pendant l'intervalle. Le rapport entre ces deux nombres correspond à la fraction de paquets perdus sur l'intervalle. Ce rapport doit être égal au champ fraction perdue si les deux rapports de réception sont consécutifs, mais pas dans les autres cas. Le taux de perte par seconde peut être obtenu en divisant la fraction de perdus par la différence des horodates NTP, exprimées en secondes. Le nombre de paquets reçus est égal au nombre de paquets prévus moins le nombre de paquets perdus. Le nombre de paquets prévus peut aussi servir à évaluer la validité statistique de toute estimation de perte. Par exemple, un paquet perdu sur cinq a moins de sens que 200 paquets perdus sur 1000.

Grâce aux informations d'émetteur, un contrôleur tiers peut calculer le débit moyen des données de charge utile et le débit moyen des paquets sur un intervalle sans recevoir les données. Le rapport entre ces deux débits donne la taille moyenne de la charge utile. Si l'on peut supposer que la perte de paquets est indépendante de la taille de paquet, le produit entre le nombre de paquets reçus par un récepteur donné et la taille moyenne de charge utile (ou la taille de paquet correspondante) donne le débit apparent au niveau de ce récepteur.

Outre les comptes totaux qui permettent de mesurer la perte de paquets sur des longues périodes à partir des différences entre les rapports, le champ fraction perdue fournit une mesure sur une courte période à partir d'un seul rapport. Cela devient plus important lorsque la taille d'une session augmente tellement que les informations d'état de réception ne peuvent pas être conservées pour tous les récepteurs ou lorsque l'intervalle entre les rapports devient si long que seul un rapport peut avoir été reçu en provenance d'un récepteur donné.

Le champ gigue interarrivée fournit une deuxième mesure sur une courte période de l'encombrement du réseau. La perte de paquets permet de repérer les encombrements persistants alors que la mesure de gigue permet de repérer les encombrements transitoires. La mesure de gigue peut signaler un encombrement avant que celui-ci ne conduise à une perte de paquets. Etant donné que le champ gigue interarrivée n'est qu'un instantané de la gigue au moment d'un rapport, il peut être nécessaire d'analyser un certain nombre de rapports d'un même récepteur au cours du temps ou un certain nombre de rapports de plusieurs récepteurs, par exemple à l'intérieur d'un même réseau.

A.6.4 SDES: paquet RTCP de description de source



Le paquet SDES est une structure à trois niveaux constituée d'un en-tête et de zéro, une ou plusieurs bribes, chacune étant composée d'éléments décrivant la source identifiée dans cette bribe. Les éléments sont décrits séparément dans des sous-paragraphes ultérieurs.

version (V), remplissage (P), longueur: ces champs sont identiques à ceux décrits pour le paquet SR (voir A.6.3.1, SR: paquet RTCP rapport d'émetteur).

type de paquet (PT): 8 bits. Ce champ contient la constante 202 indiquant qu'il s'agit d'un paquet RTCP SDES.

compte de la source (SC): 5 bits. Il s'agit du nombre de bribes SSRC/CSRC figurant dans ce paquet SDES. La valeur zéro est valable mais inutile.

Chaque bribe est constituée d'un identificateur SSRC/CSRC suivi d'une liste de zéro, un ou plusieurs éléments, comportant des informations concernant la source SSRC/CSRC. Chaque bribe commence sur une frontière à 32 bits. Chaque élément comprend: un champ de type sur 8 bits, un compte des octets sur 8 bits indiquant la longueur du texte (non compris ces deux octets d'en-tête), et le texte proprement dit. On notera que la longueur du texte ne peut pas être supérieure à 255 octets, mais cela est cohérent avec la nécessité de limiter la consommation de largeur de bande RTCP.

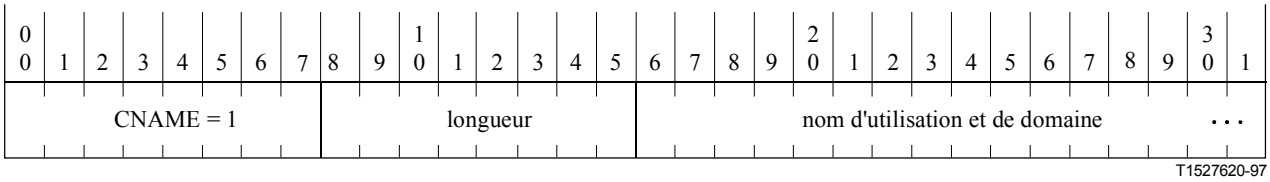
Le texte est codé conformément au codage UTF-2 spécifié dans l'Annexe F de l'ISO/CEI 10646-1 [A-10]. Ce codage est également connu sous les appellations suivantes: UTF-8 et UTF-FSS. Il est décrit dans "File System Safe UCS Transformation Format (FSS_UTF)", spécification préliminaire X/Open, document numéro P316 et rapport technique unicode n° 4. US-ASCII est une partie de ce codage qui ne nécessite aucun codage supplémentaire. La présence de codages multi-octets est indiquée par le positionnement du bit le plus significatif d'un caractère sur la valeur un.

Les éléments sont contigus, c'est-à-dire qu'ils ne sont pas complétés chacun par des bits de remplissage jusqu'à une frontière à 32 bits. Le texte n'est pas terminé par des octets vides car certains codages multi-octets incluent des octets vides. La liste des éléments de chaque bribe se termine par un ou plusieurs octets vides, le premier de ces octets étant interprété comme le type d'élément zéro qui indique la fin de la liste, le reste servant de remplissage jusqu'à la frontière à 32 bits suivante. Une bribe avec zéro élément (quatre octets vides) est valable mais inutile.

Les systèmes d'extrémité envoient un seul paquet SDES contenant leur propre identificateur de source (le même que l'identificateur SSRC dans l'en-tête RTP fixe). Un mélangeur envoie un seul paquet SDES contenant une bribe pour chaque source contributive de laquelle il reçoit des informations SDES, ou plusieurs paquets SDES complets dans le format indiqué ci-dessus si le nombre de sources contributives est supérieur à 31 (voir A.2.3, Mélangeur et traducteurs).

Les éléments SDES définis ici sont décrits dans les sous-paragraphes suivants. Seul l'élément CNAME est obligatoire. Certains éléments présentés ici peuvent être utiles uniquement pour certains profils, mais les types d'élément sont tous attribués à partir d'un espace commun pour promouvoir l'utilisation en partage et pour simplifier les applications indépendantes du profil. Il est possible de définir des éléments supplémentaires dans un profil, il suffit pour cela d'enregistrer les numéros de type auprès de l'IANA.

A.6.4.1 CNAME: élément SDES identificateur d'extrémité canonique



L'identificateur CNAME a les propriétés suivantes:

- comme l'identificateur SSRC attribué de façon aléatoire peut changer en cas de découverte d'un conflit ou de redémarrage d'un programme, l'élément CNAME doit assurer le lien entre l'identificateur SSRC et un identificateur pour la source qui reste constant;
- tout comme l'identificateur SSRC, l'identificateur CNAME doit aussi être unique parmi tous les participants à une même session RTP;
- pour assurer un lien entre les divers outils multimédia utilisés par un même participant dans un ensemble de sessions RTP connexes, le CNAME doit être fixe pour ce participant;
- afin de faciliter le contrôle par un tiers, le CNAME doit pouvoir être utilisé aussi bien par un programme que par une personne pour localiser la source.

Le CNAME doit donc être dérivé algorithmiquement et non pas entré manuellement, lorsque c'est possible. Pour répondre à ces prescriptions, il convient d'utiliser le format donné ci-après sauf si un profil spécifie une autre syntaxe ou une autre sémantique. L'élément CNAME doit avoir le format "utilisateur@serveur", ou "serveur" s'il n'y a pas de nom d'utilisateur comme dans les systèmes à un seul utilisateur. Pour les deux formats, "serveur" est soit le nom entier du domaine du serveur à partir duquel proviennent les données en temps réel, formatées conformément aux règles énoncées dans les documents RFC 1034 [A-11], RFC 1035 [A-12] et au 2.1/RFC 1123 [A-13]; soit la représentation ASCII normalisée de l'adresse numérique du serveur à l'interface utilisée pour la communication RTP. Par exemple, la représentation ASCII normalisée d'une adresse IP Version 4 est "décimale pointée", on dit aussi "dotted quad" en anglais. On prévoit que d'autres types d'adresse auront des représentations ASCII qui seront globalement uniques. Le nom entier du domaine est plus pratique pour un observateur humain et peut éviter d'avoir à envoyer un élément NAME en plus, mais il peut être difficile voire impossible à obtenir de façon fiable dans certains environnements. Dans ces environnements, les applications doivent plutôt utiliser la représentation ASCII de l'adresse.

Pour un système multi-utilisateurs, on peut citer en exemple "doe@sleepy.megacorp.com" ou "doe@192.0.2.89". Pour un système sans nom d'utilisateur, les exemples précédents deviennent "sleepy.megacorp.com" et "192.0.2.89".

Le nom d'utilisateur doit revêtir une forme qu'un programme comme "finger" ou "talk" puisse utiliser, c'est-à-dire que généralement il s'agira du nom d'entrée plutôt que du nom personnel. Le nom de serveur n'est pas nécessairement identique à celui qui figure dans l'adresse électronique du participant.

Cette syntaxe ne fournira pas un identificateur unique pour chaque source si une application autorise un utilisateur à générer plusieurs sources à partir d'un seul serveur. Pour une telle application, il faudra avoir recours à l'identificateur SSRC pour pouvoir identifier une source donnée, ou le profil de cette application devra spécifier une syntaxe supplémentaire pour l'identificateur CNAME.

Si chaque application crée son CNAME de façon indépendante, les CNAME résultants pourront ne pas être identiques à ceux requis pour assurer un lien entre les divers outils multimédia appartenant à un participant dans un ensemble de sessions RTP connexes. Si le lien entre les divers média est requis, il peut être nécessaire qu'un outil de coordination configure de façon externe le CNAME de chaque outil avec la même valeur. Les concepteurs d'application doivent savoir que l'attribution d'adresses de réseau privées comme l'attribution Net-10 proposée dans le document RFC 1597 [A-14] peut créer des adresses de réseau qui ne sont pas globalement uniques. On aurait alors des CNAME qui ne sont pas uniques si des serveurs avec des adresses privées et ne présentant pas de connectivité IP directe avec le réseau Internet public transmettaient leurs paquets RTP au réseau Internet public via un traducteur au niveau RTP. (Voir aussi le document RFC 1627 [A-15].) Pour traiter ce cas, les applications peuvent fournir un moyen de configurer un CNAME unique, mais la responsabilité repose sur le traducteur qui doit traduire les CNAME d'adresses privées en adresses publiques, lorsqu'il est nécessaire de ne pas exposer les adresses privées.

A.6.4.2 NAME: élément SDES nom d'utilisateur

Voir Appendice I.

A.6.4.3 EMAIL: élément SDES adresse électronique

Voir Appendice I.

A.6.4.4 PHONE: élément SDES numéro de téléphone

Voir Appendice I.

A.6.4.5 LOC: élément SDES situation géographique de l'utilisateur

Voir Appendice I.

A.6.4.6 TOOL: élément SDES nom d'application ou d'outil

Voir Appendice I.

A.6.4.7 NOTE: élément SDES avis/état

Voir Appendice I.

A.6.4.8 PRIV: élément SDES extensions privées

Voir Appendice I.

A.6.5 BYE: paquet RTCP au revoir

Voir Appendice I.

A.6.6 APP: paquet RTCP défini par l'application

Voir Appendice I.

A.7 Traducteurs et mélangeurs RTP

Outre les systèmes d'extrémité, le protocole RTP prend en charge des "traducteurs" et des "mélangeurs", qui peuvent être considérés comme des "systèmes intermédiaires" au niveau RTP. Cette prise en charge ajoute une certaine complexité au protocole, mais des expériences réalisées avec des applications audio et vidéo en mode multidiffusion sur Internet ont clairement démontré que ces fonctions sont nécessaires. Les exemples d'utilisation de traducteurs et de mélangeurs donnés dans le présent sous-paragraphe reposent sur la présence de barrières et de connexions à faible largeur de bande, qui risque de perdurer.

A.7.1 Description générale

Un traducteur/mélangeur RTP raccorde plusieurs "nuages" au niveau transport. Généralement, chaque nuage est défini par un protocole de réseau et de transport commun (IP/UDP par exemple), une adresse de multidiffusion ou deux adresses de monodiffusion, et un accès de destination au niveau transport. (Les traducteurs de protocole au niveau réseau, IP version 4 à IP version 6 par exemple, peuvent figurer dans un nuage de manière invisible pour le protocole RTP.) Un système peut servir de traducteur ou de mélangeur pour un certain nombre de sessions RTP, mais chacun est considéré comme une entité distincte sur le plan logique.

Afin d'éviter la création d'une boucle lorsqu'un traducteur ou un mélangeur est installé, il faut respecter les règles suivantes:

- chacun des nuages raccordés par des traducteurs et des mélangeurs participant à une session RTP doivent soit différer de tous les autres par au moins un de leurs paramètres (protocole, adresse, accès), soit être isolés des autres au niveau réseau;
- une règle découlant de la première est qu'il ne doit pas y avoir plusieurs traducteurs ou mélangeurs raccordés en parallèle sauf si, par un certain arrangement, ils créent une partition de l'ensemble des sources à transmettre.

D'une manière similaire, tous les systèmes d'extrémité RTP qui peuvent communiquer par l'intermédiaire d'un ou plusieurs traducteurs ou mélangeurs RTP utilisent le même espace SSRC, c'est-à-dire que les identificateurs SSRC doivent être uniques parmi tous ces systèmes d'extrémité. Le sous-paragraphe A.8.2 (Résolution des collisions et détection des boucles) décrit l'algorithme de résolution des collisions qui permet d'assurer l'unicité des identificateurs SSRC et de détecter les boucles.

Il peut exister de nombreuses variétés de traducteurs et de mélangeurs conçus à différentes fins et pour diverses applications: par exemple pour ajouter ou supprimer un chiffrement, pour modifier le codage des données des protocoles sous-jacents, ou encore pour transformer une adresse de multidiffusion en une ou plusieurs adresses de monodiffusion. La différence entre un traducteur et un mélangeur est qu'un traducteur traite séparément les flux de données provenant de plusieurs sources, alors qu'un mélangeur combine ces flux de données pour ne former qu'un seul nouveau flux de données:

traducteur: il transmet les paquets RTP en gardant intact leur identificateur SSRC; les récepteurs peuvent ainsi identifier chaque source même si les paquets provenant de toutes les sources passent dans le même traducteur et comportent l'adresse de source de réseau du traducteur. Certains types de traducteurs transmettront les données sans les modifier, alors que d'autres pourront modifier le codage des données et donc l'horodate et le type de charge utile de données RTP. Si plusieurs paquets de données sont codés à nouveau pour ne former plus qu'un paquet, ou inversement, le traducteur doit attribuer de nouveaux numéros de séquence aux paquets sortants. Des pertes au niveau du flux de paquets entrant peuvent provoquer des trous correspondants dans les numéros de séquence en sortie. Les récepteurs ne peuvent pas détecter la présence d'un traducteur sauf s'ils connaissent par un autre moyen l'adresse de transport ou le type de charge utile qui a été utilisé par la source d'origine;

mélangeur: il reçoit les flux de paquets de données RTP provenant d'une ou plusieurs sources, modifie éventuellement le format des données, combine les flux d'une certaine manière et transmet ensuite le flux combiné. Etant donné que le rythme entre les diverses sources d'entrée ne sera généralement pas synchronisé, le mélangeur fera des ajustements de rythme entre les flux et générera son propre rythme pour le flux combiné, il s'agit donc de la source de synchronisation du flux combiné. Tous les paquets de données transmis par un mélangeur seront donc marqués de l'identificateur SSRC propre au mélangeur. Afin de conserver l'identité des sources d'origine qui ont contribué au paquet mélangé, le mélangeur doit insérer leurs identificateurs SSRC dans la liste d'identificateurs CSRC venant à la suite de l'en-tête RTP fixe du paquet. Un mélangeur qui joue également le rôle de source contributive pour un certain paquet doit inclure explicitement son propre identificateur SSRC dans la liste CSRC correspondant à ce paquet.

Pour certaines applications, il peut être accepté qu'un mélangeur n'identifie pas les sources dans la liste CSRC au prix d'une incapacité à détecter des boucles faisant intervenir ces sources.

Pour des applications de type audio par exemple, l'avantage d'un mélangeur par rapport à un traducteur réside dans le fait que la largeur de bande de sortie est limitée à celle d'une source, même lorsque plusieurs sources sont actives du côté entrée. Cela peut être important pour les liaisons à faible largeur de bande. L'inconvénient est que les récepteurs du côté sortie ne commandent pas les sources qui sont transmises et celles qui sont écartées, sauf si un mécanisme est implémenté pour la commande à distance du mélangeur. La régénération des informations de synchronisation par les mélangeurs signifie aussi que les récepteurs ne peuvent pas faire de synchronisation intermédia des flux de données d'origine. Un mélangeur multimédia pourrait le faire.

La Figure A.3 illustre l'effet d'un ensemble de mélangeurs et de traducteurs sur les identificateurs SSRC et CSRC. Sur la figure, les systèmes d'extrémité (E) sont représentés par des rectangles, les traducteurs (T) par des losanges et les mélangeurs (M) par des ovales. La notation "M1:48 (1, 17)" désigne un paquet provenant du mélangeur M1, identifié par l'identificateur SSRC (aléatoire) du mélangeur M1 valant 48 et par les deux identificateurs CSRC, 1 et 17, qui sont les identificateurs SSRC des paquets venant de E1 et de E2.

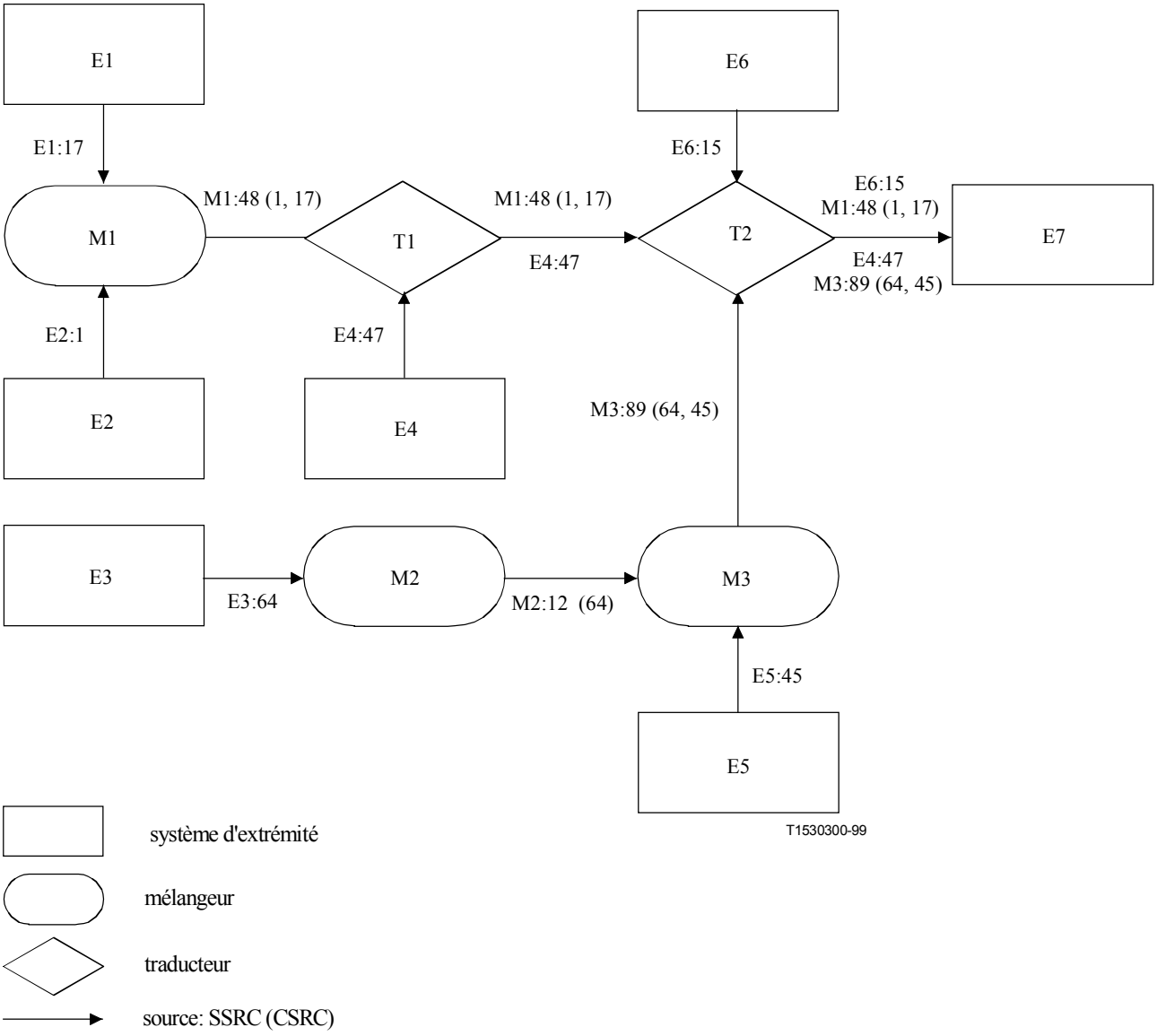


Figure A.3/H.225.0 – Exemple de réseau RTP avec des systèmes d'extrémité, des mélangeurs et des traducteurs

A.7.2 Traitement RTCP dans les traducteurs

Outre la transmission des paquets de données, éventuellement modifiés, les traducteurs et les mélangeurs doivent aussi traiter les paquets RTCP. Dans de nombreux cas, ils décomposeront les paquets RTCP composés reçus en provenance de systèmes d'extrémité afin de regrouper les informations SDES et de modifier les paquets SR ou RR. La retransmission de ces informations peut être déclenchée par l'arrivée de paquets ou à la fin d'une temporisation d'intervalle RTCP du traducteur ou du mélangeur.

Un traducteur qui ne modifie pas les paquets de données, par exemple un traducteur qui ne fait que transformer une adresse de multidiffusion en adresses de monodiffusion, peut tout simplement transmettre les paquets RTCP sans les modifier non plus. Un traducteur qui transforme la charge utile doit effectuer les transformations correspondantes dans les informations SR et RR de sorte que celles-ci continuent à refléter les caractéristiques des données et la qualité de réception. Ces traducteurs ne doivent pas se contenter de transmettre les paquets RTCP. En général, un traducteur ne doit pas regrouper en un seul paquet les paquets SR et RR provenant de plusieurs sources car cela réduirait la précision des mesures de temps de transmission basées sur les champs LSR et DLSR.

informations d'émetteur SR: un traducteur ne génère pas ses propres informations d'émetteur, mais transmet les paquets SR envoyés par un nuage aux autres nuages. L'identificateur SSRC est laissé intact mais les informations d'émetteur doivent être modifiées si c'est nécessaire pour la traduction. Si un traducteur modifie le codage des données, il doit modifier le champ "compte des octets de l'émetteur". S'il combine plusieurs paquets de données en un seul paquet de sortie, il doit modifier le champ "compte des paquets de l'émetteur". S'il modifie la fréquence d'horodatage, il doit modifier le champ "horodate RTP" du paquet SR.

blocs de rapport de réception SR/RR: un traducteur transmet les rapports de réception envoyés par un nuage aux autres nuages. On notera que ces rapports vont dans le sens opposé à celui des données. L'identificateur SSRC est laissé intact. Si un traducteur combine plusieurs paquets de données en un seul paquet de sortie, et que, par conséquent, il modifie les numéros de séquence, il doit faire la manipulation inverse pour les champs de perte de paquets et le champ "dernier numéro de séquence étendu", ce qui peut être compliqué. Dans le cas extrême, il peut ne pas y avoir de manière sensée de traduire les rapports de réception, auquel cas le traducteur peut ne pas transmettre de rapport de réception du tout ou transmettre un rapport synthétique sur la base de sa propre réception. La règle générale consiste à faire ce qui a un sens pour une traduction donnée.

Un traducteur ne nécessite pas d'identificateur SSRC propre, mais il peut choisir d'en attribuer un s'il veut envoyer des rapports concernant ce qu'il a reçu. Ces rapports seraient envoyés à tous les nuages raccordés, chaque rapport correspondant à la traduction du flux de données envoyé à un nuage, étant donné que les rapports de réception sont normalement multidiffusés à tous les participants.

SDES: les traducteurs transmettent généralement sans modification les informations SDES envoyées par un nuage aux autres nuages, mais ils peuvent par exemple décider de filtrer les informations SDES autres que les informations de CNAME si la largeur de bande est limitée. Il faut transmettre les CNAME pour que la détection des collisions d'identificateurs SSRC puisse fonctionner. Un traducteur qui génère ses propres paquets RR doit envoyer les informations SDES CNAME le concernant aux mêmes nuages auxquels il envoie ces paquets RR.

BYE: les traducteurs transmettent les paquets BYE sans les modifier. Les traducteurs possédant leur propre identificateur SSRC doivent générer des paquets BYE comportant cet identificateur SSRC s'ils sont sur le point d'arrêter de transmettre des paquets.

APP: les traducteurs transmettent les paquets APP sans les modifier.

A.7.3 Traitement RTCP dans les mélangeurs

Etant donné qu'un mélangeur génère un nouveau flux de données, il ne transmet pas les paquets SR ou RR mais il génère de nouvelles informations dans les deux sens.

informations d'émetteur SR: un mélangeur ne transmet pas les informations d'émetteur provenant des sources qu'il mélange car les caractéristiques des flux de données de la source sont perdues dans le mélange. En tant que source de synchronisation, le mélangeur génère ses propres paquets SR contenant les informations d'émetteur concernant le flux de données mélangé et les envoie dans le même sens que le flux de données mélangé.

blocs de rapport de réception SR/RR: un mélangeur génère ses propres rapports de réception pour les sources de chaque nuage et ne les envoie qu'au nuage correspondant. Il n'envoie pas ces rapports de réception aux autres nuages et ne transmet pas non plus les rapports de réception provenant d'un nuage aux autres nuages car les sources ne sont pas des SSRC (uniquement des CSRC).

SDES: les mélangeurs transmettent généralement sans modification les informations SDES envoyées par un nuage aux autres nuages, mais ils peuvent par exemple décider de filtrer les informations SDES autres que les informations de CNAME si la largeur de bande est limitée. Il faut transmettre les CNAME pour que la détection des collisions d'identificateurs SSRC puisse fonctionner. (Un identificateur d'une liste CSRC généré par un mélangeur peut entrer en collision avec un identificateur SSRC généré par un système d'extrémité.) Un mélangeur doit envoyer les

informations SDES CNAME le concernant aux mêmes nuages auxquels il envoie les paquets SR ou RR.

Etant donné que les mélangeurs ne transmettent pas les paquets SR ou RR, ils extrairont généralement les paquets SDES d'un paquet RTCP composé. Pour réduire au minimum le préfixe, des bribes de paquets SDES peuvent être regroupées en un seul paquet SDES qui est ensuite empilé sur un paquet SR ou RR provenant du mélangeur. Le débit de paquets RTCP peut être différent de chaque côté du mélangeur.

Un mélangeur qui n'insère pas d'identificateur CSRC peut aussi s'abstenir de transmettre les informations SDES CNAME. Dans ce cas, les espaces d'identificateur SSRC des deux nuages sont indépendants. Comme il a été mentionné plus haut, ce mode de fonctionnement crée le danger d'une incapacité à détecter des boucles.

BYE: il est nécessaire que les mélangeurs transmettent les paquets BYE. Ils doivent générer des paquets BYE avec leurs propres identificateurs SSRC s'ils sont sur le point d'arrêter de transmettre des paquets.

APP: le traitement des paquets APP par les mélangeurs est propre à l'application.

A.7.4 Mélangeurs en cascade

Une session RTP peut faire intervenir tout un ensemble de mélangeurs et de traducteurs comme le montre la Figure A.3. Si deux mélangeurs sont en cascade, comme M2 et M3 sur la figure, les paquets reçus par un mélangeur peuvent déjà avoir été mélangés et peuvent comporter une liste CSRC de plusieurs identificateurs. Le deuxième mélangeur doit établir la liste CSRC pour le paquet sortant à partir des identificateurs CSRC des paquets d'entrée déjà mélangés et des identificateurs SSRC des paquets d'entrée non mélangés. Cette liste est montrée en sortie du mélangeur M3 appelé M3:89 (64, 45) sur la Figure A.3. Comme dans le cas de mélangeurs qui ne sont pas en cascade, si la liste CSRC résultante comporte plus de 15 identificateurs, les identificateurs en surplus ne peuvent pas être inclus.

A.8 Attribution et utilisation des identificateurs SSRC

L'identificateur SSRC figurant dans l'en-tête RTP et dans divers champs des paquets RTCP est un nombre aléatoire sur 32 bits qui doit absolument être unique sur le plan global dans une session RTP donnée. Il est très important que ce nombre soit choisi avec soin afin que les participants qui sont sur le même réseau ou qui démarrent en même temps ne risquent pas de choisir le même nombre.

Il ne suffit pas d'utiliser l'adresse de réseau local (une adresse IPv4 par exemple) pour l'identificateur car l'adresse peut ne pas être unique. Etant donné que les traducteurs et les mélangeurs RTP permettent l'interfonctionnement de plusieurs réseaux avec des espaces d'adresses différents, les schémas d'attribution des adresses dans deux espaces pourraient conduire à un taux de collision beaucoup plus élevé que si l'attribution était aléatoire.

Plusieurs sources fonctionnant sur un même serveur entreraient également en conflit.

Il ne suffit pas d'obtenir un identificateur SSRC simplement en invoquant la fonction de randomisation sans en initialiser l'état avec soin. Le sous-paragraphe A.8.2 présente un exemple de procédure pour générer un identificateur aléatoire.

A.8.1 Probabilité de collision

Etant donné que les identificateurs sont choisis au hasard, il est possible que plusieurs sources choisissent le même nombre. La probabilité de collision est la plus élevée lorsque toutes les sources démarrent simultanément, par exemple lorsqu'elles sont déclenchées automatiquement par un événement de gestion de session. Si N est le nombre de sources et L la longueur de l'identificateur

(ici 32 bits), la probabilité pour que deux sources tirent de façon indépendante la même valeur peut être approchée pour N grand [20] par $1 - e^{-\frac{N^2}{2^{(L+1)}}}$. Pour N = 1000, la probabilité vaut environ 10^{-4}.

La probabilité de collision est généralement beaucoup plus faible que celle indiquée ci-dessus qui correspond au cas le plus défavorable. Lorsqu'une nouvelle source rejoint une session RTP dans laquelle toutes les autres sources ont déjà un identificateur unique, la probabilité de collision correspond à la fraction de nombres utilisés en dehors de l'espace. Si N est de nouveau le nombre de sources et L la longueur de l'identificateur, la probabilité de collision vaut $\frac{N}{2^L}$. Pour N = 1000, cette probabilité vaut environ $2 \cdot 10^{-7}$. La probabilité de collision est encore réduite si une nouvelle source reçoit des paquets des autres participants avant d'envoyer son propre paquet (de données ou de commande). Si la nouvelle source garde la trace des autres participants (par l'identificateur SSRC), alors, avant d'envoyer son premier paquet, elle peut vérifier que son identificateur n'entre en conflit avec aucun de ceux qu'elle a reçus; si ce n'est pas le cas, elle en choisit un autre.

A.8.2 Résolution des collisions et détection des boucles

Bien que la probabilité de collision d'identificateurs SSRC soit faible, toutes les implémentations du protocole RTP doivent pouvoir détecter les collisions et prendre des mesures appropriées pour les résoudre. Si une source découvre à un instant quelconque qu'une autre source utilise le même identificateur SSRC que le sien, elle doit envoyer un paquet RTCP BYE pour l'ancien identificateur et en choisir un autre au hasard. Si un récepteur découvre que deux autres sources sont en collision, il peut conserver les paquets de l'une et ignorer les paquets de l'autre lorsque cette collision peut être détectée par des adresses de transport de source ou des CNAME différents. On attend des deux sources qu'elles résolvent la collision afin que la situation ne s'éternise pas.

Etant donné que l'unicité globale des identificateurs aléatoires est assurée pour chaque session RTP, ces identificateurs peuvent être utilisés pour détecter les boucles qui peuvent être engendrées par les mélangeurs et les traducteurs. Une boucle entraîne la duplication des informations de données et de commande, soit non modifiées soit éventuellement mélangées, comme dans les exemples suivants:

- un traducteur peut transmettre un paquet par erreur au groupe de multidiffusion qui lui a envoyé le paquet, soit directement soit par l'intermédiaire d'une chaîne de traducteurs. Dans ce cas, le même paquet apparaît plusieurs fois, provenant de différentes sources de réseau;
- deux traducteurs mis en parallèle par erreur, c'est-à-dire avec les mêmes groupes de multidiffusion, transmettent tous les deux les paquets d'un groupe de multidiffusion à un autre. Les traducteurs unidirectionnels produiront deux copies; les traducteurs bidirectionnels forment une boucle;
- un mélangeur peut former une boucle en envoyant les paquets qu'il reçoit d'une destination de transport à cette même destination, soit directement soit par l'intermédiaire d'un autre mélangeur ou traducteur. Dans ce cas, une source peut apparaître à la fois comme une source SSRC dans un paquet de données et comme une source CSRC dans un paquet de données mélangé.

Une source peut découvrir une boucle concernant ses propres paquets ou les paquets d'une autre source (boucle causée par un tiers). Les boucles et les collisions produites par le choix aléatoire d'un identificateur de source se traduisent par des paquets arrivant avec le même identificateur SSRC mais avec une adresse de transport de source différente, qui peut être celle du système d'extrémité ayant produit ce paquet ou celle d'un système intermédiaire. Par conséquent, si une source change d'adresse de transport, elle doit aussi choisir un nouvel identificateur SSRC pour éviter d'être interprétée comme une source bouclée. Les boucles ou les collisions se produisant du côté distant d'un traducteur ou d'un mélangeur ne peuvent pas être détectées au moyen de l'adresse de transport de source si toutes les copies des paquets passent dans le traducteur ou le mélangeur; toutefois, les

collisions peuvent continuer à être détectées lorsque des bribes provenant de deux paquets RTCP SDES contiennent le même identificateur SSRC mais des CNAME différents.

Pour détecter et résoudre ces conflits, les implémentations du protocole RTP doivent intégrer un algorithme similaire à celui décrit ci-dessous. Cet algorithme ignore les paquets qui proviennent d'une nouvelle source ou d'une boucle et qui entrent en collision avec une source établie. Il résout les collisions avec l'identificateur SSRC du participant en envoyant un paquet RTCP BYE pour l'ancien identificateur et en choisissant un nouvel identificateur. Toutefois, lorsque la collision provient d'une boucle causée par les propres paquets du participant, l'algorithme ne choisira un nouvel identificateur qu'une seule fois et ignorera ensuite les paquets dont l'adresse est l'adresse de transport de la source bouclante. Cela est nécessaire pour éviter une avalanche de paquets BYE.

Dans cet algorithme, l'adresse de transport de source doit être la même pour les paquets RTP et RTCP provenant d'une même source. L'algorithme devra être modifié pour prendre en charge les applications qui ne satisfont pas cette contrainte.

Dans cet algorithme, il est nécessaire de conserver un tableau indexé par les identificateurs de source et contenant l'adresse de transport de source à partir de laquelle l'identificateur a été reçu (la première fois), ainsi qu'un autre état pour cette source. Chaque identificateur SSRC ou CSRC reçu dans un paquet de données ou de commande est recherché dans ce tableau en vue du traitement de ces informations de données ou de commande. Pour les paquets de commande, chaque élément avec son propre identificateur SSRC, par exemple une bribe SDES, nécessite une recherche distincte. (La présence d'un identificateur SSRC dans un bloc de rapport de réception est une exception.) Si l'identificateur SSRC ou CSRC n'est pas trouvé, une nouvelle rubrique est créée. Une rubrique du tableau est supprimée lorsqu'est reçu un paquet RTCP BYE contenant l'identificateur SSRC correspondant, ou après un temps relativement long pendant lequel aucun paquet n'est arrivé (voir A.6.2.1, Mise à jour du nombre de membres de la session).

Pour repérer les boucles causées par les paquets de données du participant, il est également nécessaire de conserver une liste séparée des adresses de transport de source (et pas des identificateurs) pour lesquelles on a repéré un conflit. On notera que cette liste doit être courte et la plupart du temps vide. Chaque élément de cette liste comporte l'adresse de source ainsi que l'instant auquel a été reçu le dernier paquet conflictuel. Un élément peut être supprimé de la liste lorsqu'aucun paquet conflictuel n'est arrivé en provenance de cette source pendant une période de l'ordre de 10 intervalles de rapport RTCP (voir A.6.2, Intervalle de transmission RTCP).

Dans l'algorithme présenté, on suppose que l'état et l'identificateur de source du participant figurent dans le tableau des identificateurs de source. L'algorithme pourrait être restructuré pour qu'il commence par vérifier si l'identificateur de source du participant figure dans le tableau.

si l'identificateur SSRC ou CSRC n'est pas trouvé dans le tableau des identificateurs de source:

alors créer une nouvelle rubrique contenant l'adresse de transport de source et l'identificateur SSRC ou CSRC ainsi qu'un autre état.

continuer le traitement normal.

(L'identificateur est trouvé dans le tableau.)

si l'adresse de transport de source indiquée dans le paquet correspond à l'adresse enregistrée dans le tableau pour cet identificateur:

alors continuer le traitement normal.

(Une collision d'identificateurs ou une boucle est indiquée.)

si l'identificateur de la source n'est pas celui du participant:

alors si l'identificateur de la source provient d'une bribe RTCP SDES contenant un élément CNAME qui est différent du CNAME figurant dans la rubrique du tableau:

alors (option) il s'agit d'une collision causée par un tiers.

sinon (option) il s'agit d'une boucle causée par un tiers.

abandonner le traitement du paquet de données ou de l'élément de commande.

(Il s'agit d'une collision ou d'une boucle causée par les propres données du participant.)

si l'adresse de transport de source est trouvée dans la liste des adresses conflictuelles:

alors si l'identificateur de source ne provient pas d'une bribe RTCP SDES contenant un élément CNAME, **ou** si ce CNAME est celui du participant:

alors (option) il s'agit d'une occurrence de bouclage du trafic du participant. Inscrire l'heure courante dans une rubrique de la liste des adresses conflictuelles.

abandonner le traitement du paquet de données ou de l'élément de commande.

Journaliser l'occurrence d'une collision.

Créer une nouvelle rubrique dans la liste des adresses conflictuelles et inscrire l'heure courante.

Envoyer un paquet RTCP BYE avec l'ancien identificateur SSRC.

Choisir un nouvel identificateur.

Créer dans le tableau des identificateurs de source une nouvelle rubrique comportant l'ancien identificateur SSRC ainsi que l'adresse de transport de source associée au paquet en cours de traitement.

continuer le traitement normal.

Dans cet algorithme, les paquets associés à l'adresse d'une source qui vient juste d'entrer en conflit seront ignorés et les paquets provenant de la source d'origine seront conservés. (Si la source d'origine passe par un mélangeur et si plus tard la même source est reçue directement, il peut être judicieux que le récepteur commute sauf si cela entraîne la perte d'autres sources du mélange.) Si aucun paquet n'arrive en provenance de la source d'origine pendant une longue période, la rubrique du tableau arrivera en fin de temporisation et la nouvelle source pourra prendre le relais. Cela se produira si la source d'origine détecte la collision et change d'identificateur de source, mais dans le cas général, un paquet RTCP BYE sera reçu en provenance de la source d'origine, ce qui permet de supprimer l'état sans avoir à attendre la fin de la temporisation.

Lorsqu'un nouvel identificateur SSRC est choisi par suite d'une collision, l'identificateur candidat doit d'abord être recherché dans le tableau des identificateurs de source pour voir s'il est déjà utilisé par une autre source. Si c'est le cas, il faut générer un autre identificateur candidat et répéter le processus.

Une boucle de paquets de données en direction d'une destination de multidiffusion peut causer un grave encombrement du réseau. Il est nécessaire que tous les mélangeurs et traducteurs implémentent un algorithme de détection des boucles – comme celui qui est décrit ici – de façon à pouvoir rompre les boucles. Cet algorithme doit limiter l'excès de trafic à une quantité au plus égale au trafic d'origine, ce qui permet de poursuivre la session afin de pouvoir trouver ce qui a créé la boucle. Toutefois, dans les cas extrêmes où un mélangeur ou un traducteur ne rompt pas correctement la boucle, entraînant des niveaux de trafic élevés, il peut être nécessaire que les systèmes d'extrémité cessent toute transmission de paquets de données ou de commande. Cette décision peut dépendre de l'application. Une condition d'erreur doit être indiquée, le cas échéant. La transmission peut être tentée à nouveau périodiquement après une longue durée aléatoire (de l'ordre de quelques minutes).

A.9 Sécurité

On trouvera à l'Appendice I des renseignements sur certaines méthodes de sécurité Internet. La confidentialité et les méthodes d'échange de clé H.323 sont décrites dans la Recommandation H.323.

A.10 Protocole RTP au-dessus des protocoles de réseau et de transport

Le présent sous-paragraphe traite de problèmes propres à l'acheminement de paquets RTP dans des protocoles de réseau et de transport particuliers. Les règles suivantes s'appliquent sauf si elles sont remplacées par des définitions propres au protocole qui sortent du cadre de la présente spécification.

Le protocole RTP s'appuie sur le ou les protocoles sous-jacents pour assurer le démultiplexage des flux de données RTP et des flux de commande RTCP. Pour le protocole UDP et les protocoles similaires, le numéro d'accès RTP est un numéro pair et le numéro d'accès RTCP correspondant est le numéro (impair) qui vient juste après. Si le numéro d'accès RTP fourni pour une application est impair, l'application doit remplacer ce numéro par le numéro (pair) qui le suit immédiatement.

Les paquets de données RTP ne comportent ni champ de longueur ni autre délimitation, le protocole RTP attend donc du ou des protocoles sous-jacents qu'ils fournissent une indication de longueur. La longueur maximale des paquets RTP n'est limitée que par les protocoles sous-jacents.

Si des paquets RTP sont inclus dans un flux d'octets continu plutôt que dans des messages (paquets) au niveau du protocole sous-jacent, il faut définir une encapsulation des paquets RTP pour fournir un mécanisme de tramage. Le tramage est également nécessaire si le remplissage est possible au niveau du protocole sous-jacent de sorte que la longueur de la charge utile RTP ne puisse pas être déterminée. Le mécanisme de tramage n'est pas défini ici.

Un profil peut spécifier une méthode de tramage à utiliser même lorsque des paquets RTP sont transportés dans des protocoles qui ne fournissent pas de tramage afin que plusieurs paquets RTP puissent rentrer dans une seule unité de données protocolaire de couche inférieure, un paquet UDP par exemple. Si plusieurs paquets RTP peuvent être contenus dans un même paquet de réseau ou de transport, le préfixe d'en-tête est réduit et la synchronisation entre les différents flux peut être simplifiée.

A.11 Récapitulatif des constantes protocolaires

Le présent sous-paragraphe récapitule sommairement les constantes définies dans la présente spécification.

Les constantes de type de charge utile (PT, *payload type*) RTP sont définies dans les profils plutôt que dans la présente Recommandation. Toutefois, il ne faut pas attribuer les valeurs réservées 200 et 201 (en décimal) à l'octet de l'en-tête RTP qui contient le ou les bits de marqueur ainsi que le type de charge utile pour que les paquets RTP puissent être distingués des types de paquet RTCP SR et RR dans la procédure de validation d'en-tête décrite au A.6.3. Pour la définition normalisée donnée dans la présente spécification (un bit de marqueur et 7 bits de type de charge utile), cette restriction signifie que les types de charge utile 72 et 73 sont réservés.

A.11.1 Types de paquet RTCP

Abréviations	Nom	Valeur
SR	Rapport d'émetteur	200
RR	Rapport de récepteur	201
SDES	Description de source	202
BYE	Au revoir	203
APP	Défini par l'application	204

Ces valeurs de type ont été choisies dans l'intervalle 200 à 204 afin d'améliorer le contrôle de validité d'en-tête des paquets RTCP par rapport aux paquets RTP ou aux autres paquets non connexes. Lorsque le champ de type de paquet RTCP est comparé à l'octet correspondant de l'en-tête RTP, cet intervalle correspond à un bit de marqueur valant 1 (ce qui n'est généralement pas le cas pour les paquets de données) et à un bit le plus élevé du champ de type de charge utile normalisé valant 1 (étant donné que les types de charge utile statiques sont généralement définis dans la moitié inférieure). Cet intervalle a également été choisi de façon à être numériquement éloigné des valeurs 0 et 255 étant donné que les séquences ne comportant que des zéros et que des uns sont des séquences de données courantes.

Etant donné que tous les paquets RTCP composés doivent commencer par un paquet SR ou RR, on a choisi pour les codes de ces types de paquet un couple pair/impair pour permettre au contrôle de validité RTCP de tester le plus grand nombre de bits par rapport à un masque et à une valeur.

D'autres constantes sont attribuées par l'IANA. Les personnes réalisant des expériences sont invitées à enregistrer les numéros dont elles ont besoin pour leurs expériences, et à radier ensuite les numéros qui se révèlent non nécessaires.

A.11.2 Types d'éléments SDES

Abréviations	Nom	Valeur
END	Fin de liste SDES	0
CNAME	Nom canonique	1
NAME	Nom d'utilisateur	2
EMAIL	Adresse électronique de l'utilisateur	3
PHONE	Numéro de téléphone de l'utilisateur	4
LOC	Situation géographique de l'utilisateur	5
TOOL	Nom d'application ou d'outil	6
NOTE	Avis concernant la source	7
PRIV	Extensions privées	8

D'autres constantes sont attribuées par l'IANA. Les personnes réalisant des expériences sont invitées à enregistrer les numéros dont elles ont besoin pour leurs expériences, et à radier ensuite les numéros qui se révèlent non nécessaires.

A.12 Spécifications de profil et de format de charge utile RTP

La spécification complète du protocole RTP pour une application donnée nécessitera un ou plusieurs documents associés de deux types décrits ici: spécifications de profil et de format de charge utile.

Le protocole RTP peut être utilisé pour diverses applications ayant des prescriptions qui diffèrent quelque peu. La flexibilité d'adaptation à ces prescriptions est assurée par les diverses possibilités de choix dans la partie principale de la spécification de protocole et par la possibilité de définir, dans un document de profil distinct, des extensions concernant un environnement particulier et une classe d'applications particulière. Une application ne fonctionnera généralement que conformément à un seul profil, le profil utilisé n'est donc pas indiqué explicitement. On trouvera à l'Annexe B un profil pour les applications audio et vidéo.

La spécification d'un format de charge utile, qui est le second type de document associé, définit la manière dont un type particulier de données de charge utile, des données vidéo codées H.261 par exemple, doit être transporté dans le protocole RTP. Le titre de ces documents est du type "Format de charge utile RTP pour le codage audio/vidéo XYZ". Etant donné qu'on peut avoir besoin d'un format de charge utile sous divers profils, on peut définir les formats indépendamment du profil. Les

documents de profil doivent alors spécifier un mappage par défaut entre les formats et des valeurs de type de charge utile, si nécessaire. (Voir Annexe C.)

Dans la présente spécification, les éléments suivants ont été identifiés comme pouvant être définis dans un profil, mais cette liste ne se veut pas exhaustive:

en-tête de données RTP: l'octet de l'en-tête de données RTP qui contient le bit de marqueur et le champ de type de charge utile peut être redéfini par un profil pour répondre à des prescriptions différentes, par exemple une prescription avec un nombre de bits de marqueur différent (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

types de charge utile: en supposant qu'un champ de type de charge utile est présent, le profil définira généralement un ensemble de formats de charge utile (des codages de média par exemple) et un mappage statique par défaut entre ces formats et les valeurs de type de charge utile. Certains formats de charge utile peuvent être définis par référence à des spécifications de format de charge utile séparées. Pour chaque type de charge utile défini, le profil doit spécifier la fréquence de l'horloge d'horodatage RTP à utiliser (voir A.5.1, Champs de l'en-tête fixe RTP).

ajouts à l'en-tête de données RTP: des champs supplémentaires peuvent être rattachés à l'en-tête de données RTP fixe si une fonctionnalité supplémentaire indépendante du type de charge utile est requise par une classe d'applications conformes à un profil donné (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

extensions de l'en-tête de données RTP: il faut définir le contenu des 16 premiers bits de la structure d'extension de l'en-tête de données RTP si l'utilisation de ce mécanisme pour des extensions propres à l'implémentation est accordée par le profil (voir A.5.3, Modifications de l'en-tête RTP propres au profil).

types de paquets RTCP: de nouveaux types de paquet RTCP propres à une classe d'applications peuvent être définis et enregistrés auprès de l'IANA.

intervalle de rapport RTCP: un profil doit spécifier s'il sera fait usage des valeurs proposées au A.6.2, Intervalle de transmission RTCP pour les constantes servant au calcul de l'intervalle de rapport RTCP. Il s'agit de la fraction RTCP de largeur de bande de la session, de l'intervalle de rapport minimal et du partage de la largeur de bande entre émetteurs et récepteurs. Un profil peut spécifier d'autres valeurs s'il a été démontré qu'elles pouvaient être évolutives.

extension SR/RR: il est possible de définir une section d'extension pour les paquets RTCP SR et RR si des informations supplémentaires concernant l'émetteur ou les récepteurs doivent faire l'objet de rapports réguliers (voir A.6.3.3, Extension des rapports d'émetteur et de récepteur).

utilisation d'éléments SDES: le profil peut spécifier des rangs de priorité relatifs pour les éléments RTCP SDES devant être transmis ou exclus (voir A.6.2.2, Attribution de largeur de bande pour la description de source); une autre syntaxe ou une autre sémantique pour l'élément CNAME (voir A.6.4.1, CNAME: élément SDES identificateur d'extrémité canonique); le format de l'élément LOC (voir A.6.4.5, LOC: élément SDES situation géographique de l'utilisateur); la sémantique et l'utilisation de l'élément NOTE (voir A.6.4.7, NOTE: élément SDES avis/état); ou de nouveaux types d'élément SDES à enregistrer auprès de l'IANA.

sécurité: un profil peut spécifier les algorithmes et les services de sécurité qui doivent être offerts par les applications, et peut fournir des indications sur leur utilisation (voir A.9, Sécurité).

mappage chaîne-clé: un profil peut spécifier le mappage entre un mot de passe fourni par l'utilisateur et une clé de chiffrement.

protocole sous-jacent: il peut être nécessaire d'utiliser un protocole particulier de couche Transport ou réseau sous-jacent pour acheminer les paquets RTP.

mappage de transport: il est possible de spécifier un mappage entre d'une part les protocoles RTP et RTCP et d'autre part les adresses au niveau transport (accès UDP par exemple) qui soit différente du mappage normalisé défini à l'Annexe B.

encapsulation: il est possible de définir une encapsulation des paquets RTP pour pouvoir transporter plusieurs paquets de données RTP dans un seul paquet de couche inférieure ou pour fournir un tramage au-dessus des protocoles sous-jacents qui ne fournissent pas de tramage (voir A.10, Protocole RTP au-dessus des protocoles de réseau et de transport).

A.13 Algorithmes

On trouvera le présent sous-paragraphe à l'Appendice I. Tous ces exemples d'implémentations ne sont donnés qu'à titre d'information et ne sont donc pas inclus ici.

A.14 Bibliographie

On notera que les documents cités dans la présente bibliographie ne sont donnés qu'à titre d'information, et qu'ils ne sont pas requis pour la mise en application de la présente annexe.

[A-1] CLARK (D.D.), TENNENHOUSE (D.L.): Architectural considerations for a new generation of protocols, *SIGCOMM Symposium on Communications Architectures and Protocols,* Philadelphie, Pennsylvanie, p. 200-208, *IEEE*, septembre 1990, Computer Communications Review, Vol. 20 (4), septembre 1990.

[A-2] COMER (D.E.): Internetworking with TCP/IP, Vol. 1, *Prentice Hall*, Englewood Cliffs, New Jersey 1991.

[A-3] POSTEL (J.): Internet protocol, RFC 791, *Internet Engineering Task Force*, septembre 1981.

[A-4] MILLS (D.): Network time protocol (v3), RFC 1305, *Internet Engineering Task Force*, avril 1992.

[A-5] EASTLAKE (D.), CROCKER (S.), SCHILLER (J.): Randomness recommendations for security, RFC 1750, *Internet Engineering Task Force*, décembre 1994.

[A-6] BOLOT (J.-C.), TURLETTI (T.), WAKEMAN (I.): Scalable feedback control for multicast video distribution in the internet, *SIGCOMM Symposium on Communications Architectures and Protocols*, p. 58-67, ACM, Londres, août 1994.

[A-7] BUSSE (I.), DEFFNER (B.), SCHULZRINNE (H.): Dynamic QOS control of multimedia applications based on RTP, *Computer Communications*, janvier 1996.

[A-8] FLOYD (S.), JACOBSON (V.): The synchronization of periodic routing messages, *SIGCOMM Symposium on Communications Architectures and Protocols* (D. P. Sidhu, ed.,), p. 33-44, ACM, (San Francisco Californie) septembre 1993.

[A-9] CADZOW (J. A.): Foundations of digital signal processing and data analysis, *Macmillan*, New York 1987.

[A-10] ISO/CEI 10646-1:1993, *Technologies de l'information – Jeu universel de caractères codés à plusieurs octets – Partie 1: Architecture et table multilingue*.

[A-11] MOCKAPETRIS (P.): Domain names – Concepts and facilities, STD 13, RFC 1034, *Internet Engineering Task Force*, novembre 1987.

[A-12] MOCKAPETRIS (P.): Domain names – Implementation and specification, STD 13, RFC 1035, *Internet Engineering Task Force*, novembre 1987.

[A-13] BRADEN (R.): Requirements for internet hosts – Application and support, STD 3, RFC 1123, *Internet Engineering Task Force*, octobre 1989.

[A-14] REKHTER (Y.), MOSKOWITZ (R.), KARRENBERG (D.), de GROOT (G.): Address allocation for private internets, RFC 1597, *Internet Engineering Task Force*, mars 1994.

[A-15] LEAR (E.), FAIR (E.), CROCKER (D.), KESSLER (T.): Network 10 considered harmful (some practices should not be codified), RFC 1627, *Internet Engineering Task Force*, juillet 1994.

[A-16] CROCKER (D.): Standard for the format of ARPA internet text messages, STD 11, RFC 822, *Internet Engineering Task Force*, août 1982.

[A-17] FELLER (W.): An Introduction to Probability Theory and its Applications, Vol. 1, *John Wiley and Sons*, troisième édition, New York 1968.

[A-18] BALENSON (D.): Privacy enhancement for internet electronic mail: Part III: algorithms, modes, and identifiers, RFC 1423, *Internet Engineering Task Force*, février 1993.

[A-19] VOYDOCK (V.L.), KENT (S.T.): Security mechanisms in high-level network protocols, *ACM Computing Surveys*, Vol. 15, p. 135-171, juin 1983.

[A-20] RIVEST (R.): The MD5 message-digest algorithm, RFC 1321, *Internet Engineering Task Force*, avril 1992.

ANNEXE B

Profil RTP

On se reportera à l'introduction de l'Annexe A; toutes les mises en garde contenues dans l'Annexe A sont également applicables à la présente annexe. On trouvera dans l'Appendice II, à titre d'information, la référence au document IETF complet; cependant, la présente annexe contient toutes les informations nécessaires à l'implémentation de la Recommandation H.323.

B.1 Introduction

Le présent profil définit certains aspects du protocole RTP non spécifiés dans l'Annexe A. Ce profil est destiné à être utilisé dans le cadre de conférences audio et vidéo avec gestion minimale des sessions. En particulier, aucune prise en charge pour la négociation des paramètres ou la gestion de participation n'est assurée. Ce profil est censé être utile dans les sessions où aucune négociation ou gestion de participation n'est utilisée (par exemple en utilisant les types de charge utile statiques et les indications d'appartenance fournies par le protocole RTCP), ce profil peut également être utile en association avec un protocole de commande de haut niveau.

L'utilisation du présent profil est valable pour certaines applications; il n'y a pas d'indication explicite par numéro d'accès, identificateur de protocole, etc.

D'autres profils peuvent reposer sur des choix différents pour les éléments spécifiés ici.

B.2 Formes de paquets RTP et RTCP et comportement des protocoles

Le présent sous-paragraphe "Profil RTP et spécification des formats de charge utile" donne la liste d'un nombre d'éléments qui peuvent être spécifiés ou modifiés à l'intérieur d'un profil. Le présent sous-paragraphe traite de ces éléments. En général, ce profil suit les aspects par défaut ou recommandés de la spécification RTP.

en-tête de données RTP: le format normalisé de l'en-tête de données RTP fixe est utilisé (un bit marqueur).

types de charge utile: les types de charge utile statiques sont définis au B.6. Définitions des types de charge utile.

addition aux en-têtes de données RTP: aucun champ supplémentaire n'est ajouté à l'en-tête de données RTP.

extensions des en-têtes de données RTP: aucune extension d'en-tête RTP n'est définie, mais les applications fonctionnant dans le cadre de ce profil peuvent utiliser de telles extensions d'en-tête. Ainsi, dans les applications, on ne supposera pas que le bit X est toujours égal à zéro et ces applications devront pouvoir ignorer l'extension d'en-tête. Si une extension d'en-tête est définie dans le futur, cette définition devra spécifier le contenu des 16 premiers bits de manière telle que plusieurs extensions différentes puissent être identifiées.

types de paquet RTCP: aucun type additionnel de paquet RTCP n'est défini dans la présente spécification de profil.

intervalle de rapport RTCP: constantes proposées, servant au calcul de l'intervalle entre rapports RTCP.

extension SR/RR: aucune section d'extension n'est définie pour les paquets SR ou RR du protocole RTCP.

utilisation SDES: les applications peuvent utiliser tout élément SDES décrit. L'information CNAME est envoyée à chaque intervalle de rapport, en revanche, les autres éléments doivent être envoyés seulement tous les cinq intervalles de rapport.

sécurité: les services de sécurité par défaut RTP ne sont pas les services par défaut dans le présent profil.

mappage chaîne-touche: voir Appendice II concernant ces informations.

protocole sous-jacent: la présence d'un protocole sous-jacent est autorisée et décrit dans l'Appendice IV sous réserve du respect de certaines conditions.

mappage de transport: le mappage normalisé du RTP et RTCP avec les adresses niveau de transport est utilisé.

encapsulation: aucune encapsulation des paquets RTP n'est spécifiée.

B.3 Types de charge utile

Voir Appendice II pour des renseignements concernant l'enregistrement des nouveaux types de charge utile.

Il convient de noter que tous les codages à utiliser par le RTP doivent se voir assigner un type de charge utile statique. Les moyens non-RTP hors du domaine d'application de la présente annexe (par exemple, pour les services d'annuaire et les protocoles d'invitation) peuvent être utilisés pour établir un mappage dynamique entre un type de charge utile extrait de la gamme 96-127 et un codage. Pour faciliter la tâche de la personne chargée de l'implémentation, le présent profil contient des descriptions des codages qui actuellement n'ont pas de type de charge utile statique qui leur est assigné.

L'espace pour les types de charge utile disponibles est relativement petit. Ainsi, les nouveaux types de charge utile statiques sont assignés seulement si les conditions suivantes sont satisfaites:

- le codage intéresse la communauté Internet dans son ensemble;
- il présente des avantages comparé au codage actuel ou est rendu nécessaire pour assurer l'interopérabilité avec les systèmes de conférence ou les systèmes multimédia existants, largement utilisés;
- la description est suffisante pour construire un décodeur.

B.4 Audio

B.4.1 Recommandations indépendantes du codage

Pour les applications qui n'envoient aucun paquet au cours des silences, le premier paquet de signal de parole (premier paquet après une période de silence) est repéré par la valeur du bit marqueur dans l'en-tête de données RTP. Pour les applications sans suppression de silence, le bit sera mis à zéro.

Le signal d'horloge RTP utilisé pour produire l'horodate RTP est indépendant du nombre de canaux et du codage; il est égal au nombre de périodes d'échantillonnage par seconde. Pour les codages de canal N, chaque période d'échantillonnage (par exemple, 1/8000 s) produit N échantillons. (Cette terminologie est normalisée, mais pas assez précise, étant donné que le nombre total d'échantillons par seconde est alors égal au taux d'échantillonnage que multiplie le décompte du canal.

Si l'on utilise plusieurs canaux audio, les canaux sont numérotés de gauche à droite, à partir de 1. Dans les paquets audio RTP, les informations provenant des canaux de numéro inférieur précèdent celles des canaux de numéro plus élevé.

Pour plus de deux canaux, la convention sera la suivante:

l gauche
r droit
c centre
S enviophonie
F avant
R arrière

Canaux	Description	Canaux					
		1	2	3	4	5	6
2	Stéréo	l	r				
3		l	r	c			
4	Quadriphonique	Fl	Fr	Rl	Rr		
4		l	c	r	S		
5		Fl	Fr	Fc	Sl	Sr	
6		l	lc	c	r	rc	S

Les échantillons pour tous les canaux appartenant à un même instant d'échantillonnage doivent se trouver dans le même paquet. L'entrelacement d'échantillons provenant des différents canaux dépend du codage. Des directives générales sont données au B.4.2.

La fréquence d'échantillonnage doit être extraite de l'ensemble 8000, 11 025, 16 000, 22 050, 24 000, 32 000, 44 100 et 48 000 Hz. (Les ordinateurs Apple Macintosh ont des fréquences d'échantillonnage propres de 22 254,54 et 11 127,27 qui peuvent être converties à 22 050 et 11 025 avec une qualité acceptable en éliminant 4 ou 2 échantillons dans une trame de 20 ms.) Cependant, la plupart des codages audio sont définis pour un ensemble plus restreint de fréquences d'échantillonnage. Les récepteurs doivent pouvoir accepter de l'information audio multicanal, mais peuvent choisir seulement de reproduire un seul canal.

Les recommandations ci-dessous sont des paramètres de fonctionnement par défaut. Les applications doivent pouvoir accepter d'autres valeurs. Les gammes indiquées ici sont simplement destinées à donner des indications aux rédacteurs d'applications, afin de permettre à un ensemble d'applications conformes à ces directives d'interfonctionner sans négociation supplémentaire. Ces directives ne sont pas destinées à limiter les paramètres de fonctionnement pour les applications qui peuvent négocier un jeu de paramètres "interfonctionnables", par exemple par l'intermédiaire d'un protocole de gestion de conférence.

Pour l'audio en mode paquet, l'intervalle de mise en paquets par défaut doit avoir une durée de 20 ms, sauf indication contraire lors de la description du codage. L'intervalle de mise en paquets détermine le délai minimal de bout en bout; les paquets plus longs donnent lieu à moins de préfixes d'en-tête mais à des délais plus élevés et rendront plus facilement repérable la perte de paquets. Pour les applications non interactives (par exemple, pour les cours ou les liaisons avec des contraintes sévères de largeur de bande) un délai de mise en paquets plus élevé peut être tout à fait approprié. Un récepteur doit accepter des paquets représentant de 0 à 200 ms de données audio. Cette restriction permet un dimensionnement raisonnable des tampons pour le récepteur.

B.4.2 Directives pour les codages audio à échantillonnage

Dans les codages à échantillonnage, chaque échantillon audio est représenté par un nombre fixe de bits. Dans les données audio comprimées, les codes pour chaque échantillon peuvent dépasser les limites d'octets. Un paquet audio RTP peut contenir un nombre quelconque d'échantillons audio, étant soumis à la contrainte que le nombre de bits par échantillon multiplié par le nombre d'échantillons par paquet donne un nombre entier d'octets. Les codages fractionnaires donnent moins d'un octet par échantillon.

La durée d'un paquet audio est déterminée par le nombre d'échantillons dans le paquet.

Pour les codages à échantillonnage produisant un ou plusieurs octets par échantillon, les échantillons provenant de différents canaux échantillonnés au même instant d'échantillonnage sont mis en paquets dans des octets consécutifs. Par exemple, pour un codage à deux canaux, la séquence d'octets est (canal de gauche, premier échantillon), (canal de droite, premier échantillon), (canal de gauche, deuxième échantillon), (canal de droite, deuxième échantillon) et ainsi de suite. Pour des codages multioctets, les octets sont transmis dans l'ordre d'octet du réseau (c'est-à-dire, l'octet de plus fort poids d'abord).

La mise en paquets des codages à échantillonnage produisant moins d'un octet par échantillon est propre au codage.

B.4.3 Directives pour les codages audio à trame

Les codages à trame codent un bloc de longueur fixe audio en un autre bloc de données comprimées, en général de longueur fixe aussi. Pour les codages à trame, l'expéditeur peut choisir de combiner plusieurs de ces trames en un seul message. Le récepteur peut indiquer le nombre de trames contenues dans un message étant donné que la durée de trame est définie dans le cadre du codage.

Pour les codecs à trame, l'ordre des canaux est défini pour l'ensemble du bloc. C'est-à-dire, pour l'audio à deux canaux, les échantillons droit et gauche sont codés de manière indépendante, la trame codée pour le canal de gauche précédant celle pour le canal de droite.

Tous les codecs audio à trame doivent être en mesure de coder et de décoder plusieurs trames consécutives dans un seul paquet. Comme la taille de trame pour les codecs à trame est donnée, il n'est pas nécessaire d'utiliser une désignation distincte pour le même codage, mais avec différents numéros de trame par paquet.

B.4.4 Codages audio

Les caractéristiques des codages audio normalisés sont indiquées dans le Tableau B.1 et leurs types de charge utile sont donnés dans le Tableau B.2.

Voir Appendice II pour des informations concernant les codages non indiqués dans le Tableau B.1. La prise en charge de ces codages ne fait pas partie de la Recommandation H.323.

Tableau B.1/H.225.0 – Caractéristiques des codages audio

Codage	Echantillon/trame	Bits/échantillon	ms/Trame
G722	Echantillon	8	
G728	Trame	Sans objet	2,5
PCMA	Echantillon	8	
PCMU	Echantillon	8	
G723	Trame	Sans objet	30
G729	Trame	Sans objet	10
GSM	Trame	Sans objet	20

B.4.4.1 Codage G722

Le codage G722 est spécifié dans la Recommandation G.722, "Codage audiofréquence à 7 kHz à un débit inférieur ou égal à 64 kbit/s".

B.4.4.2 Codage G728

Le codage G728 est défini dans la Recommandation G.728, "Codage de la parole à 16 kbit/s en utilisant la prédiction linéaire à faible délai avec excitation par code".

B.4.4.3 Codage MIC-A

Le codage MIC-A (codage MIC en loi A) est spécifié dans la Recommandation G.711. Les données audio sont codées à raison de huit bits par échantillon, après l'application d'une mise à l'échelle logarithmique.

B.4.4.4 Codage MIC-U

Le codage MIC (codage MIC en loi μ) est spécifié dans la Recommandation G.711. Les données audio sont codées sur huit bits par échantillon, après mise à l'échelle logarithmique.

B.5 Vidéo

Les codages vidéo ci-dessous sont actuellement définis, ainsi que leurs noms abrégés utilisés pour l'identification. Voir Appendice II pour tout codage non décrit ici. Ces codages ne font pas partie de la Recommandation H.323.

H261

Le codage est spécifié dans la Recommandation H.261. La mise en paquets et les propriétés spécifiques au RTP sont décrites dans l'Annexe C.

H263

Le codage est spécifié dans la Recommandation H.263. La mise en paquets et les propriétés spécifiques au RTP sont décrites dans l'Annexe E.

La procédure suivante doit être suivie par des entités H.323 souhaitant transmettre des flux vidéo H.263 (1996 ou 1998):

- dans un message **OpenLogicalChannel** H.245, un expéditeur qui souhaite utiliser le format de charge utile léguée, largement utilisé dans l'industrie pour les flux H.263 (1996), ne doit signaler que les éléments H.263 (1996) et doit omettre la mise en paquets multimédias des paramètres de voie logique H.225.0;

- dans un message **OpenLogicalChannel** H.245, un expéditeur qui souhaite utiliser le format de charge utile défini dans le commentaire RFC 2190 pour les flux H.263 (1996) doit spécifier comme suit le type de capacité utile par protocole RTP dans les paramètres de voie logique H.225.0: {rfc-number = 2190, payloadType = 34};
- en général, dans un message **OpenLogicalChannel** H.245, l'expéditeur doit spécifier le format de charge utile conformément à la sémantique définie dans la Recommandation H.245. Cette procédure doit en particulier être suivie pour signaler le format de charge utile H.263 + (1998) qui est défini dans l'Annexe A, ainsi que ses éventuels successeurs.

B.6 Définitions des types de charge utile

Le Tableau B.2 définit ces valeurs de type de charge utile statique du profil pour le champ PT de l'en-tête de données RTP. De plus, les valeurs de type de charge utile dans la plage 96-127 peuvent être définies de manière dynamique par l'intermédiaire du protocole de gestion de la conférence, qui sort du domaine d'application de la présente Recommandation. Par exemple, un directeur de session peut spécifier que pour une session donnée, le type de charge utile 96 indique un codage PCMU, une fréquence d'échantillonnage de 8000 Hz, 2 canaux. La gamme de types de charge utile indiquée "réservée" a été mise à part de sorte que les paquets RTCP et RTP peuvent être distingués de manière fiable (voir A.11, Récapitulatif des constantes protocolaires).

Une source RTP émet un seul type de charge utile RTP à un instant donné; l'entrelacement de plusieurs types de charge utile RTP en une session RTP n'est pas autorisé, mais plusieurs sessions RTP peuvent être utilisées en parallèle pour envoyer plusieurs médias. Les types de charge utile actuellement définis dans le présent profil acheminent soit de l'audio ou de la vidéo mais pas les deux. Cependant, on est autorisé à définir des types de charge utile qui combinent plusieurs supports, par exemple audio et vidéo, avec une distinction appropriée dans le format de charge utile. Les participants à une session décident au moyen de mécanismes hors du domaine d'application de la présente spécification, de l'ensemble de types de charge utile autorisé dans une session donnée. Cet ensemble peut, par exemple, être défini par les capacités des applications utilisées, négocié par un protocole de gestion de conférence ou être fixé par accord entre les participants.

Tous les codages vidéo actuels utilisent une fréquence d'horodatage de 90 000 Hz, la même que la fréquence d'horodatage de la présentation MPEG. Cette fréquence conduit à des incréments d'horodatage entiers exacts pour fréquence de trame type de 24 (TVHD), 25 (PAL) et 29,97 (NTSC) et 30 Hz (TVHD) et des fréquences images de 50, 59,94 et 60 Hz. La fréquence de 90 kHz est recommandée pour les futurs codages vidéo utilisés dans le cadre du présent profil, mais d'autres fréquences sont possibles. Cependant il n'est pas suffisant d'utiliser une fréquence de trame vidéo (en général comprise entre 15 et 30 Hz), car sa résolution adéquate ne correspond pas aux impératifs de synchronisation type, lorsqu'on calcule l'horodatage RTP correspondant à l'horodatage NTP dans un paquet SR RTPC (voir l'Annexe A). La résolution d'horodatage doit aussi être suffisante pour l'évaluation de gigue figurant dans les rapports de récepteur.

Les codages vidéo normalisés et leurs types de charge utile sont donnés dans le Tableau B.2.

Tableau B.2/H.225.0 – Types de charge utile (PT) pour les codages normalisés audio et vidéo

PT	Nom du codage	Audio/vidéo (A/V)	Fréquence d'horloge (Hz)	Canaux (audio)
0	PCMU	A	8 000	1
8	PCMA	A	8 000	1
9	G722	A	8 000	1
4	G723	A	8 000	1
15	G728	A	8 000	1
18	G729	A	8 000	1
31	H261	V	90 000	
34	H263	V	90 000	
3	GSM	A	8 000	1
96-127	dynamique	?		
NOTE – Les types de charge utile 1-7, 10-14, 16-30 et 30-95 sont réservés. Voir Appendice II pour de plus amples détails.				

B.7 Assignation des accès

Comme spécifié dans la définition du protocole RTP, les données RTP doivent être acheminées sur un numéro pair d'accès UDP et les paquets RTCP correspondants doivent être acheminés sur le numéro d'accès immédiatement supérieur (impair).

Les applications fonctionnant dans le cadre du présent profil peuvent utiliser toutes les paires d'accès UDP de ce type. Par exemple, la paire d'accès peut être attribuée de façon aléatoire par un programme de gestion de session. Une seule paire de numéros d'accès fixe ne peut pas être requise car les applications utilisant le présent profil fonctionneront vraisemblablement sur le même serveur, et il existera certains systèmes d'exploitation qui ne permettent pas à plusieurs processus d'utiliser le même accès UDP avec différentes adresses de multidiffusion.

Toutefois, les numéros d'accès 5004 et 5005 ont été enregistrés pour être utilisés dans le présent profil pour les applications qui choisissent de les utiliser comme paire par défaut. Les applications qui fonctionnent dans le cadre de plusieurs profils peuvent utiliser cette paire d'accès comme indication pour choisir le présent profil, si elles ne sont pas soumises aux contraintes décrites dans le précédent alinéa. Les applications doivent avoir une valeur par défaut et peuvent nécessiter la spécification explicite de la paire d'accès. Les numéros d'accès particuliers ont été choisis pour se trouver au-dessus de 5000 afin d'être compatibles avec la pratique d'attribution des numéros d'accès dans le cadre du système d'exploitation Unix, où les numéros d'accès inférieurs à 1024 ne peuvent être utilisés que par des processus privilégiés et les numéros d'accès compris entre 1024 et 5000 sont automatiquement assignés par le système d'exploitation.

ANNEXE C

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

On se reportera à l'Annexe A; toutes les mises en garde contenues dans l'Annexe A sont également applicables à la présente annexe. On trouvera dans l'Appendice III à titre d'information la référence du Document IETF complet; cependant, la présente annexe contient toutes les informations nécessaires à l'implémentation de la Recommandation H.323.

C.1 Introduction

La Recommandation H.261 [C-2] spécifie les codages utilisés par les codecs de visioconférence conformes aux normes de l'UIT-T. Bien que ces codages aient été à l'origine spécifiés pour des circuits du RNIS à débit de données fixe, les expériences ont montré qu'ils pouvaient être également utilisés sur des réseaux à commutation par paquets tel Internet.

L'objet de la présente annexe est de spécifier un format de charge utile RTP pour l'encapsulation des flux vidéo H.261 (voir l'Annexe A).

C.2 Structure du flux de paquets

C.2.1 Aperçu général de la Recommandation H.261

Le codage H.261 est organisé en hiérarchie de groupements. Le flux vidéo est composé de séquences d'images, ou trames, qui sont elles-mêmes organisées en ensembles de groupes de blocs (GOB, *group of blocks*). Il convient de noter que les "images" H.261 sont appelées "trames" dans la présente Recommandation. Chaque GOB contient trois lignes de 11 macroblocs (MB, *macro blocks*). Chaque macrobloc transporte des informations sur un groupe de 16×16 pixels: l'information de luminance est spécifiée pour 4 blocs de 8×8 pixels, tandis que l'information de chrominance est donnée par deux composantes de différence de couleur "rouge" et "bleu" avec une résolution de seulement 8×8 pixels. Ces composantes et les codecs représentant leurs valeurs échantillonnées sont définis dans la Recommandation UIT-R BT.601 [C-3].

Ce groupement est utilisé pour spécifier l'information à chaque niveau de la hiérarchie:

– au niveau de la trame, on spécifie l'information telle que le décalage temporel par rapport à la trame précédente, le format d'image et divers indicateurs;

– au niveau du groupe de blocs, on spécifie le numéro du groupe de blocs et le quantificateur par défaut qui sera utilisé pour les macroblocs;

– au niveau du macrobloc, on spécifie les blocs qui sont présents et qui ne seront pas modifiés, et facultativement un quantificateur et les vecteurs de mouvement.

Les blocs qui ont été modifiés sont codés par calcul de la transformée en cosinus discrète (DCT, *discrete cosine transform*) de leurs coefficients, qui sont ensuite quantifiés et codés avec un code Huffman (codes de longueur variable).

Le codage de Huffman H.261 inclut une séquence spéciale de début de groupe de blocs, composée de 15 zéros suivis d'un seul 1, qui ne peut pas être imitée par les autres mots de code. Cette séquence est incluse au début de chaque en-tête de groupe de blocs (et également au début de chaque en-tête de trame) pour indiquer la séparation entre deux groupes de blocs, et est en réalité utilisée pour indiquer que le groupe de blocs courant est terminé. Le codage inclut aussi une séquence de bourrage, composée de 7 zéros suivis par 4 "uns", cette séquence de bourrage peut être seulement introduite entre le codage de macroblocs ou juste avant le séparateur de groupe de blocs.

C.2.2 Mise en paquets

Les codecs H.261 conçus pour fonctionner sur des circuits RNIS produisent un flux binaire composé de plusieurs niveaux de codage spécifiés dans la Recommandation H.261 et les Recommandations associées. Les bits résultant du codage de Huffman sont disposés en trames de 512 bits, contenant 2 bits de synchronisation, 492 bits de données, 18 bits de code de correction d'erreur. Les trames de 512 bits sont ensuite entrelacées avec un flux audio et transmises sur des circuits $p \times 64$ kbit/s conformément à la Recommandation H.221 [C-1].

Dans le cas d'une transmission sur Internet, on prendra en considération la sortie du codage de Huffman. Tous les bits produits lors de l'étape de codage de Huffman seront inclus dans le paquet. Les trames de 512 bits ne seront pas transportées, étant donné que la protection contre les erreurs

binaires peut être obtenue par d'autres moyens. De même, les signaux audio et vidéo ne seront pas multiplexés dans les mêmes paquets, étant donné que les protocoles UDP et RTP disposent d'un moyen bien plus efficace de multiplexage.

La transmission directe du résultat du codage de Huffman sur un flux non fiable de datagrammes UDP se traduirait toutefois, par une très faible fiabilité. Le résultat de la structure hiérarchique du flux binaire H.261 est tel qu'il est nécessaire de recevoir l'information présente dans l'en-tête de trame pour décoder les groupes de blocs, ainsi que l'information présente dans l'en-tête du groupe de blocs pour décoder les macroblocs. Si aucune précaution n'était prise, cela signifierait qu'il faut recevoir tous les paquets qui transportent une image afin de décoder de manière convenable ses composantes.

Si chaque image pouvait être transportée dans un seul paquet, cette exigence ne créerait pas de problème. Cependant, une image vidéo ou même un groupe de blocs lui-même peut parfois être trop volumineux pour être contenu dans un seul paquet. Par conséquent, le macrobloc est pris comme unité de fragmentation. Les paquets doivent débuter et se terminer sur une limite de macrobloc, c'est-à-dire qu'un macrobloc ne peut pas être fragmenté sur plusieurs paquets. Plusieurs macroblocs peuvent être transportés sur un seul paquet lorsqu'ils s'insèrent convenablement dans leur taille maximale des paquets autorisés. Cette pratique est recommandée afin de réduire le débit de paquet à l'émission et les préfixes de paquet.

Pour permettre à chaque paquet d'être traité indépendamment en vue d'une resynchronisation efficace en présence de pertes de paquets, certaines informations d'état extraites de l'en-tête de trame et de l'en-tête du groupe de blocs sont acheminées avec chaque paquet pour permettre de décoder les macroblocs contenus dans le paquet. Cette information d'état inclut le numéro du groupe de blocs en vigueur au début du paquet, le prédicteur d'adresse de macrobloc (c'est-à-dire la dernière adresse MBA codée dans le précédent paquet), la valeur du quantificateur au début de ce paquet (GQUANT, MQUANT ou zéro dans le cas d'un commencement de groupe de blocs) et les données du vecteur de mouvement de référence (MVD, *motion vector data*) pour calculer les vraies données MVD contenues dans ce paquet. Le flux binaire ne peut être fragmenté entre un en-tête GOB et le macrobloc MB 1 de ce groupe de blocs.

De plus, comme le macrobloc comprimé peut ne pas être contenu dans un nombre entier d'octets, l'en-tête données contient deux entiers à trois bits, SBIT et EBIT, pour indiquer le nombre de bits inutilisés dans le premier et le dernier octet des données H.261 respectivement.

C.3 Spécification du système de mise en paquets

C.3.1 Utilisation du protocole RTP

L'information H.261 est transportée comme des données de charge utile à l'intérieur du protocole RTP. Les champs suivants de l'en-tête RTP sont spécifiés:

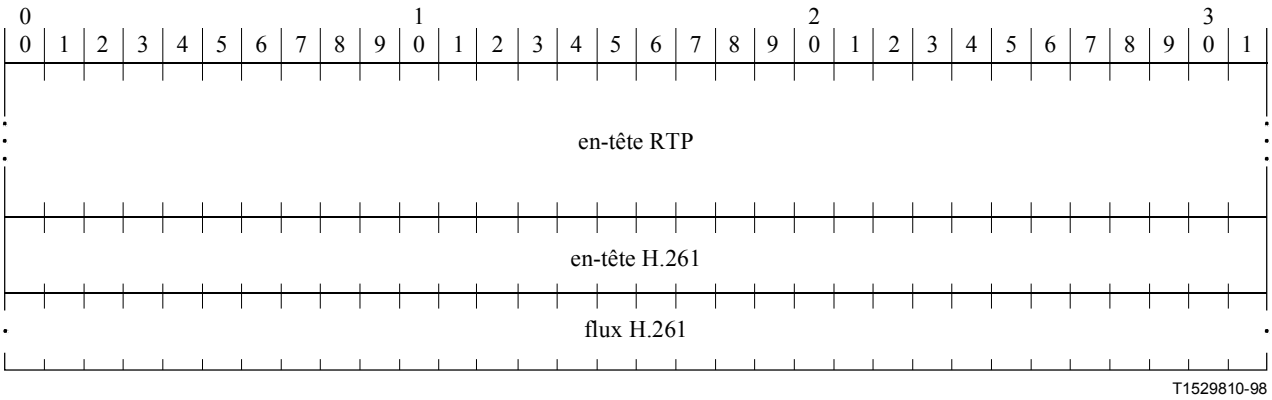
– le type de charge utile doit spécifier le format de la charge utile H.261 (voir Annexe B);

– l'horodate RTP code l'instant d'échantillonnage de la première image vidéo contenue dans le paquet de données RTP. L'horodate RTP devra être la même dans des paquets successifs lorsqu'une image vidéo de la même trame vidéo occupe plusieurs paquets. Pour les flux vidéo H.261, l'horodate RTP est établie à partir d'une horloge à 90 kHz. Cette fréquence d'horloge est un multiple de la fréquence de trame H.261 (c'est-à-dire 30 000/1001 ou approximativement 29,97 Hz). De cette manière, pour chaque instant image, l'horloge est simplement incrémentée par le multiple et ceci supprime l'imprécision dans le calcul de l'horodate. De plus, la valeur initiale de l'horodate est aléatoire (imprévisible) pouvant rendre plus difficiles les attaques en texte clair dans le chiffrement (voir RTP) (Annexe A). Il convient de noter que si plusieurs trames sont codées dans un paquet (par exemple lorsqu'il y a de faibles modifications entre deux images), il est nécessaire de calculer les temps d'affichage pour les trames après la première en utilisant l'information de synchronisation

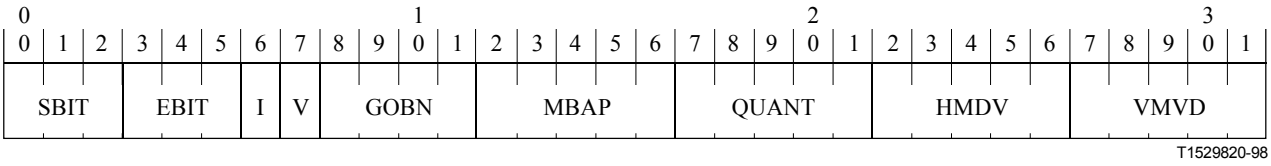
contenue dans l'en-tête de trame H.261. Cela est nécessaire car l'horodate RTP donne seulement le temps d'affichage de la première trame dans le paquet;

– le bit de marqueur de l'en-tête RTP est mis à "un" dans le dernier paquet d'une trame vidéo. Dans les autres cas il sera égal à "zéro". Ainsi, il n'est pas nécessaire d'attendre un paquet suivant (qui contient le code de démarrage qui termine la trame actuelle) pour savoir qu'une nouvelle trame doit être affichée.

Les données H.261 suivront l'en-tête RTP comme suit:



L'en-tête H.261 est défini comme suit:



Les champs dans l'en-tête H.261 ont les significations suivantes:

position bit de démarrage (SBIT, start bit position): 3 bits – Nombre de bits qui doivent être ignorés dans le premier octet de données.

position fin de bit (EBIT, end bit position): 3 bits – Nombre de bits qui doivent être ignorés dans le dernier octet de données.

données codées INTRA-trame (I): 1 bit – Mis à 1 si le flux contient seulement des blocs codés INTRA-trame. Mis à 0 si ce flux peut ou peut ne pas contenir des blocs codés INTRA-trame. La signification de ce bit peut ne pas changer au cours d'une même session.

drapeau vecteur de mouvement (V, motion vector flag): 1 bit – Mis à 0 si les vecteurs de mouvement ne sont pas utilisés dans ce flux. Mis à 1 si les vecteurs de mouvement peuvent ou peuvent ne pas être utilisés dans ce flux. La signification de ce bit peut ne pas changer au cours d'une même session.

numéro de GOB (GOBN): 4 bits – Code le numéro de groupe de bloc utilisé au début du paquet. Mis à 0 si le paquet commence avec un en-tête GOB.

prédicteur d'adresse de macrobloc (MBAP, macroblock address predictor): 5 bits – Code le prédicteur d'adresse de macrobloc (c'est-à-dire la dernière adresse MBA codée dans le paquet précédent). La valeur de ce prédicteur est comprise entre 0 et 32 (pour prévoir les adresses MBA valides 1-33), et comme ce flux binaire ne peut pas être fragmenté entre un en-tête GOB et un macrobloc 1, le prédicteur au début du paquet ne peut jamais avoir la valeur 0. Par conséquent, la gamme de valeurs est 1-32, qui est biaisée de –1 pour pouvoir être contenue dans 5 bits. Ainsi, si

MBAP = 0, la valeur du prédicteur MBA est 1. Il est mis à zéro si le paquet commence par un en-tête GOB.

quantificateur (QUANT): 5 bits – La valeur du quantificateur (MQUANT ou GQUANT) en cours avant le début du paquet. Mis à 0 si le paquet commence par un en-tête GOB.

données du vecteur de mouvement horizontal (HMVD, horizontal motion vector data): 5 bits – Données relatives au vecteur de mouvement horizontal de référence (MVD, *motion vector data*). Mis à 0 si le fanion V est à 0 ou si le paquet commence par un en-tête GOB. Les valeurs HMVD sont des nombres en complément à 2 sur 5 bits appartenant à l'intervalle [−16, +15], où −16 n'est pas utilisé.

données de vecteur de mouvement vertical (VMVD, vertical motion vector data): 5 bits – Données relatives au vecteur de mouvement vertical de référence (MVD). Mis à 0 si le fanion V est égal à 0 ou si le paquet commence par un en-tête GOB. Les valeurs VMVD sont des nombres en complément à 2 sur 5 bits appartenant à l'intervalle [−16, +15], où −16 n'est pas utilisé.

Il convient de noter que les fanions I et V sont des fanions dissimulés, c'est-à-dire qu'ils peuvent être déduits à partir du flux binaire. Ils sont inclus pour permettre aux décodeurs de procéder à des optimisations qui ne seraient pas possibles si ces fanions dissimulés n'étaient pas fournis avant le décodage du flux binaire. Par conséquent, ces bits ne peuvent pas être modifiés pendant la durée du flux. Une implémentation conforme peut toujours fixer V = 1 et I = 0.

Les données des vecteurs de mouvement horizontal et vertical doivent être mises à zéro lorsque le MTYPE du dernier macrobloc codé dans le paquet précédent était "mouvement non compensé".

C.3.2 Recommandations relatives au fonctionnement des codecs matériels

Les dispositifs de mise en paquets pour les codecs matériels peuvent de manière triviale être représentés en dehors des limites des groupes de blocs en utilisant la séquence de début de groupe de blocs incluse dans les données H.261. (Il convient de noter que les codeurs à logiciel connaissent déjà les limites.) La mise en paquets la moins coûteuse est celle qui consiste à effectuer cette opération au niveau du groupe de blocs pour tous les groupes de blocs qui peuvent être contenus dans un paquet. Cependant, lorsqu'un groupe de blocs est trop grand, le dispositif de mise en paquets doit l'examiner pour opérer une fragmentation en macroblocs. (Il convient de noter que seul le codage de Huffman peut être analysé et qu'il n'est pas nécessaire de décompresser totalement le flux, ainsi cela ne nécessite que relativement peu de traitement; des exemples d'implémentation se trouvent dans l'Appendice III. Il est recommandé d'utiliser la fragmentation au niveau des macroblocs lorsque cela est réalisable afin d'obtenir une mise en paquets plus efficace. Ce système de fragmentation réduit le débit de paquets en sortie et par conséquent réduit les préfixes.

Du côté du récepteur, le flux de données peut être dépaquetisé et acheminé à l'entrée d'un codec matériel. Si ce codec fonctionne à débit fixe, la synchronisation peut être maintenue en insérant une séquence de bourrage entre les macroblocs (c'est-à-dire entre les paquets) lorsque le débit d'arrivée des paquets est inférieur au débit.

C.3.3 Perte des paquets

Sur Internet, la plupart des pertes de paquets sont dues à l'encombrement du réseau et moins aux erreurs de transmission. Avec le protocole UDP, l'expéditeur ne dispose pas d'un mécanisme lui permettant de savoir si un paquet a été bien reçu. Il appartient à l'application, c'est-à-dire au codeur et au décodeur, de traiter la perte des paquets. Chaque paquet RTP inclut un champ numéro de séquence qui peut être utilisé pour détecter la perte de paquets.

La Recommandation H.261 utilise une redondance temporelle de la vidéo pour effectuer la compression. Ce codage différentiel (ou codage INTERtrame) est sensible à la perte de paquets. Après une perte de paquets, des parties de l'image peuvent rester erronées jusqu'à ce que les macroblocs correspondants aient été codés en mode INTRA-trame (c'est-à-dire codés indépendamment des trames antérieures). Il y a plusieurs façons de pallier la perte des paquets:

1) une des façons consiste à utiliser le codage INTRA-trame et le remplissage conditionnel au niveau du macrobloc. C'est-à-dire, seuls les macroblocs qui changent (au-delà d'un certain seuil) sont transmis;

2) une autre façon consiste à régler le débit de rafraîchissement du codage INTRA-trame en fonction de la perte de paquets observée par les récepteurs. La Recommandation H.261 spécifie qu'un macrobloc est codé INTRA-trame au moins toutes les 132 fois qu'il est transmis. Cependant, le taux de rafraîchissement INTRA-trame peut être augmenté afin d'accélérer la récupération lorsque le taux de perte mesuré est important;

3) la façon la plus rapide de corriger une image erronée est de demander le rafraîchissement d'image codée INTRA-trame après la détection d'une perte de paquets. Une façon de réaliser cette opération est pour le décodeur d'envoyer au codeur la liste des paquets perdus. Le codeur peut décider de coder chaque macrobloc de chaque groupe de blocs de la trame vidéo suivante dans le mode INTRA-trame (c'est-à-dire le codage INTRA-trame intégral), ou si le codeur peut déduire des numéros de séquence des paquets les macroblocs qui sont concernés par la perte, il peut économiser de la largeur de bande en envoyant seulement ces macroblocs dans le mode INTRA-trame. Ce mode est particulièrement efficace pour une connexion point à point ou lorsque le nombre de décodeurs est faible. Le sous-paragraphe suivant spécifie comment la fonction de rafraîchissement peut être implémentée.

C.3.4 Utilisation des paquets de commande H.261 spécifiques facultatifs

La présente spécification définit deux paquets de commande RTCP spécifiques à la Recommandation H.261, les paquets "Full INTRA-frame Request" (demande mode INTRA-trame intégral) et "Negative Acknowledgement" (accusé de réception négatif), décrits dans le présent sous-paragraphe. Leur objet est d'accélérer le rafraîchissement de la vidéo dans les situations où leur utilisation est réalisable. La prise en charge de ces paquets de commande spécifiques H.261 par l'expéditeur H.261 est facultative; en particulier, les premières expériences ont montré que l'utilisation de cette caractéristique pouvait avoir des effets négatifs lorsque le nombre de sites est très important. Aussi, ces paquets de commande doivent-ils être utilisés avec précaution.

Les paquets de commande spécifiques H.261 diffèrent des paquets normaux RTCP dans le sens où ils ne sont pas transmis vers l'adresse de transport de destination RTCP normale pour la session RTP (qui est souvent une adresse de multidiffusion). Ainsi, ces paquets de commande sont envoyés directement par monodiffusion du décodeur au codeur. L'accès de destination pour ces paquets de commande est le même accès que le codeur utilise comme accès source pour la transmission des paquets RTP (données). Par conséquent, ces paquets peuvent être considérés comme des paquets de commande "inverses".

En conséquence, ces paquets de commande peuvent seulement être utilisés lorsque aucun mélangeur ou expéditeur RTP n'intervient dans le trajet du codeur au décodeur. Si des systèmes intermédiaires de ce type venaient à intervenir, l'adresse du codeur ne serait plus présente comme adresse source au niveau réseau dans les paquets reçus par le décodeur, et en fait, il ne serait pas possible pour le décodeur d'envoyer des paquets directement au codeur.

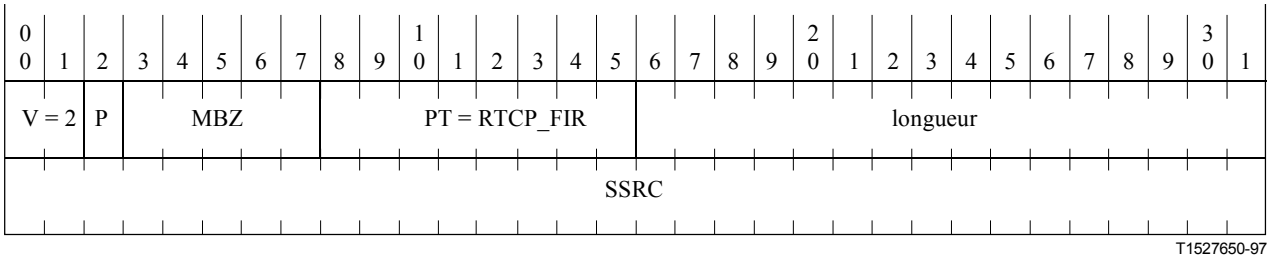
Certains protocoles de multidiffusion fiables utilisent des paquets de commande NACK similaires transmis suivant le canal de distribution de multidiffusion normal, mais ils utilisent en général des délais aléatoires pour éviter les problèmes liés à l'implosion de paquets NACK. L'objectif de ces protocoles est d'assurer une remise des paquets de multidiffusion fiable au détriment des délais, ce qui est tout à fait justifié pour des applications comme par exemple un tableau blanc utilisé en partage.

Par ailleurs, la transmission vidéo interactive est plus sensible au temps de transmission et n'exige pas une fiabilité totale. Pour les applications vidéo, il semble plus efficace d'envoyer des paquets de commande NACK dès que possible, c'est-à-dire dès qu'une perte est détectée, sans ajouter de délai aléatoire. Dans ce cas, la multidiffusion des paquets de commande NACK produirait un trafic inutile

entre les récepteurs étant donné que seul le codeur les utilisera. Mais cette méthode est seulement efficace lorsque le nombre de récepteurs est faible. Par exemple, si les paquets de commande spécifiques H.261 sont utilisés dans des connexions point à point ou dans des connexions point à multipoint lorsqu'il y a moins de 10 participants à la conférence.

C.3.5 Définition des paquets de commande

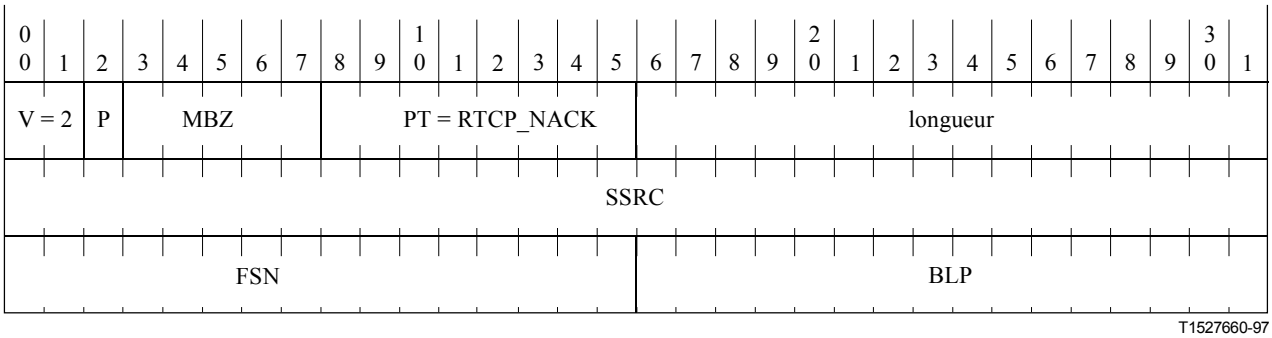
C.3.5.1 Paquet demande mode INTRA-trame intégral (FIR, *full INTRA-frame request*)



Ce paquet indique qu'un récepteur demande une image totalement codée afin de soit commencer le décodage avec une image entière, soit de rafraîchir cette image et d'accélérer la récupération après une rafale de paquets perdus. Le récepteur demande à la source de forcer l'image suivante à passer dans le mode de codage INTRA-trame intégral, c'est-à-dire sans utilisation du codage différentiel. Les différents champs sont définis dans la spécification du protocole RTP (Annexe A). L'identificateur SSRC est l'identificateur de source de synchronisation pour l'expéditeur de ce paquet. La valeur de l'identificateur de type de paquet (PT, *packet type*) est la constante RTCP_FIR (192).

C.3.5.2 Paquet accusé de réception négatif (NACK, *negative acknowledgements*)

Le format du paquet NACK est le suivant:



Les différents champs T, P, PT, longueur et SSRC sont définis dans la spécification du protocole RTP (Annexe A). La valeur de l'identificateur type de paquet (PT) est la constante RTCP_NACK (193). L'identificateur SSRC est l'identificateur de source de synchronisation pour l'expéditeur de ce paquet.

Les deux champs restants ont les significations suivantes:

premier numéro de séquence (FSN, first sequence number): 16 bits – Identifie le premier numéro de la séquence perdue;

masque binaire des paquets perdus suivants (BLP, bitmask of following lost packet): 16 bits – Un bit est mis à 1 si le paquet correspondant a été perdu et mis à 0 dans le cas contraire. Le champ BLP est mis à 0 si et seulement si aucun paquet autre que celui ayant fait l'objet d'un accusé de réception négatif (NACK) (en utilisant le champ FSN) a été perdu. Le champ BLP est mis à 0x00001 si le paquet correspondant au numéro FSN et le paquet suivant ont été perdus, etc.

C.4 Bibliographie

[C-1] Recommandation UIT-T H.221 (1997), *Structure de trame pour un canal d'un débit de 64 à 1920 kbit/s pour les téléservices audiovisuels*.

[C-2] Recommandation UIT-T H.261 (1993), *Codec vidéo pour services audiovisuels à p x 64 kbit/s*.

[C-3] Recommandation UIT-R BT.601 (1997), *Techniques numériques de transmission de l'information télévisuelle*.

ANNEXE D

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

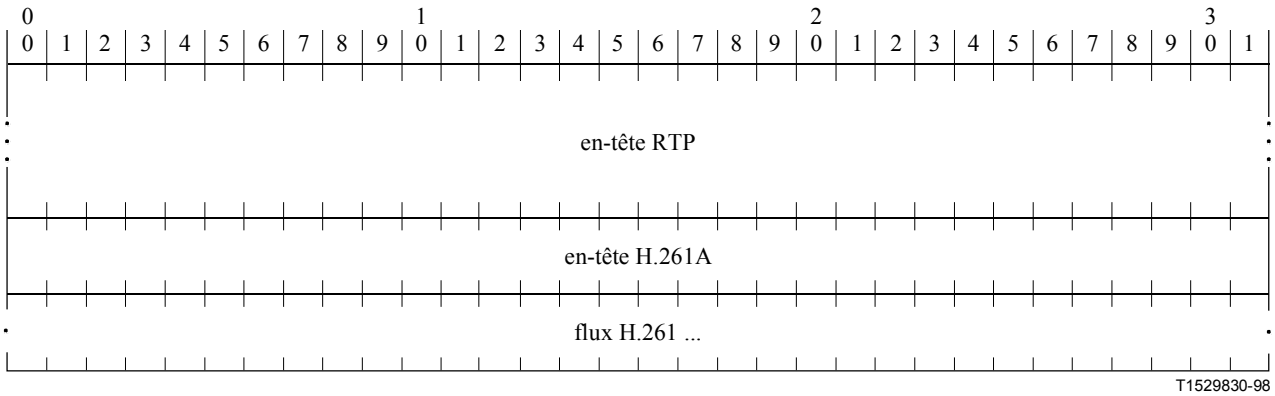
D.1 Introduction

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

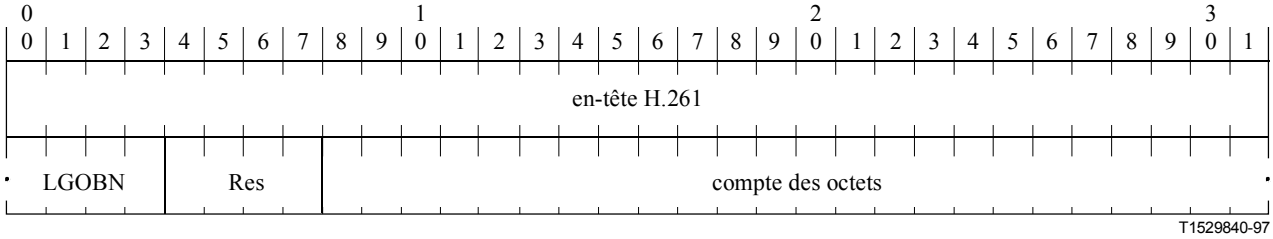
D.2 Mise en paquets RTP H.261A

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 Recommandation 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 seront 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

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 Recommandation 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 Recommandation H.263, sera spécifié ultérieurement. Un format de mise en paquets par capacité léguée, largement utilisé dans l'industrie mais non conforme à la spécification RFC 2190 de l'IETF, ne pourra être utilisé que si l'extrémité homologue a signalé sa prise en charge de ce format, lors de l'échange des capacités.

Le sous-paragraphe B.5 décrit la procédure à utiliser pour signaler les flux vidéo H.263.

ANNEXE F

Mise en paquets de données audio

La présente annexe décrit en détail la mise en paquets RTP pour les codecs audio normalisés par l'UIT-T.

F.1 G.723.1

La Recommandation G.723.1 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 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 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 débit 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.

NOTE – 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.

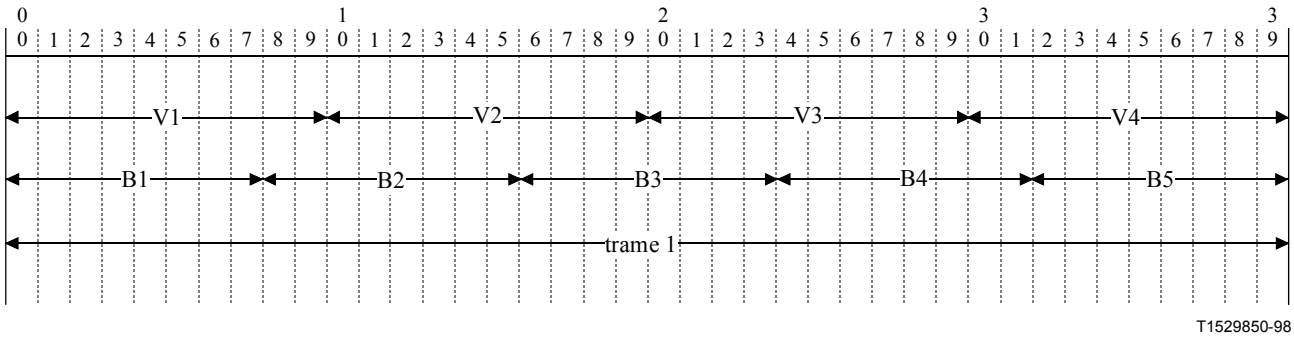
La mise en paquets G.723.1 est conforme à l'Annexe B sauf pour l'intervalle de mise en paquets (30 ms et non 20 ms, qui est la valeur par défaut):

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 deux bits devenant le bit MSB de B2 – et les 6 bits de plus fort poids de V2 – le bit de plus fort poids parmi ces bits restant aussi le bit de plus fort poids parmi ces bits dans 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.

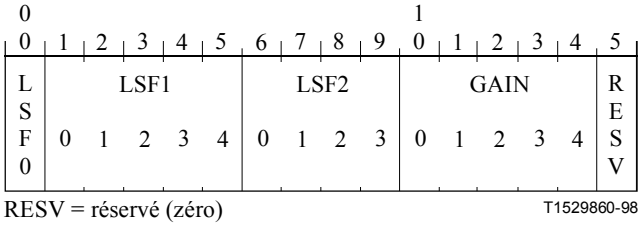
L'horodate tiendrait compte du temps de saisie du premier échantillon, dans le premier vecteur (V1) de la première trame (les informations les plus anciennes du paquet).

3) Le bit de marqueur devra conserver la même signification que celle qui lui est donnée dans la Recommandation H.225.0.

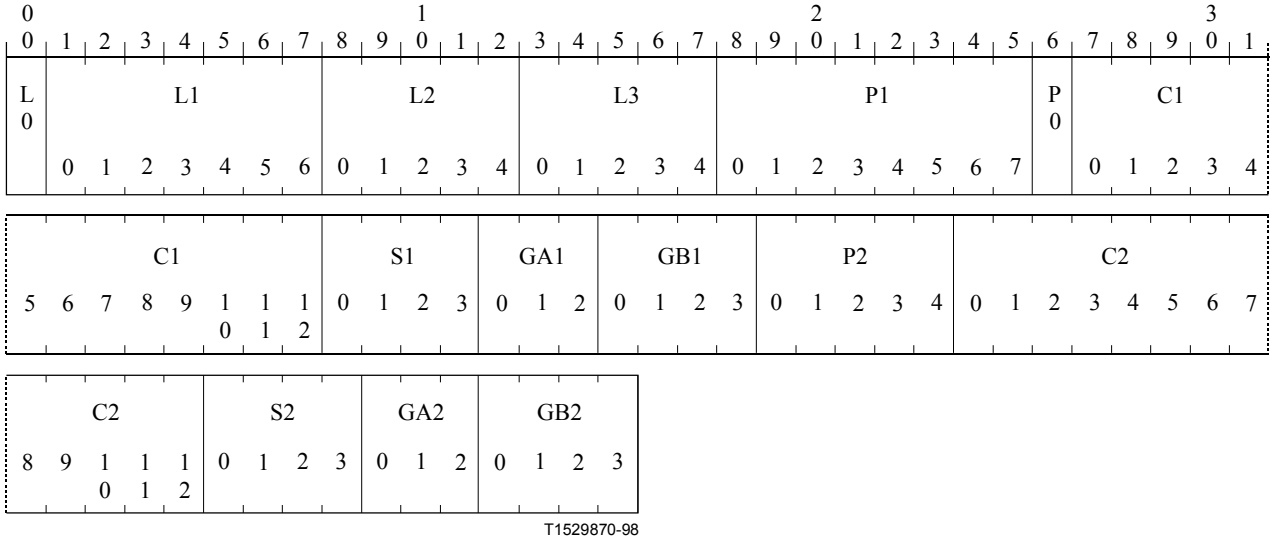
F.3 G.729

La Recommandation G.729 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. Les algorithmes de codage de la parole donnés dans le corps de la Recommandation G.729 et dans l'Annexe A/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 pour les applications numériques voix et données simultanées et peut être utilisé avec l'algorithme de la Recommandation G.729 ou de l'Annexe A/G.729. Une trame G.729 ou Annexe A/G.729 contient 10 octets, tandis que la trame de bruit de confort Annexe B/G.729 occupe 2 octets:



Les paramètres transmis d'une trame de 10 ms G.729 et G.729 Annexe A, constituée de 80 bits, sont définis dans le Tableau 8/G.729. Le mappage de ces paramètres est donné ci-dessous. Les bits sont numérotés dans l'ordre Internet, c'est-à-dire que le bit de plus fort poids est le bit 0.



Un paquet RTP peut être constitué de zéro, une ou plusieurs trames G.729 ou G.729 Annexe A, suivies de zéro ou une charge utile G.729 Annexe B. 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 Recommandation H.225.0, 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. Ceci 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-3] et les codecs GSM plein débit amélioré (EFR, *enhanced full rate*) [F-2]. 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 son mis à 1;
- trames de description d'insertion de silence (SID, *silence insertion 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. Tous les points d'extrémité doivent être en mesure de recevoir et d'identifier une trame inactive. Une trame vocale GSM inactive est remplie de uns linéaires.

Si un point d'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} trame. 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 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 bits/s, sans compter le bit supplémentaire de groupage par paquets.

Codec plein débit amélioré

Le codec GSM plein débit amélioré (EFR, *enhanced full rate*) envoie une trame de 244 bits (30,5 octets) toutes les 20 ms. Cette information doit être compactée dans un en-tête RTP avec un préfixe de quatre bits (0xC ou 1100 binaire), appelé signature. Par conséquent, la charge utile du codec GSM EFR doit être de 31 octets. La trame SID est marquée dans la bande par un mot de code SID enregistré dans les paramètres du codec décrits dans la [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): *Full rate speech; transcoding*.

[F-2] GSM 06.60 (ETS 300 726): *Enhanced Full Rate (EFR) speech transcoding*.

[F-3] GSM 06.20 (ETS 300 969): *Half rate speech; Half rate speech transcoding*.

[F-4] ETSI, TIPHON 03 001 (TS 101 318): *Using GSM speech codecs within H.323*.

[F-5] GSM 06.31 (ETS 300 963): *Full rate speech; Discontinuous Transmission (DTX) for full rate speech traffic channels*.

[F-6] GSM 06.81 (ETS 300 729): *Discontinuous Transmission (DTX) for Enhanced Full Rate (EFR) speech traffic channels*.

[F-7] GSM 06.41 (ETS 300 972): *Half rate speech; Discontinuous Transmission (DTX) for half rate speech traffic channels*.

[F-8] GSM 06.12 (ETS 300 963): *Full rate speech; Comfort noise aspect for full rate speech traffic channels*.

[F-9] GSM 06.62 (ETS 300 728): *Comfort noise aspects for Enhanced Full rate (EFR) speech traffic channels*.

[F-10] GSM 06.22 (ETS 300 971): *Half rate speech; Comfort noise aspect for half rate speech traffic channels*.

[F-11] GSM 08.60 (ETS 300 737): *Inband control of remote transcoders and rate adaptors for Enhanced Full Rate (EFR) and full rate channels*.

ANNEXE G

Communication entre domaines administratifs

G.1 Domaine d'application

Il est prévu que le réseau global H.323 se constituera de sous-ensembles d'équipement plus restreints correspondant à certain type d'organisation, par exemple sous la forme de domaines administratifs. Compte tenu du nombre important d'équipements H.323 susceptibles d'exister au sein de réseaux H.323, il est nécessaire de disposer d'un protocole efficace permettant d'établir des appels entre domaines administratifs. L'exemple le plus simple est celui d'un utilisateur (point de terminaison) appartenant à un domaine administratif qui tente d'atteindre un utilisateur (point de terminaison) desservi par un autre domaine administratif. Bien qu'il convienne à un grand nombre de besoins de communication entre domaines administratifs, le protocole RAS H.225.0 n'est ni complet, ni efficace à cet effet.

La présente annexe décrit des méthodes permettant de traiter la résolution d'adresse, l'autorisation d'accès, et la communication de rapports d'utilisation entre des domaines administratifs de systèmes H.323 en vue de l'établissement d'appels entre domaines. Un domaine administratif se présente vis-à-vis d'autres domaines administratifs sous la forme d'un élément logique appelé "élément frontière". Un élément frontière peut occuper le même emplacement que toute autre entité (par exemple un portier). L'Annexe G n'impose pas qu'un domaine administratif divulgue les détails de son organisation ou de son architecture. Elle ne prescrit aucune architecture spécifique au sein d'un domaine administratif. Elle prend de plus en charge l'utilisation de tout modèle d'appel (acheminement par un portier ou directement vers le point de terminaison).

La procédure générale consiste, pour les éléments frontière, à échanger des informations portant sur les adresses que chacun d'eux est en mesure de résoudre. Il est possible de spécifier les adresses d'une manière générale ou sous des formes de plus en plus spécifiques. Des informations

supplémentaires permettent aux éléments situés au sein d'un domaine administratif de déterminer quel est le domaine administratif le plus approprié comme destination de l'appel. Des éléments frontière peuvent gérer l'accès à leurs adresses exposées et exiger la production de rapports sur l'utilisation faite pendant les appels à ces adresses.

La Figure G.1 indique un certain nombre de points de référence qui représentent la signalisation entre divers éléments au sein d'un réseau H.323. Les domaines administratifs de cette figure appartiennent à un réseau global par paquets sans bords. Il convient de noter que cette figure ne fournit pas de définition explicite de l'architecture système H.323 et qu'elle a uniquement pour objet l'illustration des points de référence de signalisation.

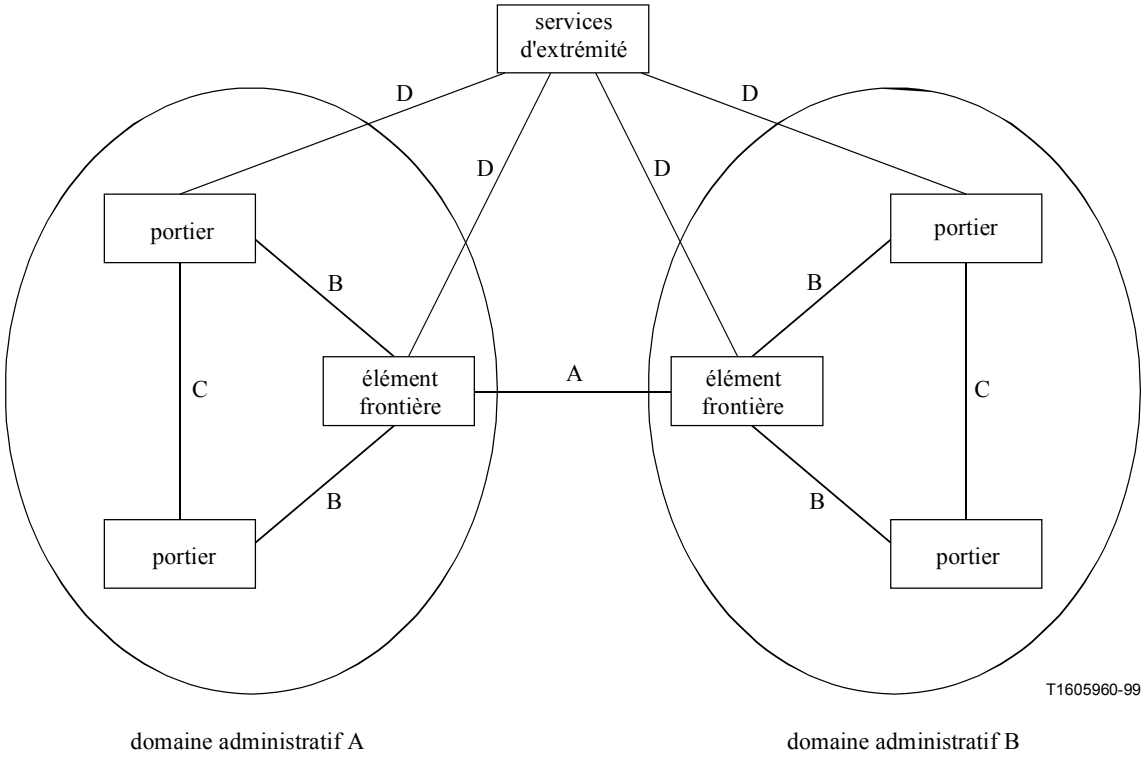


Figure G.1/H.225.0 – Points de référence système

La figure met en évidence les points de référence suivants:

A – entre éléments frontière;

B – entre élément frontière et portiers;

C – entre portiers;

D – entre éléments H.323 et services d'extrémité (en dehors du domaine d'application de la présente annexe).

L'Annexe G s'intéresse principalement au point de référence A. L'utilisation du protocole décrit dans l'Annexe G à des fins de communication entre portiers au sein d'un domaine administratif appelle une étude ultérieure. Le point de référence B appelle une étude ultérieure, car on estime à l'heure actuelle que l'élément frontière sera situé au même emplacement qu'un autre élément H.323.

Le paragraphe G.9 "Exemples de signalisation" fournit quelques exemples destinés à faciliter la compréhension.

G.2 Définitions

La présente Recommandation définit les termes suivants:

G.2.1 domaine administratif: un domaine administratif est un ensemble d'entités H.323 gérées par une même entité administrative. Il peut se constituer d'un ou de plusieurs portiers (c'est-à-dire, d'une ou plusieurs zones).

G.2.2 services d'extrémité: les services d'extrémité sont des fonctions concernant l'utilisateur, telles l'authentification ou l'autorisation, la comptabilité, la facturation, la tarification, etc.. Les services d'extrémité et le protocole permettant l'échange d'informations avec de tels services sont en dehors du domaine d'application de la présente annexe (lorsqu'ils diffèrent de ceux décrits dans ce document).

G.2.3 élément frontière: l'élément frontière est un élément fonctionnel qui prend en charge l'accès public au sein d'un domaine, à des fins d'établissement d'appel ou d'autres services impliquant une communication multimédia avec d'autres éléments situés au sein du domaine administratif. L'élément frontière gère la vue externe du domaine administratif. Un élément frontière communique avec d'autres éléments frontière au moyen du protocole spécifié dans la présente annexe. Un élément frontière peut en outre, selon son implémentation, communiquer avec d'autres entités au sein de son domaine administratif. Cet élément peut se présenter en association avec d'autres éléments H.323, par exemple comme la combinaison d'un élément frontière, d'un portier et d'une passerelle. Un domaine administratif peut contenir un nombre quelconque d'éléments frontière.

G.2.4 résolveur d'adressage: service (se présentant éventuellement sous la forme d'un élément frontière) qui est en mesure de fournir une résolution pour toutes les adresses (c'est-à-dire, un type de point d'agrégation).

G.3 Abréviations

La présente Recommandation 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*)

T terminal

G.4 Références

[1] Recommandation UIT-T H.225.0 (1998), *Protocoles de signalisation d'appel et mise en paquets d'un train multimédia pour des systèmes de communication multimédias en mode paquet*.

[2] Recommandation UIT-T H.235 (1998), *Sécurité et cryptage des terminaux multimédias de la série H (terminaux H.323 et autres terminaux de type H.245)*.

[3] Recommandation UIT-T H.323 (1998), *Systèmes de communication multimédia en mode paquet*.

[4] Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

[5] Recommandation UIT-T X.680 (1997)/Amd.1 (1999) | ISO/CEI 8824-1:1998/Amd.1:1999, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base – Amendement 1: identificateurs d'objets relatifs*.

[6] Recommandation UIT-T X.691 (1997) | ISO/CEI 8825-2:1998, *Technologies de l'information – Règles de codage ASN.1 – Spécification des règles de codage compact*.

G.5 Modèles système

L'Annexe G ne prescrit pas d'architecture système particulière entre des domaines administratifs ou au sein d'un domaine administratif. Les sous-paragraphes qui suivent fourniront quelques exemples d'architecture qui sont considérés comme des illustrations sans aucun caractère exhaustif.

On considère en général qu'un domaine administratif se constitue d'un nombre quelconque de zones et d'un nombre quelconque d'éléments frontière. Rappelons qu'un élément frontière est un élément fonctionnel qui peut coexister avec tout autre élément H.323. La Figure G.2 présente quelques exemples d'implémentation d'élément frontière en combinaison avec d'autres éléments.

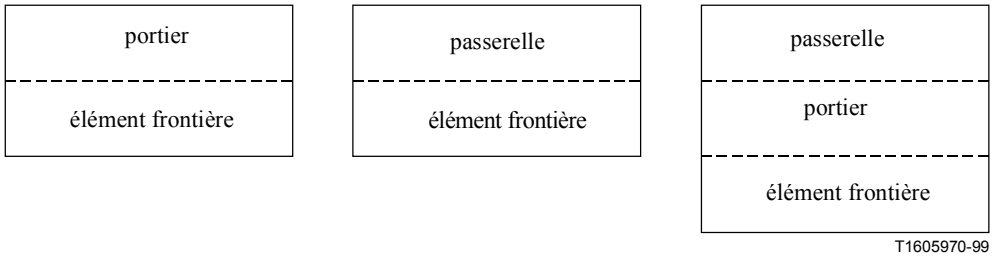


Figure G.2/H.225.0 – Exemples de localisation d'élément frontière

Les relations entre les domaines administratifs peuvent être organisées de diverses manières. Les sous-paragraphes qui suivent en fournissent des exemples.

G.5.1 Hiérarchique

La Figure G.3 présente une structure hiérarchique simple de domaines administratifs. Un élément frontière d'une telle structure consultera un élément frontière dans un domaine administratif supérieur de la hiérarchie pour résoudre une adresse.

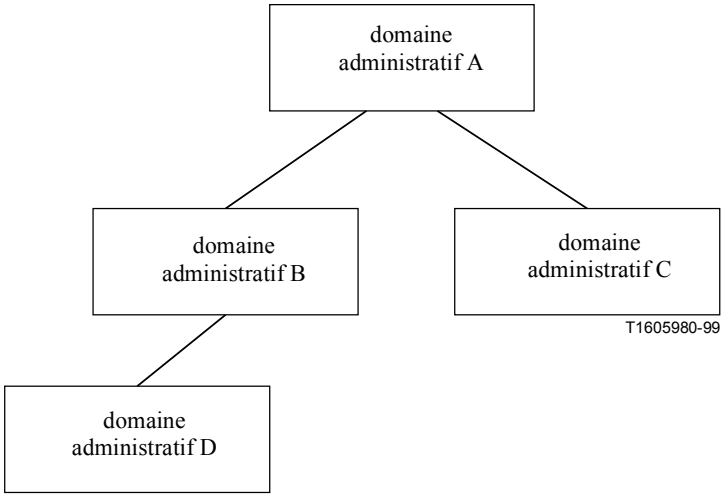


Figure G.3/H.225.0 – Structure hiérarchique simple

G.5.2 Répartition ou maillage total

Il est possible d'utiliser un modèle entièrement réparti ou maillé, comme indiqué dans la Figure G.4. Dans cet exemple, un élément frontière de chaque domaine administratif communique avec des éléments frontière appartenant aux autres domaines administratifs connus.

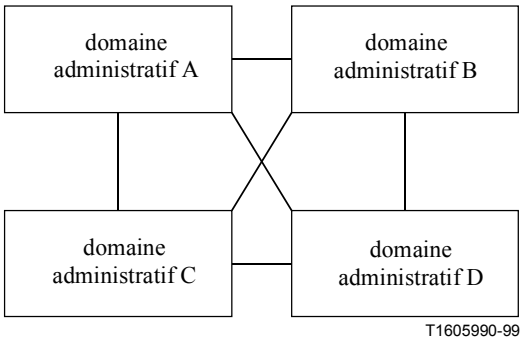


Figure G.4/H.225.0 – Exemple de structure répartie

G.5.3 Résolveur d'adressage

La Figure G.5 présente un exemple de structure de résolveur d'adressage. Dans cette structure, tout domaine administratif consulte le résolveur d'adressage pour résoudre des adresses.

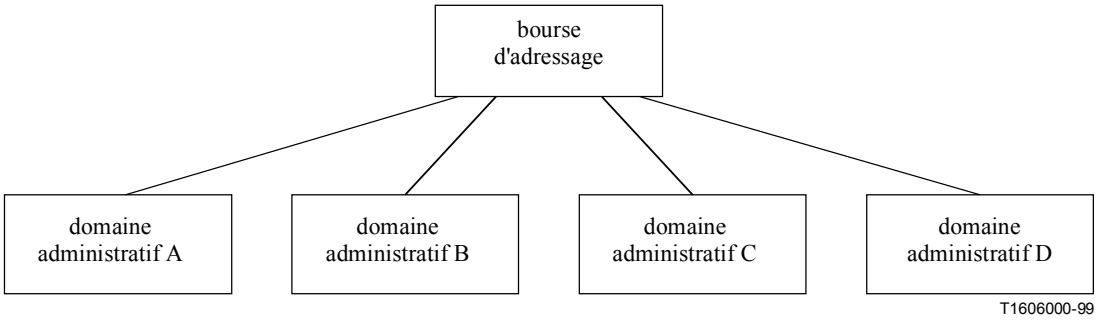


Figure G.5/H.225.0 – Exemple de structure de résolveur d'adressage

G.5.4 Point d'agrégation

La Figure G.6 présente un exemple de point d'agrégation. Le domaine administratif B de cet exemple est un point d'agrégation qui est en mesure de fournir la résolution d'adresse pour lui-même ainsi que pour les domaines administratifs C et D. Le domaine administratif B peut, par exemple, retransmettre à destination du domaine administratif C des demandes de résolution en provenance du domaine administratif A ou peut donner à ce dernier l'instruction de consulter directement le domaine administratif C en ce qui concerne certaines destinations. Si le domaine administratif B retransmet une demande du domaine administratif A à destination du domaine administratif C, il peut alors mémoriser dans son cache la réponse fournie par ce dernier.

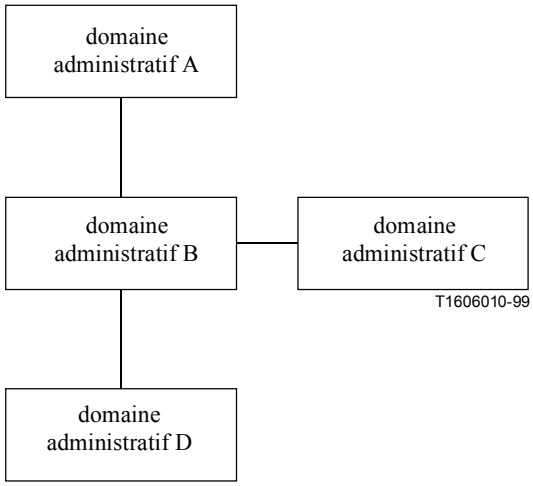


Figure G.6/H.225.0 – Exemple de point d'agrégation

G.5.5 Chevauchement de domaines administratifs

Il se peut que plusieurs domaines soient en mesure de résoudre une adresse donnée. Des domaines administratifs multiples peuvent, par exemple, contenir des passerelles capables d'établir un appel à destination d'un terminal du RTGC. Le choix du domaine administratif de destination adéquat est de la responsabilité du domaine administratif d'origine. L'algorithme utilisé pour ce choix est une affaire d'implémentation.

G.6 Conventions d'adressage

Il est important, pour la fourniture de l'interfonctionnement entre domaines, que les formats d'adressage utilisés dans des messages H.323 émis soient compris par le système récepteur. Un élément frontière prendra en charge les deux types d'alias "identificateur de messagerie électronique" et "numéro d'abonné" (le numéro public étant utilisé avec un type de numéro public égal à "numéro international"). Il convient de noter que cette prescription implique la prise en charge de la version (1998) de la Recommandation H.225.0 ou d'une version ultérieure. Lors de la communication avec d'autres éléments frontière, seuls les types d'alias "identificateur de messagerie électronique" et "numéro d'abonné" doivent être utilisés dans le champ "adresse de destination" d'un message LRQ ou Setup, sauf accord préalable entre les domaines administratifs concernés. Si, par exemple, les domaines administratifs d'un groupe ont conclu un accord sur l'interprétation de numéros locaux privés, ces numéros peuvent alors être utilisés dans leurs messages mutuels.

G.7 Fonctionnement

G.7.1 Canevas et descripteurs d'adresse

Un canevas d'adresse (en abrégé un "canevas") définit un ensemble d'identificateurs d'adresse d'alias, des informations de tarification pour l'établissement d'appels à destination de ces adresses et le protocole devant être utilisé pour atteindre les adresses de cet ensemble. Un domaine administratif indique les appels qu'il est en mesure de résoudre en publiant des canevas. Les canevas sont regroupés par un identificateur appelé "descripteur". Une fois qu'un canevas a été associé à un descripteur, toute modification de ce canevas implique une modification du descripteur "groupe". Les informations de canevas peuvent permettre la mise en commun d'informations d'adressage si le schéma d'adressage est organisé de manière hiérarchique ou pouvant faire l'objet d'un acheminement. Une zone donnée peut, par exemple, traiter le numéro 1303538* qui équivaut à tous les numéros débutant par les chiffres 1303538. (Il convient de noter que le caractère "*" est significatif et que le canevas contient dans ce cas un fanion booléen indiquant si l'adresse est spécifique ou non. Les exemples qui suivent utilisent le caractère "*" pour indiquer un "joker", mais la représentation effective dans le canevas se fait au moyen du fanion booléen.)

Le canevas suivant est donné à titre d'exemple:

pour le numéro 1 555 123 4567	émettre un message de demande d'accès à destination de l'élément frontière A;
pour les chiffres 1 555 987*	émettre un message de demande d'accès à destination de l'élément frontière B;
pour le numéro 1 555 987 6543	émettre un message d'établissement à destination de la passerelle X;
pour *@example.org	émettre un message de demande d'accès à destination de l'élément frontière A;
pour les chiffres 1*	émettre un message de demande d'accès à destination de l'élément frontière B;
pour les chiffres privés 31*	émettre un message de demande d'accès à destination de l'élément frontière C;
pour les chiffres 44 171 112*	n'existe pas.

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

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

G.7.1.1 Configuration statique

Un élément frontière gérera des canevas pour toutes les zones dont il est responsable. Ces canevas peuvent être fournis de manière explicite dans l'élément frontière ou obtenus en résumant des informations fournies par des portiers présents au sein de son domaine. L'élément frontière peut mettre ces informations à la disposition d'autres éléments frontière par le biais d'un mécanisme d'échange de demandes et de réponses. Un domaine administratif peut déterminer le niveau de détail fourni par ses éléments frontière, par exemple:

- un élément frontière souhaitant masquer sa structure interne peut fournir un descripteur unique (avec une indication d'émission de message "demande d'accès") qui représente la totalité de sa zone et fait référence à un portier pour le traitement des appels 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 "établissement";
- un élément frontière dont le domaine contient des trous (correspondant à des numéros qui ont été déplacés vers un autre domaine administratif) fournit des canevas avec une marque "émission de demande d'accès" indiquant l'élément frontière qui doit être utilisé pour contacter l'autre domaine administratif;
- un élément frontière résolveur d'adressage (possédant, par exemple une copie complète de 44) peut contenir un canevas avec une marque "émission de demande d'accès" pour chaque domaine administratif au sein de 44.

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

G.7.1.2 Réception de descripteurs

Un élément frontière peut demander à un autre élément frontière de lui fournir des canevas de configuration statique. Ce dernier décide de la réponse donnée à la demande.

L'élément frontière demandeur émet un message "demande de descripteur" indiquant les descripteurs qu'il souhaite recevoir. S'il est en mesure de transférer les descripteurs, l'élément frontière propriétaire répond alors au moyen d'un message "confirmation de descripteur" qui contient tous les canevas.

L'élément frontière demandeur peut mémoriser dans un cache une copie d'un canevas obtenu de cette manière et la conserver jusqu'à la fin de la durée de vie du canevas, moment auquel la copie sera effacée. S'il modifie ses canevas de configuration statique avant la fin de leur durée de vie, l'élément frontière propriétaire doit alors émettre un message "mise à jour de descripteur" à destination des éléments frontière dont il a connaissance. Lorsqu'il reçoit un message "mise à jour de descripteur", un élément frontière doit supprimer, ajouter ou modifier dans son cache tous les canevas concernés ou demander au propriétaire de lui fournir les copies des descripteurs indiqués.

Un élément frontière intermédiaire (situé entre les domaines administratifs d'origine et de destination, par exemple un résolveur d'adressage ou un point d'agrégation) peut publier ses propres descripteurs en fonction de ceux qu'il reçoit. Un résolveur d'adressage peut, par exemple, indiquer qu'elle est elle-même le contact pour une "demande d'accès", même si elle a reçu d'un autre élément frontière des descripteurs indiquant ce dernier comme contact.

Un élément frontière peut indiquer dans un canevas la nécessité pour l'auteur d'un appel de recevoir l'autorisation d'appeler un domaine administratif. Si un fanion **callSpecific** est placé dans un canevas et si le type de message spécifie la nécessité d'envoyer un message "demande d'accès", alors l'auteur de l'appel doit fournir des informations pour chaque appel dans le message de demande d'accès. Lorsqu'un élément frontière reçoit un message de demande d'accès ne contenant d'informations propres à l'appel, et si la stratégie adoptée consiste à exiger des informations pour chaque appel, l'élément frontière répondra par un message de "rejet d'accès" portant la mention explicative **needCallInformation**.

Un élément frontière peut émettre un message "mise à jour de descripteur" à destination d'autres éléments frontière connus individuellement ou émettre ce message en diffusion multiple. L'élément frontière doit déterminer le domaine de diffusion dans le cas de diffusion multiple d'un tel message. Le message "mise à jour de descripteur" peut contenir les descripteurs qui ont été modifiés. Il peut,

par ailleurs, fournir uniquement les identificateurs des descripteurs modifiés, ce qui permet au destinataire de demander de nouvelles informations. Si un grand nombre de descripteurs a été modifié, les informations doivent alors être émises dans plusieurs messages "mise à jour de descripteur", de manière à éviter que la taille des messages devienne supérieure au maximum pris en charge par les paquets de transport.

G.7.1.3 Réception de réponses à des demandes spécifiques

Un élément frontière peut émettre un message "demande d'accès" à destination d'un autre élément frontière pour demander la résolution d'une adresse qualifiée de manière totale ou partielle. La demande d'accès est émise en général au moyen d'un protocole de transport non fiable (par exemple, le protocole UDP) mais elle peut utiliser également un protocole de transport fiable (par exemple, le protocole TCP).

Lorsqu'il reçoit la demande d'accès, l'élément frontière effectue une recherche dans sa base de données et fournit dans sa réponse le canevas le plus spécifique pour la destination demandée. Il renverra la totalité des canevas s'il en existe plusieurs qui satisfont à la demande. S'il est effectivement responsable pour l'adresse d'alias spécifiée, l'élément frontière répondra alors en général avec un canevas indiquant qu'un message "demande d'accès" ou un message "établissement" doit être émis. S'il joue le rôle d'un résolveur d'adressage, l'élément frontière de destination répondra alors normalement avec un canevas indiquant que le message "demande d'accès" doit être émis.

L'élément frontière de destination peut également ajouter à la réponse d'autres canevas dont il estime qu'ils peuvent être utiles dans le futur. L'ajout de ces canevas ne doit pas faire passer la taille de la réponse au-delà de la limite qui conduirait le réseau de transport à effectuer une fragmentation (par exemple, 576 octets pour le protocole IPv4 ou de 1200 octets pour le protocole IPv6).

Un élément frontière qui est, par exemple, couplé étroitement à un serveur pare-feu peut fournir deux canevas dans sa réponse à un message "demande d'accès": un canevas de courte durée de vie (de quelques minutes ou secondes) spécifiant l'emplacement vers lequel un message "établissement" doit être émis et des canevas supplémentaires spécifiant que des messages "demande d'accès" doivent être émis à destination de l'élément frontière pour d'autres adresses d'alias au sein du domaine administratif.

Un élément frontière peut mémoriser dans un cache, et jusqu'à l'expiration de sa durée de vie, un canevas reçu dans un message "confirmation d'adresse".

G.7.2 Découverte d'un élément frontière ou d'un ensemble d'éléments frontière

G.7.2.1 Statique

Un élément frontière peut disposer d'un ensemble administré d'autres éléments frontière qu'il peut contacter à des fins de résolution d'adresse. La détermination de cet ensemble administré peut se faire par le biais d'accords bilatéraux entre domaines administratifs. Les domaines administratifs peuvent utiliser de manière optionnelle les services d'un résolveur d'adressage.

G.7.2.2 Dynamique

Sur les réseaux Internet, le système DNS définit les propriétaires d'adresses du type "identificateur de messagerie électronique". Il s'ensuit qu'un élément frontière effectuera, en l'absence d'informations plus complètes, une recherche d'enregistrements SRV sur le serveur DNS en utilisant la partie de l'identificateur de messagerie électronique situé à la droite du caractère "@" (il effectuera, par exemple, une recherche de l'enregistrement "**_h2250-annexe-g udp.example.org**" sur le serveur DNS pour la résolution de l'adresse "**person@example.org**"). La réponse à cette recherche permettra de générer un canevas "envoi de demande d'accès" qui peut être utilisé par le processus de résolution. Les canevas générés à partir des demandes de serveur DNS ne doivent pas rester en cache pendant une durée supérieure à la durée de vie indiquée dans la réponse du serveur DNS.

G.7.2.3 Autres méthodes

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

G.7.3 Procédures de résolution

G.7.3.1 Procédure de résolution au sein d'un domaine administratif

L'élément frontière trouve un canevas correspondant lorsqu'il fait l'objet d'une demande de résolution d'alias d'adresse (par exemple, de la part d'une passerelle ou d'un portier situé au même emplacement).

Si plusieurs canevas conviennent, les canevas adéquats sont sélectionnés et triés en fonction d'une stratégie locale. Les canevas peuvent, par exemple, être triés tout d'abord selon le critère de longueur du joker (pour donner la préférence aux canevas les plus spécifiques), puis selon le critère du type de protocole spécifié (l'émission d'une demande d'établissement est préférable à l'émission d'une demande d'accès).

L'élément frontière renverra tous les canevas conformes si plusieurs satisfont à la demande.

Si la procédure de sélection de canevas ne fournit pas de canevas avec une marque "émission d'établissement", l'élément frontière émettra alors, à destination de l'adresse spécifiée dans le canevas, un message "demande d'accès" contenant une adresse de destination spécifique. Il peut mémoriser dans son cache la réponse reçue de l'élément frontière et renvoyer au demandeur l'adresse vers laquelle doit être émis le message "établissement".

G.7.3.2 Procédure de résolution entre domaines administratifs

L'élément frontière effectue une recherche parmi les canevas mémorisés dans son cache et trouve un canevas correspondant à l'adresse de la demande lorsqu'il fait l'objet d'une demande d'accès.

Si plusieurs canevas conviennent, ils sont triés tout d'abord selon le critère de longueur du joker (pour donner la préférence aux canevas les plus spécifiques), puis selon le critère du type de protocole spécifié (l'émission d'une demande d'établissement est préférable à l'émission d'une demande d'accès). On rejette lors de chaque tri les canevas qui ne fournissent pas la correspondance la plus spécifique.

Si les canevas conformes sont marqués "émission de demande d'accès", l'élément frontière peut alors choisir de transmettre le message "demande d'accès" vers un ou plusieurs éléments frontière spécifiés dans un ou plusieurs canevas ou il peut choisir de renvoyer les canevas tels quels. L'élément frontière ne doit pas retransmettre le message "demande d'accès" vers un autre élément frontière si le compteur de bonds du message "demande d'accès" reçu a atteint la valeur nulle; il doit dans ce cas renvoyer à la place du message tout canevas conforme. L'élément frontière répondra au moyen d'un message "rejet d'accès" indiquant le dépassement du nombre de bonds si le compteur de bonds a atteint la valeur nulle et si l'élément frontière ne dispose d'aucune information pouvant être fournie dans un message "confirmation d'accès".

A ce stade, l'élément frontière peut utiliser un élément frontière d'un troisième domaine administratif (par exemple un résolveur d'adressage) afin d'autoriser la demande d'accès. A cet effet, il envoie un message "de demande de validation", contenant des jetons d'accès fournis par l'élément frontière demandeur dans la demande d'accès. L'élément frontière récepteur valide les jetons et renvoie une "confirmation de validation".

L'élément frontière renvoie ensuite un message "confirmation d'accès" contenant les canevas qu'il a trouvé (ces canevas auront des champs "adresse" et "type de message" identiques) ainsi que tout autre canevas dont il estime qu'il sera utile.

L'élément frontière renverra tous les canevas conformes s'il en existe plusieurs qui satisfont à la demande.

Si la demande d'accès contient des informations spécifiques propres à l'appel, la validité des canevas renvoyés est limitée à l'appel demandé. Cette disposition est utilisée lorsqu'un domaine administratif souhaite autoriser l'accès pour chaque appel. Dans ce cas le domaine administratif peut demander l'inclusion d'informations propres à l'appel pour chaque demande d'accès transmise. Il positionne alors un fanion dans les canevas qui lui correspondent.

G.7.4 Echange d'information d'utilisation

Les domaines administratifs peuvent demander aux autres domaines de leur communiquer des informations d'utilisation lors de l'établissement d'appels particuliers. Des messages "d'indication d'utilisation" peuvent être communiqués à n'importe quel stade de l'établissement de l'appel. De plus, plusieurs indications d'utilisation peuvent être transmises pour le même appel, chacune contenant des informations plus récemment mises à jour.

Les échanges d'indications d'utilisation ne sont admis que s'il existe une relation de service entre les deux éléments frontière.

Des demandes d'indication d'utilisation seront envoyées, si un élément frontière l'exige, dans les canevas pour lesquels il fait office d'élément à contacter ou s'il l'indique dans l'un des messages de demande d'utilisation, de demande d'accès, de demande de validation et de confirmation de validation, envoyés en rapport avec l'appel faisant l'objet d'une demande d'indication d'utilisation.

G.8 Protocole

Les messages du protocole de l'Annexe G peuvent être émis en utilisant un service de transport non fiable (par exemple, le protocole UDP) ou un service fiable (par exemple, le protocole TCP) vers une adresse bien connue. Sur les réseaux Internet, il convient d'utiliser le port 2099 bien connu, pour les protocoles TCP et UDP, sauf si un autre port a été indiqué à l'expéditeur. Les éléments frontière doivent être à l'écoute sur les ports TCP et UDP.

Si les messages sont émis au moyen d'un service de transport fiable, il est alors possible d'émettre des messages multiples en respectant les contraintes de taille définies pour l'unité de données du protocole (PDU) de transport fiable, dans la mesure où des messages entiers sont émis. (Cette unité PDU est définie par un paquet TPKT dans les implémentations du protocole IP, telles qu'elles sont résumées dans l'Appendice IV/H.225.0.)

En cas d'utilisation d'un service de transport non fiable, les messages de demande peuvent être retransmis. La valeur par défaut du temporisateur de retransmission doit être déterminée par une méthode adaptative sensible au délai (telle que celle qu'utilise le protocole TCP). Des temps d'attente exponentiels seront alors utilisés pour les retransmissions subséquentes. Le nombre de retransmissions ne dépassera pas 5. Les réponses ne seront pas retransmises.

Dans les implémentations du protocole IP, les messages comporteront un préfixe constitué d'en-têtes de paquet TPKT, permettant l'émission de messages multiples. Le champ longueur de paquet UDP contiendra la longueur totale de la charge utile, y compris tous les messages et leurs en-têtes TPKT.

G.8.1 Considérations relatives à la sécurité

Le fonctionnement de sécurité du protocole IP, tel qu'il est décrit dans la Norme IETF RFC 1825 ("architecture de sécurité pour le protocole Internet") et complété par la Norme IETF RFC 1826 ("en-tête d'authentification du protocole IP") ou par la Norme IETF RFC 1827 ["charge utile d'encapsulation de sécurité IP" (ESP, *IP encapsulating security payload*)] ou par les deux, sera mis en œuvre lorsque l'authentification, l'intégrité et le chiffrement sont souhaités pour des messages échangés entre éléments frontière.

Les procédures et expressions de la Recommandation H.235 seront utilisées le cas échéant pour la prise en charge de la sécurité au niveau application. Ceci concerne plus précisément les formats de

jeton et les échanges d'authentification. Les jetons et les jetons chiffrés reçus dans les messages de réponse seront utilisés dans une demande connexe ultérieure.

G.8.2 Définitions des messages

Tout message contient un ensemble de champs communs en plus des informations propres au message. Les champs communs sont les suivants:

Champ	Description
numéro de séquence [*sequenceNumber*]	Tout message de demande ou de mise à jour contient un numéro de séquence unique. Le message émis en réponse à un message de demande (message de confirmation ou de rejet) utilise le numéro de séquence du message de demande. Les messages retransmis auront le même numéro de séquence.
adresse de réponse [*ReplyAddress*]	Adresse à laquelle il faut envoyer la réponse à un message de demande. Tout message de demande contiendra une adresse de réponse, sauf si la demande a été envoyée par un service de transport bidirectionnel orienté connexion (par ex. TCP). Aucun message, hormis les messages de demande, ne doit contenir d'adresse de réponse.
Version	Version de protocole utilisée par l'émetteur du message.
comptage de bonds [*hopCount*]	Définition du nombre d'éléments frontière à travers lesquels ce message peut se propager. Un élément frontière commence par décrémenter le compteur de bonds lorsqu'il reçoit un message et qu'il décide de le retransmettre à destination d'un autre élément frontière. Si la nouvelle valeur du compteur est supérieure à 0, l'élément insère alors cette valeur dans le message à retransmettre. L'élément frontière ne retransmettra pas le message si le comptage a atteint la valeur 0. Si le message est une demande, l'élément frontière répondra alors au moyen d'un message de confirmation contenant toute information pertinente. Il répondra au moyen d'un message de rejet si aucune information n'est disponible.
valeur de vérification d'intégrité [*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*, chaque octet de ce champ doit être mis à zéro. Après le calcul, l'expéditeur place la valeur de vérification d'intégrité calculée dans le champ *integrityCheckValue* et transmet le message.
jetons [*token*]	Il s'agit de données qui peuvent être nécessaires pour l'opération. Si ces données sont disponibles, elles doivent être insérées dans le message.
jetons chiffrés [*cryptoTokens*]	Jetons chiffrés.
NonStandard	Informations non normalisées.

G.8.2.1 **Descripteur** [*descriptor*]

Le descripteur n'est pas un message, mais un élément de message utilisé comme étiquette pour un ensemble de canevas.

Le descripteur contient les informations suivantes:

Champ	**Description**
informations de descripteur [*descriptorInfo*]	Identificateur non ambigu du descripteur et de l'instant de sa dernière modification (se référer aux informations de descripteur ci-dessous).
Canevas [*templates*]	Ensemble de canevas qui définit les adresses pouvant être résolues par ce descripteur.
identificateur de portier [*gatekeeperID*]	Identificateur du type texte indiquant le propriétaire du descripteur (c'est-à-dire, le portier qui a créé ce message).

G.8.2.2 **Informations de descripteur** [*descriptorInfo*]

Les informations de descripteur identifient de manière non ambiguë le descripteur et indiquent l'instant de sa dernière modification.

Champ	**Description**
identificateur de descripteur [*descriptorID*]	Identificateur globalement non ambigu utilisé pour distinguer ce descripteur parmi un grand nombre d'autres descripteurs possibles.
dernière modification [*lastChanged*]	Date et heure de la dernière modification de ce descripteur.

G.8.2.3 **Canevas** [*template*]

Le canevas d'adresse décrit un ensemble d'une ou plusieurs adresses d'alias. Le canevas n'est pas un message, mais un élément utilisé par d'autres éléments comme bloc de construction. Le canevas se compose d'autres structures qui sont décrites dans les sous-paragraphes qui suivent.

Champ	**Description**
Modèle [*pattern*]	Liste de modèles (voir Modèle ci-dessous).
informations d'acheminement [*routeInfo*]	Liste d'informations d'acheminement pour ce canevas (se référer aux informations d'acheminement ci-dessous).
durée de vie [*timeToLive*]	Durée de validité de ce canevas, exprimée en secondes.

G.8.2.3.1 **Informations d'acheminement** [*routeInfo*]

La structure "informations d'acheminement" figurant dans l'élément "canevas" (champ "informations d'acheminement") contient les champs suivants:

Champ	**Description**
type de message [*messageType*]	Type de message à émettre lors d'une tentative de résolution d'une adresse spécifique utilisant ce canevas. Les types possibles sont les suivants "émission de demande d'accès" [*sendAccessRequest*], "émission d'établissement" [*sendSetup*] ou "inexistant" [*nonExistent*] (indique que l'adresse n'existe pas).

spécificité d'appel [*CallSpecific*]	Si ce champ est mis à la valeur Vrai, chaque appel vers cet acheminement doit obligatoirement faire l'objet d'une autorisation, ce qui implique la nécessité pour le message de demande d'accès de contenir les informations propres à l'appel. Ce champ booléen a une signification uniquement quand le champ *messageType* prend la valeur sendAccessRequest ; sinon *callSpecific* sera mis à la valeur Faux.
spécification d'utilisation [*UsageSpec*]	Lorsqu'il est présent ce champ spécifie les messages d'"indication d'utilisation" qui seront envoyés en ce qui concerne les appels vers cet acheminement.
informations de tarification [*priceInfo*]	Liste d'informations de tarification pour cet acheminement particulier (se référer aux informations de tarification ci-dessous). Il convient de noter que des structures de tarification différentes sont décrites dans des structures "informations d'acheminement" multiples.
contacts	Informations de contact concernant l'élément qui acceptera le message tel qu'il est spécifié dans le champ "type de message" des informations d'acheminement. Les informations de contact peuvent être fournies sous la forme d'une liste de contacts possibles (se référer à la description des informations de contact ci-dessous).
type	Type de point de terminaison pouvant recevoir l'appel. Dans les cas d'acheminement par portier, ce champ indique les points de terminaison desservis par le portier plutôt que le portier lui-même.

G.8.2.3.2 Informations de tarification [*priceInfo*]

Les informations de tarification constituent un élément figurant dans la structure "informations d'acheminement" (champ "informations de tarification"). Les informations de tarification sont définies par les structures "spécification d'informations de tarification" [*PriceInfoSpec*] et "élément de prix" [*PriceElement*].

La structure "spécification d'informations de tarification" contient les champs suivants:

Champ	Description
devise [*currency*]	Désignation de devise conformément à l'ISO-4217.
échelle de devise [*currencyScale*]	Position de la virgule implicite, par rapport à la gauche d'un champ de prix. Si, par exemple, la devise est spécifiée en dollars US, un champ *currencyScale* égal à 2 indique que le montant indiqué dans le champ "élément de prix" est exprimé en cents US.
valable à partir de [*validFrom*]	Date et heure de début de validité de ces informations.
valable jusqu'à [*validUntil*]	Date et heure de fin de validité de ces informations.
heures depuis [*hoursFrom*]	Heure dans la journée à partir de laquelle ce tarif s'applique.
heures jusqu'à [*hoursUntil*]	Heure dans la journée à partir de laquelle ce tarif ne s'applique plus. Ce champ peut être inférieur au champ "heures depuis", ce qui indique un tarif d'étendue horaire nulle.

élément de prix [*priceElement*]	Liste optionnelle d'"éléments de prix" additifs constituant le tarif.
formule de prix [*priceFormula*]	Chaîne optionnelle contenant une formule de tarification utilisée en variante à la structure "élément de prix".

La structure "élément de prix" [*PriceElement*] contient les champs suivants:

Champ	Description
montant [*amount*]	Incrément de comptage. Le compteur est incrémenté d'une unité pour chaque *quantum* ou fraction de *quantum*.
quantum	Nombre d'unités auxquelles s'applique le "montant". Une valeur de 60, avec des "unités" exprimées en secondes indique, par exemple, que l'appel est taxé toutes les minutes ou fractions de minute. Si le champ unités est mis à l'une des valeurs "initial", "minimum" ou "maximum", alors le champ "unités" est sans intérêt, et sa valeur ne doit pas être prise en compte par le destinataire.
unités [*units*]	Unité utilisée pour exprimer le quantum: • secondes – secondes de durée de l'appel; • paquets – nombre de paquets émis ou reçus; • octets – nombre d'octets émis ou reçus; • initial – taxation initiale au moment de la connexion; • minimum – taxation minimale d'un appel; • maximum – taxation maximale d'un appel.

G.8.2.3.3 Informations de contact [*ContactInformation*]

La structure "informations de contact" est un élément de la structure "informations d'acheminement" (champ "contacts").

Champ	Description
adresse de transport [*transportAddress*]	Adresse (par exemple, une adresse de transport ou un identificateur URL) à destination de laquelle doit être émis le message spécifié dans le champ "type de message" de la structure "informations d'acheminement". Dans tous les cas où cela est possible, l'utilisation d'une adresse de transport est obligatoire.
priorité [*priority*]	Lorsque plusieurs contacts figurent dans la liste, le champ "priorité" spécifie l'ordre des tentatives de contact multiple. Les contacts de la liste peuvent partager une même priorité, par exemple lorsqu'il n'existe pas de préférence pour l'ordre des tentatives de contact. Une priorité égale à 0 indique la priorité la plus élevée (premier choix).
qualité de service du transport [*transportQoS*]	Indique qui a la responsabilité de la réservation des ressources pour l'appel établi par ce contact.
Sécurité [*Security*]	Mécanisme de sécurité indiquant l'ordre de préférence à observer pour communiquer avec le contact.

jetons d'accès [*AccessTokens*]	Ensemble de jetons qui seront transmis dans le message adressé à ce contact ("Etablissement" ou "Demande d'accès"). Ces jetons seront également envoyés dans les messages "Indication d'utilisation" relatifs aux appels utilisant ce canevas.

G.8.2.3.4 Modèle

Une structure de modèle apparaît dans le canevas d'adresse. Le modèle permet de spécifier une adresse alias, une adresse alias joker ou une série d'adresses alias:

Champ	Description
Spécifique [*specific*]	Désigne une adresse alias spécifique.
Joker [*Wildcard*]	Forme de définition hiérarchique qui représente des extensions possibles de la chaîne. Cette extension est possible à la fin d'un numéro E.164 ou en début de chaîne dans le cas d'une adresse de messagerie électronique. Par exemple si ce champ est "+1 303", le modèle peut représenter tout numéro dans la zone de code de Denver.
Série [*range]*	Il s'agit d'une série d'adresses, y compris la première et la dernière de la série.

G.8.2.4 Structures communes

Les structures définies dans le présent sous-paragraphe apparaissent dans un certain nombre de messages.

G.8.2.4.1 Autre élément frontière [*AlternateBE*]

Champ	Description
adresse de contact [*contactAddress*]	Adresse de transport d'un autre élément frontière (à destination de laquelle peuvent être émis des messages définis dans l'Annexe G).
priorité [*priority*]	Lorsque des variantes multiples figurent dans la liste, le champ "priorité" spécifie l'ordre dans lequel ces variantes doivent être essayées. Les variantes dans la liste peuvent partager une même priorité, par exemple lorsqu'il n'existe pas de préférence pour le choix des variantes. Une priorité égale à 0 indique la priorité la plus élevée (premier choix).
identificateur d'élément [*elementIdentifier*]	Chaîne de caractères Unicode identifiant cet autre élément frontière.

G.8.2.4.2 Information sur un participant

Cette structure contient des informations relatives à un participant (à l'origine ou à destination) de l'appel.

Champ	Description
adresse logique [*LogicalAddress*]	Adresses de format type adresse de messagerie électronique ou numéro E.164 permettant d'identifier le participant.

identificateur de domaine [*DomainIdentifier*]	Adresse alias identifiant le domaine administratif à l'origine ou à destination de l'appel. Lorsque plusieurs domaines sont impliqués dans l'établissement d'un appel, il faut indiquer le domaine qui a fait office d'origine ou de destination du point de vue de l'expéditeur.
adresse de transport [*TransportAddress*]	Adresse de transport du point d'extrémité.
type de point d'extrémité [*EndpointType*]	Indications détaillées concernant le type de point d'extrémité et les capacités correspondantes.
information d'utilisateur [*UserInfo*]	Information concernant l'utilisateur à l'origine de l'appel. Cette identification peut se faire sous un format d'adresse de messagerie électronique ou d'un numéro de réseau public et le cas échéant au moyen de justificatifs d'identité.
fuseau horaire [*TimeZone*]	Indication du fuseau horaire du participant, dans la mesure où elle est utilisée à des fins de tarification. Si l'appel provient d'une passerelle, alors le fuseau horaire de la passerelle doit être transmis. Exprimé en secondes par rapport au temps universel coordonné.

G.8.2.4.3 Information sur l'appel

Information permettant d'identifier un appel particulier.

Champ	Description
identificateur d'appel [*CallIdentifier*]	Permet d'identifier l'appel de façon univoque. Constituera l'identificateur d'appel associé au même appel que dans les messages RAS et de signalisation d'appel.
identificateur de conférence [*ConferenceID*]	Permet d'identifier de façon univoque la conférence à laquelle participe l'appel. Constituera l'identificateur de conférence associé au même appel que dans les messages RAS et de signalisation d'appel.

G.8.2.4.4 Information d'utilisateur

Information permettant d'identifier l'utilisateur ou un participant quelconque à l'appel.

Champ	Description
Identificateur d'utilisateur [*UserIdentifier*]	Identification univoque de l'utilisateur.
authentificateur d'utilisateur [*UserAuthenticator*]	Jetons chiffrés permettant une authentification sécurisée.

G.8.2.4.5 Spécification d'utilisation

Cet élément décrit les paramètres dont la notification est nécessaire dans les messages d'"indication d'utilisation". Les appels auxquels cette spécification est applicable sont déterminés par le contexte du message qui contient l'élément "Spécification d'utilisation".

Champ	Description
Envoyer à [*SendTo*]	Elément frontière vers lequel les messages d'indication d'utilisation doivent être envoyés. Puisque l'expéditeur doit avoir des relations de service avec cet élément frontière, il s'agit de l'identificateur d'élément renvoyé dans le message de confirmation de service.
Quand [*When*]	Spécifie les étapes de l'appel pour lesquelles les indications en question doivent être envoyées ainsi que la fréquence de ces envois: • jamais – arrêter d'envoyer des messages; • commencer – au début de l'appel; • cesser – à la fin de l'appel, ou après; • période – envoyer périodiquement les messages en question pendant la durée de l'appel. Période indiquée en secondes; • échec – notifier les tentatives d'appel ayant échoué.
champs obligatoires [*Required*]	Liste d'identificateurs des champs qui *doivent* obligatoirement être présents dans les messages "UsageIndication". L'expéditeur du message information sur l'utilisation des ressources doit rejeter ou ignorer le message qui contient ce message, s'il ne peut fournir les informations relatives à ces champs.
champs théoriquement présents [*Preferred*]	Liste d'identificateurs des champs qui *devraient* être présents dans les messages *UsageIndication*.

G.8.2.4.6 Mode sécurité

Cet élément décrit un profil de sécurité spécifique à utiliser dans les communications auxquelles l'Annexe G est consacrée.

Champ	Description
authentification [*Authentication*]	Champ indiquant le mécanisme d'authentification devant être utilisé pour cette relation de service. Le mécanisme d'authentification doit être choisi dans l'ensemble fourni dans le message "demande de service".
Intégrité [*Integrity*]	Indique le mécanisme d'intégrité devant être utilisé. Si ce champ est présent, tous les messages ultérieurs renseigneront le champ "valeur de vérification d'intégrité"; le champ "mode d'authentification" décrit alors le mode de génération des clés secrètes (par échange de clés Diffie-Hellman ou *a priori*).
identificateurs d'algorithme [*AlgorithmOID*]	Indique l'algorithme de cryptage pour le mécanisme de sécurité.

G.8.2.5 Demande de service [*ServiceRequest*]

Un élément frontière peut établir une relation de service par l'émission d'un message "demande de service" à destination d'un autre élément frontière. La relation définit les mécanismes de sécurité devant être utilisés entres les éléments frontière et permet l'identification d'éléments frontière de

remplacement ou de secours. Il convient de noter que cette relation est unidirectionnelle. La sécurité entre les deux éléments est utilisée pour des demandes émises par l'élément qui est à l'origine de la demande de service et pour des réponses émises par le récepteur de la demande de service. Des clés de session peuvent être créées pendant le processus d'établissement de la relation de service. Ces clés resteront valides pendant la durée de vie de la relation de service. Il est possible d'utiliser des jetons à cet effet, tel qu'indiqué dans la Recommandation H.235.

Le destinataire d'une demande de service peut indiquer d'autres éléments frontière pouvant être utilisés par le récepteur de la demande pour une demande de service de secours. L'établissement d'une relation de service est obligatoire pour pouvoir échanger des messages d'indication d'utilisation. Sinon, il s'agit d'une procédure optionnelle, mais il se peut que la stratégie d'un élément frontière impose une telle relation.

Un élément frontière peut émettre un message "demande de service" à destination d'un élément frontière avec lequel il a déjà établi une relation dans le but de remplacer les caractéristiques de la relation existante par de nouvelles caractéristiques. Les relations de service peuvent avoir une durée de vie limitée. Un élément frontière peut alors régénérer la relation en envoyant une nouvelle demande de service.

Champ	**Description**
Identificateur d'élément [*ElementIdentifier*]	Chaîne identifiant l'élément frontière qui envoie la demande.
Identificateur de domaine [*DomainIdentifier*]	Indique le domaine administratif qui demande la relation de service.
capacité de sécurité [*SecurityCapability*]	Ensemble de mécanismes de sécurité que cet élément frontière peut prendre en charge.
durée de vie [*TimeToLive*]	Durée de vie proposée, en secondes, pour la relation de service. En l'absence de cette indication, on suppose une durée de vie infinie.

G.8.2.6 Confirmation de service [*ServiceConfirmation*]

Lorsqu'il reçoit un message "demande de service", un élément frontière répond au moyen d'un message "confirmation de service" pour indiquer qu'il accepte l'établissement d'une relation de service. Si l'élément frontière possède déjà une relation de service avec l'élément frontière qui a émis le message "demande de service", l'émission du message "confirmation de service" indique que les caractéristiques de la relation initiale ne sont plus valides et ont été remplacées par les nouvelles caractéristiques.

Champ	**Description**
identificateur d'élément [*elementIdentifier*]	Chaîne identifiant l'élément frontière.
Variantes [*alternates*]	Liste d'autres éléments frontière pouvant être contactés en cas d'absence de réponse de cet élément frontière.
identificateur de domaine [*DomainIdentifier*]	Désigne le domaine administratif qui répond à la demande.

mode sécurité [*SecurityMode*]	Indique le mécanisme de sécurité à utiliser pour cette relation de service. Le mécanisme de sécurité doit être choisi dans l'ensemble de mécanismes mentionnés dans le message de demande de service.
durée de vie [*TimeToLive*]	Durée de vie exprimée en secondes de la relation de service déterminée par l'élément frontière serveur.

G.8.2.7 Rejet de service [*ServiceRejection*]

Lorsqu'il reçoit un message "demande de service", un élément frontière répond au moyen d'un message "rejet de service" pour indiquer qu'il refuse l'établissement d'une relation de service. Si l'élément frontière possède déjà une relation de service avec l'élément frontière qui a émis le message "demande de service", l'émission du message "rejet de service" indique que les nouvelles caractéristiques ont été rejetées, mais la relation initiale reste valide.

Champ	Description
motif [*reason*]	Motif du rejet de la demande de service par l'élément frontière. Les valeurs possibles sont les suivantes: • service indisponible [*serviceUnavailable*] – Cet élément frontière n'est pas disponible actuellement pour le service; • renvoi du service [*serviceRedirected*] – La tentative doit être renouvelée en utilisant la liste des autres éléments frontière; • sécurité [*security*] – Cet élément frontière ne peut pas prendre en charge les mécanismes de sécurité proposés dans le message "demande de service"; • continuation [*continue*] – indique le message suivant de demande de service à envoyer, afin de poursuivre la procédure à plusieurs étapes d'échange de clés; • non défini [*undefined*] – Le motif du rejet de la demande de service ne correspond à aucun des choix précédents.
Variantes [*alternates*]	Liste d'autres éléments frontière qui sont éventuellement en mesure d'établir une relation de service. Au moins une liste doit être fournie si le motif est "renvoi du service".

G.8.2.8 Libération du service [*ServiceRelease*]

L'un ou l'autre des éléments frontière engagés dans une relation de service peut y mettre fin par l'émission d'un message "libération du service".

Champ	Description
motif [*reason*]	Motif pour lequel cet élément frontière a mis fin à la relation de service. Les valeurs possibles sont les suivantes: • hors service [*outOfService*] – L'élément frontière passe dans l'état "hors service"; • maintenance – L'élément frontière est mis hors service pour des raisons de maintenance; • terminé [*terminated*] – L'élément frontière a décidé de mettre fin à la relation;

	• expiration [*expired*] – La durée de vie de la relation de service s'est écoulée.
Variantes [*alternates*]	Liste d'autres éléments frontière qui sont éventuellement en mesure d'établir une relation de service.

G.8.2.9 Demande de descripteur [*DescriptorRequest*]

Le message "demande de descripteur" permet à une entité de demander à un élément frontière la fourniture de descripteurs spécifiques.

Champ	Description
identificateur de descripteur [*descriptorID*]	Identificateur d'un ou plusieurs descripteurs particuliers demandés par l'émetteur de ce message.

G.8.2.10 Confirmation de descripteur [*DescriptorConfirmation*]

Le message "confirmation de descripteur" constitue la réponse positive d'un élément frontière à une demande de descripteur, lorsque cet élément peut interpréter la demande et que les règles d'implémentation permettent l'échange d'informations.

Champ	Description
descripteur [*descriptor*]	Champ "descripteur" décrit ci-dessus.

G.8.2.11 Rejet de descripteur [*DescriptorRejection*]

Un élément frontière peut rejeter une demande de descripteur pour divers motifs.

Champ	Description
motif [*reason*]	Motif du rejet de la demande de descripteur. Les valeurs possibles sont les suivantes: • taille de paquet dépassée [*packetSizeExceeded*] – La taille de la réponse est supérieure à la taille maximale de paquet, de sorte que le demandeur devrait émettre la demande en utilisant un autre mécanisme de transport (il doit, par exemple, utiliser le protocole TCP à la place du protocole UDP); • identificateur non valide [*illegalID*] – Le destinataire de la demande de descripteur ne possède aucun enregistrement au sujet du descripteur concerné; • sécurité [*security*] – La demande de descripteur ne répond pas aux contraintes de sécurité du destinataire; • dépassement du compteur de bonds [*hopCountExceeded*] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible; • non disponible [*unavailable*] – Le destinataire ne peut fournir les descripteurs. Il convient alors d'utiliser une méthode de fourniture statique ou hors-bande;

	• pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service; • non défini [*undefined*] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.
identificateur de descripteur [*descriptorID*]	Identificateur du descripteur particulier fourni dans cette réponse.

G.8.2.12 Demande d'identificateur de descripteur [*DescriptorIDRequest*]

La demande d'identificateur de descripteur permet à une entité d'interroger un élément frontière au sujet de la liste des identificateurs de descripteur de son domaine administratif.

G.8.2.13 Confirmation d'identificateur de descripteur [*DescriptorIDConfirmation*]

Un message "confirmation d'identificateur de descripteur" constitue la réponse positive d'un élément frontière à un message "demande d'identificateur de descripteur". Un élément frontière peut, une fois qu'il a reçu le message "demande d'identificateur de descripteur", émettre le message "demande de descripteur" pour demander la transmission des descripteurs.

Champ	Description
informations de descripteur [*descriptorInfo*]	Liste d'informations de descripteur dont chaque élément identifie sans ambiguïté le descripteur et l'instant de sa dernière modification.

G.8.2.14 Rejet d'identificateur de descripteur [*DescriptorIDRejection*]

Un élément frontière peut rejeter une demande d'identificateur de descripteur pour divers motifs.

Champ	Description
motif [*reason*]	Motif du rejet de la demande d'identificateur de descripteur. Les valeurs possibles sont les suivantes: • pas de descripteurs [*noDescriptors*] – L'élément frontière ne dispose pas de descripteurs dont il peut rendre compte; • sécurité [*security*] – La demande d'identificateur de descripteur ne répond pas aux contraintes de sécurité du destinataire; • dépassement du compteur de bonds [*hopCountExceeded*] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible; • non disponible [*unavailable*] – Le destinataire ne peut fournir les descripteurs. Il convient alors d'utiliser une méthode de fourniture statique ou hors-bande; • pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service; • non défini [*undefined*] – Le motif du rejet de la demande d'identificateur de descripteur ne correspond à aucun des choix précédents.

G.8.2.15 Mise à jour de descripteur [*DescriptorUpdate*]

Le message "mise à jour de descripteur" constitue la notification d'un élément frontière indiquant que les informations d'adresse ont été modifiées. Un élément frontière peut également émettre ce message pendant l'initialisation. Un élément frontière qui reçoit le message "mise à jour de descripteur" peut demander des informations à l'élément identifié dans ce message.

Champ	Description
émetteur [*sender*]	Un élément qui reçoit la mise à jour de descripteur peut émettre une demande à destination de cette adresse (par exemple, une adresse de transport ou un identificateur URL).
informations de mise à jour [*updateInfo*]	Liste de mises à jour dont chaque élément fournit le descripteur ou l'identificateur de descripteur mis à jour. L'élément de la liste indique également si le descripteur a été modifié, ajouté ou supprimé.

G.8.2.16 Accusé de réception de mise à jour de descripteur [*DescriptorUpdateAck*]

Un élément frontière doit accuser réception d'un message "mise à jour de descripteur" en émettant le message "accusé de réception de mise à jour de descripteur". Le numéro de séquence utilisé dans l'accusé de réception doit être identique à celui reçu dans le message "mise à jour de descripteur". Un élément frontière ne doit pas fournir d'accusé de réception d'un message "mise à jour de descripteur" transmis par multidiffusion.

G.8.2.17 Demande d'accès [*AccessRequest*]

Un élément frontière peut émettre un message "demande d'accès" à destination d'un autre élément frontière pour demander la résolution d'une adresse d'alias spécifique.

Champ	Description
informations de destination [*destinationInfo*]	Adresse devant être résolue.
informations d'origine [*sourceInfo*]	Informations sur le participant qui a déclenché l'appel faisant l'objet d'une demande d'accès.
informations propres à l'appel [*CallInfo*]	Champ permettant d'identifier l'appel particulier faisant l'objet d'une demande d'autorisation d'accès. En l'absence de cette indication la demande porte sur des appels non définis vers les destinations spécifiées.
spécification d'utilisation [*UsageSpec*]	Champ indiquant les messages d'utilisation que le participant à l'origine de l'appel demande au participant qui répond et qui concernent l'appel faisant l'objet d'une demande d'accès dans ce message. Utilisé uniquement en présence du champ *CallInfo.*

G.8.2.18 Confirmation d'accès [*AccessConfirmation*]

Un élément frontière renvoie dans le message "confirmation d'accès" les informations demandées dans le message "demande d'accès".

Champ	Description
canevas [*template*]	Liste des canevas correspondant aux attributs de la demande d'accès.

réponse partielle [*partialResponse*]	Ce message contient, en cas de réponse positive, certaines parties des informations disponibles. La totalité des informations n'a pas été émise pour respecter la taille maximale de paquet. La totalité des informations doit être récupérée en utilisant un autre type de transport (par exemple, le protocole TCP).

G.8.2.19 Rejet d'accès [*AccessRejection*]

Un élément frontière peut rejeter une demande de descripteur pour divers motifs.

Champ	Description
motif [*reason*]	Motif du rejet de la demande de descripteur. Les valeurs possibles sont les suivantes: • pas de correspondance [*noMatch*] – La destination spécifiée dans la demande d'accès ne peut pas être résolue; • taille de paquet dépassée [*packetSizeExceeded*] – La taille de la réponse dépasse la taille maximale de paquet, de sorte que le demandeur doit émettre la demande en utilisant un autre mécanisme de transport (il doit, par exemple, utiliser le protocole TCP à la place du protocole PDU); • sécurité [*security*] – La demande de descripteur ne répond pas aux contraintes de sécurité du destinataire; • dépassement du compteur de bonds [*hopCountExceeded*] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible; • pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service; • absence d'informations propres à l'appel [*CallInfoNeeded*] – La demande ne contenait pas d'informations spécifiques propres à l'appel; • non défini [*undefined*] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.

G.8.2.20 Demande en cours [*RequestInProgress*]

Un élément frontière peut renvoyer le message "demande en cours" dans le but d'indiquer que la durée nécessaire pour répondre à une demande peut dépasser la durée prévue normalement pour une réponse. Le numéro de séquence doit être identique à celui figurant dans la demande concernée par l'envoi de ce message.

Champ	Description
retard [*delay*]	Laps de temps, exprimé en millisecondes, prévu pour la réponse de l'élément frontière à la demande initiale.

G.8.2.21 Demande non normalisée [*NonStandardRequest*]

Le message "demande non normalisée" peut être émis par un élément frontière dans le cas d'un message de demande qui n'est pas défini dans l'Annexe G. Les informations non normalisées sont véhiculées dans l'élément "non normalisé" défini dans le champ "informations communes" lequel est défini par l'Annexe G.

G.8.2.22 Confirmation non normalisée [*NonStandardConfirmation*]

Le message "confirmation non normalisée" peut être émis par un élément frontière en réponse à un message "demande non normalisée". Les informations non normalisées sont véhiculées dans l'élément "non normalisé" définit dans le champ "informations communes" de l'Annexe G.

G.8.2.23 Rejet non normalisé [*NonStandardRejection*]

Le message "rejet non normalisé" peut être émis par un élément frontière en réponse à un message "demande non normalisée". Les informations non normalisées sont véhiculées dans l'élément "non normalisé" définit dans le champ "informations communes" de l'Annexe G.

Champ	Description
motif [*raison*]	Motifs du rejet de la demande. Les valeurs possibles sont les suivantes: • non prise en charge [*notSupported*] – Le destinataire sait qu'il s'agit d'une demande non normalisée, mais ne comprend pas ou ne prend pas en charge les données non normalisées; • pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service; • non défini [*undefined*] – Le motif du rejet de la demande non normalisée ne correspond à aucun des choix précédents.

G.8.2.24 Réponse à un message non reconnu [*UnknownMessageResponse*]

Un élément frontière qui ne comprend pas un message reçu doit répondre à l'émetteur au moyen d'un message "réponse à un message non reconnu". L'élément frontière ne doit pas utiliser ce message si un autre message défini dans l'Annexe G fournit une réponse adéquate (un message "rejet de descripteur" constitue, par exemple, une réponse adéquate à une demande de descripteur contenant un identificateur de descripteur non valide).

Champ	Description
message non reconnu [*unknownMessage*]	Contenu du message non reconnu.
Motif [*reason*]	Motif d'utilisation du message "réponse à un message non reconnu". Les valeurs possibles sont les suivantes: • non compris [*notUnderstood*] – Le message n'a pas été compris; • non défini [*undefined*] – Le motif de l'émission du message "réponse à un message non reconnu" ne correspond à aucun des choix précédents.

G.8.2.25 Demande d'utilisation

Ce message permet de demander au destinataire d'envoyer des messages d'indication d'utilisation concernant un appel particulier.

Champ	Description
informations propres à l'appel [*CallInfo*]	Appel pour lequel la demande d'indications est formulée.
spécification d'utilisation [*UsageSpec*]	Spécifie quand les indications doivent parvenir et quel doit être leur contenu.

G.8.2.26 Confirmation d'utilisation

Le message de confirmation d'utilisation est envoyé en réponse à un message de demande d'utilisation, afin d'indiquer que le destinataire a accepté la demande et enverra des indications d'utilisation.

G.8.2.27 Rejet d'utilisation

Le message de rejet d'utilisation est envoyé en réponse à un message de demande d'utilisation, afin d'indiquer que le destinataire a rejeté la demande et n'enverra pas subséquemment d'indications d'utilisation.

Champ	Description
motif [*reason*]	Motif du rejet de la demande d'utilisation par l'élément frontière. Les valeurs possibles sont les suivantes: • appel non valide [*InvalidCall*]; • sécurité [*Security*]; • non disponible [*Unavailable*]; • pas de relation de service [*noServiceRelationship*]; • non défini [*undefined*].

G.8.2.28 Indication d'utilisation

Ce message notifie des indications détaillées sur l'appel ainsi que des information concernant l'utilisation des ressources. Il est envoyé en rapport avec le dernier élément de spécification d'utilisation *UsageSpecification* reçu par l'élément frontière et concernant l'appel.

Champ	Description
informations propres à l'appel [*CallInfo*]	Appel auquel l'indication se rapporte.
jetons d'accès [*AccessTokens*]	Jetons d'accès relatifs à l'appel. Ces jetons ont été reçus dans le canavas d'adresse utilisé pour l'appel et communiqué dans le message de demande d'accès/d'établissement relatif au même appel.
rôle de l'expéditeur [*SenderRole*]	Rôle de l'expéditeur de l'indication: • origine – participant à l'origine de l'appel; • destinatation – participant destinataire; • non normalisé – autre.

état d'appel d'utilisation [*UsageCallStatus*]	Etat actuel de l'appel: • pré-connexion; • appel en cours; • appel achevé.
adresse de la source [*SourceAddress*]	Numéro E.164 ou adresse de messagerie électronique du demandeur. Dans le cas d'un numéro E.164 il s'agit de l'identification ANI/CLI.
adresse de la destination [*DestAddress*]	Numéro E.164 ou adresse de messagerie électronique du demandé.
instant de début [*StartTime*]	Instant du début de l'appel exprimé en temps universel coordonné (UTC) . Concerne uniquement les appels ayant franchi l'étape de l'établissement.
instant de fin [*EndTime*]	Instant de fin de l'appel exprimé en temps universel coordonné (UTC) . Concerne uniquement les appels aboutis.
cause de fin [*terminationCause*]	Motif de la fin de l'appel. Concerne uniquement les appels aboutis.
information d'utilisation [*usageInformation*]	Ensemble de champs d'information. Chaque champ est représenté par un "champ d'utilisation" susceptible d'être normalisé ou non. Les champs d'utilisation normalisés doivent faire l'objet d'un complément d'étude.

G.8.2.29 Confirmation d'indication d'utilisation

Le message de confirmation d'indication d'utilisation est envoyé en réponse à un message de d'indication d'utilisation, afin d'indiquer que le destinataire a accepté l'indication notifiée.

G.8.2.30 Rejet d'indication d'utilisation

Le message de rejet d'indication d'utilisation est envoyé en réponse à un message de d'indication d'utilisation, afin d'indiquer que le destinataire a rejeté l'indication notifiée et n'en tiendra pas compte.

Champ	**Description**
motif [*reason*]	Motif du rejet de la demande d'utilisation par l'élément frontière. Les valeurs possibles sont les suivantes: • appel non valide [*InvalidCall*]; • sécurité [*Security*]; • pas de relation de service [*noServiceRelationship*]; • non défini [*undefined*].

G.8.2.31 Demande de validation

Un élément frontière qui termine un appel peut envoyer un message de demande de validation à un autre élément frontière pour vérifier la validité de l'origine de l'appel.

Champ	Description
information de destination [*DestinationInfo*]	Informations détaillées concernant la destination de l'appel.
informations d'origine [*sourceInfo*]	Informations concernant le point d'extrémité à l'origine de l'appel.
informations propres à l'appel [*CallInfo*]	Ce champ identifie l'appel particulier qui fait l'objet d'une demande d'autorisation d'accès.
Spécification d'utilisation [*UsageSpec*]	Lorsqu'il est présent ce champ indique à l'élément frontière adressant les demandes de message qu'il doit recevoir un message d'indication d'utilisation concernant l'appel validé.
jetons d'accès [*AccessTokens*]	Jetons reçus du participant à l'origine de l'appel attestant l'autorisation d'accès relative à l'appel considéré.

G.8.2.32 Confirmation de validation

Message indiquant que l'appel est validé. L'élément frontière demandeur peut terminer l'appel. L'élément frontière qui effectue la validation peut indiquer des alias afin de terminer l'appel.

Champ	Description
information de destination [*DestinationInfo*]	Autres paramètres relatifs à la destination à utiliser par l'élément frontière destinataire.
spécification d'utilisation [*UsageSpec*]	Lorsqu'il est présent ce champ indique à l'élément frontière adressant les demandes de confirmation qu'il doit recevoir un message d'indication d'utilisation concernant l'appel validé.

G.8.2.33 Rejet de validation

Message indiquant que l'appel n'est pas valide. L'élément frontière demandeur peut ne pas établir l'appel.

Champ	Description
motif [*reason*]	Motif du rejet de la demande. Les valeurs possibles sont les suivantes: • jeton non valide [*tokenNotValid*] – les jetons d'accès fournis ne sont pas valides pour l'appel en question; • sécurité [*security*] – La demande de validation ne répond pas aux contraintes de sécurité du destinataire; • dépassement du compteur de bonds [*hopCountExceeded*] – Le compteur de bonds a atteint la valeur nulle et aucune information n'est disponible; • information d'origine manquante [*MissingSourceInfo*] – l'information d'origine fournie ne suffit pas pour valider l'appel;

- information de destination manquante [*MissingDestInfo*] – l'information de destination fournie ne suffit pas pour valider l'appel;
- pas de relation de service [*noServiceRelationship*] – Le destinataire échangera cette information seulement après établissement d'une relation de service;
- non défini [*undefined*] – Le motif du rejet de la demande de descripteur ne correspond à aucun des choix précédents.

G.9 Exemples de signalisation

Les exemples de signalisation qui suivent illustrent le mode opératoire de base. On fait l'hypothèse que les domaines administratifs ont conclu des accords bilatéraux, de sorte que les éléments frontière ont reçu des informations mutuelles (concernant, par exemple, des ports TCP). Nombre des exemples ci-dessous font état de messages RAD LRQ/LCF qui sont échangés entre un portier et un élément drontière à l'intérieur du même domaine administratif. Ces exemples sont présentés à titre purement indicatif, puisque le protocole relatif au point de référence B n'a pas été déterminé (voir G.1).

G.9.1 Répartition ou maillage total

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

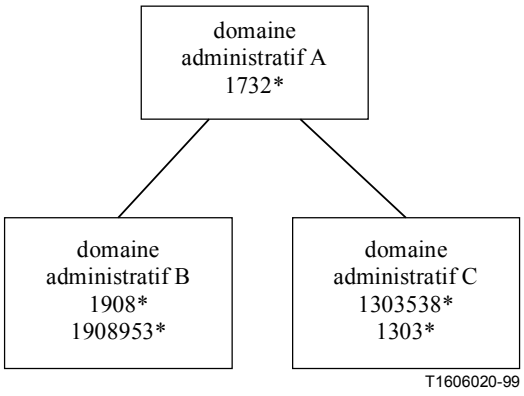


Figure G.7/H.225.0 – Réseau réparti pour les exemples de signalisation

On fait l'hypothèse, 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 adminis-tratif	Définitions de canevas	Commentaire
A	Descripteur "d1": Modèle = 1732* Adresse de transport = adresse de signal d'appel BE_A Type de message = sendSetup	La signalisation pour tout appel dans le domaine administratif A se fera à travers l'élément frontière de ce domaine.
B	Descripteur "d1": Modèle = 1908* Adresse de transport = adresse annexe g BE_B Type de message = sendAccessRequest Descripteur "d2": Modèle = 1908953* Adresse de transport = adresse de SIGNALISATION D'APPEL GW_{B1} Type de message = sendSetup	Un message "demande d'accès" est nécessaire pour faire aboutir à leur adresse de signalisation d'appel de destination (c'est-à-dire, une passerelle) des messages concernant les numéros 1908*. Le message d'établissement peut être émis directement à destination de l'adresse particulière dans le cas d'appels concernant les numéros 1908953*.
C	Descripteur "d1": Modèle = 1303538* Adresse de transport = adresse de signal d'appel GK_{C1} Type de message = sendSetup Descripteur "d2": Modèle = 1303* Adresse de transport = adresse annexe g BE_C Type de message = sendAccessRequest	Les appels concernant les numéros 1303538*, seront acheminés par l'intermédiaire de ce portier particulier. Les appels concernant les numéros 1303* peuvent faire l'objet d'une signalisation directe vers la passerelle de destination, mais une demande d'accès doit être émise pour obtenir l'adresse de signalisation d'appel de la passerelle.

G.9.1.1 Echange d'informations de zone

L'organisation de chaque domaine administratif dans la structure répartie ou totalement maillée a connaissance de chacun des autres domaines administratifs, sans doute à la suite d'un certain nombre d'accords bilatéraux. Un élément frontière d'un domaine administratif peut demander à tout instant à un autre domaine administratif de lui fournir des informations d'adressage. La Figure G.8 présente un exemple de la signalisation correspondante.

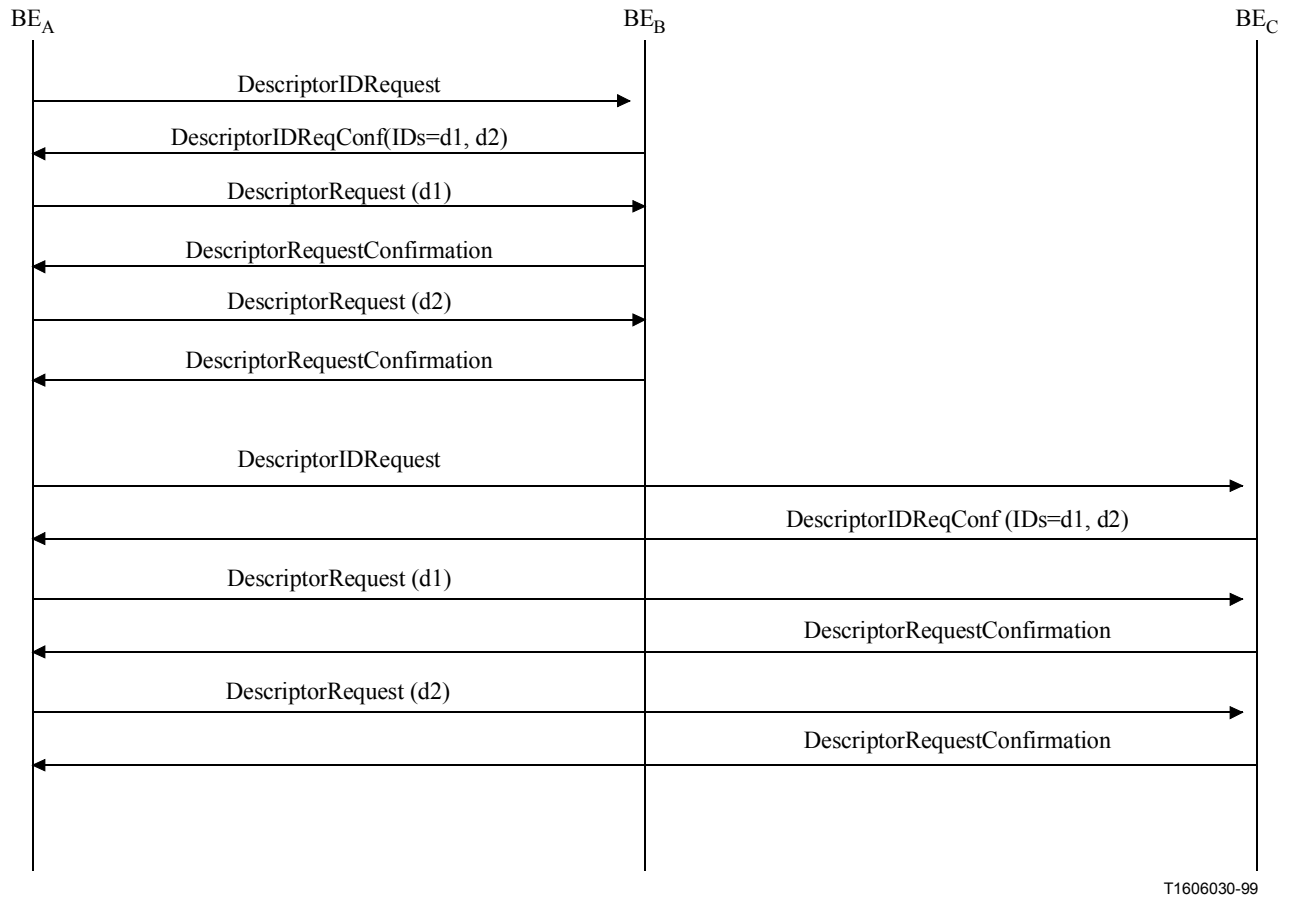


Figure G.8/H.225.0 – Exemple d'échange de descripteurs

L'élément frontière BE_B interroge de même les éléments frontière BE_A et BE_C et l'élément frontière BE_C, les éléments frontière BE_A et BE_B.

G.9.1.2 Etablissement d'un appel

Supposons que le terminal T1 du domaine administratif A lance un appel vers le numéro 19085551515 (terminal T2). Le portier du terminal T1 émet un message LRQ lorsqu'il reçoit le message ARQ émis par le terminal T1. Un élément frontière BE_A du domaine administratif A a reçu au préalable des descripteurs de zone et connaît le traitement qui doit être appliqué à la requête. Comme indiqué dans la Figure G.9, l'élément frontière BE_A émet un message "demande d'accès" à destination de l'élément frontière BE_B qui est spécifié dans le descripteur BE_A reçu de l'élément frontière BE_B. Ce dernier renvoie une réponse contenant l'adresse de signalisation d'appel du terminal T2 (le terminal T2 de cet exemple peut être tout type de point de terminaison). Le terminal T1 émet ensuite le message Setup (établissement) H.225.0 à destination de l'adresse de signalisation d'appel du terminal T2 en appliquant les procédures normales définies par la Recommandation H.323 ou ses annexes.

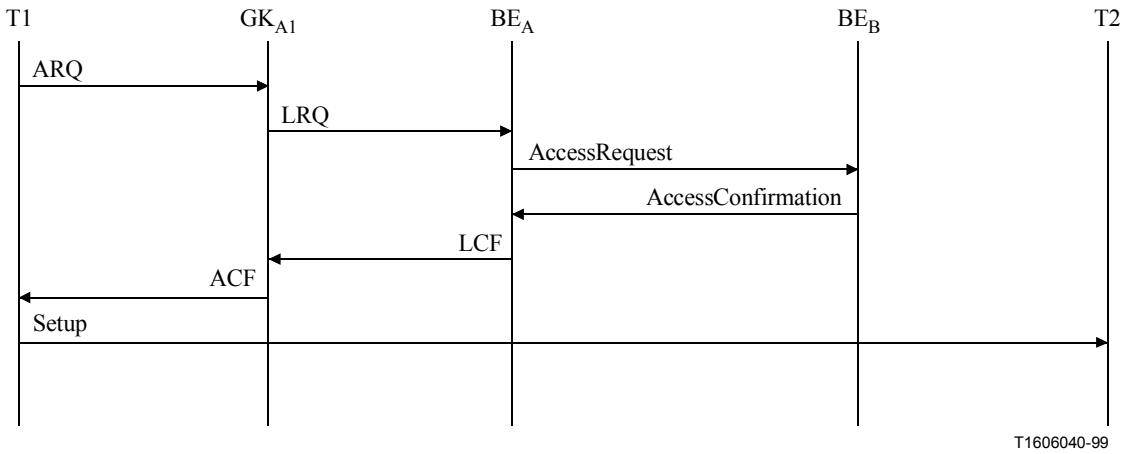


Figure G.9/H.225.0

Supposons maintenant que le terminal T1 lance un appel vers le numéro 19089532000. L'élément frontière BE_A de cet exemple a obtenu au préalable l'adresse de signalisation d'appel d'une passerelle située dans un domaine administratif qui acceptera l'appel. Comme indiqué dans la Figure G.10, l'élément frontière BE_A peut répondre au message LRQ sans aucun échange de message dans le domaine administratif B, ce qui permet au terminal T1 d'émettre le message Setup directement à destination de la passerelle GW.

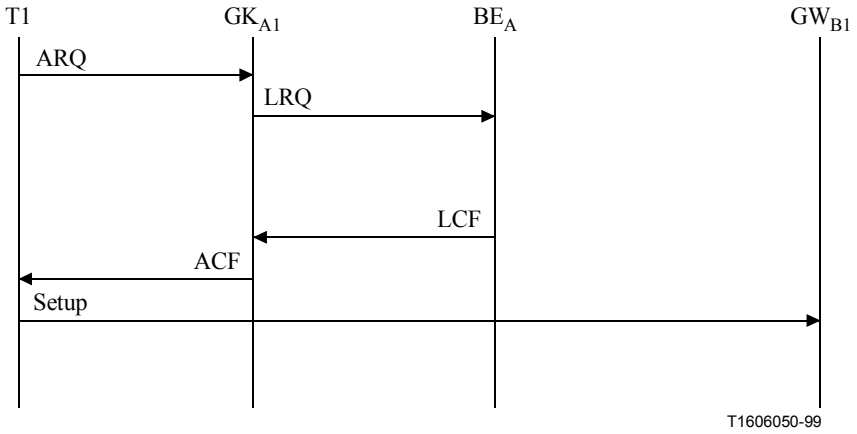


Figure G.10/H.225.0

Supposons que, dans un autre exemple, le terminal T1 lance un appel vers le numéro 13035382899. Le domaine administratif C a publié sa capacité de prise en charge d'un appel à destination de ce numéro et acceptera la signalisation d'appel par le biais de son portier en implémentant le modèle 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.

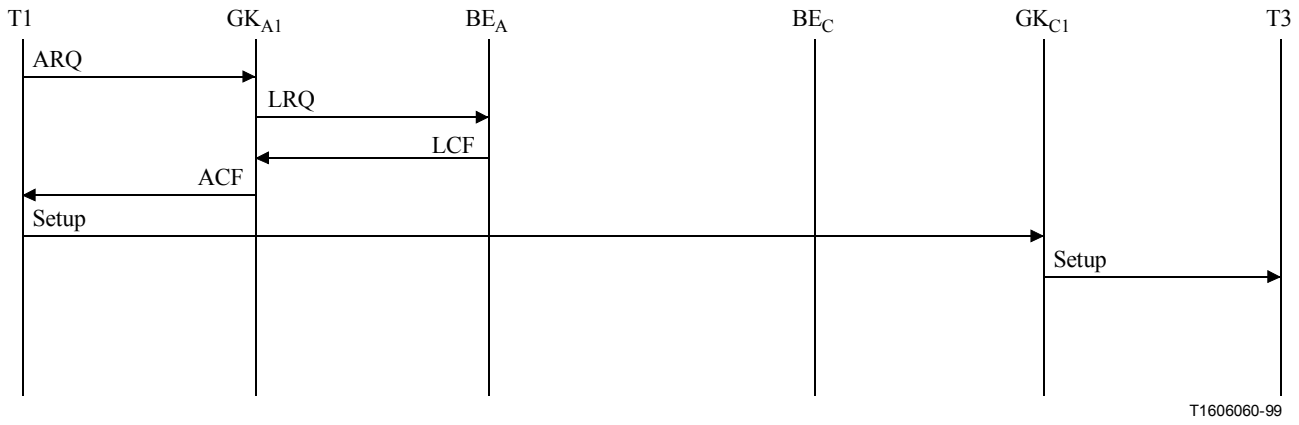


Figure G.11/H.225.0

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

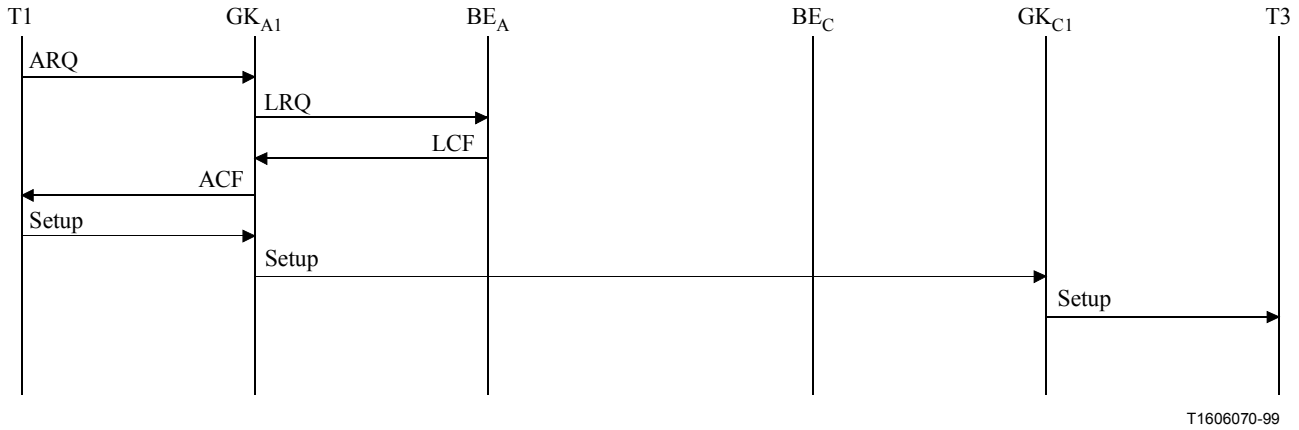


Figure G.12/H.225.0

G.9.2 Résolveur d'adressage

La Figure G.13 présente un exemple qui utilise un résolveur d'adressage. Cette figure sert de référence pour les exemples qui suivent. Le résolveur d'adressage conserve des informations d'adressage pour tous les domaines administratifs auxquels elle fournit des services.

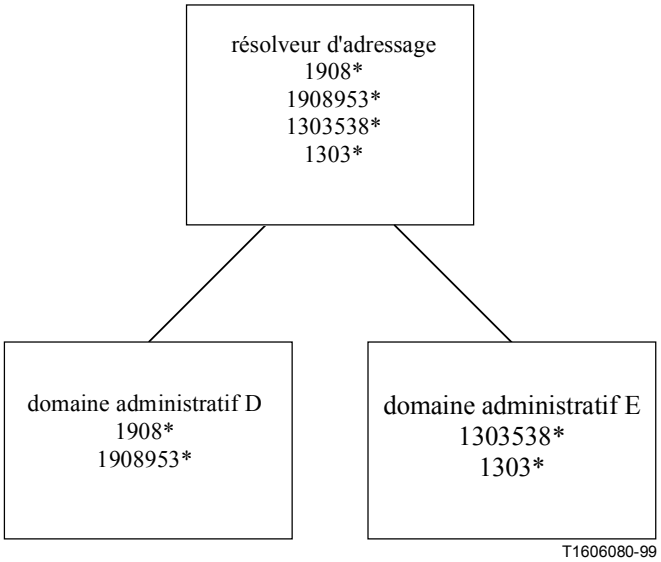


Figure G.13/H.225.0 – Exemple de configuration avec résolveur d'adressage

Les éléments frontière des domaines administratifs D et E et le résolveur d'adressage détiennent les informations suivantes dans l'exemple ci-dessous:

Domaine adminis-tratif	**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	Les appels concernant les numéros 1908*, un message "demande d'accès" est nécessaire pour obtenir l'adresse de signalisation d'appel de destination (c'est-à-dire, une passerelle). Les appels concernant les numéros 1908953* seront acheminés par l'intermédiaire de cette passerelle particulière.
E	Descripteur "d1": Modèle = 1303538* Adresse de transport = adresse de signal d'appel GK_{E1} Type de message = sendSetup Descripteur "d2": Modèle = 1303* Adresse de transport = adresse annexe g BE_E Type de message = sendAccessRequest	Les appels concernant les numéros 1303538* seront acheminés par l'intermédiaire de ce portier particulier. Les appels concernant les numéros 1303* peuvent faire l'objet d'une signalisation directe vers la passerelle de destination, mais une demande d'accès doit être émise pour obtenir l'adresse de signalisation d'appel de la passerelle.

Domaine adminis-tratif	Définitions de canevas	Commentaire
CH	Descripteur "d1": Modèle = 1908* Adresse de transport = BE_D adresse annexe g Type de message = sendAccessRequest Descripteur "d2": Modèle = 1908953* Adresse de transport = adresse de signalisation d'appel GW_{D1} Type de message = sendSetup Descripteur "d3": Modèle = 1303538* Adresse de transport = adresse de signal d'appel GK_{E1} Type de message = sendSetup Descripteur "d4": Modèle = 1303* Adresse de transport = adresse annexe g BE_E Type de message = sendAccessRequest	Le résolveur d'adressage obtient des descripteurs d'autres domaines administratifs et conserve ces informations durant l'échange de descripteurs.

G.9.2.1 Echange d'informations de zone

Un résolveur d'adressage échange dans cet exemple des informations avec des domaines administratifs qui se sont abonnés à ses services. Le résolveur d'adressage conserve les informations qu'elle reçoit de chaque domaine administratif et les relaye à destination d'autres domaines administratifs. Le résolveur d'adressage est vue par le domaine administratif D comme le domaine administratif E, mais les domaines administratifs D et E n'ont pas nécessairement connaissance de leur existence mutuelle. Voir Figure G.14.

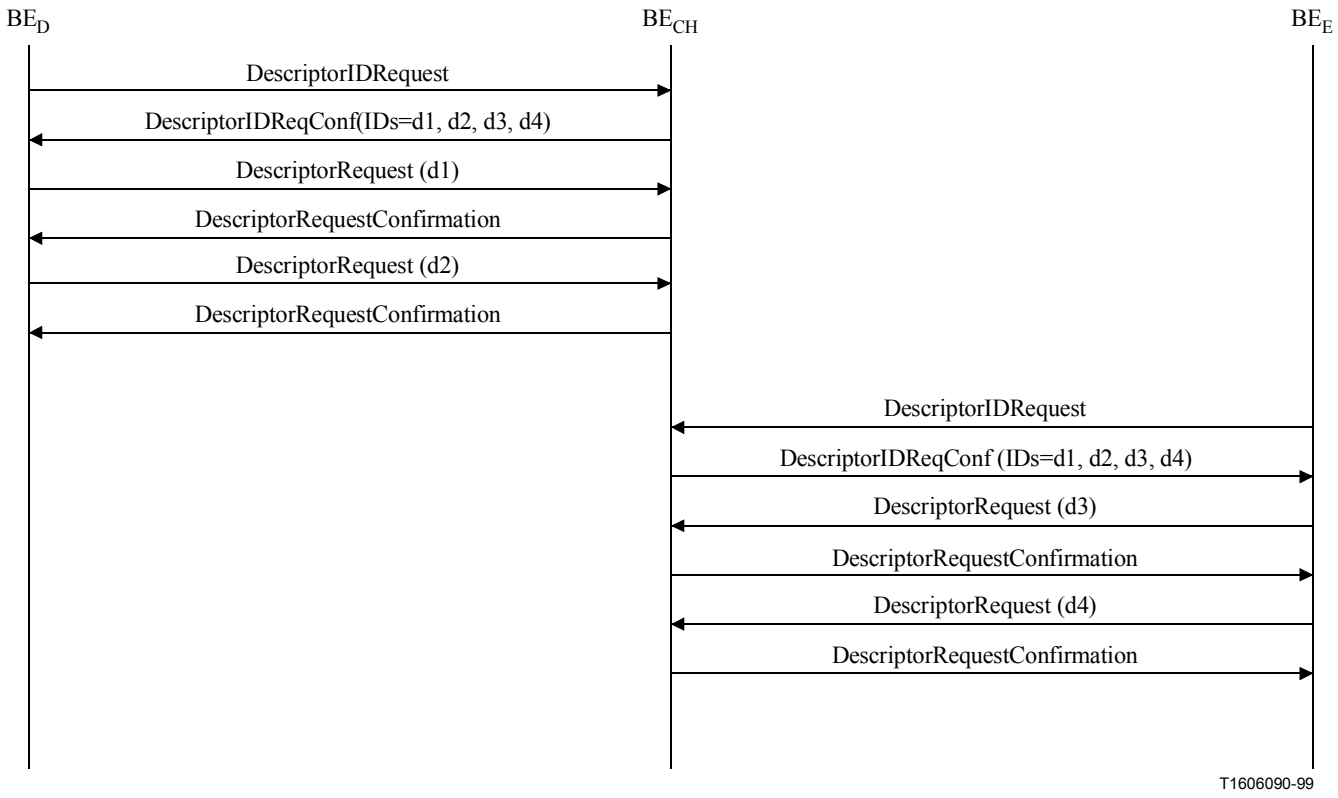


Figure G.14/H.225.0 – Exemple d'échange de descripteur avec un résolveur d'adressage

G.9.2.2 Etablissement d'un appel

Supposons que le terminal T1 du domaine administratif E lance un appel vers le numéro 19085551515. L'élément frontière du domaine administratif E a reçu du résolveur d'adressage des descripteurs indiquant que cette dernière doit être consultée pour un tel appel. L'élément émet une demande d'accès à destination de l'élément frontière du résolveur d'adressage. Les descripteurs que l'élément frontière du résolveur d'adressage a reçu de l'élément frontière du domaine administratif D lui permettent d'émettre une demande d'accès à destination de l'élément frontière du domaine administratif D. Lorsque l'élément frontière du résolveur d'adressage renvoie la confirmation à destination de l'élément frontière du domaine administratif E, cette confirmation contient les informations émises par l'élément frontière du domaine administratif D. Le portier du terminal T1 renvoie un message ACF contenant l'adresse de signalisation d'appel du terminal T1 qui doit être utilisée pour émettre le message *Setup* à destination du terminal T2. Voir Figure G.15.

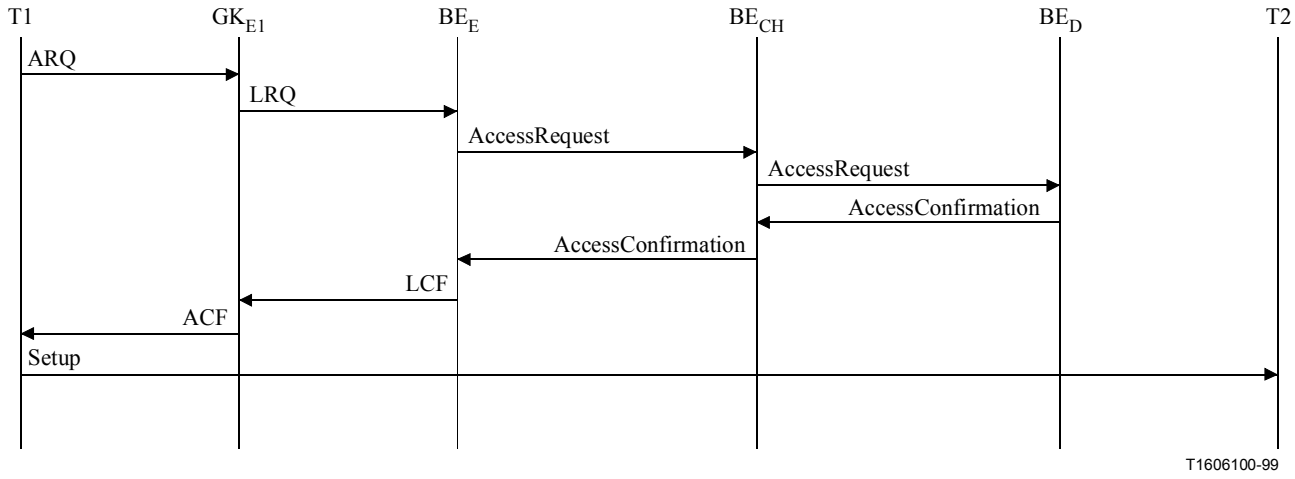


Figure G.15/H.225.0

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

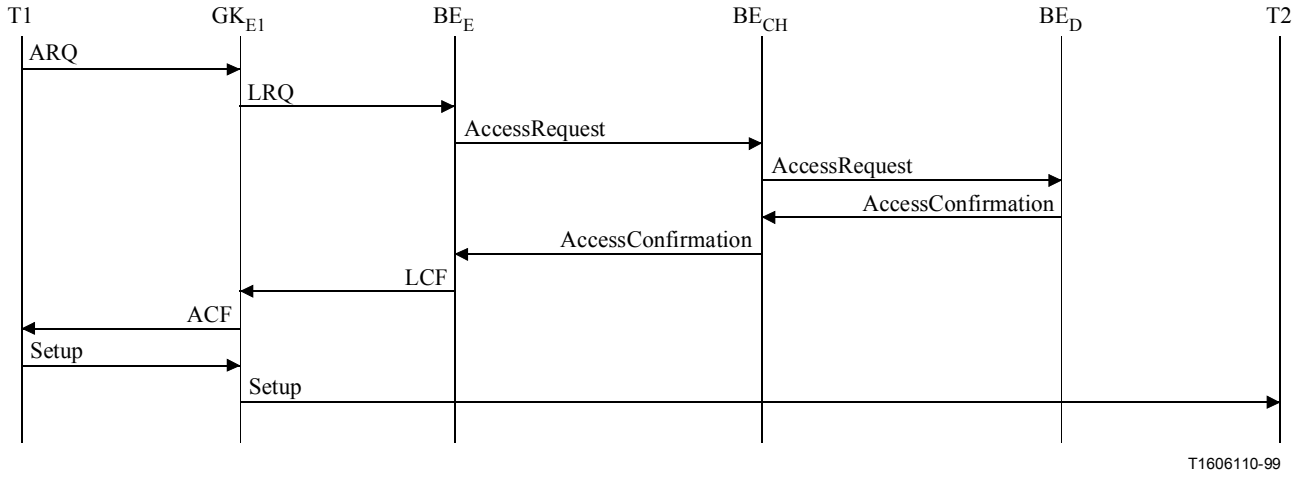


Figure G.16/H.225.0

Une autre possibilité pour le résolveur d'adressage consiste à répondre à l'élément frontière du domaine administratif E en lui fournissant les informations de contact pour l'élément frontière du domaine administratif D, comme indiqué dans la Figure G.17.

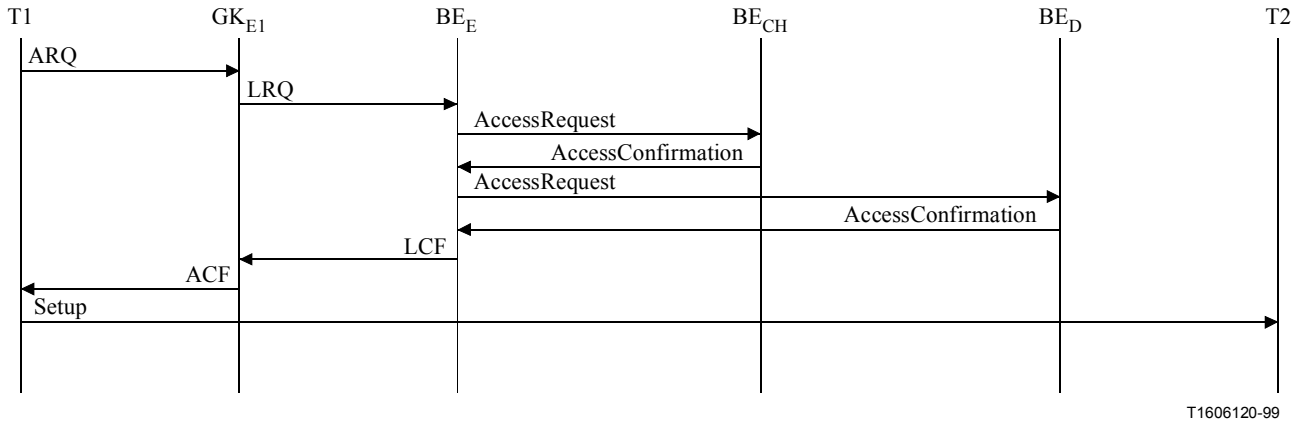


Figure G.17/H.225.0

Supposons maintenant que le terminal T1 lance un appel vers le numéro 19089532000. Les descripteurs échangés au préalable permettent à l'élément frontière de renvoyer au terminal T1 l'adresse de signalisation d'appel sans consulter le résolveur d'adressage, comme indiqué dans la Figure G.18.

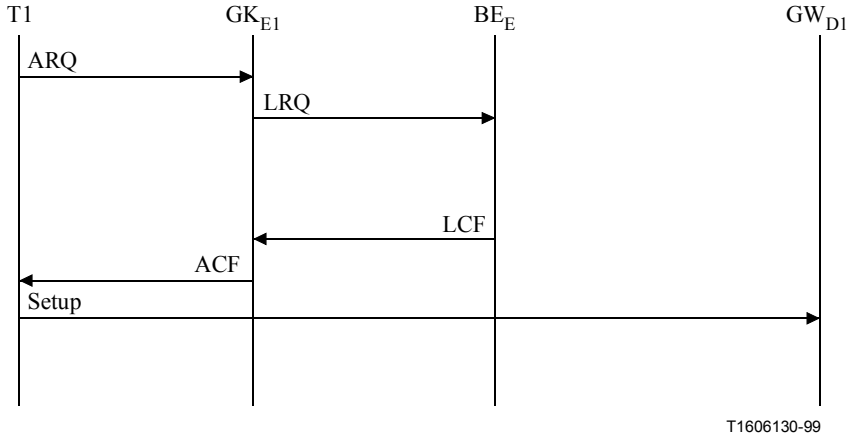


Figure G.18/H.225.0

Examinons maintenant un scénario dans lequel le terminal T1 lance un appel vers le numéro 13035382899. L'élément frontière du domaine administratif E a publié au préalable le fait que les appels à destination des numéros 1303538* peuvent être 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 cette dernière n'a pas l'obligation, dans cet exemple, de fournir la résolution d'adresse pour ces appels.

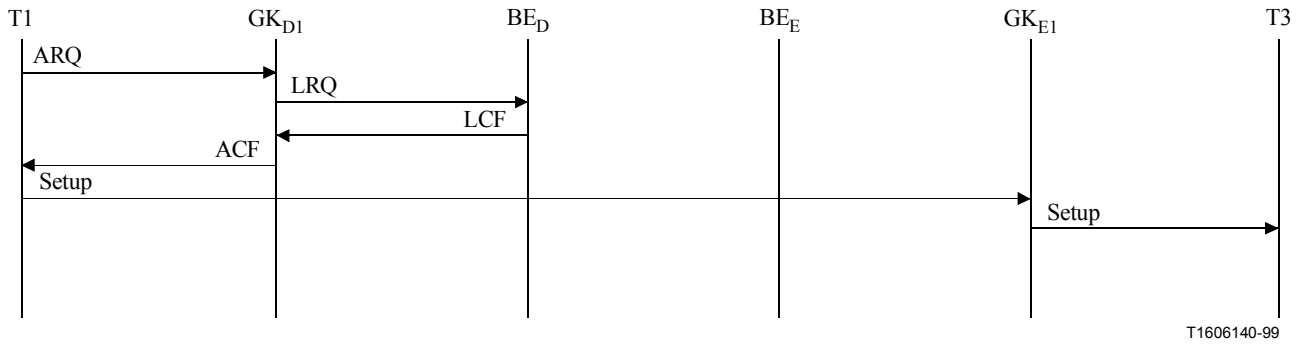


Figure G.19/H.225.0

Rappelons qu'un élément frontière peut être combiné avec un portier et peut également 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 si les descripteurs sont configurés en conséquence.

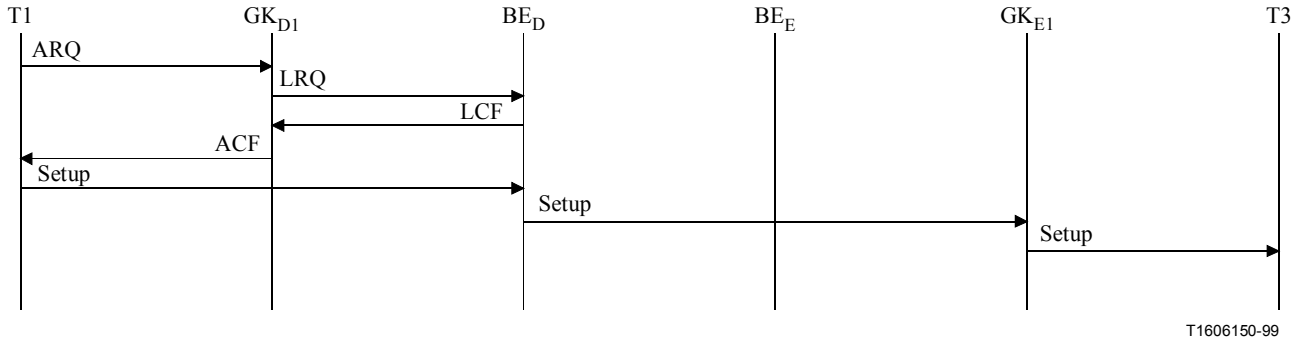


Figure G.20/H.225.0

Dans l'exemple de la Figure G.21 le résolveur d'adressage valide l'appel pour le domaine administratif de destination. Le résolveur d'adressage demande en outre aux éléments frontière d'origine et de destination d'envoyer des indications d'utilisation concernant l'appel.

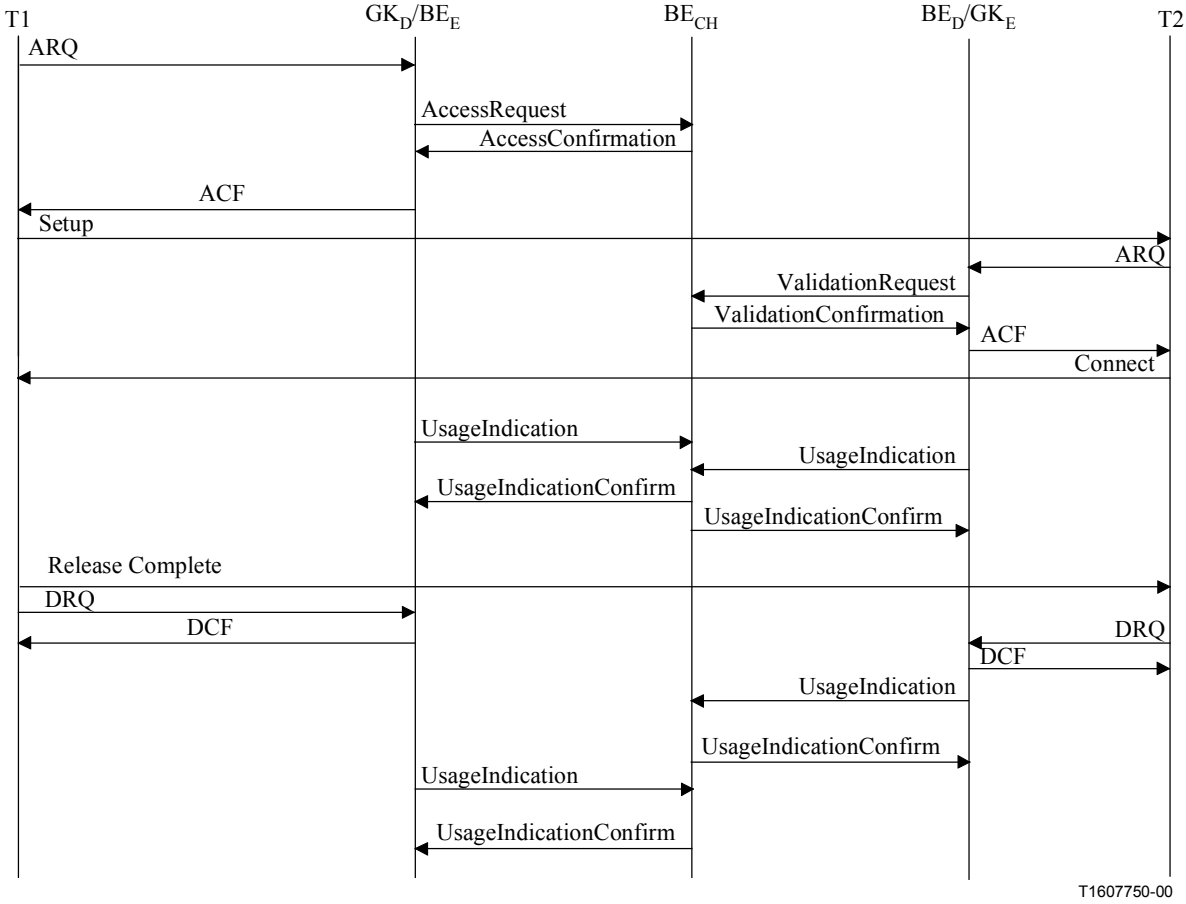


Figure G.21/H.225.0

Syntaxe des messages

```
ANNEXG-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
IMPORTS
      AuthenticationMechanism,
      TimeStamp,
      ClearToken
      FROM H235-SECURITY-MESSAGES

      AliasAddress,
      TransportAddress,
      ReleaseCompleteReason,
      ConferenceIdentifier,   CallIdentifier,   CryptoH323Token,  CryptoToken,

      EndpointType,
      GatekeeperIdentifier,
      GloballyUniqueID,
      NonStandardParameter,
      NumberDigits,
      PartyNumber,
      TransportQOS,
      VendorIdentifier,
      IntegrityMechanism,
      ICV
      FROM H323-MESSAGES;
```

```
Message ::= SEQUENCE
{
      body  AnnexGMessageBody,
      common      AnnexGCommonInfo,
      ...
}

AnnexGMessageBody ::= CHOICE
{
      serviceRequest                ServiceRequest,
      serviceConfirmation           ServiceConfirmation,
      serviceRejection              ServiceRejection,
      serviceRelease                ServiceRelease,
      descriptorRequest             DescriptorRequest,
      descriptorConfirmation        DescriptorConfirmation,
      descriptorRejection           DescriptorRejection,
      descriptorIDRequest           DescriptorIDRequest,
      descriptorIDConfirmation      DescriptorIDConfirmation,
      descriptorIDRejection         DescriptorIDRejection,
      descriptorUpdate              DescriptorUpdate,
      descriptorUpdateAck           DescriptorUpdateAck,
      accessRequest                 AccessRequest,
      accessConfirmation            AccessConfirmation,
      accessRejection               AccessRejection,
      requestInProgress             RequestInProgress,
      nonStandardRequest            NonStandardRequest,
      nonStandardConfirmation       NonStandardConfirmation,
      nonStandardRejection          NonStandardRejection,
      unknownMessageResponse        UnknownMessageResponse,
      usageRequest                  UsageRequest,
      usageConfirmation             UsageConfirmation,
      usageIndication               UsageIndication,
      usageIndicationConfirmation   UsageIndicationConfirmation,
      usageIndicationRejection      UsageIndicationRejection,
      usageRejection                UsageRejection,
      validationRequest             ValidationRequest,
      validationConfirmation        ValidationConfirmation,
      validationRejection           ValidationRejection,
      ...
}

AnnexGCommonInfo ::= SEQUENCE
{
      sequenceNumber          INTEGER (0..65535),
      version                 AnnexGVersion,
      hopCount                INTEGER (1..255),
      replyAddress            SEQUENCE OF TransportAddress OPTIONAL,
                              -- Doit être présent dans la demande
      integrityCheckValue     ICV OPTIONAL,
      tokens                  SEQUENCE OF ClearToken OPTIONAL,
      cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
      nonStandard             SEQUENCE OF NonStandardParameter OPTIONAL,
      ...
}



--
-- messages de l'Annexe G
--
```

```
ServiceRequest ::= SEQUENCE
{
      elementIdentifier       ElementIdentifier OPTIONAL,
      domainIdentifier  AliasAddress OPTIONAL,
      securityMode      SEQUENCE OF SecurityMode OPTIONAL,
      timeToLive        INTEGER (1..4294967295) OPTIONAL,
      ...
}

SecurityMode ::= SEQUENCE
{
      authentication    AuthenticationMechanism OPTIONAL,
      integrity         IntegrityMechanism OPTIONAL,
      algorithmOIDs     SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
      ...
}

ServiceConfirmation ::= SEQUENCE
{
      elementIdentifier ElementIdentifier,
      domainIdentifier  AliasAddress,
      alternates        AlternateBEInfo OPTIONAL,
      securityMode      SecurityMode OPTIONAL,
      timeToLive        INTEGER (1..4294967295) OPTIONAL,
      ...
}

ServiceRejection ::= SEQUENCE
{
      reason            ServiceRejectionReason,
      alternates        AlternateBEInfo OPTIONAL,
      ...
}

ServiceRejectionReason ::= CHOICE
{
      serviceUnavailable      NULL,
      serviceRedirected       NULL,
      security                NULL,
      continue                NULL,
      undefined               NULL,
      ...
}


ServiceRelease ::= SEQUENCE
{
      reason            ServiceReleaseReason,
      alternates        AlternateBEInfo OPTIONAL,
      ...
}

ServiceReleaseReason ::= CHOICE
{
      outOfService      NULL,
      maintenance       NULL,
      terminated        NULL,
      expired           NULL,
      ...
}
```

```
DescriptorRequest ::= SEQUENCE
{
      descriptorID      SEQUENCE OF DescriptorID,
      ...
}


DescriptorConfirmation ::= SEQUENCE
{
      descriptor        SEQUENCE OF Descriptor,
      ...
}


DescriptorRejection ::= SEQUENCE
{
      reason                  DescriptorRejectionReason,
      descriptorID            DescriptorID OPTIONAL,
      ...
}

DescriptorRejectionReason ::= CHOICE
{
      packetSizeExceeded      NULL, -- utiliser un autre type de transport
      illegalID               NULL, -- pas de descripteur pour l'identificateur
                                    -- fourni
      security          NULL, -- la demande n'a pas satisfait aux prescriptions
                              -- de sécurité
      hopCountExceeded  NULL,
      noServiceRelationship NULL,
      undefined         NULL,
      ...
}




DescriptorIDRequest ::= SEQUENCE
{
      ...
}


DescriptorIDConfirmation ::= SEQUENCE
{
      descriptorInfo          SEQUENCE OF DescriptorInfo,
      ...
}


DescriptorIDRejection ::= SEQUENCE
{
      reason            DescriptorIDRejectionReason,
      ...
}

DescriptorIDRejectionReason ::= CHOICE
{
      noDescriptors           NULL, -- pas de descripteurs pour le compte rendu
      security          NULL, -- la demande n'a pas satisfait aux prescriptions
                              -- de sécurité
      hopCountExceeded  NULL,
      noServiceRelationship   NULL,
```

```
      undefined         NULL,
      ...
}


DescriptorUpdate ::= SEQUENCE
{
      sender            AliasAddress,
      updateInfo        SEQUENCE OF UpdateInformation,
      ...
}

UpdateInformation ::=   SEQUENCE
{
      descriptorInfo    CHOICE      {
            descriptorID      DescriptorID,
            descriptor        Descriptor,
            ...
      },
      updateType  CHOICE
      {
            added             NULL,
            deleted           NULL,
            changed           NULL,
            ...
      },
      ...
}

DescriptorUpdateAck ::= SEQUENCE
{
      ...
}


AccessRequest ::= SEQUENCE
{
      destinationInfo   PartyInformation,
      sourceInfo        PartyInformation OPTIONAL,
      callInfo          CallInformation OPTIONAL,
      usageSpec         UsageSpecification OPTIONAL,  ...
}


AccessConfirmation ::= SEQUENCE
{
      templates         SEQUENCE OF AddressTemplate,
      partialResponse   BOOLEAN,
      ...
}


AccessRejection ::= SEQUENCE
{
      reason                  AccessRejectionReason,
      ...
}

AccessRejectionReason ::= CHOICE
{
      noMatch                 NULL, -- aucun canevas ne convient pour
                                    -- l'information de destination
      packetSizeExceeded      NULL, -- utiliser un autre type de transport
```

```
      security          NULL, -- la demande n'a pas satisfait aux prescriptions
                              -- de sécurité
      hopCountExceeded  NULL,
      needCallInformation     NULL,       -- l'information d'appel doit être
                                          -- spécifiée
      noServiceRelationship   NULL,
      undefined         NULL,
      ...
}

UsageRequest ::= SEQUENCE
{
      callInfo    CallInformation,
      usageSpec   UsageSpecification,
      ...
}


UsageConfirmation ::= SEQUENCE
{
      ...
}

UsageRejection ::= SEQUENCE
{
      reason                  UsageRejectReason,
      ...
}

UsageIndication ::= SEQUENCE
{
      callInfo                CallInformation,
      accessTokens            SEQUENCE OF AccessToken OPTIONAL,
      senderRole              Role,
      usageCallStatus         UsageCallStatus,
      srcInfo                 PartyInformation OPTIONAL,
      destAddress             PartyInformation,
      startTime               TimeStamp OPTIONAL,
      endTime                 TimeStamp OPTIONAL,
      terminationCause        TerminationCause OPTIONAL,
      usageFields             SEQUENCE OF UsageField,
      ...
}

UsageField ::= SEQUENCE
{
      id                      OBJECT IDENTIFIER,
      value                   OCTET STRING,
      ...
}



UsageRejectReason ::= CHOICE
{
      invalidCall             NULL,
      unavailable             NULL,
      security                NULL,
      noServiceRelationship   NULL,
      undefined               NULL,
      ...
}
```

```
UsageIndicationConfirmation ::= SEQUENCE
{
      ...
}

UsageIndicationRejection ::= SEQUENCE
{
      reason                  UsageIndicationRejectionReason,
      ...
}

UsageIndicationRejectionReason ::= CHOICE
{
      unknownCall       NULL,
      incomplete        NULL,
      security          NULL,
      noServiceRelationship   NULL,
      undefined         NULL,
      ...
}

ValidationRequest ::= SEQUENCE
{
      accessToken       SEQUENCE OF AccessToken OPTIONAL,
      destinationInfo   PartyInformation OPTIONAL,
      sourceInfo        PartyInformation OPTIONAL,
      callInfo          CallInformation,
      usageSpec         UsageSpecification OPTIONAL,
      ...
}


ValidationConfirmation ::= SEQUENCE
{
      destinationInfo   PartyInformation OPTIONAL,
      usageSpec         UsageSpecification OPTIONAL,
      ...
}


ValidationRejection ::= SEQUENCE
{
      reason                  ValidationRejectionReason,
      ...
}

ValidationRejectionReason ::= CHOICE
{
      tokenNotValid           NULL,
      security                NULL, -- la demande n'a pas satisfait aux
                                    -- prescriptions de sécurité
      hopCountExceeded        NULL,
      missingSorceInfo        NULL,
      missingDestInfo         NULL,
      noServiceRelationship   NULL,
      undefined               NULL,
      ...
}

RequestInProgress ::= SEQUENCE
{
      delay       INTEGER (1..65535),
      ...
}
```

```
NonStandardRequest ::= SEQUENCE
{
      ...
}

NonStandardConfirmation ::= SEQUENCE
{
      ...
}

NonStandardRejection ::= SEQUENCE
{
      reason            NonStandardRejectionReason,
      ...
}

NonStandardRejectionReason ::= CHOICE
{
      notSupported            NULL,
      noServiceRelationship   NULL,
      undefined         NULL,
      ...
}



UnknownMessageResponse ::= SEQUENCE
{
      unknownMessage          OCTET STRING,
      reason                  UnknownMessageReason,
      ...
}

UnknownMessageReason ::= CHOICE
{
      notUnderstood           NULL,
      undefined         NULL,
      ...
}







--
-- structures communes à plusieurs messages
--

AddressTemplate ::= SEQUENCE
{
      pattern           SEQUENCE OF Pattern,
      routeInfo         SEQUENCE OF RouteInformation,
      timeToLive        INTEGER (1..4294967295),
      ...
}
```

```
Pattern ::= CHOICE
{
      specific          AliasAddress,
      wildcard          AliasAddress,
      range       SEQUENCE          {
            startOfRange      PartyNumber,
            endOfRange  PartyNumber
      },
      ...
}

RouteInformation ::= SEQUENCE
{
      messageType CHOICE
      {
            sendAccessRequest NULL,
            sendSetup         NULL,
            nonExistent       NULL,
            ...
      },
      callSpecific      BOOLEAN,
      usageSpec   UsageSpecification OPTIONAL,
      priceInfo   SEQUENCE OF PriceInfoSpec OPTIONAL,
      contacts    SEQUENCE OF ContactInformation,
      type        EndpointType OPTIONAL,
                  -- doit être présent si messageType = sendSetup
      ...}
ContactInformation ::= SEQUENCE{    transportAddress  AliasAddress,     priority
            INTEGER (0..127), transportQoS      TransportQOS OPTIONAL,
      security          SEQUENCE OF SecurityMode OPTIONAL,
      accessTokens            SEQUENCE OF AccessToken OPTIONAL,
      ...
}


PriceInfoSpec ::= SEQUENCE
{
      currency                IA5String (SIZE(3)),    -- par exemple "USD"
      currencyScale                 INTEGER(-127..127),
      validFrom               GlobalTimeStamp OPTIONAL,
      validUntil              GlobalTimeStamp OPTIONAL,
      hoursFrom               IA5String (SIZE(6)) OPTIONAL, -- heure UTC "HHMMSS"
      hoursUntil              IA5String (SIZE(6)) OPTIONAL, -- heure UTC "HHMMSS"
      priceElement                  SEQUENCE OF PriceElement OPTIONAL,
      priceFormula                  IA5String (SIZE(1..2048)) OPTIONAL,
      ...
}

PriceElement ::= SEQUENCE
{
      amount                        INTEGER(0..4294967295), -- incrément de
                                                            -- compteur
      quantum                       INTEGER(0..4294967295), -- pour chaque
                                                            -- quantum ou
                                                            -- fraction de
                                                            -- quantum
      units CHOICE
      {
            seconds                 NULL,
            packets                 NULL,
            bytes                   NULL,
            initial                 NULL,
            minimum                 NULL,
```

```
            maximum                 NULL,
            ...
      },
      ...
}



Descriptor ::= SEQUENCE
{
      descriptorInfo          DescriptorInfo,
      templates         SEQUENCE OF AddressTemplate,
      gatekeeperID            GatekeeperIdentifier OPTIONAL,
      ...
}

DescriptorInfo ::= SEQUENCE
{
      descriptorID            DescriptorID,
      lastChanged       GlobalTimeStamp,
      ...
}


AlternateBEInfo ::= SEQUENCE
{
      alternateBE       SEQUENCE OF AlternateBE,
      alternateIsPermanent    BOOLEAN,
      ...
}

AlternateBE ::= SEQUENCE
{
      contactAddress          AliasAddress,
      priority          INTEGER (1..127),
      elementIdentifier ElementIdentifier OPTIONAL,
      ...
}

AccessToken ::= CHOICE
{
      token       ClearToken,
      cryptoToken CryptoH323Token,
      ...
}

CallInformation ::= SEQUENCE
{
      callIdentifier          CallIdentifier,
      conferenceID            ConferenceIdentifier,
      ...
}


UsageCallStatus ::= CHOICE
{
      preConnect              NULL, -- La communication n'a pas commencé
      callInProgress                NULL, -- Appel en cours
      callEnded               NULL, -- Fin d'appel
      ...
}
```

```
UserInformation ::= SEQUENCE
{
      userIdentifier          AliasAddress,
      userAuthenticator SEQUENCE OF CryptoH323Token OPTIONAL,
      ...
}


UsageSpecification ::= SEQUENCE
{
      sendTo                  ElementIdentifier,
      when SEQUENCE
      {
            never       NULL OPTIONAL,
            start       NULL OPTIONAL,
            end         NULL OPTIONAL,
            period            INTEGER(1..65535) OPTIONAL,   -- en secondes
            failures    NULL OPTIONAL,
            ...
      },
      required          SEQUENCE OF OBJECT IDENTIFIER,
      preferred         SEQUENCE OF OBJECT IDENTIFIER,
      ...
}


PartyInformation ::= SEQUENCE
{
      logicalAddresses  SEQUENCE OF AliasAddress,
      domainIdentifier  AliasAddress OPTIONAL,
      transportAddress  AliasAddress OPTIONAL,
      endpointType            EndpointType OPTIONAL,
      userInfo          UserInformation OPTIONAL,
      timeZone          TimeZone OPTIONAL,
      ...
}


Role ::= CHOICE
{
      originator        NULL,
      destination       NULL,
      nonStandardData   NonStandardParameter,
      ...
}

TimeZone ::=  INTEGER (-43200..43200)
                                          -- nombre de secondes relatives au
                                          -- temps UTC
                                          -- y compris le DST le cas échéant


TerminationCause ::= SEQUENCE
{
      releaseCompleteReason         ReleaseCompleteReason,
      causeIE                       INTEGER (1..65535) OPTIONAL,
      nonStandardData               NonStandardParameter OPTIONAL,
      ...
}
```

```
AnnexGVersion           ::=   OBJECT IDENTIFIER
                        -- doit être positionné sur la valeur
                        -- {itu-t (0) recommendation (0) h(8) h225.0(2250)
                        -- Annex (1) G (7) version (0) 1 (0)}
DescriptorID            ::=   GloballyUniqueID

ElementIdentifier ::=   BMPString (SIZE(1..128))

GlobalTimeStamp   ::=   IA5String (SIZE(14))    -- sous la forme YYYYMMDDHHmmSS
                                  -- avec YYYY = année, MM = mois, DD = jour,
                                  -- HH = heure, mm = minute, SS = seconde
                                  -- (par exemple, 19981219120000 pour minuit
                                  -- le 19 décembre 1998)



END   -- des messages de l'Annexe G
```

ANNEXE H

Syntaxe des messages H.225.0 (ASN.1)

La présente Recommandation définit les protocoles pour les messages RAS (essentiellement un protocole de portier) et pour la signalisation d'appel (essentiellement des unités de données de protocolaires qui résident dans un élément d'information d'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 antérieurs.

```
H323-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN


IMPORTS
              SIGNED{},
              ENCRYPTED{},
              HASHED{},
              ChallengeString,
              TimeStamp,
              RandomVal,
              Password,
              EncodedPwdCertToken,
              ClearToken,
              CryptoToken,
              AuthenticationMechanism
       FROM H235-SECURITY-MESSAGES
              DataProtocolCapability,
              T38FaxProfile
       FROM MULTIMEDIA-SYSTEM-CONTROL;

H323-UserInformation ::= SEQUENCE           -- root for all Q.931 related ASN.1
{
       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            -- used when a FACILITY message is sent,
                                                        -- but the Facility-UUIE is not to be invoked
                                                        -- (possible when transporting supplementary
                                                        -- services messages)
        },
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        h4501SupplementaryService       SEQUENCE OF OCTET STRING OPTIONAL,
                                                -- each sequence of octet string is defined as one
                                                -- H4501SupplementaryService APDU as defined in
                                                -- Table 3/H.450.1

        h245Tunneling                   BOOLEAN,
                                                -- if TRUE, tunneling of H.245 messages is enabled
        h245Control                     SEQUENCE OF OCTET STRING OPTIONAL,
                                                -- each octet string may contain exactly
                                                -- one H.245 PDU
        nonStandardControl              SEQUENCE OF NonStandardParameter 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,
        screeningIndicator      ScreeningIndicator
}

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
}

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,
                                        -- RFC1766 language tag
        connectedAddress        SEQUENCE OF AliasAddress OPTIONAL,
        presentationIndicator   PresentationIndicator,
```

```
        screeningIndicator      ScreeningIndicator
}

Information-UUIE ::= SEQUENCE
{
        protocolIdentifier      ProtocolIdentifier,
        ...,
        callIdentifier          CallIdentifier,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart               SEQUENCE OF OCTET STRING 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,
        screeningIndicator      ScreeningIndicator
}

ReleaseCompleteReason ::= CHOICE
{
        noBandwidth                     NULL,   -- bandwidth taken away or ARQ denied
        gatekeeperResources             NULL,   -- exhausted
        unreachableDestination          NULL,   -- no transport path to the destination
        destinationRejection            NULL,   -- rejected at destination
        invalidRevision                 NULL,
        noPermission                    NULL,   -- called party's gatekeeper rejects
        unreachableGatekeeper           NULL,   -- terminal cannot reach gatekeeper for ARQ
        gatewayResources                NULL,
        badFormatAddress                NULL,
        adaptiveBusy                    NULL,   -- call is dropping due to LAN crowding
        inConf                          NULL,   -- no address in AlternativeAddress
        undefinedReason                 NULL,
        ...,
        facilityCallDeflection          NULL,   -- call was deflected using a Facility message
        securityDenied                  NULL,   -- incompatible security settings
        calledPartyNotRegistered        NULL,   -- used by gatekeeper when endpoint has
                                                -- preGrantedARQ to bypass ARQ/ACF
        callerNotRegistered             NULL,   -- used by gatekeeper when endpoint has
                                                -- preGrantedARQ to bypass ARQ/ACF
        newConnectionNeeded             NULL,   -- indicates that the Setup was not accepted on this
                                                -- connection, but that the Setup may be accepted on
                                                -- a new connection
        nonStandardReason               NonStandardParameter,
        replaceWithConferenceInvite     ConferenceIdentifier  -- call dropped due to subsequent
                                                              -- invitation to a conference
                                                              -- (see 8.4.3.8/H.323)
}

Setup-UUIE ::= SEQUENCE
{
        protocolIdentifier              ProtocolIdentifier,
        h245Address                     TransportAddress OPTIONAL,
        sourceAddress                   SEQUENCE OF AliasAddress OPTIONAL,
        sourceInfo                      EndpointType,
        destinationAddress              SEQUENCE OF AliasAddress OPTIONAL,
        destCallSignalAddress           TransportAddress OPTIONAL,
        destExtraCallInfo               SEQUENCE OF AliasAddress OPTIONAL,          -- Note 1
        destExtraCRV                    SEQUENCE OF CallReferenceValue OPTIONAL,    -- Note 1
        activeMC                        BOOLEAN,
        conferenceID                    ConferenceIdentifier,
        conferenceGoal                  CHOICE
        {
                create                          NULL,
                join                            NULL,
                invite                          NULL,
                ...,
                capability-negotiation          NULL,
                callIndependentSupplementaryService         NULL
        },
        callServices                    QseriesOptions  OPTIONAL,
        callType                        CallType,
        ...,
```

```
        sourceCallSignalAddress       TransportAddress OPTIONAL,
        remoteExtensionAddress        AliasAddress OPTIONAL,
        callIdentifier                CallIdentifier,
        h245SecurityCapability        SEQUENCE OF H245Security OPTIONAL,
        tokens                        SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        fastStart                     SEQUENCE OF OCTET STRING OPTIONAL,
        mediaWaitForConnect           BOOLEAN,
        canOverlapSend                BOOLEAN,
        endpointIdentifier            EndpointIdentifier OPTIONAL,
        multipleCalls                 BOOLEAN,
        maintainConnection            BOOLEAN,
        connectionParameters          SEQUENCE       -- additional gateway parameters
        {
                connectionType                ScnConnectionType,
                numberOfScnConnections        INTEGER (0..65535),
                connectionAggregation         ScnConnectionAggregation,
                ...
        } OPTIONAL,
        language                      SEQUENCE OF IA5String(SIZE (1..32)) OPTIONAL,
                                              -- RFC1766 language tag
        presentationIndicator         PresentationIndicator,
        screeningIndicator            ScreeningIndicator
}

ScnConnectionType ::= CHOICE
{
        unknown         NULL,   -- should be seleceted when connection type is unknown
        bChannel        NULL,   -- each individual connection on the SCN is 64kbps.
                                        -- Note that where SCN delivers 56kbps usable data, the
                                        -- actual bandwidth allocated on SCN is still 64kbps.
        hybrid2x64      NULL,   -- each connection is a 128kbps hybrid call
        hybrid384       NULL,   -- each connection is an H0 (384kbps) hybrid call
        hybrid1536      NULL,   -- each connection is an H11 (1536kbps) hybrid call
        hybrid1920      NULL,   -- each connection is an H12 (1920kbps) hybrid call
        multirate       NULL,   -- bandwidth supplied by SCN using multirate.
                                        -- In this case, the information transfer rate octet in the
                                        -- bearer capability shall be set to multirate and the rate
                                        -- multiplier octet shall denote the number of B channels.
        ...
}

ScnConnectionAggregation ::= CHOICE
{
        auto            NULL,   -- aggregation mechanism is unknown
        none            NULL,   -- call produced using a single SCN connection
        h221            NULL,   -- use H.221 framing to aggregate the connections
        bonded-mode1    NULL,   -- use ISO/IEC 13871 bonding mode 1.
                                        -- Use bonded-mode1 to signal a bonded call if the precise
                                        -- bonding mode to be used is unknown.
        bonded-mode2    NULL,   -- use ISO/IEC 13871 bonding mode 2
        bonded-mode3    NULL,   -- use ISO/IEC 13871 bonding mode 3
        ...
}

PresentationIndicator ::= CHOICE
{
        presentationAllowed           NULL,
        presentationRestricted        NULL,
        addressNotAvailable           NULL,
        ...
}

ScreeningIndicator ::= ENUMERATED
{
        userProvidedNotScreened (0),
                -- number was provided by a remote user
                -- and has not been screened by a gatekeeper
        userProvidedVerifiedAndPassed (1),
                -- number was provided by a user
                -- equipment (or by a remote network), and has
                -- been screened by a gatekeeper
        userProvidedVerifiedAndFailed (2),
                -- not used, value reserved
        networkProvided (3),
                -- number was provided by a gatekeeper
        ...
}
```

```
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
}

ConferenceList ::= SEQUENCE
{
        conferenceID                    ConferenceIdentifier OPTIONAL,
        conferenceAlias                 AliasAddress OPTIONAL,
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...
}

FacilityReason ::= CHOICE
{
        routeCallToGatekeeper           NULL,           -- call must use gatekeeper model
                                                        -- gatekeeper is alternativeAddress
        callForwarded                   NULL,
        routeCallToMC                   NULL,
        undefinedReason                 NULL,
        ...,
        conferenceListChoice            NULL,
        startH245                       NULL,           -- recipient should connect to h245Address
        noH245                          NULL            -- endpoint does not support H.245
}

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
}

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,
        ...
}

EndpointType ::= SEQUENCE
{
        nonStandardData         NonStandardParameter OPTIONAL,
        vendor                  VendorIdentifier OPTIONAL,
        gatekeeper              GatekeeperInfo OPTIONAL,
        gateway                 GatewayInfo OPTIONAL,
        mcu                     McuInfo OPTIONAL,       -- mc must be set as well
        terminal                TerminalInfo OPTIONAL,
        mc                      BOOLEAN,                -- shall not be set by itself
        undefinedNode           BOOLEAN,
        ...,
        set                     BIT STRING (SIZE(32)) OPTIONAL
                                        -- shall not be used with mc, gatekeeper
                                        -- code points for the various SET devices
                                        -- are defined in the respective SET Annexes
}

GatewayInfo ::= SEQUENCE
{
        protocol                        SEQUENCE OF SupportedProtocols OPTIONAL,
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...
}

SupportedProtocols ::= CHOICE
{
        nonStandardData                 NonStandardParameter,
        h310                            H310Caps,
        h320                            H320Caps,
        h321                            H321Caps,
        h322                            H322Caps,
        h323                            H323Caps,
        h324                            H324Caps,
        voice                           VoiceCaps,
        t120-only                       T120OnlyCaps,
        ...,
        nonStandardProtocol             NonStandardProtocol,
        t38FaxAnnexbOnly                T38FaxAnnexbOnlyCaps
}

H310Caps ::= SEQUENCE
{
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported              SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes               SEQUENCE OF SupportedPrefix
}

H320Caps ::= SEQUENCE
{
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported              SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes               SEQUENCE OF SupportedPrefix
}

H321Caps ::= SEQUENCE
{
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported              SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes               SEQUENCE OF SupportedPrefix
}

H322Caps ::= SEQUENCE
{
```

```
        nonStandardData             NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported          SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes           SEQUENCE OF SupportedPrefix
}

H323Caps ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported          SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes           SEQUENCE OF SupportedPrefix
}

H324Caps ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported          SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes           SEQUENCE OF SupportedPrefix
}

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

T120OnlyCaps ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...,
        dataRatesSupported          SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes           SEQUENCE OF SupportedPrefix
}

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

T38FaxAnnexbOnlyCaps ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        dataRatesSupported          SEQUENCE OF DataRate OPTIONAL,
        supportedPrefixes           SEQUENCE OF SupportedPrefix,
        t38FaxProtocol              DataProtocolCapability,
        t38FaxProfile               T38FaxProfile,
        ...
}

McuInfo ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...
}

TerminalInfo ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...
}

GatekeeperInfo ::= SEQUENCE
{
        nonStandardData             NonStandardParameter OPTIONAL,
        ...
}

VendorIdentifier ::= SEQUENCE
{
        vendor                  H221NonStandard,
        productId               OCTET STRING (SIZE(1..256)) OPTIONAL, -- per vendor
        versionId               OCTET STRING (SIZE(1..256)) OPTIONAL, -- per product
```

```
        ...
}

H221NonStandard ::= SEQUENCE
{       t35CountryCode          INTEGER(0..255),        -- country, as per T.35
        t35Extension            INTEGER(0..255),        -- assigned nationally
        manufacturerCode        INTEGER(0..65535),      -- assigned nationally
        ...
}

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

NonStandardIdentifier          ::= CHOICE
{
        object                 OBJECT IDENTIFIER,
        h221NonStandard        H221NonStandard,
        ...
}

AliasAddress ::= CHOICE
{
        e164                   IA5String (SIZE (1..128)) (FROM ("0123456789#*,")),
        h323-ID                BMPString (SIZE (1..256)),    -- Basic ISO/IEC 10646-1 (Unicode)
        ...,
        url-ID                 IA5String (SIZE(1..512)),     -- URL style address
        transportID            TransportAddress,
        email-ID               IA5String (SIZE(1..512)),     -- rfc822-compliant email address
        partyNumber            PartyNumber
}

PartyNumber ::= CHOICE
{
        publicNumber                    PublicPartyNumber,
                                                -- the numbering plan is according to
                                                -- Recommendations E.163 and E.164.
        dataPartyNumber                 NumberDigits,
                                                -- not used, value reserved.
        telexPartyNumber                NumberDigits,
                                                -- not used, value reserved.
        privateNumber                   PrivatePartyNumber,
        nationalStandardPartyNumber     NumberDigits,
                                                -- not used, value reserved.
        ...
}

PublicPartyNumber ::= SEQUENCE
{
        publicTypeOfNumber     PublicTypeOfNumber,
        publicNumberDigits     NumberDigits
}

PrivatePartyNumber ::= SEQUENCE
{
        privateTypeOfNumber    PrivateTypeOfNumber,
        privateNumberDigits    NumberDigits
}

NumberDigits ::= IA5String (SIZE (1..128)) (FROM ("0123456789#*,"))

PublicTypeOfNumber ::= CHOICE
{
        unknown                NULL,
                                        -- if used number digits carry prefix indicating type
                                        -- of number according to national recommendations.
        internationalNumber    NULL,
        nationalNumber         NULL,
        networkSpecificNumber  NULL,
                                        -- not used, value reserved
        subscriberNumber       NULL,
        abbreviatedNumber      NULL,
                                        -- valid only for called party number at the outgoing
                                        -- access, network substitutes appropriate number.
...
}
```

```
PrivateTypeOfNumber ::= CHOICE
{
        unknown                 NULL,
        level2RegionalNumber    NULL,
        level1RegionalNumber    NULL,
        pISNSpecificNumber      NULL,
        localNumber             NULL,
        abbreviatedNumber       NULL,
        ...
}

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,
        ...
}

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,
        ...                             -- can be extended with other specific modes
}

SecurityCapabilities ::= SEQUENCE
{
        nonStandard                     NonStandardParameter OPTIONAL,
        encryption                      SecurityServiceMode,
        authenticaton                   SecurityServiceMode,
        integrity                       SecurityServiceMode,
        ...
}

H245Security ::= CHOICE
{
        nonStandard                     NonStandardParameter,
        noSecurity                      NULL,
        tls                             SecurityCapabilities,
        ipsec                           SecurityCapabilities,
        ...
}


QseriesOptions ::= SEQUENCE
{
        q932Full                BOOLEAN,        -- if true, indicates full support for Q.932
        q951Full                BOOLEAN,        -- if true, indicates full support for Q.951
        q952Full                BOOLEAN,        -- if true, indicates full support for Q.952
        q953Full                BOOLEAN,        -- if true, indicates full support for Q.953
        q955Full                BOOLEAN,        -- if true, indicates full support for Q.955
        q956Full                BOOLEAN,        -- if true, indicates full support for Q.956
        q957Full                BOOLEAN,        -- if true, indicates full support for 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)      -- in 100s of bits
CallReferenceValue      ::=     INTEGER (0..65535)
EndpointIdentifier      ::=     BMPString (SIZE(1..128))
ProtocolIdentifier      ::=     OBJECT IDENTIFIER
                                -- shall be set to
                                -- {itu-t (0) recommendation (0) h (8) 2250 version (0) 3}
TimeToLive              ::=     INTEGER (1..4294967295)       --in seconds

CallIdentifier ::= SEQUENCE
{
        guid                    GloballyUniqueID,
        ...
}


EncryptIntAlg ::= CHOICE
{       -- core encryption algorithms for RAS message integrity
        nonStandard             NonStandardParameter,
        isoAlgorithm            OBJECT IDENTIFIER,     -- defined in ISO/IEC 9979
        ...
}
NonIsoIntegrityMechanism ::= CHOICE
{       -- HMAC mechanism used, no truncation, tagging may be necessary!
        hMAC-MD5                NULL,
        hMAC-iso10118-2-s       EncryptIntAlg, -- according to ISO/IEC 10118-2 using
                                               -- EncryptIntAlg as core block encryption algorithm
                                               -- (short MAC)
        hMAC-iso10118-2-l       EncryptIntAlg, -- according to ISO/IEC 10118-2 using
                                               -- EncryptIntAlg as core block encryption algorithm
                                               -- (long MAC)
        hMAC-iso10118-3         OBJECT IDENTIFIER,     -- according to ISO/IEC 10118-3 using
                                                       -- OID as hash function (OID is SHA-1,
                                                       -- RIPE-MD160,
                                                       -- RIPE-MD128)
        ...
}

IntegrityMechanism ::= CHOICE
{       -- for RAS message integrity
        nonStandard             NonStandardParameter,
        digSig                  NULL,                  -- indicates to apply a digital signature
        iso9797                 OBJECT IDENTIFIER,     -- according to ISO/IEC 9797 using OID as
                                                       -- core encryption algorithm (X-CBC MAC)
        nonIsoIM                NonIsoIntegrityMechanism,
        ...
}

ICV ::= SEQUENCE
{
        algorithmOID    OBJECT IDENTIFIER,     -- the algorithm used to compute the signature
        icv             BIT STRING             -- the computed cryptographic integrity check value
                -- or signature
}

FastStartToken ::= ClearToken (WITH COMPONENTS {..., timeStamp PRESENT, dhkey PRESENT, generalID
PRESENT -- set to 'alias' -- })
EncodedFastStartToken ::= TYPE-IDENTIFIER.&Type (FastStartToken)
CryptoH323Token::= CHOICE
{
        cryptoEPPwdHash SEQUENCE
        {
                alias           AliasAddress,  -- alias of entity generating hash
                timeStamp       TimeStamp,             -- timestamp used in hash
                token           HASHED  { EncodedPwdCertToken -- generalID set to 'alias' -- }
        },
```

```
        cryptoGKPwdHash  SEQUENCE
        {
                gatekeeperId   GatekeeperIdentifier,  -- GatekeeperID of GK generating hash
                timeStamp      TimeStamp,             -- timestamp used in hash
                token          HASHED  { EncodedPwdCertToken  -- generalID set to Gatekeeperid -- }
        },
        cryptoEPPwdEncr        ENCRYPTED { EncodedPwdCertToken -- generalID set to Gatekeeperid --},
        cryptoGKPwdEncr        ENCRYPTED { EncodedPwdCertToken -- generalID set to Gatekeeperid --},
        cryptoEPCert           SIGNED { EncodedPwdCertToken -- generalID set to Gatekeeperid -- },
        cryptoGKCert           SIGNED { EncodedPwdCertToken -- generalID set to alias -- },
        cryptoFastStart        SIGNED { EncodedFastStartToken },
        nestedcryptoToken      CryptoToken,
        ...
}


DataRate ::= SEQUENCE
{
        nonStandardData                NonStandardParameter OPTIONAL,
        channelRate                    BandWidth,
        channelMultiplier              INTEGER (1..256) OPTIONAL,
        ...
}

SupportedPrefix ::= SEQUENCE
{
        nonStandardData                NonStandardParameter OPTIONAL,
        prefix                         AliasAddress,
        ...
}


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
}

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
}

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
}

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
}

GatekeeperRejectReason ::= CHOICE
{
        resourceUnavailable             NULL,
        terminalExcluded                NULL,   -- permission failure, not a resource failure
        invalidRevision                 NULL,
        undefinedReason                 NULL,
        ...,
        securityDenial                  NULL
}

RegistrationRequest ::= SEQUENCE --(RRQ)
{
        requestSeqNum                   RequestSeqNum,
        protocolIdentifier              ProtocolIdentifier,
        nonStandardData                 NonStandardParameter OPTIONAL,
        discoveryComplete               BOOLEAN,
        callSignalAddress               SEQUENCE OF TransportAddress,
        rasAddress                      SEQUENCE OF TransportAddress,
        terminalType                    EndpointType,
        terminalAlias                   SEQUENCE OF AliasAddress OPTIONAL,
        gatekeeperIdentifier            GatekeeperIdentifier  OPTIONAL,
        endpointVendor                  VendorIdentifier,
        ...,
        alternateEndpoints              SEQUENCE OF Endpoint OPTIONAL,
        timeToLive                      TimeToLive OPTIONAL,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL,
        keepAlive                       BOOLEAN,
        endpointIdentifier              EndpointIdentifier OPTIONAL,
        willSupplyUUIEs                 BOOLEAN,
        maintainConnection              BOOLEAN,
        supportsAnnexECallSignalling    BOOLEAN
}

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,  -- in seconds; not
                                                                        -- present if GK
                                                                        -- does not want IRRs
                totalBandwidthRestriction               BandWidth OPTIONAL,    -- total limit for all
                                                                        -- concurrent calls
                useAnnexECallSignalling                 BOOLEAN
        } OPTIONAL,
        maintainConnection              BOOLEAN
}

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
}

RegistrationRejectReason ::= CHOICE
{
        discoveryRequired               NULL,
        invalidRevision                 NULL,
        invalidCallSignalAddress        NULL,
        invalidRASAddress               NULL,   -- supplied address is invalid
        duplicateAlias                  SEQUENCE OF AliasAddress,
                                                -- alias registered to another endpoint
        invalidTerminalType             NULL,
        undefinedReason                 NULL,
        transportNotSupported           NULL,   -- one or more of the transports
        ...,
        transportQOSNotSupported        NULL,   -- endpoint QOS not supported
        resourceUnavailable             NULL,   -- gatekeeper resources exhausted
        invalidAlias                    NULL,   -- alias not consistent with gatekeeper rules
        securityDenial                  NULL,
        fullRegistrationRequired        NULL    -- registration permission has expired
}

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
}

UnregRequestReason ::= CHOICE
{
        reregistrationRequired          NULL,
        ttlExpired                      NULL,
```

```
        securityDenial                NULL,
        undefinedReason               NULL,
        ...
}

UnregistrationConfirm ::= SEQUENCE --(UCF)
{
        requestSeqNum                 RequestSeqNum,
        nonStandardData               NonStandardParameter OPTIONAL,
        ...,
        tokens                        SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue           ICV 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
}

UnregRejectReason ::= CHOICE
{
        notCurrentlyRegistered        NULL,
        callInProgress                NULL,
        undefinedReason               NULL,
        ...,
        permissionDenied              NULL,   -- requesting user not allowed to unregister
                                              -- specified user
        securityDenial                NULL
}

AdmissionRequest ::= SEQUENCE --(ARQ)
{
        requestSeqNum                 RequestSeqNum,
        callType                      CallType,
        callModel                     CallModel OPTIONAL,
        endpointIdentifier            EndpointIdentifier,
        destinationInfo               SEQUENCE OF AliasAddress OPTIONAL,    -- Note 1
        destCallSignalAddress         TransportAddress OPTIONAL,            -- Note 1
        destExtraCallInfo             SEQUENCE OF AliasAddress OPTIONAL,
        srcInfo                       SEQUENCE OF AliasAddress,
        srcCallSignalAddress          TransportAddress OPTIONAL,
        bandWidth                     BandWidth,
        callReferenceValue            CallReferenceValue,
        nonStandardData               NonStandardParameter OPTIONAL,
        callServices                  QseriesOptions  OPTIONAL,
        conferenceID                  ConferenceIdentifier,
        activeMC                      BOOLEAN,
        answerCall                    BOOLEAN,       -- answering a call
        ...,
        canMapAlias                   BOOLEAN,       -- can handle alias address
        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
}

CallType ::= CHOICE
{
        pointToPoint          NULL,          -- Point-to-point
        oneToN                NULL,          -- no interaction (FFS)
        nToOne                NULL,          -- no interaction (FFS)
        nToN                  NULL,          -- interactive (multipoint)
        ...
}
```

```
CallModel ::= CHOICE
{
        direct                  NULL,
        gatekeeperRouted        NULL,
        ...
}

TransportQOS ::= CHOICE
{
        endpointControlled      NULL,
        gatekeeperControlled    NULL,
        noControl               NULL,
        ...
}

AdmissionConfirm ::= SEQUENCE --(ACF)
{
        requestSeqNum                   RequestSeqNum,
        bandWidth                       BandWidth,
        callModel                       CallModel,
        destCallSignalAddress           TransportAddress,
        irrFrequency                    INTEGER (1..65535) OPTIONAL,
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        destinationInfo                 SEQUENCE OF AliasAddress OPTIONAL,
        destExtraCallInfo               SEQUENCE OF AliasAddress OPTIONAL,
        destinationType                 EndpointType OPTIONAL,
        remoteExtensionAddress          SEQUENCE OF AliasAddress OPTIONAL,
        alternateEndpoints              SEQUENCE OF Endpoint OPTIONAL,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL,
        transportQOS                    TransportQOS OPTIONAL,
        willRespondToIRR                BOOLEAN,
        uuiesRequested                  UUIEsRequested,
        language                        SEQUENCE OF IA5String(SIZE (1..32)) OPTIONAL, -- RFC1766
language tag
        useAnnexECallSignalling         BOOLEAN
}

UUIEsRequested ::= SEQUENCE
{
        setup                           BOOLEAN,
        callProceeding                  BOOLEAN,
        connect                         BOOLEAN,
        alerting                        BOOLEAN,
        information                     BOOLEAN,
        releaseComplete                 BOOLEAN,
        facility                        BOOLEAN,
        progress                        BOOLEAN,
        empty                           BOOLEAN,
        ...
}

AdmissionReject ::= SEQUENCE --(ARJ)
{
        requestSeqNum           RequestSeqNum,
        rejectReason            AdmissionRejectReason,
        nonStandardData         NonStandardParameter OPTIONAL,
        ...,
        altGKInfo               AltGKInfo OPTIONAL,
        tokens                  SEQUENCE OF ClearToken OPTIONAL,
        callSignalAddress       SEQUENCE OF TransportAddress OPTIONAL,
        cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue     ICV OPTIONAL
}

AdmissionRejectReason ::= CHOICE
{
        calledPartyNotRegistered        NULL,   -- cannot translate address
        invalidPermission               NULL,   -- permission has expired
        requestDenied                   NULL,   -- no bandwidth available
        undefinedReason                 NULL,
        callerNotRegistered             NULL,
        routeCallToGatekeeper           NULL,
        invalidEndpointIdentifier       NULL,
        resourceUnavailable             NULL,
        ...,
        securityDenial                  NULL,
        qosControlNotSupported          NULL,
```

```
        incompleteAddress               NULL,
        routeCallToSCN                  SEQUENCE OF PartyNumber,
        aliasesInconsistent             NULL    -- multiple aliases in request identify distinct people
}

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
}

BandwidthConfirm ::= SEQUENCE --(BCF)
{
        requestSeqNum                   RequestSeqNum,
        bandWidth                       BandWidth,
        nonStandardData                 NonStandardParameter OPTIONAL,
        ...,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV 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
}

BandRejectReason ::= CHOICE
{
        notBound                        NULL,   -- discovery permission has aged
        invalidConferenceID             NULL,   -- possible revision
        invalidPermission               NULL,   -- true permission violation
        insufficientResources           NULL,
        invalidRevision                 NULL,
        undefinedReason                 NULL,
        ...,
        securityDenial                  NULL
}

LocationRequest ::= SEQUENCE --(LRQ)
{
        requestSeqNum                   RequestSeqNum,
        endpointIdentifier              EndpointIdentifier OPTIONAL,
        destinationInfo                 SEQUENCE OF AliasAddress,
        nonStandardData                 NonStandardParameter OPTIONAL,
        replyAddress                    TransportAddress,
        ...,
        sourceInfo                      SEQUENCE OF AliasAddress OPTIONAL,
        canMapAlias                     BOOLEAN,        -- can handle alias address
        gatekeeperIdentifier            GatekeeperIdentifier OPTIONAL,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV 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,
        supportsAnnexECallSignalling  BOOLEAN
}

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
}

LocationRejectReason ::= CHOICE
{
        notRegistered         NULL,
        invalidPermission     NULL,   -- exclusion by administrator or feature
        requestDenied         NULL,   -- cannot find location
        undefinedReason       NULL,
        ...,
        securityDenial        NULL,
        routeCallToSCN        SEQUENCE OF PartyNumber,
        aliasesInconsistent   NULL    -- multiple aliases in request identify distinct people
}

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
}

DisengageReason ::= CHOICE
{
        forcedDrop                    NULL,   -- gatekeeper is forcing the drop
        normalDrop                    NULL,   -- associated with normal drop
        undefinedReason               NULL,
        ...
}

DisengageConfirm ::= SEQUENCE --(DCF)
{
        requestSeqNum                 RequestSeqNum,
        nonStandardData               NonStandardParameter OPTIONAL,
        ...,
        tokens                        SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                  SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue           ICV 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
}

DisengageRejectReason ::= CHOICE
{
        notRegistered           NULL,   -- not registered with gatekeeper
        requestToDropOther      NULL,   -- cannot request drop for others
        ...,
        securityDenial          NULL
}

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
}

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         -- TRUE is sent, FALSE is received
                } OPTIONAL
        } OPTIONAL,
        ...,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL,
        needResponse                    BOOLEAN
}

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),
        ...
}

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,   -- not registered with gatekeeper
        securityDenial          NULL,
        undefinedReason         NULL,
        ...
}

NonStandardMessage ::= SEQUENCE
{
        requestSeqNum                   RequestSeqNum,
        nonStandardData                 NonStandardParameter,
        ...,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL
}

UnknownMessageResponse ::= SEQUENCE   -- (XRS)
{
        requestSeqNum                   RequestSeqNum,
        ...,
        tokens                          SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                    SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue             ICV OPTIONAL
}

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,
        ...
}
```

```
ResourcesAvailableConfirm ::= SEQUENCE --(RAC)
{
        requestSeqNum                RequestSeqNum,
        protocolIdentifier           ProtocolIdentifier,
        nonStandardData              NonStandardParameter OPTIONAL,
        tokens                       SEQUENCE OF ClearToken OPTIONAL,
        cryptoTokens                 SEQUENCE OF CryptoH323Token OPTIONAL,
        integrityCheckValue          ICV OPTIONAL,
        ...
}

END             -- of ASN.1
```

ANNEXE I

Groupage par paquets vidéo "H.263+"

La norme IETF RFC 2429 spécifie le format de charge utile RTP des trains de lits vidéo H.263 qui contiennent les nouvelles caractéristiques "H.263+" adoptées dans la version 2 (1998) de la Recommandation H.263 (avec les caractéristiques utilisant PLUSTYPE ou les Annexes I à T de la Recommandation 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 de la présente Recommandation est exigée pour les trains de bits H.263 qui n'utilisent pas les nouvelles caractéristiques de la version 2 de la Recommandation 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 trains 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.), JACOBSON (V.): RTP: A Transport Protocol for Real-Time Applications (RTP: un protocole de transport pour les applications en temps réel), RFC 1889, *Internet Engineering Task Force*, 1996.

APPENDICE II

Profil RTP

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

– SCHULZRINNE (H.): RTP Profile for Audio and Video Conferences with Minimal Control (profil RTP pour les conférences audio et vidéo avec commande minimale), RFC 1890, *Internet Engineering Task Force*, 1996.

APPENDICE III

Mise en paquets H.261

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

– TURLETTI (T.), HUITEMA (C.): RTP Payload Format for H.261 Video Stream (format de charge utile RTP pour les flux vidéo H.261), RFC 2032, *Internet Engineering Task Force*, 1996.

APPENDICE IV

Fonctionnement du mode H.225.0 sur différentes piles protocolaires de réseau en mode paquet

Le présent appendice donne des détails complémentaires concernant le fonctionnement du mode H.225.0 sur diverses piles protocolaires réelles de réseau à commutation par paquets. Les réseaux à commutation par paquets utilisés dans la présente Recommandation doivent fournir un mode de fonctionnement fiable et un mode de fonctionnement non fiable, comprenant un moyen de repérer les frontières de paquet.

IV.1 TCP/IP/UDP

Il convient de noter que le protocole UDP peut fragmenter et réassembler de grands paquets vidéo, mais que l'échec de la mise en paquets des macroblocs peut conduire à la perte d'un groupe de blocs entier.

La multidiffusion IP peut être utilisée pour la distribution GRQ par opposition à la diffusion de couche d'accès média.

<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</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 train TCP, qui assure lui-même un train continu d'octets sans limites précises. Un TPKT est constitué d'un champ de numéro de version d'un octet, suivi d'un champ réservé d'un octet, d'un champ de longueur de deux octets et des données effectives. Le champ de numéro de version doit contenir la valeur "3" et le champ réservé doit contenir la valeur "0". Le champ de longueur doit contenir la longueur de tout le paquet, y compris le numéro de version, le champ réservé et le champ de longueur sous forme de mot gros-boutiste de 16 bits.

IV.1.1 Recherche du portier

IV.1.1.1 Recherche au moyen d'une adresse de multidiffusion ou d'un accès connu

Après l'exécution des procédures de recherche et d'enregistrement du portier décrites au paragraphe 7/H.323, les points d'extrémité doivent utiliser l'adresse de multidiffusion ou l'accès connu suivant lorsqu'elle recherche le portier en fonction de leur configuration de réseau:

– adresseUDP 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

NOTE – Les termes "autre accord" peuvent désigner l'enregistrement d'un point d'extrémité auprès d'un portier.

A noter que, lors des implémentations, il faut faire attention à la portée de la multidiffusion de manière à ne pas inonder Internet avec des messages de recherche.

A supposer qu'un portier ait une adresse IP de type 134.134.12.1, par exemple, la signalisation suivante peut être établie:

arrivée d'un message LRQ ou GRQ à l'adresse 134.134.12.1: accès 1719.

arrivée d'un message LRQ ou GRQ à l'adresse 134.134.12.1 : accès 1718.
(à noter que cette situation peut se produire avec des portiers de version 1).

arrivée d'un message LRQ ou GRQ à l'adresse 224.0.1.41: accès 1718.

Le portier peut transmettre un message LRQ aux adresses suivantes:

224.0.1.41: accès 1718 (multidiffusion à destination de tous les portiers).

X.X.X.X: accès 1719 (à destination d'un portier donné).

L'accès 1719 ne doit être utilisé que lorsqu'une demande est envoyée en mode unidiffusion. Cela permet au destinataire de savoir s'il doit envoyer un refus (xRJ) à l'expéditeur (il doit le faire dans tous les cas)

L'accès 1718 ne doit être utilisé que lorsqu'une demande est envoyée en mode multidiffusion. Le destinataire doit envoyer une réponse appropriée, selon le message. Pour un message LRQ, aucun refus n'est nécessaire. Le destinataire ne répond pas aux demandes de multidiffusion. Pour un message GRQ, un message GRJ dirigé doit être envoyé à la source dont provient le message GRQ.

IV.1.1.2 Recherche au moyen d'un système DNS (à titre d'information)

IV.1.1.2.1 Adresse URL pour les portiers

D'abord, il convient de noter qu'un portier est identifié par une adresse de transport et un identificateur gatekeeperIdentifier, qui est une chaîne. Un portier étant une ressource particulière sur Internet, il est raisonnable de le spécifier avec un identificateur uniforme de ressources (URL, *uniform resource locator*). Le protocole utilisé par le portier étant le protocole RAS, l'adresse URL d'un portier pourrait être donnée par:

ras://gkID@domainname

gkID est l'identificateur gatekeeperIdentifier et domainname est un nom de domaine DNS qui identifie le domaine du portier. A noter qu'il ne s'agit pas nécessairement d'un nom de domaine complet (FQDN, *fully qualified domain name*) avec un enregistrement A – il n'est pas exigé que ce nom de domaine ait une interface de transport physique avec un numéro IP enregistré dans le système DNS. Toutefois, s'il s'agit d'un nom FQDN, il est raisonnable d'exiger que le numéro IP soit celui du portier auquel l'adresse URL se rapporte. Dans ce cas, l'adjonction d'un numéro d'accès facultatif à l'adresse URL est autorisée:

ras://gkID@domainname:port_no.

Si aucun numéro d'accès n'est donné, la valeur connue 1719 est prise comme valeur par défaut.

Le cas le plus intéressant se présente lorsqu'il ne s'agit pas d'un nom FQDN et que le nom de domaine ne se rapporte pas à une adresse de transport énumérée dans le système DNS. Le nom de domaine peut alors renvoyer à une simple "zone d'autorité du portier". Le sous-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 à la norme RFC 822 aux points d'extrémité 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 enregistrement de localisation de service (SRV, *service location record*). Etant donné un nom de domaine, une interrogation relative aux enregistrements SRV sera 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 gatekeeperIdentifier.

Cette solution simple sera bientôt normalisée. Le problème est que pratiquement aucune implémentation actuelle de serveur ou de client DNS ne prend encore en charge l'enregistrement de ressources SRV. A moins que le client DNS soit au courant du type d'enregistrement de ressources SRV, il lui est impossible de faire des interrogations relatives à de tels enregistrements de ressources. Tant que cette prise en charge ne sera pas généralisée, il y a de fortes chances pour que les interrogations relatives aux enregistrements SRV échouent.

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 seront renvoyés lorsqu'une interrogation sera 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 à la norme 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. Ceci 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 valables:

```
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 le corps de la présente Recommandation par la structure ASN.1 AlternateGK.

A noter que ce que le serveur renvoie dans une telle interrogation pourrait être une vraie adresse de transport en notation décimale pointée, ou un nom FQDN qui, lui-même, nécessite une interrogation relative aux enregistrements A dans le système DNS pour déterminer l'adresse de transport. L'avantage lié à l'utilisation d'un nom FQDN tient à la dissimulation habituelle des numéros IP effectifs. L'avantage lié à l'utilisation de numéros IP tient à ce qu'une seconde interrogation dans le système DNS est évitée, d'où une accélération du temps de préétablissement d'appel.

IV.1.1.2.3 Traitement par le portier des identificateurs email-ID pendant les demandes ARQ et LRQ

Lorsque le champ **destinationInfo** d'un message ARQ ou LRQ contient une adresse alias **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 générer un domaine par défaut. Le domaine sert alors à localiser un ou plusieurs portiers, au moyen des procédures du IV.1.1.2.2. Le portier peut alors interroger tous les portiers trouvés avec un échange de messages LRQ/LCF/LRJ.

A noter que plusieurs portiers peuvent avoir des enregistrements TXT correspondants dans un même domaine du système DNS. En conséquence, un même domaine du système DNS peut "contenir" plusieurs zones H.323. Ainsi, même si un portier ne peut pas résoudre un identificateur email-ID

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 RFC 822, le portier peut interpréter l'alias comme s'il s'agissait d'un identificateur email-ID 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 email-ID 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 point d'extrémité à point d'extrémité

Les points d'extrémité 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 le canal de signalisation d'appel:

– accès de signalisation d'appel TCP de point 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 points d'extrémité en dehors de la zone relevant du portier, sauf via une passerelle dans la zone.

IV.2 SPX/IPX

Il convient de noter que compte tenu de l'absence de réassemblage dans le réseau des grands paquets, l'utilisation de la fragmentation des macroblocs est essentielle.

Applications avec remise non fiable	**Canal H.245 et canal de signalisation d'appel**
PXP	SPX
IPX	
Couche Liaison	
Couche Physique	

IV.2.1 Découverte du portier

Dans la terminologie IPX, une "prise" ("socket") est équivalente à un accès dans IP et un "identificateur TSAP" dans la présente Recommandation et dans la Recommandation H.323.

Sur les réseaux de type IPX, les portiers doivent faire connaître le "type de service portier" défini ci-après pour permettre aux points d'extrémité de les localiser dans un réseau. De même, les points d'extrémité doivent demander à connaître le "type de service portier" pour localiser le portier le plus proche.

– type de service portier à étudier

NOTE – Le type de service est appelé prise SAP dans certains documents IPX.

IV.2.2 Communication de point d'extrémité à point d'extrémité

Les points d'extrémité 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 de point d'extrémité à 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 points d'extrémité en dehors de la zone relevant du portier, sauf lorsque cela s'effectue via une passerelle située dans la zone.

SERIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	**Systèmes audiovisuels et multimédias**
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux pour données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication