UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T J.162

(03/2001)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE J: RÉSEAUX CÂBLÉS ET TRANSMISSION DES SIGNAUX RADIOPHONIQUES, TÉLÉVISUELS ET AUTRES SIGNAUX MULTIMÉDIAS

IPCablecom

Protocole réseau de signalisation d'appel pour la fourniture de services à temps critique sur les réseaux de télévision par câble utilisant des câblo-modems

Recommandation UIT-T J.162

RECOMMANDATIONS UIT-T DE LA SÉRIE J

RÉSEAUX CÂBLÉS ET TRANSMISSION DES SIGNAUX RADIOPHONIQUES, TÉLÉVISUELS ET AUTRES SIGNAUX MULTIMÉDIAS

Recommandations générales	J.1–J.9
Spécifications générales des transmissions radiophoniques analogiques	J.10–J.19
Caractéristiques de fonctionnement des circuits radiophoniques analogiques	J.20–J.29
Equipements et lignes utilisés pour les circuits radiophoniques analogiques	J.30–J.39
Codeurs numériques pour les signaux radiophoniques analogiques	J.40–J.49
Transmission numérique de signaux radiophoniques	J.50–J.59
Circuits de transmission télévisuelle analogique	J.60–J.69
Transmission télévisuelle analogique sur lignes métalliques et interconnexion avec les faisceaux hertziens	J.70–J.79
Transmission numérique des signaux de télévision	J.80–J.89
Services numériques auxiliaires propres aux transmissions télévisuelles	J.90–J.99
Prescriptions et méthodes opérationnelles de transmission télévisuelle	J.100–J.109
Services interactifs pour la distribution de télévision numérique	J.110–J.129
Transport des signaux MPEG-2 sur les réseaux par paquets	J.130–J.139
Mesure de la qualité de service	J.140–J.149
Distribution de la télévision numérique sur les réseaux locaux d'abonnés	J.150–J.159
IPCablecom	**J.160–J.179**
Divers	J.180–J.199
Application à la télévision numérique interactive	J.200–J.209

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

Recommandation UIT-T J.162

Protocole réseau de signalisation d'appel pour la fourniture de services à temps critique sur les réseaux de télévision par câble utilisant des câblo-modems

Résumé

De nombreux exploitants de télévision par câble mettent actuellement leurs équipements à niveau afin d'intégrer une capacité bidirectionnelle qui permette de fournir des services de données IP à haut débit conformément aux Recommandations UIT-T J.83 et J.112. Ces exploitants souhaitent désormais élargir la capacité de cette plate-forme de diffusion pour y inclure divers services à temps critique. La présente Recommandation fait partie d'une série de Recommandations nécessaires pour atteindre cet objectif. Elle contient la description d'un protocole de signalisation d'appel par le réseau nécessaire pour établir des connexions.

Source

La Recommandation J.162 de l'UIT-T, élaborée par la Commission d'études 9 (2001-2004) de l'UIT-T, a été approuvée le 9 mars 2001 selon la procédure définie dans la Résolution 1 de l'AMNT.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas 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 UIT-T J.162

Protocole réseau de signalisation d'appel pour la fourniture de services à temps critique sur les réseaux de télévision par câble utilisant des câblo-modems

1 Domaine d'application

La présente Recommandation décrit une interface de programmation, nommée MGCI c'est-à-dire interface de contrôleur de passerelle média (*media gateway controller interface*), et le protocole correspondant, MGCP c'est-à-dire protocole de commande de passerelle média (*media gateway control protocol*), permettant de contrôler des clients intégrant la voix sur IP (VoIP, *voice-over-IP*), à partir d'éléments externes de contrôle d'appel. Le protocole MGCP suppose une architecture de commande d'appel, dans laquelle l'intelligence de commande d'appel se situe en dehors des passerelles et est gérée par des éléments externes de commande d'appel. Le profil décrit dans la présente Recommandation est désigné sous le terme protocole NCS de signalisation d'appel par le réseau (*network-based call signalling*).

La présente Recommandation repose sur la RFC 2705 intitulée "Media Gateway Control Protocol" (MGCP) 1.0 (protocole de commande de passerelle média), elle-même issue de la fusion entre le protocole "Simple Gateway Control Protocol" (protocole simple de contrôle de passerelle) et la famille de protocoles (IPDC, *IP device control*), et des informations délivrées par les personnes ayant mis au point ce profil.

2 Références

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.

2.1 Références normatives

- UIT-T J.83 (1997), *Systèmes numériques multiprogrammes pour la distribution par câble des services de télévision, son et données*.
- UIT-T J.112 Annexe A (2001), *Diffusion vidéonumérique: canal d'interaction pour les systèmes de télédistribution par câble*.
- UIT-T J.112 Annexe B (2001), *Spécifications de l'interface du service de transmission de données par câble: interface radioélectrique*.
- UIT-T J.160 (projet), *Cadre architectural pour la fourniture de services à temps critique sur les réseaux de télévision par câble utilisant des câblo-modems*.
- UIT-T J.161 (2001), *Caractéristiques codecs audio destinés au service audio bidirectionnel sur les réseaux de télévision par câble utilisant des câblo-modems*.
- UIT-T J.163 (2001), *Qualité de service dynamique pour la fourniture de services en temps réel sur les réseaux de télévision par câble utilisant des câblo-modems*.
- IETF RFC 821 (1982), *Protocole de transfert de courrier simple*.
- IETF RFC 1034 (1987), *Noms de domaine – Concepts de base*.

– IETF RFC 1889 RTP (1996), *Protocole de transport pour applications en temps réel*.

– IETF RFC 2045 (1996), *Extensions polyvalentes de polypostage pour l'Internet (MIME) Partie 1: Format des corps de messages pour l'Internet*.

– IETF RFC 2234 (1997), *Forme BNF augmentée pour les spécifications de syntaxe: ABNF*.

– IETF RFC 2327 SDP (1998), *Protocole de description de session*.

– IETF RFC 2543 SIP (1999), *Protocole d'ouverture de session*.

NOTE – La référence faite à un document dans la présente Recommandation ne lui confère pas, en tant que document autonome, le statut de Recommandation.

2.2 Références informatives

– IETF RFC 1890 RTP (1996), *Profil pour conférences audiovisuelles avec commande minimale*.

– IETF RFC 2705 (1999), *Protocole MGCP (MGCP) Version 1.0*.

3 Termes et définitions

La présente Recommandation définit les termes suivants:

3.1 modem-câble; câblo-modem: dispositif terminal de couche 2 formant l'extrémité client de la connexion J.112.

3.2 IPCablecom: projet de l'UIT-T comprenant une architecture et une série de Recommandations permettant l'acheminement de services interactifs à temps critique sur les réseaux de télévision par câble.

3.3 DOIT ou **NE DOIT PAS**: convention utilisée dans la présente Recommandation pour indiquer une disposition absolument obligatoire de la spécification.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

API interface de programmation d'application (*application programming interface*)

CPE équipement des locaux client (*customer premises equipment*)

DTMF multifréquence bitonalité (*dual tone multi-frequency*)

IP protocole Internet (*Internet protocol*)

MGCI interface de commande de passerelle de support (*media gateway controller interface*)

MGCP protocole MGCP; protocole de commande de passerelle de support (*media gateway control protocol*)

MIB base de données MIB; base d'informations de gestion (*management information base*)

MTA adaptateur de terminal multisupport (*media terminal adaptor*)

MWD temps d'attente maximale (*maximum waiting delay*)

NCS signalisation d'appel par le réseau (*network call signalling*)

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

RTPC réseau téléphonique public commuté

SDP protocole SDP; protocole de description de session (*session description protocol*)

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

5 Introduction

La présente Recommandation décrit le profil NCS d'une interface de programmation d'application (MGCI, *media gateway controller interface*) et un protocole MGCP connexe, dédié au contrôle de clients intégrés à partir d'éléments externes à l'appel. Un client intégré est un élément du réseau qui fournit:

- deux ou plusieurs lignes traditionnelles d'accès analogique à un réseau Voix sur IP (VoIP, *voice-over-IP*);
- une ou plusieurs lignes vidéo d'accès à un réseau Voix sur IP, en cours d'élaboration.

Les clients intégrés ne doivent pas être limités uniquement à une utilisation résidentielle. Ils peuvent également être utilisés dans une entreprise, par exemple. Les clients intégrés permettent un accès côté ligne, et en tant que tels, sont censés être dotés d'un équipement côté ligne, par exemple, des lignes d'accès analogique destinées aux téléphones conventionnels qui leur sont associés, contrairement aux passerelles de communication interurbaines.

Le protocole MGCP suppose une architecture de commande d'appel, dans laquelle l'intelligence de commande d'appel se situe en dehors des passerelles et est gérée par des éléments externes de commande d'appel, désignés sous le terme agents d'appel. Le protocole MGCP implique que ces éléments externes de commande d'appel, ou agents d'appel (CA, *call agent*), se synchronisent les uns avec les autres pour envoyer des commandes cohérentes aux passerelles qu'ils contrôlent. Le protocole MGCP défini dans la présente Recommandation ne spécifie pas un mécanisme de synchronisation des agents d'appel. Toutefois, des spécifications IPCablecom à venir pourraient spécifier de tels mécanismes.

Le protocole MGCP suppose un modèle de connexion où les constructions de base sont les points d'extrémité et les connexions. Une passerelle comporte un ensemble de points d'extrémité, physiques ou virtuels, qui constituent les sources ou les collecteurs de données.

Un point d'extrémité physique est par exemple l'interface d'une passerelle qui termine une connexion du service téléphonique ordinaire analogique à un appareil téléphonique, un système de clé, un autocommutateur privé (PBX), etc. Une passerelle qui termine les lignes de service téléphonique ordinaire résidentiel (à des appareils téléphoniques) est appelée une *passerelle résidentielle,* un *client intégré* ou un adaptateur *MTA*. Les clients intégrés peuvent éventuellement prendre en charge la vidéo, aussi.

Un exemple de point d'extrémité virtuel est une source audio, ou un serveur de contenu audio. Par conséquent, la création de points d'extrémité implique l'installation de matériel, tandis que les points d'extrémité virtuels peuvent être créés à l'aide d'un logiciel. Toutefois, le profil NCS du protocole MGCP n'adresse que les points d'extrémité physiques.

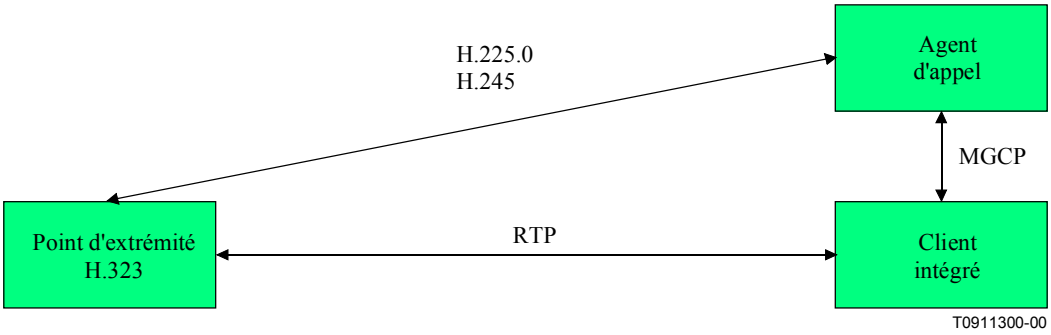
Les connexions sont du type point à point. Une connexion point à point est une association entre deux points d'extrémité dont le but est de transmettre des données entre ces deux points d'extrémité. Une fois cette association entre les deux points d'extrémité établie, le transfert de données peut avoir lieu. L'association est établie en créant une connexion en deux moitiés: l'une du côté du point d'extrémité d'origine, l'autre du côté du point d'extrémité de terminaison.

Les agents d'appel ordonnent aux passerelles de créer des connexions point à point et de détecter certains événements, tels que le décrochage, et de générer certains signaux, tels qu'une sonnerie. C'est uniquement à l'agent d'appel qu'incombe la tâche de spécifier comment et quand les connexions sont effectuées, ainsi que de décider quels événements et quels signaux sont générés sur les points d'extrémité. Quant à la passerelle, elle ne fait plus qu'office de simple dispositif, dénué d'état d'appel,

qui reçoit des instructions génériques de l'agent d'appel sans avoir besoin de savoir ou comprendre les concepts qui sous-tendent les appels, les états d'appels, les caractéristiques, ou les interactions entre les caractéristiques. Lors de l'introduction de nouveaux services, les profils des utilisateurs sont modifiés, etc. (ces modifications sont transparentes pour la passerelle.) L'agent d'appel met en oeuvre ces modifications et génère la nouvelle série d'instructions appropriées relatives à ces modifications, et l'envoie à la passerelle. A chaque redémarrage de la passerelle, elle s'ouvre dans un état propre et se contente d'exécuter les instructions de l'agent d'appel, au fur et à mesure qu'elle les reçoit.

5.1 Relations avec les normes H.323

Le protocole est conçu sous forme d'un protocole interne au sein d'un système distribué qui apparaît, à l'extérieur, comme une passerelle VoIP unique. Ce système se compose d'un agent d'appel, qui peut être distribué ou non sur plusieurs plates-formes, et d'un ensemble de passerelles. Dans une configuration H.323, ce système de passerelles distribuées peut s'interfacer d'un côté avec une ou plusieurs lignes de service téléphonique ordinaire, et de l'autre côté avec des compatibles H.323, comme illustré ci-dessous:



Dans le modèle MGCP, les passerelles sont dédiées à la traduction des signaux audio, tandis que l'agent d'appel gère les fonctions de signalisation et de traitement des appels. Par conséquent, l'agent d'appel implémente les couches "signalisation" de la norme H.323, et se présente aux systèmes H.323 comme étant un "portier H.323 " ou comme étant un ou plusieurs "points d'extrémité H.323". La signalisation d'appel H.225.0 et la signalisation multimédia H.245 sont donc acheminées vers l'agent d'appel.

5.2 Relation avec les normes de l'IETF

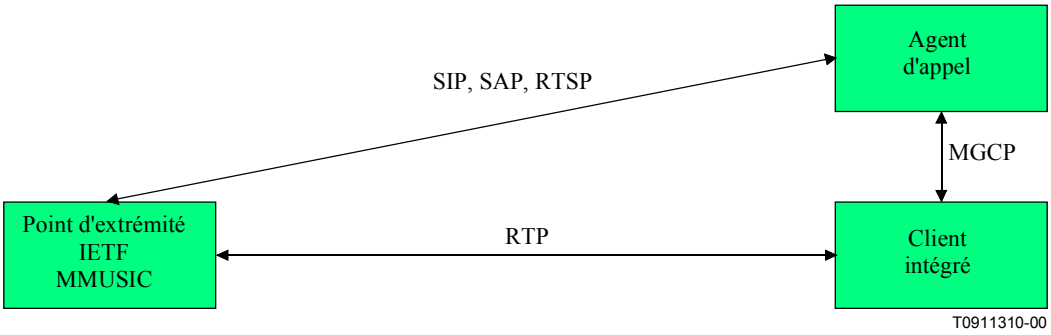
Tandis que H.323 constituait auparavant la norme reconnue pour les terminaux voix sur IP, l'IETF a également publié des spécifications relatives à d'autres types d'applications multimédias. Ces autres spécifications comprennent:

- le protocole de description de session; protocole SDP (*session description protocol*), RFC 2327;
- le protocole d'annonce de session; protocole SAP (*session announcement protocol*), RFC 2974: en cours d'élaboration;
- le protocole d'initialisation de session; protocole SIP (*session initiation protocol*), RFC 2543;
- le protocole de flux en temps réel; protocole RTSP (*real-time streaming protocol*), RFC 2326.

Les trois dernières spécifications sont en fait des normes de signalisation de remplacement qui prévoient la transmission d'une description de session à un tiers concerné. Le protocole SAP est utilisé par des gestionnaires de session en multidiffusion, afin de distribuer une description de session en multidiffusion à un grand nombre de destinataires. Le protocole SIP sert à inviter un

utilisateur individuel à participer à une session point à point ou monodiffusion. Le protocole RTSP sert d'interface avec un serveur qui distribue des données en temps réel. Quel que soit le protocole, la description de session est réalisée conformément au protocole SDP; lorsque ce sont des données audio qui sont transmises, elles le sont via les protocoles de transport en temps réel (RTP et RTCP).

Les systèmes de passerelles distribués combinés au protocole MGCP permettront aux usagers des communications vocales du réseau téléphonique public commuté et aux utilisateurs des clients imbriqués d'accéder aux sessions établies à l'aide des protocoles SAP, SIP ou RTSP définis par le Groupe de travail MMUSIC de l'IETF. Ce sera l'agent d'appel qui se chargera de la conversion de la signalisation, comme illustré ci-dessous:



La norme SDP joue un rôle essentiel dans cette architecture. En effet, nous découvrirons au paragraphe suivant qu'elle sert également à transporter les descriptions de session dans le protocole MGCP.

6 Interface MGCI (*media gateway controller interface*)

Les fonctions de l'interface MGCI permettent le contrôle de connexion, le contrôle de point d'extrémité, l'audit, et l'élaboration de rapports d'état. Elles utilisent toutes le même modèle de système et les mêmes conventions de nommage.

6.1 Modèle et conventions de nommage.

Le protocole MGCP suppose implicitement la présence d'un modèle de connexion où les constructions de base sont les points d'extrémité et les connexions. Les connexions sont regroupées en appels. Une ou plusieurs connexions peuvent appartenir à un seul appel. Les connexions et les appels sont élaborés à l'instigation d'un ou plusieurs agents d'appel.

6.1.1 Noms de point d'extrémité

Les noms des points d'extrémité, également appelés identificateurs de points d'extrémité, comportent deux composantes, lesquelles ne sont pas sensibles à la casse, ici:

- le nom de domaine de la passerelle qui gère le point d'extrémité;
- un nom de point d'extrémité local, au sein de la passerelle.

Les noms des points d'extrémité se présentent sous la forme

```
Point-extrême-local-nom@nom-domaine
```

Où `nom-domaine` est un nom de domaine complet, conformément à la norme RFC 1034 et comporte une partie hôte; voici donc un exemple de nom de domaine:

```
MonClientIntegre.cablelabs.com
```

De plus, `nom-domaine` peut correspondre à une adresse IPv4 au format décimal séparé par un point sous la forme d'une chaîne de texte et entourée de crochets ("[" et "]"), comme par exemple "[128.96.41.1]"; pour plus de détails, voir RFC 821. Toutefois, il est généralement déconseillé d'utiliser des adresses IP.

Les clients intégrés peuvent être associés à un ou plusieurs points d'extrémité (par exemple, un pour chaque prise RJ11 des téléphones noirs), et chacun de ces points d'extrémité est identifié par un nom de point d'extrémité local distinct. A l'instar du nom de domaine, le nom du point d'extrémité local n'est pas sensible à la casse. Un type de point d'extrémité tel qu'un téléphone analogique ou un vidéophone est associé au nom du point d'extrémité local. Le type de point d'extrémité peut dériver du nom du point d'extrémité local. Ce dernier est un nom hiérarchique, pour lequel l'élément le moins spécifique du nom se trouve à gauche, et le plus spécifique se trouve à droite. Plus formellement, le nom du point d'extrémité local doit respecter les règles suivantes:

- les éléments individuels du nom du point d'extrémité local doivent être séparés par une barre oblique unique ("/", ASCII 2F hex);
- les éléments individuels sont des chaînes de caractère ASCII composées de lettres, chiffres ou autres caractères imprimables, sauf les caractères utilisés comme délimiteurs dans les noms de points d'extrémité ("/", "@"), les caractères génériques ("*", "$"), et les espaces blancs;
- les caractères génériques sont représentés par un astérisque ("*") ou un signe dollar ("$") qui remplacent les termes à rechercher. Ainsi, si le nom complet d'un point d'extrémité local se présente sous la forme:

  ```
  term1/term2/term3
  ```

  et que l'un des éléments du nom du point d'extrémité local est remplacé par un caractère générique, alors le nom du point d'extrémité local prendra la forme:

`term1/term2/*`	si `term3` est remplacé par un caractère générique.
`term1/*/*`	si `term2` et `term3` sont remplacés par un caractère générique.

  Dans chacun de ces exemples, il est possible de remplacer l'astérisque par le signe dollar;
- les caractères génériques ne peuvent être insérés qu'à partir de la droite de la chaîne; par conséquent, si un élément est remplacé par un caractère générique, tous les éléments à sa droite doivent l'être également;
- si des signes dollar et astérisque de remplacement sont mélangés, les signes dollar ne sont autorisés qu'à partir de la droite de la chaîne; par conséquent, si un élément contient un signe dollar de remplacement, tous les éléments à sa droite doivent en contenir également;
- un élément représenté par un astérisque sera interprété de la façon suivante:

  "utiliser *toutes* les valeurs de cet élément qui sont connues dans le cadre du client intégré concerné";
- un élément représenté par un signe dollar sera interprété de la façon suivante:

  "utiliser une seule valeur *quelconque* de cet élément dans le cadre du client intégré concerné";
- chaque type de point d'extrémité peut spécifier des informations complémentaires relatives aux règles de nommage de ce type de point d'extrémité; toutefois, ces règles ne doivent pas entrer en conflit avec les règles citées ci-dessus.

Il convient de noter que la présence de types différents de points d'extrémité, ou même de sous-éléments différents, des "lignes" par exemple, dans le même type de point d'extrémité générera des noms de points d'extrémité locaux différents. Par conséquent, chaque "ligne" doit être traitée comme un point d'extrémité distinct.

6.1.1.1 Noms de points d'extrémité dans des clients intégrés

Les points d'extrémités dans des clients intégrés DOIVENT prendre en charge les conventions de nommage complémentaires spécifiées dans le présent paragraphe.

Les clients intégrés PEUVENT prendre en charge un ou plusieurs types de points d'extrémité, et notamment les suivants:

- `Analog Telephone` – Le téléphone analogique est représenté comme une ligne d'accès analogique (aaln, *analogue access line*). Il s'agit en fait de l'équivalent d'une ligne téléphonique analogique du réseau téléphonique public commuté.
- `Video` – Les informations relatives au type d'appareil vidéo appellent un complément d'étude.
- `Basic Access ISDN` (*RNIS d'accès de base*) – Les informations relatives aux types de dispositif RNIS appellent un complément d'étude.

6.1.1.1.1 Points d'extrémité d'une ligne d'accès analogique

En plus des conventions de nommage spécifiées ci-dessus, les noms des points d'extrémité locaux du type de points d'extrémité "ligne d'accès analogique" (`aaln`) des clients intégrés doivent respecter les règles suivantes:

- Les noms des points d'extrémité locaux doivent contenir au moins un élément et au plus, deux éléments.
- Term1 DOIT être l'élément "`aaln`" ou un caractère générique. Il convient de noter que l'utilisation d'un caractère générique pour term1 peut faire référence à tous les types de points d'extrémité ou à un type quelconque dans le client intégré, quel que soit leur type. L'utilisation de cette fonction est généralement conseillée pour des raisons administratives (l'audit ou le redémarrage, par exemple).
- Term2 DOIT être un nombre compris entre 1 et le nombre de lignes d'accès analogique prises en charge par le client intégré concerné. Par conséquent, le nombre identifie une ligne d'accès analogique spécifique du client intégré.
- Si un nom de point d'extrémité local ne se compose que d'un élément, cet élément sera term1.
- Si term1 *n'est pas* un caractère générique, il est donc présumé que le signe dollar (faisant référence à "n'importe lequel") remplace term2, c'est-à-dire que "`aaln`" équivaut à "`aaln/$`".
- Si term1 *est* un caractère générique, il est donc présumé que le caractère générique astérisque (faisant référence à "all") remplace term2, c'est-à-dire que "`*`" et "`$`" équivalent respectivement à "`*/*`" et à "`$/*`".

Voici quelques exemples de points d'extrémité de ligne d'accès analogique:

– `aaln/1` première ligne d'accès analogique du client intégré concerné.

– `aaln/2` seconde ligne d'accès analogique du client intégré concerné.

– `aaln/$` n'importe quelle ligne analogique du client intégré concerné.

– `aaln/*` toutes les lignes analogiques du client intégré concerné.

– `*` tous les points d'extrémité (quel que soit le type de point d'extrémité) du client intégré concerné.

Le processus de mise en service/configuration (automatique) est chargé d'obtenir et de fournir des informations concernant le nombre de points d'extrémité dont dispose un client intégré, ainsi que le type de chaque point d'extrémité. Même s'ils diffèrent par leur logique, il faut remarquer que le *type de point d'extrémité* peut être dérivé de la partie locale du nom du point d'extrémité.

6.1.2 Noms d'appel

Les appels sont identifiés par des identificateurs uniques, indépendants des plates-formes ou des agents sous-jacent(e)s. Les identificateurs d'appels sont des chaînes hexadécimales, créées par l'agent d'appel. Les identificateurs d'appel d'une longueur maximale de 32 DOIVENT être pris en charge.

A tout le moins, les identificateurs d'appel DOIVENT être uniques au sein d'une collection d'agents d'appel qui contrôlent les mêmes passerelles. Toutefois, la coordination de ces identificateurs d'appel entre les agents d'appel ne s'inscrit pas dans le cadre de la présente Recommandation. Lorsqu'un agent d'appel établit plusieurs connexions appartenant au même appel, sur une même passerelle ou sur des passerelles différentes, ces connexions sont toutes liées au même appel, grâce à l'identificateur d'appel. Ce dernier peut être utilisé par des procédures de comptage ou d'administration, lesquelles ne s'inscrivent pas dans le cadre du protocole MGCP.

6.1.3 Noms de connexion

Les identificateurs de connexion sont créés par la passerelle lorsqu'on lui demande de créer une connexion. Ils identifient la connexion dans le contexte d'un point d'extrémité. Les identificateurs de connexion sont traités comme des chaînes hexadécimales par le protocole MGCP. La passerelle DOIT garantir qu'un délai d'attente approprié d'au moins trois minutes est respecté entre la fin de la connexion qui utilisait cet identificateur et son utilisation dans le cadre d'une nouvelle connexion, pour le même point d'extrémité. Les noms de connexion d'une longueur maximale de 32 caractères DOIVENT être pris en charge.

6.1.4 Noms des agents d'appel et d'autres entités

Le protocole MGCP a été conçu pour améliorer la fiabilité du réseau et permettre l'implémentation d'agents d'appel redondants. Cela signifie qu'aucun lien statique n'est imposé entre les entités et les plates-formes matérielles ou les interfaces réseau.

A l'instar des noms des points d'extrémité, les noms des agents d'appel se composent de deux parties. La partie locale du nom n'indique aucune structure interne spécifique. Voici un exemple de nom d'agent d'appel:

```
ca1@ca.quelconque.net
```

La fiabilité est assurée par les précautions suivantes:

- des entités telles que des clients intégrés ou des agents d'appel sont identifiés à l'aide de leurs noms de domaine, et non par leur adresse réseau. En effet, il est possible d'associer plusieurs adresses réseau à un seul nom de domaine. Si une commande ne peut pas être transférée vers l'une des adresses réseau, les implémentations DOIVENT réessayer la retransmission en utilisant une autre adresse;
- les entités peuvent se déplacer vers d'autres plates-formes. L'association entre un nom logique (nom de domaine) et la plate-forme réelle est conservée dans le service de nom de domaine (DNS, *domain name service*). Les agents d'appel et les passerelles DOIVENT garder la trace de la durée de vie de l'enregistrement issue du DNS. Ils DOIVENT interroger le DNS afin de rafraîchir les informations si la durée de vie a expiré.

En plus de l'adressage indirect fourni par l'utilisation de noms de domaine et du DNS, le concept "d'entité avisée" est essentiel pour la fiabilité et la reprise sur défaillance dans le contexte du protocole MGCP. "L'entité avisée" d'un point d'extrémité correspond à l'agent d'appel qui contrôle actuellement ce point d'extrémité. A tout moment, le point d'extrémité est associé à une et une seule "entité avisée"; ainsi lorsque le point d'extrémité a besoin d'envoyer une commande à l'agent d'appel, il DOIT envoyer la commande à "l'entité avisée" courante pour le ou les points d'extrémité auxquels la commande se rapporte. Au démarrage, "l'entité avisée" DOIT être définie sur une valeur mise en service. La plupart des commandes envoyées par l'agent d'appel sont capables de nommer de manière explicite "l'entité avisée", à l'aide du paramètre "NotifiedEntity". "L'entité avisée" DOIT

rester la même jusqu'à la réception d'un nouveau paramètre "NotifiedEntity" ou jusqu'au redémarrage du point d'extrémité. Si "l'entité avisée" d'un point d'extrémité est vide ou n'a pas été définie explicitement[1], elle prendra alors la valeur par défaut de l'adresse source de la dernière commande de gestion de connexion ou de la dernière requête de notification reçue concernant le point d'extrémité. Par conséquent, l'audit ne modifiera pas "l'entité avisée".

Le paragraphe 6.4 présente une description plus détaillée de la fiabilité et de reprise sur défaillance.

6.1.5 Scripts de numérotation

L'agent d'appel peut ordonner à la passerelle de recueillir les chiffres composés par l'utilisateur. Cette fonction a été conçue pour être utilisée par les lignes d'accès analogique associées à des passerelles résidentielles afin de pouvoir recueillir les numéros composés par l'utilisateur; elle peut également servir à recueillir les codes d'accès, numéros de cartes de crédit et autres numéros demandés par les services de contrôle d'appel. Les points d'extrémité DOIVENT prendre en charge les scripts de numérotation, comme expliqué dans le présent paragraphe.

Une autre procédure implique que la passerelle transmette à l'agent d'appel les chiffres numérotés dès qu'ils le sont; cette procédure se nomme envoi avec chevauchement. Toutefois, une telle procédure génère un grand nombre d'interactions. Il est donc préférable d'accumuler les chiffres numérotés dans une mémoire tampon, puis de les transmettre dans un message unique.

Le problème créé par cette approche cumulative, toutefois, est que la passerelle a des difficultés à prévoir le nombre de chiffres qu'elle doit accumuler avant de les transmettre. Par exemple, à l'aide du téléphone de votre bureau, vous pouvez composer les numéros suivants:

0	Opérateur local
00	Opérateur longue distance
xxxx	Numéro de poste local
8xxxxxxx	Numéro local
#xxxxxxx	Numéro raccourci correspondant à un numéro local d'un autre site de l'entreprise
*xx	Services confort
91xxxxxxxxxx	Numéro longue distance
9011 + un maximum de 15 chiffres	Numéro international

La solution à ce problème consiste à charger dans la passerelle un script de numérotation correspondant au plan de numérotation de la zone de résidence de la passerelle. Par conséquent, le script de numérotation utilisé peut varier d'une région à l'autre. Celui-ci s'exprime sous la forme d'une syntaxe dérivée de la commande *egrep* du système UNIX. Par exemple, le plan de numérotation décrit ci-dessus génère le script de numérotation suivant:

```
(0T| 00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
```

La syntaxe formelle du script de numérotation est décrite à l'aide de la notation BNF suivante:

```
Digit          ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
Timer          ::= "T" | "t"     -- correspond à la détection d'une temporisation
Letter         ::= Digit | Timer | "#" | "*" | "A" | "a" | "B" | "b" | "C" | "c" | "D" | "d"
Range          ::= "X" | "x"         -- correspond à un chiffre quelconque
                 | "[" Letters "]"    -- correspond à l'une des lettres spécifiées
Letters        ::= Subrange | Subrange Letters
```

[1] Cela pourrait être le cas si un paramètre NotifiedEntity vide était spécifié.

```
Subrange      ::= Letter           -- correspond à la lettre spécifiée
              | Digit "-" Digit    -- correspond à un chiffre quelconque entre
                                   -- le premier et le dernier
Position      ::= Letter | Range
StringElement ::= Position         -- correspond à une occurrence de la position
              | Position "."       -- correspond à un nombre aléatoire
                                   -- d'occurences de la position, y compris 0
     String    ::= StringElement | StringElement String
     StringList::= String | String "|" StringList
     DigitMap  ::= String | "(" StringList ")"
```

Selon cette syntaxe, un script de numérotation est défini soit par une "chaîne" (insensible à la casse), ou par une "liste de chaînes". Indépendamment de la syntaxe ci-dessus, une temporisation n'est généralement admise que si elle apparaît en dernière position dans une chaîne[2]. Chaque chaîne de la liste est un modèle de numérotation de remplacement. Une passerelle capable de détecter les chiffres, les lettres ou les temporisations, pourra:

1) ajouter le code du paramètre événement correspondant au chiffre, à la lettre ou à la temporisation, tel qu'un jeton à la fin de la variable d'état interne "chaîne de numérotation courante";

2) appliquer la "chaîne de numérotation courante" à la table de script de numérotation, en cherchant une correspondance, dans l'ordre lexical, pour chaque expression régulière du script de numérotation;

3) si le résultat est sous-qualifié (il correspond partiellement à au moins une entrée du script de numérotation), elle ne fait rien d'autre.

Si le résultat correspond, ou s'il est surqualifié (c'est-à-dire qu'aucun autre chiffre n'a pu générer de correspondance), la passerelle envoie la liste des chiffres à l'agent[3] d'appel et vide la "chaîne de numérotation courante".

Timer T est une temporisation d'entrée de chiffres qui peut être utilisée de deux manières:

- lorsque timer T est utilisée conjointement avec un script de numérotation[4], la temporisation n'est démarrée que lorsque le premier chiffre est saisi, puis elle est redémarrée après chaque saisie d'un chiffre, jusqu'à ce que le script de numérotation trouve une correspondance ou une non-correspondance. Dans ce cas, timer T fonctionne comme une temporisation entre les chiffres;
- lorsque timer T est utilisée sans script de numérotation, la temporisation est démarrée immédiatement et annulée (mais pas redémarrée) dès qu'un chiffre est entré. Dans ce cas, timer T peut être utilisée comme une temporisation entre les chiffres lors de l'utilisation de l'envoi avec chevauchement.

Lorsqu'elle est utilisée conjointement avec un script de numérotation, timer T prend l'une des deux valeurs, T_{par} ou T_{crit}. Lorsqu'un chiffre de plus au minimum est requis pour que la chaîne de chiffres corresponde à l'un des modèles du script de numérotation, la temporisation T prend la valeur T_{par}, ce qui correspond à une temporisation partielle de la numérotation. S'il suffit d'une temporisation pour produire une correspondance, timer T prend la valeur T_{crit} correspondant à la temporisation critique. Lorsque timer T est utilisée sans script de numérotation, elle prend la valeur T_{crit}. La valeur par défaut de T_{par} est de 16 secondes, tandis que la valeur par défaut de T_{crit} est de quatre secondes. Le processus de mise en service peut modifier ces deux valeurs.

---

[2] Par exemple, "123T" et "123[1-2T5]" respectent cette règle, mais "12T3" ne la respecte pas.

[3] La liste de chiffres peut également contenir d'autres événements – voir 6.4.3.1.

[4] Techniquement, il s'agit de l'action "accumulation en fonction du script de numérotation".

Les scripts de numérotation peuvent être fournis à la passerelle par l'agent d'appel, chaque fois que l'agent d'appel ordonne à la passerelle de détecter les chiffres. Une fois encore, il faut noter que le script de numérotation utilisé dépend de la zone dans laquelle la passerelle réside et qu'il est par conséquent programmable. Les scripts de numérotation, lorsqu'ils sont fournis par l'agent d'appel, DOIVENT être tels que définis dans le présent paragraphe.

6.1.6 Evénements et signaux

Le concept d'événements et de signaux est essentiel au protocole MGCP. Un agent d'appel peut effectivement demander à être avisé de certains événements se produisant à un point d'extrémité, des événements de décrochage, par exemple. Un agent d'appel peut également demander l'application de certains signaux sur un point d'extrémité, une tonalité, par exemple.

Les événements et les signaux sont regroupés en paquets au sein desquels ils partagent le même espace nominatif, que nous appellerons noms d'événement dans la suite de la présente Recommandation. Un paquet est un groupe d'événement et de signaux pris en charge par un type de point d'extrémité particulier. Par exemple, un paquet peut prendre en charge un certain groupe d'événements et de signaux pour les lignes d'accès analogique, tandis qu'un autre paquet prendra en charge un autre groupe d'événements et de signaux pour les lignes vidéo. Il peut exister un ou plusieurs paquets pour un type de point d'extrémité donné, et chaque type de point d'extrémité est associé à un paquet par défaut.

Les noms d'événement se composent d'un nom de paquet, et d'un code d'événement et, du fait que chaque paquet définit un espace nominatif distinct, les mêmes codes d'événement peuvent être utilisés dans des paquets différents. Les noms de paquets et les codes d'événement sont des chaînes de lettres insensibles à la casse, de chiffres et de tirets, avec une restriction: le tiret NE DOIT PAS être le premier ou le dernier caractère d'un nom. Certains codes d'événement doivent être paramétrés à l'aide de données complémentaires; pour cela, il suffit d'ajouter les paramètres entre parenthèses. Le nom du paquet est séparé du code d'événement par une barre oblique ("/"). Le nom du paquet peut être exclu du nom d'événement, auquel cas, c'est le nom du paquet par défaut du type de point d'extrémité concerné qui est supposé. Par exemple, pour une ligne d'accès analogique, le paquet ligne exemple (nom de paquet "X") étant le paquet par défaut, les deux événements suivants sont considérés égaux:

- X/dl tonalité dans le paquet ligne exemple pour une ligne d'accès analogique;
- dl tonalité dans le paquet ligne exemple (par défaut) pour une ligne d'accès analogique.

L'Annexe A définit un jeu initial de paquets. Des noms de paquets et des codes d'événement complémentaires peuvent être définis et/ou enregistrés auprès de l'IPCablecom. Toute modification apportée aux paquets définis dans la présente Recommandation DOIT entraîner la modification du nom du paquet ou du numéro de version du nom du profil NCS, ou les deux.

Chaque paquet DOIT posséder une définition de paquet, qui DOIT définir le nom du paquet, ainsi que la définition de chaque événement appartenant au paquet. La définition d'événement DOIT contenir le nom précis de l'événement, c'est-à-dire, le code d'événement, sa définition en texte brut ainsi que, si nécessaire, la définition précise des signaux correspondants; par exemple les fréquences exactes des signaux audio tels qu'une tonalité de numérotation ou des tonalités multifréquences (DTMF). De plus, les événements doivent spécifier s'ils sont persistants (décrochage, par exemple; voir 6.3.1) et s'ils contiennent des états d'événements qu'il est possible d'auditer (décrochage, par exemple; voir 6.3.8.1). Les types des signaux DOIVENT également être définis (On/Off, Time-out ou Brief). En outre, les signaux temporisés DOIVENT avoir une valeur par défaut définie – voir 6.3.1.

En plus des paquets IPCablecom, les personnes chargées de l'implémentation PEUVENT consolider leur expérience en définissant des paquets expérimentaux. Le nom de ces paquets expérimentaux DOIT commencer par les deux caractères "x-" ou "X-"; en effet, IPCablecom NE DOIT PAS enregistrer des paquets dont le nom commence par ces deux caractères. Si un client intégré reçoit

une commande faisant référence à un paquet non pris en charge, il DOIT retourner une erreur (code d'erreur 518 – paquet non pris en charge).

Les noms de paquets et les codes d'événement prennent chacun en charge une notation avec caractère générique. Le caractère générique "*" (astérisque) peut être utilisé pour référencer tous les paquets pris en charge par le point d'extrémité concerné; le code d'événement "all" permet de référencer tous les événements du paquet concerné. Par exemple:

- X/all référence tous les événements dans ce paquet de ligne exemple pour une ligne d'accès analogique;
- */all pour une ligne d'accès analogique, réfère tous les paquets et événements contenus dans les paquets pris en charge par le point d'extrémité concerné.

Par conséquent, le nom de paquet "*" NE DOIT PAS être attribué à un paquet; le code d'événement "all" NE DOIT PAS être utilisé dans un paquet.

Par défaut, les événements et les signaux sont détectés et générés sur les points d'extrémité. Toutefois, il est possible que certains événements et signaux soient détectés et générés sur des connexions en remplacement ou en plus d'un point d'extrémité. Ainsi, il est possible de demander à un point d'extrémité de fournir une tonalité de retour d'appel sur une connexion. Afin qu'un événement ou un signal puisse être détecté ou généré sur une connexion, la définition de l'événement ou du signal DOIT définir explicitement que l'événement ou le signal peut être détecté et généré sur une connexion.

Lorsqu'un signal va être appliqué sur une connexion, le nom de la connexion est ajouté au nom de l'événement, à l'aide du caractère arobase "(@)" qui fait office de délimiteur, comme dans:

```
X/rt@0A3F58
```

Le caractère générique "*" (astérisque) peut être utilisé pour indiquer "toutes les connexions" sur les points d'extrémité concernés. Lors de l'utilisation de cette convention, la passerelle DOIT générer ou détecter l'événement sur toutes les connexions établies aux points d'extrémités. Voici un exemple de la convention:

```
X/rt@*
```

Le caractère générique "$" (signe dollar) peut être utilisé pour indiquer "la connexion en cours" Cette convention NE DOIT PAS être utilisée, à moins que la requête de notification d'événement soit "encapsulée" dans une commande CreateConnection ou ModifyConnection. Lors de l'utilisation de cette convention, la passerelle DOIT générer et détecter l'événement sur la connexion en cours de création ou de modification. Voici un exemple de la convention:

```
X/rt@$
```

L'id de connexion, ou un caractère générique de remplacement, peut être utilisé conjointement avec les conventions "tous les paquets" et " tous les événements". Par exemple, la notation:

```
*/all@*
```

peut être utilisée pour désigner tous les événements sur toutes les connexions des points d'extrémité concernés.

6.2 Utilisation du protocole SDP

L'agent d'appel utilise le protocole MGCP pour transmettre aux passerelles la description des paramètres de connexion, tels que les adresses IP, le port UDP, et les profils RTP. Sauf notification ou implication contraire dans la présente Recommandation, les descriptions SDP DOIVENT respecter les conventions présentées dans le protocole SDP, qui correspond désormais à la norme RFC 2327 proposée par l'IETF.

Le protocole SDP permet la description des conférences multimédias. Le profil NCS ne prendra en charge que la configuration des connexions audio et vidéo faisant appel aux types de médias "audio" et "vidéo". Actuellement, seules les connexions "audio" ont été spécifiées.

6.3 Fonctions de contrôle de passerelle

Le présent paragraphe décrit les commandes du protocole MGCP sous la forme d'appel de procédure à distance (RPC, *remote procedure call*) comme une API, que nous désignerons sous le terme d'interface MGCI (*media gateway controller interface*). Une fonction MGCI est définie pour chaque commande MGCP, où la fonction MGCI prend et retourne les mêmes paramètres que la commande MGCP correspondante. Les fonctions décrites dans le présent paragraphe fournissent une description de haut niveau du fonctionnement du protocole MGCP, et fournissent un exemple de l'API ressemblant à RPC qui PEUT être utilisée pour implémenter le protocole MGCP. Bien que l'API MGCI soit une simple API exemple, le comportement sémantique défini par l'interface MGCI fait partie intégrante de la Recommandation. Par conséquent, toutes les implémentations DOIVENT se conformer aux sémantiques spécifiées par l'interface MGCI. Les messages MGCP réellement échangés, incluant les formats de messages et les codages utilisés sont définis dans le paragraphe traitant du protocole (paragraphe 7). Les clients intégrés DOIVENT les implémenter exactement tels que spécifiés.

Le service MGCI est constitué des commandes de gestion de connexions et des commandes de gestion des points d'extrémité. Voici un aperçu général des commandes:

- l'agent d'appel peut envoyer une commande NotificationRequest vers une passerelle, afin de lui ordonner de détecter, sur un point d'extrémité spécifique, des événements spécifiques, tels que les actions de crochet ou des tonalités multifréquences;
- la passerelle utilise ensuite la commande Notify pour informer l'agent d'appel lorsque les événements concernés se produisent sur le point d'extrémité spécifié;
- l'agent d'appel peut alors faire appel à la commande CreateConnection pour établir une connexion qui se termine sur un point d'extrémité, à l'intérieur de la passerelle;
- l'agent d'appel dispose de la commande ModifyConnection pour modifier les paramètres associés à une connexion préalablement établie;
- l'agent d'appel peut utiliser la commande DeleteConnection pour supprimer une connexion existante. Dans certaines circonstances, la commande DeleteConnection peut également servir à une passerelle pour indiquer qu'une connexion ne peut plus être maintenue;
- l'agent d'appel peut utiliser les commandes AuditEndpoint et AuditConnection pour auditer l'état d'un "point d'extrémité" ainsi que de toutes les connexions qui lui sont associées. Une administration réseau allant au-delà des fonctionnalités offertes par ces commandes est généralement conseillée, afin d'obtenir, par exemple, des informations relatives à l'état du client intégré. De telles fonctionnalités devraient être prises en charge grâce à l'utilisation du protocole simple de gestion de réseau (SNMP, *simple network management protocol*) et à la définition d'une base de données MIB, ce qui sort du cadre de la présente Recommandation;
- la passerelle peut utiliser la commande RestartInProgress pour notifier l'agent d'appel que le point d'extrémité ou le groupe de points d'extrémité qu'elle gère est mis hors service ou remis en service.

Ces services permettent à un contrôleur (normalement l'agent d'appel) d'envoyer des instructions à la passerelle sur la création de connexions qui se terminent sur un point d'extrémité rattaché à la passerelle, et d'être informé en retour des événements se produisant sur ce point d'extrémité. Actuellement, un point d'extrémité est limité à une ligne d'accès analogique spécifique, au sein d'un client intégré.

Les connexions sont regroupées en "appels". Par conséquent, plusieurs connexions, qu'elles appartiennent ou non au même appel, peuvent se terminer sur le même point d'extrémité. Chaque connexion est qualifiée par un paramètre "mode", qui peut être défini sur "envoyer uniquement" (*sendonly*), "recevoir uniquement" (*recvonly*), "envoyer/recevoir" (*sendrecv*), "conférence" (*confrnce*), "inactive" (*inactive*), "répliquer" (*replcate*), "boucle réseau" (*netwloop*) ou "test de continuité réseau" (*netwtest*). Le paramètre "mode" détermine si les paquets de média peuvent être envoyés et/ou reçus sur la connexion, sans pour autant affecter le protocole RTCP.

Les signaux audio reçus d'un point d'extrémité seront envoyés sur toute connexion de ce point d'extrémité dont le mode est soit "envoyer uniquement", "envoyer/recevoir", "conférence" ou "répliquer".

La gestion des signaux audio reçus sur ces connexions est également conditionnée par les paramètres mode:

- les signaux audio reçus dans les paquets de données via des connexions en mode "inactive" ou "répliquer" sont ignorés;
- les signaux audio reçus dans les paquets de données via des connexions en mode "recevoir uniquement", "conférence" ou "envoyer/recevoir"sont mélangés puis envoyés sur le point d'extrémité;
- les signaux audio émanant du point d'extrémité sont transmis via toutes les connexions dont le mode est "envoyer uniquement", "conférence" ou "envoyer/recevoir";
- en plus d'être envoyés sur le point d'extrémité, les signaux audio reçus dans des paquets de données via des connexions en mode "conférence" sont répliqués sur toutes les autres connexions au point d'extrémité dont le mode est "conférence". Les détails de ce transfert (c'est-à-dire le convertisseur RTP ou mélangeur, etc.) ne s'inscrivent pas dans le cadre de la présente Recommandation;
- les signaux audio envoyés vers et reçus d'un point d'extrémité sont mélangés puis transmis via toutes les connexions dont le mode est "répliquer". Il CONVIENT d'y inclure les signaux audio générés par des signaux;
- les signaux audio reçus sous forme de paquets de données via des connexions en mode "boucle réseau" ou "test de continuité réseau" sont renvoyés vers la connexion, comme décrit ci-dessous.

Si le mode est défini sur "boucle réseau", les signaux audio reçus via la connexion seront renvoyés en écho à la même connexion. Il CONVIENT que le mode "boucle réseau" fonctionne uniquement en tant que réflecteur de paquets RTP.

Le mode "test de continuité réseau" est utilisé pour vérifier la continuité du réseau IP. A cet effet, un signal d'un type de point d'extrémité spécifique est envoyé aux points d'extrémité sur le réseau IP, puis le point d'extrémité est supposé renvoyer le signal en écho sur le réseau IP après qu'il sera passé à travers le matériel interne de la passerelle, afin d'en vérifier le bon fonctionnement. Le signal DOIT subir un décodage puis un recodage interne avant de revenir. Dans le cas des lignes d'accès analogique, il s'agira d'un signal audio, qui NE DOIT PAS passer par un téléphone connecté à la ligne d'accès analogique, quel que soit l'état courant du crochet de cet appareil, c'est-à-dire, raccroché ou décroché.

Les connexions nouvelles et existantes du point d'extrémité NE DOIVENT PAS être affectées par les connexions placées en mode "boucle réseau" ou "test de continuité réseau". Cependant, les contraintes de ressources locales peuvent limiter le nombre de nouvelles connexions qu'il est possible de créer.

Le mode "répliquer" DOIT au minimum prendre en charge la réplication d'un flux à partir du point d'extrémité et une autre connexion, quelle que soit la méthode d'encodage utilisée pour cette autre connexion. Il est seulement exigé de la connexion "répliquée" qu'elle prenne en charge un flux média

obtenu conformément au codage G.711[5]. La prise en charge du mode "conférence" est facultative. Pour des illustrations des modes d'interactions, voir l'Appendice IV.

6.3.1 NotificationRequest

La commande NotificationRequest sert à demander à la passerelle d'envoyer une notification lors de l'apparition d'événements spécifiés dans un point d'extrémité. Par exemple, une notification peut être demandée lorsque des tonalités associées à des communications par fax sont détectées sur le point d'extrémité. L'entité recevant cette notification, généralement l'agent d'appel, peut décider qu'il convient d'utiliser un type différent de codage sur les connexions liées à ce point d'extrémité et en informer la passerelle[6].

```
ReturnCode
          ← NotificationRequest(EndpointId
                                [, NotifiedEntity]
                                [, RequestedEvents]
                                , RequestIdentifier
                                [, DigitMap]
                                [, SignalRequests]
                                [, QuarantineHandling]
                                [, DetectEvents])
```

EndpointId est l'identificateur des points d'extrémité dans la passerelle sur laquelle s'exécute NotificationRequest. Le paramètre EndpointId DOIT respecter les règles de nommage des points d'extrémité, spécifiées au 6.1.1. Le caractère générique "any of" NE DOIT PAS être utilisé.

NotifiedEntity est un paramètre facultatif qui spécifie une nouvelle "entité avisée" pour le point d'extrémité.

RequestIdentifier est utilisé pour corréler cette requête avec la notification qu'elle peut déclencher. Ce paramètre sera répété dans la commande Notify correspondante.

SignalRequests est un paramètre qui contient l'ensemble des signaux que la passerelle doit appliquer. Sauf indication contraire, les signaux sont appliqués au point d'extrémité; toutefois, certains signaux peuvent être appliqués à une connexion. Ci-dessous, vous trouverez des exemples de signaux[7]:

- sonnerie;
- tonalité d'occupation;
- tonalité d'appel en attente;
- tonalité d'avertissement de décrochage;
- tonalités de retour d'appel sur une connexion.

Les signaux sont divisés en différents types, en fonction de leur comportement:

- `On/off (OO)` – Une fois appliqués, ces signaux durent jusqu'à ce qu'ils soient désactivés. Cela ne peut qu'être le résultat d'une nouvelle commande SignalRequests, dans laquelle le signal est désactivé (voir plus loin). Les signaux de type OO se définissent comme idempotents, et donc plusieurs requêtes visant à activer un signal OO (ou à désactiver) sont parfaitement valides et NE DOIVENT générer aucune erreur. Un signal On/Off pourrait être un indicateur visuel de message en attente (VMWI, *visual message waiting indicator*). Une

---

[5] La connexion "répliquée" peut, par exemple, être utilisée pour prendre en charge la "vérification d'occupation de ligne", avec un impact minime sur les ressources du client intégré.

[6] La nouvelle instruction pourrait être une commande ModifyConnection.

[7] Se reporter à 0 pour une liste complète de signaux.

fois activé, il NE DOIT PAS être désactivé avant que l'agent d'appel ne l'ordonne explicitement ou que le point d'extrémité redémarre.

- `Time-out (TO)` – Une fois appliqués, ces signaux durent jusqu'à ce qu'ils soient annulés (par l'irruption d'un événement ou parce qu'ils n'ont pas été inclus dans une liste ultérieure de signaux [éventuellement vide]), ou jusqu'à ce qu'un délai spécifique à chaque signal se soit écoulé. Un signal dont le délai expire déclenche un événement "opération terminée" (voir l'Annexe A pour une définition plus détaillée de cet événement). Un signal TO pourrait être une "tonalité de retour d'appel", dont le délai s'écoule au bout de 180 secondes. Si un événement se produit avant les 180 secondes, le signal sera par défaut arrêté[8]. S'il n'est pas arrêté, le signal sera temporisé, arrêté et déclenchera un événement "opération terminée", dont l'agent d'appel peut ou non avoir demandé à être avisé. Si l'agent d'appel a demandé la notification de l'événement "opération terminée", cet événement qui lui est envoyé inclut les noms des signaux temporisés.[9]. Les signaux générés sur une connexion comprennent le nom de cette connexion. Les signaux temporisés sont associés à une valeur de temporisation par défaut, qui peut être modifiée par le processus de mise en service. Par ailleurs, la période de temporisation peut être fournie comme un paramètre au signal. Une valeur zéro indique que la période de temporisation est infinie. Un signal TO qui échoue après avoir été initié, mais avant d'avoir déclenché l'événement "opération terminée", va générer un événement "échec de l'opération", qui comprendra également les noms des signaux temporisés[9].
- `Brief (BR)` – La durée de ces signaux est tellement courte qu'ils s'arrêtent d'eux-mêmes. Si un événement d'arrêt du signal survient, ou qu'une nouvelle commande SignalRequests est appliquée, un signal BR actuellement actif ne s'arrête pas. Cependant, tous les signaux BR en cours non encore appliqués sont annulés. Une tonalité brève (*brief)* pourrait être un chiffre d'une tonalité multifréquence. Si c'est le chiffre de tonalité multifréquence, "1" qui est actuellement lu, et qu'un événement d'arrêt du signal survient, le "1" finirait de jouer.

Par défaut, les signaux s'appliquent aux points d'extrémité. Si un signal appliqué à un point d'extrémité génère un flux média (audio, vidéo, etc.), ce dernier NE DOIT PAS être transféré sur une connexion associée à ce point d'extrémité, quel que soit le mode de la connexion. Par exemple, si une tonalité d'appel en attente est appliquée sur un point d'extrémité déjà impliqué dans un appel actif, seul le correspondant utilisant ce point d'extrémité entendra la tonalité de message en attente. Toutefois, les signaux individuels peuvent adopter un comportement différent.

Lorsqu'un signal est appliqué à une connexion qui a reçu une commande RemoteConnectionDescriptor (voir 6.3.3), le flux média généré par ce signal DOIT être transféré vers la connexion, *quel que soit* le mode courant de la connexion. Si aucune commande RemoteConnectionDescriptor n'a été reçue, la passerelle DOIT retourner une erreur (code d'erreur 527 – RemoteConnectionDescriptor manquant).

Lorsqu'une liste de signaux (éventuellement vide) est fournie, celle-ci remplace intégralement la liste courante des signaux temporisés actifs. Les signaux temporisés actifs qui ne sont pas inclus dans la nouvelle liste DOIVENT être arrêtés pour que les nouveaux signaux fournis deviennent actifs. Les signaux temporisés actifs qui se trouvent dans la nouvelle liste DOIVENT rester actifs sans interruption; ainsi le temporisateur qui leur est associé ne sera pas affecté. Par conséquent, il n'existe actuellement aucun moyen de redémarrer la temporisation pour un signal temporisé actuellement actif, sans arrêter celui-ci au préalable. Si le signal temporisé est paramétré, l'ensemble des paramètres d'origine DOIT rester effectif, quelles que soient les valeurs fournies par la suite. Un signal donné NE DOIT PAS figurer plus d'une fois dans une commande SignalRequests.

On trouvera dans l'Annexe A les signaux actuellement définis.

---

[8] L'action "Maintenir actifs le ou les signaux" peut écraser ce comportement.

[9] Si des paramètres étaient transmis au signal, ils ne seront pas rapportés.

RequestedEvents est une liste d'événements que la passerelle doit détecter sur un point d'extrémité. Sauf indication contraire, les événements sont détectés sur le point d'extrémité; toutefois il arrive que certains événements soient détectés sur une connexion. Voici des exemples d'événements[10]:

- transition de raccrochage (qui se produit sur des postes téléphoniques classiques lorsque l'utilisateur raccroche le combiné);
- transition de décrochage (qui se produit sur des postes téléphoniques classiques lorsque l'utilisateur décroche le combiné);
- chiffres de tonalités multifréquences (ou chiffres d'impulsion).

Les événements définis ici sont décrits à l'Annexe A.

A chaque événement, on associe une ou plusieurs **actions** qui définissent l'action que la passerelle doit exécuter lorsque l'événement en question se produit. Les actions possibles sont les suivantes:

- notifier l'événement immédiatement, ainsi que la liste cumulative de tous les événements observés;
- cumuler l'événement;
- cumuler en fonction du script de numérotation;
- ignorer l'événement;
- garder le signal ou les signaux actifs;
- Embedded NotificationRequest (*NotificationRequest intégrée*);
- Embedded ModifyConnection (*ModifyConnection intégrée*).

Deux ensembles d'événements demandés seront détectés par le point d'extrémité: qu'ils soient persistants ou non.

Les événements persistants sont toujours détectés sur le point d'extrémité. Si un événement persistant ne fait pas partie de la liste des RequestedEvents, et que l'événement se produit, celui-ci sera quand même détecté, puis traité comme les autres événements, comme s'il avait été requis à l'aide d'une action Notify[11]. Par conséquent, de manière informelle, les événements peuvent être considérés comme s'ils avaient toujours été inclus dans la liste des RequestedEvents avec une action à notifier, même si aucune détection de collision d'appel, etc., ne sera exécutée[12]. Les événements persistants sont identifiés en tant que tels grâce à leur définition (voir l'Annexe A).

Les événements non persistants sont ceux qui doivent être explicitement inclus dans la liste des RequestedEvents. La liste (éventuellement vide) des événements demandés remplacent intégralement la liste précédente. En plus des événements persistants, seuls les événements spécifiés dans la liste des événements demandés seront détectés par le point d'extrémité. Si un événement persistant est inclus dans la liste de RequestedEvents, l'action spécifiée remplace alors l'action par défaut associée avec l'événement pour la durée de vie de la liste RequestedEvents, à la fin de laquelle l'action par défaut est restaurée. Par exemple, si "ignorer décrochage" est spécifié, et qu'une nouvelle requête sans aucune instruction de décrochage est reçue, l'opération par défaut "Notifier décrochage" est restaurée. Un événement donné NE DOIT PAS figurer plus d'une fois dans une liste RequestedEvents.

Plusieurs actions peuvent être spécifiées pour un événement, cependant, une action donnée ne peut pas figurer plus d'une fois pour un événement donné. Le tableau suivant spécifie les combinaisons valides d'actions:

---

[10] Il s'agit de simples exemples inspirés du paquet ligne exemple dans l'Appendice I.

[11] Ainsi, le RequestIdentifier sera le RequestIdentifier de la NotificationRequest courante.

[12] Normalement, si une requête de recherche (recherche de décrochage, par exemple) est effectuée, celle-ci n'aboutit que si le téléphone n'est pas déjà décroché.

	Notifier	Cumuler	Cumuler en fonction du script de numérotation	Ignorer	Garder le (les) signal (signaux) actif(s)	Notification Request intégrée	Modify Connection intégrée
Notifier	–	–	–	–	√	–	√
Cumuler	–	–	–	–	√	√	√
Cumuler en fonction du script de numérotation	–	–	–	–	√	–	√
Ignorer	–	–	–	–	√	–	√
Garder le ou les signaux actifs	√	√	√	√	–	√	√
Notification Request incorporée	–	√	–	–	√	–	√
Modify Connection incorporée	√	√	√	√	√	√	–

Si un client reçoit une requête contenant une action non valide ou une combinaison non valide d'actions, il DOIT retourner une erreur à l'agent d'appel (code d'erreur 523 – combinaison d'actions inconnue ou non valide).

Lorsque plusieurs actions sont spécifiées, par exemple "garder le ou les signaux actifs" et "Notifier", les actions individuelles sont supposées se produire simultanément.

L'agent d'appel peut envoyer à la passerelle une NotificationRequest avec une liste RequestedEvents vide. Il peut le faire, par exemple, à un client intégré, s'il ne souhaite pas recueillir des chiffres de tonalité multifréquences. Toutefois, les événements persistants sont toujours détectés et notifiés.

DigitMap est un paramètre optionnel qui permet à l'agent d'appel de mettre en service un point d'extrémité avec un script de numérotation, en fonction des chiffres qui seront cumulés lorsque l'agent d'appel soumet un paramètre RequestedEvents avec l'action "cumuler en fonction du script de numérotation", pour ce point d'extrémité. Le script de numérotation fourni est persistant; par conséquent, il n'est pas utile de le fournir à nouveau si une nouvelle requête "cumuler en fonction du script de numérotation" est émise Toutefois, les agents d'appel peuvent fournir un script de numérotation à tout moment. Un script de numérotation DOIT être fourni au point d'extrémité pas plus tard qu'avec la première requête "cumuler en fonction du script de numérotation". Si une passerelle doit "cumuler en fonction du script de numérotation" et qu'elle ne dispose d'aucun script de numérotation correspondant au point d'extrémité concerné, elle DOIT retourner une erreur (code d'erreur 519 – le point d'extrémité ne possède pas de script de numérotation).

Chaque point d'extrémité possède une variable nommée "chaîne de numérotation courante" qui permet de recueillir les chiffres pour les comparer au script de numérotation, comme spécifié au 6.1.5. Chaque fois qu'un Notify est envoyé ou qu'une NotificationRequest doit être traitée, la "chaîne de numérotation courante" est initialisée sur une chaîne "null" (néant). Les chiffres à traiter peuvent désormais être détectés comme saisie, ou extraits d'une zone de conservation de saisies d'événement, nommée "tampon de quarantaine"; pour plus de détails, voir 6.4.3.1.

Les signaux appliqués par les SignalRequests sont synchronisés avec la collection d'événements spécifiés ou inclus dans le paramètre RequestedEvents, sauf si l'action "Garder le ou les signaux actifs les en empêche. Par exemple, si la NotificationRequest a demandé un signal de "sonnerie" alors que la requête a demandé à rechercher un événement "décrochage", la sonnerie devrait s'arrêter

par défaut dès que la passerelle a détecté un événement "décrochage". Si l'événement "décrochage" était défini comme événement persistant et que la requête d'événement n'a pas demandé à chercher un événement "décrochage", la sonnerie devrait quand même s'arrêter car l'événement est alors inséré implicitement dans le paramètre RequestedEvents. La définition formelle spécifie que la génération des signaux de "temporisation" DOIT cesser dès que l'un des événements demandés est détecté, à moins qu'une action "Garder le ou les signaux actifs" soit associée à l'événement spécifié. S'il s'agit d'une action "cumuler en fonction du script de numérotation", le comportement par défaut est d'arrêter tous les signaux de temporisation actifs lorsque le premier chiffre[13] est cumulé; c'est sans rapport avec cette synchronisation si le chiffre cumulé correspond, ne correspond pas, ou ne correspond que partiellement au script de numérotation.

Si les signaux de temporisation souhaités continuent lorsqu'un événement recherché se produit, l'action "Garder le ou les signaux actifs" peut être utilisée. Cette action permet de garder actifs tous les signaux de temporisation couramment actifs, empêchant ainsi l'arrêt par défaut des signaux de temporisation lorsque l'événement se produit.

Si les signaux doivent commencer lorsqu'un événement recherché se produit, l'action "Embedded NotificationRequest" peut être utilisée. La NotificationRequest intégrée peut contenir une nouvelle liste de RequestedEvents, de SignalRequests ainsi qu'un script de numérotation. Toutefois, "Embedded NotificationRequest" ne peut pas inclure une autre "Embedded NotificationRequest". Lorsque "Embedded NotificationRequest" est activée, la "chaîne courante de numérotation" est effacée; sans pour autant affecter les événements observés ni le tampon de quarantaine (voir 6.4.3.1).

L'action NotificationRequest intégrée permet à l'agent d'appel de créer un "mini-script" qui sera traité par la passerelle dès la détection de l'événement associé. Toute commande SignalRequest spécifiée dans la NotificationRequest intégrée commence immédiatement. Cette procédure doit être maniée avec beaucoup de soin pour éviter tout conflit entre l'agent d'appel et la passerelle. Cependant, aucun conflit à long terme ne peut se produire puisque les nouvelles SignalRequests remplacent intégralement l'ancienne liste des signaux de temporisation actifs, et que les signaux de type BR s'arrêtent toujours d'eux-mêmes. Toutefois, il est conseillé de limiter le nombre de signaux du type On/Off. Il est considéré comme une bonne habitude pour un agent d'appel d'activer parfois tous les signaux On/Off qui devraient l'être et de désactiver tous les signaux On/Off qui devraient être désactivés.

Si les modes de connexion doivent être modifiés lorsqu'un événement recherché se produit, l'action "Embedded ModifyConnection" peut être utilisée. L'action ModifyConnection intégrée peut contenir la liste des modifications apportées aux modes de connexion, chacune d'entre elles étant constituée de la modification du mode et de l'ID de connexion affecté. Le caractère générique "$" peut être utilisé pour indiquer la "connexion courante", cependant cette notation NE DOIT PAS être utilisée en dehors d'une commande de gestion des connexions; le caractère générique fait référence à la connexion concernée pour la commande de gestion des connexions.

L'action ModifyConnection intégrée permet à l'agent d'appel d'ordonner au point d'extrémité de changer de mode de connexion d'une ou plusieurs connexions suivant immédiatement la détection de l'événement associé. Chaque changement de mode de connexion fonctionne de la même manière qu'une commande ModifyConnection correspondante[14]. Lorsqu'une liste de changements de modes de connexion est fournie, ceux-ci DOIVENT être appliqués l'un après l'autre, de gauche à droite. Une fois tous les changements de modes de connexion terminés, un événement "opération terminée", paramétré avec le nom de l'action terminée, est généré (pour plus de détails, voir l'Annexe A). En cas

---

[13] Chiffre comme défini dans des scripts de numérotation, c'est-à-dire comprenant une astérisque ou une temporisation, etc.

[14] Ainsi, si par exemple une D-QoS est utilisée sur la connexion, l'action D-QoS par défaut sera toujours effectuée lorsque l'action ModifyConnection imbriquée sera exécutée.

d'échec de l'un des changements de mode de connexion, un événement "échec de l'opération", paramétré avec le nom de l'action échouée et le nom du changement de mode de connexion qui a échoué, est généré (pour plus de détails, voir l'Annexe A); les autres changements de modes de connexion NE DOIVENT PAS être tentés. De plus, les changements de modes de connexion réussis dans la liste NE DOIVENT PAS être modifiés.

Enfin, l'action Ignorer peut être utilisée pour ignorer un événement, par exemple, pour empêcher la notification d'un événement persistant. Toutefois, la synchronisation entre l'événement et un signal actif se produira toujours par défaut.

Le paragraphe 6.4.3.1 contient des détails supplémentaires quant à la sémantique de la détection d'événement et à l'élaboration de rapports. Le lecteur est encouragé à la lire attentivement.

La définition précise des actions demandées via ces SignalRequests (par exemple, la durée et la fréquence d'un chiffre DTMF) ne s'inscrit pas dans le cadre de la principale spécification NCS. Cette définition peut varier d'un endroit à l'autre, et donc, d'une passerelle à l'autre. Par conséquent, ces définitions sont fournies dans des paquets d'événements, qui peuvent être spécifié en dehors de la spécification principale. Une liste initiale de paquets d'événements se trouve à l'Annexe A.

Les RequestedEvents et les SignalRequests font généralement référence aux même événements. Dans le premier cas, la passerelle doit détecter l'irruption d'un événement, et dans le second, elle doit le générer. Cette règle comporte des exceptions, par exemple, les tonalités de fax et de modem, qui peuvent être détectées mais pas signalées. Toutefois, nous n'attendons pas forcément de tous les points d'extrémité qu'ils détectent tous les événements. Les événements et signaux spécifiques qu'un point d'extrémité donné peut détecter sont déterminées par la liste des paquets d'événements qui sont pris en charge par ce point d'extrémité. Chaque paquet contient une liste d'événements et signaux qui peuvent être détectés et appliqués. Une passerelle qui doit détecter ou appliquer un événement non pris en charge par le point d'extrémité concerné DOIT retourner une erreur (code d'erreur 512 ou 513 – pas équipé pour détecter un événement ou générer un signal). Lorsque le nom de l'événement n'est pas qualifié par un nom de paquet, c'est le nom de paquet par défaut du point d'extrémité qui est supposé. Si le nom de l'événement n'est pas inscrit dans le paquet par défaut, la passerelle DOIT retourner un code d'erreur (code d'erreur 522 – événement ou signal introuvable).

L'agent d'appel peut envoyer une NotificationRequest dont la liste de signaux demandés est vide. Ceci a pour effet d'arrêter tous les signaux de temporisation actifs. Cela peut arriver, par exemple, lorsque la génération d'une tonalité, par exemple de retour d'appel, doit s'arrêter.

QuarantineHandling est un paramètre facultatif qui spécifie les options de gestion du tampon de quarantaine (voir 6.4.3.1). Il permet à l'agent d'appel de spécifier si les événements mis en quarantaine devraient être traités ou ignorés. Si le paramètre est omis, les événements mis en quarantaine DOIVENT être traités.

DetectEvents est un paramètre facultatif qui spécifie une liste minimale d'événements que la passerelle doit détecter dans l'état "notification" ou "verrouillé". La liste est persistante jusqu'à ce qu'une nouvelle valeur soit spécifiée. Pour en savoir plus sur ce paramètre, voir 6.4.3.1.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

6.3.2 Notifications

Les notifications sont envoyées via la commande Notify par la passerelle lorsqu'un événement observé doit être notifié:

```
ReturnCode
          ← Notify(EndpointId
                        [, NotifiedEntity]
                        , RequestIdentifier
                        , ObservedEvents)
```

EndpointId est le nom du point d'extrémité dans la passerelle, qui émet la commande Notify, comme défini au 6.1.1. L'identificateur DOIT être un nom complet de point d'extrémité, comprenant le nom de domaine de la passerelle. La partie locale du nom NE DOIT PAS utiliser la convention de remplacement par caractères génériques.

NotifiedEntity est un paramètre facultatif qui identifie l'entité à laquelle la notification est envoyée. Ce paramètre équivaut au paramètre NotifiedEntity de la NotificationRequest qui a déclenché cette notification. Le paramètre est omis si aucun paramètre de ce type n'était présent dans la requête. Quelle que soit la valeur du paramètre NotifiedEntity, la notification DOIT être envoyée au paramètre "entité avisée" courant du point d'extrémité.

RequestIdentifier est un paramètre qui répète le paramètre RequestIdentifier de la NotificationRequest à l'origine de cette notification. Il est utilisé pour mettre en relation cette notification avec la requête de notification qui l'a générée. Les événements persistants seront considérés ici comme s'ils avaient été inclus dans la dernière NotificationRequest. Lorsque aucune NotificationRequest n'a été reçue, le RequestIdentifier utilisé est zéro ("0").

ObservedEvents est une liste d'événements détectés et cumulés par la passerelle, à l'aide d'une action "cumuler", "cumuler en fonction du script de numérotation" ou "notifier". Une notification unique peut reporter une liste d'événements dans l'ordre dans lequel ils ont été détectés. La liste ne peut contenir que des événements persistants et des événements qui ont été demandés dans le paramètre RequestedEvents de la NotificationRequest qui a généré la notification. Les événements qui ont été détectés sur une connexion comprennent le nom de cette connexion. La liste contient les événements qui ont été cumulés (mais pas notifiés) ou cumulés en fonction du script de numérotation (sans aucune correspondance encore), ainsi que l'événement final qui a déclenché la notification ou fourni une dernière correspondance dans le script de numérotation. A noter que les chiffres sont ajoutés à la liste des événements observés comme ils sont cumulés, sans prendre en compte le fait qu'ils aient été cumulés en fonction du script de numérotation ou non. Par exemple, si un utilisateur saisit les chiffres "1234" et qu'un événement E est cumulé entre les chiffres "3" et "4" en cours de saisie, la liste des événements observés donnera "1, 2, 3, E, 4".

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

6.3.3 CreateConnection

Cette commande est utilisée pour créer une connexion.

```
ReturnCode
, ConnectionId
[, SpecificEndPointId]
, LocalConnectionDescriptor
[, ResourceID]
          ← CreateConnection(CallId
                    , EndpointId
                                        [, NotifiedEntity]
                                        , LocalConnectionOptions
                                        , Mode
                                        [, RemoteConnectionDescriptor]
                                        [, RequestedEvents]
                                        [, RequestIdentifier]
                                        [, DigitMap]
                                        [, SignalRequests]
                                        [, QuarantineHandling]
                                        [, DetectEvents])
```

Cette fonction est utilisée lors de la configuration d'une connexion entre deux points d'extrémité. Une connexion est définie par ses attributs et les points d'extrémité qu'elle relie. Les paramètres d'entrée de CreateConnection fournissent les données nécessaires pour construire l'une des deux "vues" des points d'entrées d'une connexion.

CallId est un paramètre qui sert à identifier l'appel (ou session) auquel appartient la connexion. Ce paramètre doit être au minimum unique au sein d'une collection d'agents d'appel qui contrôle les mêmes passerelles; les connexions appartenant au même appel partagent le même id d'appel. L'id d'appel peut être utilisé pour identifier des appels à des fins de rapport ou de comptabilisation.

EndpointId est l'identificateur du point d'extrémité dans la passerelle dans laquelle CreateConnection s'exécute. Le EndpointId peut être spécifié complètement en affectant une valeur sans caractère générique au paramètre EndpointId dans l'appel de fonction, ou il peut être sous-spécifié en utilisant la convention de remplacement par caractère générique "anyone". Si le point d'extrémité est sous-spécifié, l'identificateur de point d'extrémité sera attribué par la passerelle et sa valeur complète retournée dans le paramètre **SpecificEndPointId** de la réponse. La convention de remplacement "all" NE DOIT PAS être utilisée.

NotifiedEntity est un paramètre facultatif qui spécifie une nouvelle "entité avisée" pour le point d'extrémité.

LocalConnectionOptions est une structure qui décrit les caractéristiques d'une connexion de données médias du point de vue de la passerelle qui exécute la commande CreateConnection. Elle informe le point d'extrémité des caractéristiques d'envoi et de réception de la connexion média. Les principaux champs contenus dans le paramètre LocalConnectionOptions sont les suivants:

- **Encoding Method**: liste de noms littéraux pour l'algorithme de compression (méthode de codage/décodage) utilisé pour envoyer et recevoir des médias sur la connexion qui DOIT être spécifiée dans au moins une valeur. Les entrées de la liste sont classées par préférence. Le point d'extrémité DOIT choisir exactement l'un des codecs, et le codec DEVRAIT être choisi en fonction de la préférence indiquée. Si le point d'extrémité reçoit un média quelconque sur la connexion codée avec une méthode de codage différente, il PEUT l'ignorer. Le point d'extrémité DOIT en plus indiquer lequel des algorithmes de compression restants qu'il souhaite prendre en charge comme variante; pour plus de détails, voir 7.4.1. Une liste de méthodes de codage admises est spécifiée dans un document IPCablecom distinct.
- **Packetization Period**: période de mise en paquet exprimée en millisecondes, telle que définie par la norme SDP (RFC 2327), qui DOIT être spécifiée à l'aide d'une seule valeur. La valeur n'appartient qu'au média envoyé. Une liste des périodes de mise en paquet admises est spécifiée dans un document IPCablecom distinct.
- **Echo Cancellation**: permet de spécifier si l'annulation de l'écho doit être utilisée ou non du côté ligne[15]. Le paramètre peut prendre la valeur "on" (lorsque l'annulation de l'écho est demandée) ou "off" (lorsqu'elle est désactivée). Ce paramètre est facultatif. Lorsque le paramètre est omis, le client intégré DOIT appliquer l'annulation de l'écho.
- **Type of Service**: spécifie la classe de service qui sera utilisée pour envoyer des médias sur la connexion, en codant sur deux chiffres hexadécimaux le paramètre valeur du type de service 8 bits de l'en-tête IP. Ce paramètre est facultatif. Lorsque ce paramètre est omis, la valeur par défaut $A0_H$ s'applique, ce qui correspond à une priorité IP de cinq bits.

[15] L'annulation d'écho côté paquet n'est pas prise en charge.

- **Silence Suppression**: spécifie si la suppression du silence doit être utilisée ou non dans le sens envoi. Le paramètre peut prendre la valeur "on" (lorsque le silence doit être supprimé) ou "off" (lorsque le silence ne doit pas être supprimé). Ce paramètre est facultatif. Lorsque le paramètre est omis, la valeur par défaut est de ne pas utiliser la suppression du silence.

Les champs suivants du paramètre LocalConnectionOptions servent à prendre en charge la qualité de service dynamique (D-QoS, *dynamic quality of service*); pour plus de détails, voir l'Annexe B:

- **D-QoS GateID**: le GateID de la passerelle qui a été configurée au niveau du routeur d'accès. Le Gate-ID est un identificateur 32 bits codé sous forme de chaîne pouvant contenir jusqu'à 8 caractères hexadécimaux. En général, ce paramètre est facultatif, mais obligatoire lorsque la réservation et/ou l'affectation de ressource D-QoS vont être exécutées. La présence de ce paramètre implique que la D-QoS va être exécutée sur cette commande, tandis que son absence indique qu'aucune D-QoS ne va être exécutée.
- **D-QoS Resource Reservation**: permet un contrôle explicite pour savoir si la réservation et/ou l'affectation de ressource D-QoS doivent être exécutées ou non, dans le sens envoi et/ou réception. Ce paramètre est facultatif et peut prendre l'une ou plusieurs des valeurs suivantes:

Valeurs de réserve:

- "SendReserve" Les ressources sont réservées dans le sens envoi uniquement.
- "ReceiveReserve" Les ressources sont réservées dans le sens réception uniquement.
- "SendReceiveReserve" Les ressources sont réservées dans les sens envoyer et recevoir.

Valeurs d'affectation:

- "SendCommit" Les ressources sont affectées dans le sens envoi uniquement.
- "ReceiveCommit" Les ressources sont affectées dans le sens réception uniquement.
- "SendReceiveCommit" Les ressources sont affectées dans les sens envoyer et recevoir.

Ce paramètre est facultatif; des valeurs multiples sont séparées par des virgules. Lorsque D-QoS doit être effectuée, et que le paramètre est omis ou qu'il ne prend aucune valeur, la réservation de ressources DOIT être effectuée pour les sens envoi et réception. Les ressources réservées sont déterminées par les paramètres de codage appliquées à la connexion, c'est-à-dire, la méthode de codage, la période de mise en paquets, la suppression du silence, le système cryptographique, etc. Des paramètres externes, tels que l'utilisation de la suppression d'un en-tête de charge utile peut également affecter la quantité des ressources réservées; pour plus de détails, voir la spécification IPCablecom sur la qualité dynamique de service, UIT-T J.163.

Les ressources reçues peuvent être réservées et affectées sans avoir obtenu un RemoteConnectionDescriptor, tandis que les ressources envoyées peuvent être réservées mais pas affectées, jusqu'à ce qu'un RemoteConnectionDescriptor soit fourni. Lorsque la réservation D-QoS va être effectuée, et que le paramètre est omis ou qu'il ne prend aucune valeur, les ressources DOIVENT être affectées par défaut, en fonction du mode de connexion spécifié dans le tableau suivant:

Mode de connexion	D-QoS
"inactive"	Ne pas affecter
"envoyer uniquement", "répliquer"	Affecter dans le sens envoyer
"recevoir uniquement"	Affecter dans le sens recevoir
"envoyer/recevoir", "conférence", "boucle réseau", "test de continuité réseau"	Affecter dans les sens envoyer et recevoir

Si une opération d'affectation différente est demandée, la valeur d'affectation appropriée est fournie et sera utilisée à la place. Si une opération d'affectation a été exécutée, mais qu'aucune réservation n'a été effectuée, ou qu'une réservation existante ne satisfait pas complètement aux ressources à affecter[16], une réservation sera automatiquement exécutée. Si une valeur de réserve est spécifiée mais qu'aucune valeur d'affectation ne l'est, aucune opération d'affectation ne sera exécutée.

- **ResourceID**: resourceID existant pour les ressources déjà réservées sur le routeur d'accès. L'utilisation du ResourceID permet à différentes réservations de réserver la même ressource, toutefois une seule des réservations peut être active à un moment donné. Le paramètre ResourceID est un identificateur de 32 bits codé sous forme de chaîne pouvant contenir jusqu'à 8 caractères hexadécimaux. Ce paramètre est facultatif.
- **ReserveDestination**: ce paramètre facultatif peut spécifier une adresse IPv4, éventuellement suivie de deux points et d'un numéro de port UDP, qui indique la destination de la réservation de ressource. Si le numéro de port UDP n'est pas spécifié, c'est la valeur par défaut 9 qui s'applique. Le paramètre ReserveDestination est généralement utilisé lorsque la réservation de ressource va être exécutée, et qu'aucun RemoteConnectionDescriptor n'a encore été défini pour la connexion. Cela permet aux réservations et aux affectations dans le sens descendant d'être envoyées au routeur d'accès lorsque la source du flux de médias n'est pas encore connue[17]. Une fois le RemoteConnectionDescriptor spécifié, ce paramètre est ignoré.

Les champs suivants du paramètre LocalConnectionOptions sont utilisés pour prendre en charge les services de sécurité IPCablecom:

- **Secret**: le paramètre secret facultatif est une valeur essentielle qui DOIT être utilisée pour dériver des clés de chiffrement de bout en bout destinées aux services de sécurité RTP et RTCP, comme spécifié dans la spécification de sécurité IPCablecom (en cours d'élaboration). Le secret DEVRAIT être codé comme texte en clair s'il ne contient que des valeurs comprises dans la plage des caractères ASCII 21_H à $7E_H$. Sinon, le secret DOIT être codé à l'aide du codage de base64. Si aucune valeur n'est spécifiée, ou si le paramètre est omis et que les services de sécurité doivent être utilisés, le point d'extrémité DOIT générer un secret lui-même[18]. Si un secret est fourni par l'agent d'appel, le secret DEVRAIT être utilisé.
- **Système cryptographique RTP**: liste de systèmes cryptographiques destinée à la sécurité RTP dans l'ordre de préférence. Les entrées de la liste sont classées par ordre de préférence, le premier système cryptographique étant le préféré. Le point d'extrémité DOIT choisir un seul des systèmes cryptographiques. En outre, il DEVRAIT indiquer lesquels des

---

[16] Cela n'est pas possible pour la commande CreateConnection mais figure ici dans un souci d'exhausitivité. C'est toutefois possible pour la commande ModifyConnection (voir 6.3.4).

[17] A noter que cela autorisera certains scénarios de vol de services. Reportez-vous au document "Dynamic Quality of Service Specification (UIT-T J.163)" pour plus de détails.

[18] Cela comprend à la fois le nouveau secret et l'utilisation d'un secret fourni dans un RemoteConnectionDescriptor.

systèmes cryptographiques restants il souhaite prendre en charge comme solutions de remplacement (pour plus de détails, voir 7.4.1). Chaque système cryptographique est représenté sous forme d'une chaîne ASCII composée de deux sous-chaînes (éventuellement vides), séparées par une barre oblique ("/"), où la première sous-chaîne identifie l'algorithme d'authentification et la seconde sous-chaîne identifie l'algorithme de chiffrement. Une liste des systèmes cryptographiques admis figure dans la spécification de sécurité IPCablecom UIT-T J.170 (en cours d'élaboration).

- **Système cryptographique RTCP**: liste de systèmes cryptographiques destinés à la sécurité RTCP dans l'ordre de préférence. Les entrées de la liste sont classées par ordre de préférence, le premier système cryptographique étant le préféré. Le point d'extrémité DOIT choisir un et un seul des systèmes cryptographiques. En outre, il DEVRAIT indiquer lesquels des systèmes cryptographiques restants il souhaite prendre en charge comme solutions de remplacement (pour plus de détails, voir 7.4.1). Chaque système cryptographique est représenté sous forme d'une chaîne ASCII composée de deux sous-chaînes (éventuellement vides), séparées par une barre oblique ("/"), où la première sous-chaîne identifier l'algorithme d'authentification et la seconde sous-chaîne identifie l'algorithme de chiffrement. Une liste des systèmes cryptographiques admis figure dans la spécification de sécurité IPCablecom UIT-T J.170 (en cours d'élaboration).

Le client intégré DOIT retourner une erreur (code d'erreur 524 – Incohérence dans le paramètre LocalConnectionOptions) si l'une des règles énoncées ci-dessous est enfreinte. Toutes les valeurs par défaut indiquées ci-dessus peuvent être modifiées par le processus de mise en service.

RemoteConnectionDescriptor est le descripteur de connexion pour le côté distant d'une connexion, de l'autre côté du réseau IP. Il comporte les mêmes champs que le paramètre LocalConnectionDescriptor (à ne pas confondre avec LocalConnectionOptions), c'est-à-dire, les champs qui décrivent une session en fonction de la norme SDP. Le paragraphe 7.4 décrit en détail l'utilisation prise en charge de of SDP dans le profil NCS. Ce paramètre peut prendre une valeur nulle si l'information relative à l'extrémité distante n'est pas connue. Cela se produit car l'entité qui établit une connexion commence par envoyer une commande CreateConnection à l'une des deux passerelles concernées. Lors de la première émission de la commande CreateConnection, aucune information n'est disponible concernant l'autre extrémité de la connexion. Ces informations peuvent être fournies ultérieurement via un appel ModifyConnection.

Le profil NCS suppose pour l'instant que les mêmes paramètres médias s'appliquent à une connexion dans les deux sens: envoi et réception. Une partie des informations contenues dans le paramètre RemoteConnectionDescriptor est alors redondante; par conséquent, il existe un risque d'incohérence avec le paramètre LocalConnectionOptions. Toutefois, il appartient totalement à l'agent d'appel de s'assurer de la cohérence des commandes qu'il envoie à chaque point d'extrémité afin de garantir la cohérence des paramètres médias spécifiés. Si toutefois un conflit est détecté par la passerelle, le paramètre LocalConnectionOptions aura tout simplement priorité. Lorsque les codecs sont modifiés au cours d'un appel, il peut y exister de courtes périodes de temps durant lesquelles les points d'extrémité utilisent des codes différents. Comme indiqué ci-dessus, les clients imbriqués PEUVENT ignorer tout média reçu s'il est codé à l'aide d'un codec différent de celui spécifié dans le paramètre LocalConnectionOptions d'une connexion.

Mode indique le mode de fonctionnement de ce côté-ci de la connexion. Les options sont "envoyer uniquement", "recevoir uniquement", "envoyer/recevoir", "conférence", "inactive", "répliquer", "boucle réseau", ou "test de continuité réseau". La gestion de ces modes est spécifiée au début du 6.3. Certains points d'extrémité ne seront peut-être pas capables de prendre en charge tous ces modes. Si la commande spécifie un mode non pris en charge par le point d'extrémité, une erreur DOIT être retournée (code d'erreur 517 – mode non pris en charge). De plus, si une connexion n'a pas encore reçu de RemoteConnectionDescriptor, une erreur DOIT être retournée si la connexion est

tentée dans l'un de ces modes "envoyer uniquement", "envoyer/recevoir", "répliquer" ou "conférence" (code d'erreur 527 – RemoteConnectionDescriptor manquant).

ConnectionId est un paramètre retourné par la passerelle qui permet d'identifier de manière unique la connexion dans le contexte du point d'extrémité en question.

LocalConnectionDescriptor est un paramètre retourné par la passerelle, qui est une description de session contenant des informations telles que les adresses et les ports RTP pour les connexions "IN" définies dans SDP. Il est similaire au RemoteConnectionDescriptor, excepté qu'il spécifie ce côté-ci de la connexion. Le paragraphe 7.4 décrit en détail l'utilisation prise en charge de SDP dans le profil NCS.

Une fois qu'elle a reçu la commande "CreateConnection" qui ne comprend pas de paramètre RemoteConnectionDescriptor, une passerelle se trouve dans une situation ambiguë pour la connexion en question. Du fait qu'elle a exporté un paramètre LocalConnectionDescriptor, elle peut potentiellement recevoir des paquets sur cette connexion. Mais comme elle n'a pas encore reçu le paramètre RemoteConnectionDescriptor de l'autre passerelle, elle ne sait pas si les paquets qu'elle reçoit ont été autorisés ou non par l'agent d'appel. Ainsi, elle doit choisir entre deux risques: supprimer des annonces importantes ou écouter des données non valides. Le comportement de la passerelle est déterminé par la valeur du paramètre mode (qui dépend de la sécurité):

- si le mode est défini sur "recevoir uniquement", la passerelle DOIT accepter les signaux vocaux sur la connexion, puis les transmettre au point d'extrémité;
- si le mode est défini sur "inactive", la passerelle DOIT (comme toujours) ignorer les signaux vocaux reçus sur la connexion;
- si le mode est défini sur "boucle réseau" ou sur "test de continuité réseau", la passerelle DOIT exécuter l'écho ou la réponse attendue. Le média renvoyé en écho ou généré DOIT ensuite être envoyé à la source du média reçu;
- à noter que lorsque le point d'extrémité ne possède pas de RemoteConnectionDescriptor pour la connexion, celle-ci ne peut, par définition, prendre aucun des modes "envoyer uniquement", "envoyer/recevoir", "répliquer" ou "conférence".

Les paramètres **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** et **DetectEvents** sont tous facultatifs. Ils peuvent être utilisés par l'agent d'appel pour inclure une requête de notification qui est exécutée en même temps que la connexion est établie. Si l'un de ces paramètres ou plusieurs sont présents, RequestIdentifier DOIT être l'un d'entre eux. Ainsi, l'inclusion d'une requête de notification peut être reconnue par la présence d'un RequestIdentifier. Le reste des paramètres peut être présent ou non. Si l'un des paramètres est omis, il DOIT être traité comme s'il s'agissait d'une NotificationRequest normale, le paramètre en question étant omis. Cela peut entraîner l'annulation de signaux et l'arrêt d'événement de recherche.

Comme exemple d'utilisation, considérons un agent d'appel qui souhaite émettre un appel vers un client intégré. L'agent d'appel devrait:

- demander au client intégré de créer une connexion afin de s'assurer que l'utilisateur peut commencer à parler dès que le téléphone est décroché;
- demander au client intégré de commencer à sonner;
- demander au client intégré d'aviser l'agent d'appel lorsque le téléphone est décroché.

Toutes les actions décrites ci-dessus peuvent être accomplies à l'aide d'une seule commande CreateConnection en y intégrant une requête de notification assortie des paramètres RequestedEvents de l'événement décrochage et du paramètre SignalRequests pour le signal de sonnerie.

Lorsque ces paramètres sont présents, la création d'une connexion et la requête de notification DOIVENT être synchronisées, ce qui signifie qu'elles sont toutes deux soit acceptés, soit refusées. Dans notre exemple, la commande CreateConnection doit être refusée si la passerelle ne possède pas suffisamment de ressources ou ne peut pas obtenir les ressources appropriées du réseau local. La requête de notification de décrochage doit être refusée en cas de collision d'appel, si l'utilisateur a déjà décroché le combiné. Dans cet exemple, le téléphone ne doit pas sonner si la connexion ne peut pas être établie, et la connexion ne doit pas être établie si l'utilisateur a déjà décroché le combiné. Une erreur est retournée à la place (code d'erreur 401 – téléphone décroché), qui informe l'agent d'appel de la condition de collision d'appels.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

ResourceID est un paramètre D-QoS qui peut être retourné par la passerelle. Lorsqu'une réservation de ressource D-QoS réussie est exécutée, le paramètre ResourceID fournit l'identificateur pour les ressources réservées.

6.3.4 ModifyConnection

Cette commande est utilisée pour modifier les caractéristiques de la "vue" qu'a la passerelle sur une connexion. Cette "vue" de l'appel comporte le descripteur de connexion locale et le descripteur de connexion distante.

```
ReturnCode
     [, LocalConnectionDescriptor]
     [, ResourceID]
               ← ModifyConnection(CallId
                         , EndpointId
                         , ConnectionId
                         [, NotifiedEntity]
                         [, LocalConnectionOptions]
                         [, Mode]
                         [, RemoteConnectionDescriptor]
                         [, RequestedEvents]
                         [, RequestIdentifier]
                         [, DigitMap]
                         [, SignalRequests]
                         [, QuarantineHandling]
                         [, DetectEvents])
```

Les paramètres utilisés sont identiques à ceux de la commande CreateConnection, avec en plus un paramètre **ConnectionId** qui identifie de manière unique la connexion par rapport au point d'extrémité. Ce paramètre est retourné par la commande CreateConnection avec le descripteur de connexion local. Il permet d'identifier de manière unique la connexion dans le contexte du point d'extrémité.

Le **EndpointId** DOIT être un nom de point d'extrémité complet. Le nom local NE DOIT PAS utiliser la convention par remplacement de caractère.

La commande ModifyConnection peut être utilisée pour définir des paramètres de connexion; elle est soumise aux même règles et contraintes que la commande CreateConnection:

- fournir des informations relatives à l'autre extrémité de la connexion, à l'aide du **RemoteConnectionDescriptor**;
- activer ou désactiver la connexion en changeant la valeur du paramètre **mode**. Cela peut se produire à tout moment au cours de la connexion, avec des valeurs de paramètre arbitraires. Une activation peut être définie par exemple, sur le mode "recevoir uniquement";
- modifier les paramètres de la connexion à l'aide du paramètre **LocalConnectionOptions**, par exemple, en passant à un système de codage différent, en modifiant la période de mise en paquets, ou en modifiant la gestion de l'annulation de l'écho.

Les détails de l'opération de D-QoS ont été spécifiés dans la commande CreateConnection. En règle générale, les mêmes règles s'appliquent ici, sauf celles énumérées ci-dessous:

- **D-QoS GateID**: un GateID de la D-QoS est obligatoire lorsque l'opération de D-QoS est demandée, à moins qu'elle n'ait été déjà effectuée pour la connexion concernée. Dans ce dernier cas, c'est le dernier GateID de D-QoS GateID qui sera utilisé.
- **D-QoS Resource Reservation**: permet un contrôle explicite pour savoir si la réservation et/ou l'affectation de ressource D-QoS doivent être exécutées ou non, dans le sens envoi et/ou réception. Le paramètre est facultatif et peut prendre plusieurs valeurs. Lorsque le paramètre est omis et que la réservation D-QoS doit être effectuée, par défaut la réservation se fait dans les deux sens (envoi et réception), à moins qu'une réservation acceptable pour la connexion n'ait été déjà effectuée (voir Annexe B). Dans ce cas, aucune nouvelle réservation ne sera faite. Les ressources sont affectées de la même manière que pour CreateConnection, sauf que la connexion passe en mode " inactive ". Dans ce cas, les ressources affectées DOIVENT être égales à zéro. Toutefois, une réservation de ressource existante est conservée.
- **ResourceID**: ce paramètre est facultatif. S'il est fourni, il remplace le ResourceID conservé par le client intégré pour la connexion.
- **ReserveDestination**: ce paramètre est facultatif. Lorsqu'il est fourni, il remplace le ReserveDestination conservé par le client intégré pour la connexion. Si un RemoteConnectionDescriptor a été spécifié pour la connexion, le paramètre est ignoré.

Cette commande ne retourne qu'un **LocalConnectionDescriptor** si les paramètres de connexion locale, tels que les ports RTP, etc. sont modifiés. Ainsi, si par exemple, seul le mode de connexion est modifié, aucun LocalConnectionDescriptor ne sera retourné. Si un paramètre de connexion est omis (par exemple, mode ou suppression du silence), l'ancienne valeur de ce paramètre sera retenue si possible. Si la modification d'un paramètre nécessite la modification d'un ou plusieurs paramètres *non spécifiés*, la passerelle est libre de choisir les valeurs adaptées aux paramètres non spécifiés qui doivent changer[19].

L'information d'adresse RTP fournie par le RemoteConnectionDescriptor spécifie l'adresse RTP distante du récepteur de média pour la connexion. Cette information d'adresse RTP peut avoir été modifiée par l'agent d'appel[20]. Lorsque l'adresse RTP est fournie au client intégré pour la connexion, celui-ci ne DEVRAIT accepter que des flux de médias (et RTCP) de l'adresse RTP spécifiée. Il CONVIENT d'ignorer tout flux de média émanant d'une autre adresse. Il est conseillé de consulter la spécification de sécurité IPCablecom UIT-T J.170 (en cours d'élaboration) pour en savoir plus sur la sécurité.

Les paramètres **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** et **DetectEvents** sont facultatifs. Ils peuvent être utilisés par l'agent d'appel pour intégrer une notification de requête qui est liée et exécutée en même temps que la modification de la connexion. Si un ou plusieurs paramètres sont spécifiés, le RequestIdentifier DOIT être l'un d'eux. Par exemple, lorsqu'un appel est accepté, la passerelle appelante doit recevoir l'ordre de placer la connexion en mode "envoyer/recevoir" et d'arrêter de fournir des tonalités de retour d'appel. Cette action peut être exécutée par une seule commande ModifyConnection en lui intégrant une requête de notification, avec les paramètres RequestedEvents pour l'événement de raccrochage, et un paramètre SignalRequests vide pour arrêter l'exécution de tonalités de retour d'appel.

---

[19] Cela peut se produire, par exemple, si la modification d'un codec est spécifiée, et que l'ancien codec utilisait la suppression du silence, mais que le nouveau ne la prend pas en charge. Si, par exemple, une nouvelle période de mise en paquets n'était pas spécifiée alors que le nouveau codec prenait en charge l'ancienne période de mise en paquets, la valeur de ce paramètre ne changerait pas, car cela serait inutile.

[20] Par exemple, si le média doit traverser un pare-feu.

Lorsque ces paramètres sont présents, la modification de la connexion et la requête de notification DOIVENT être synchronisées, ce qui signifie qu'elles sont toutes deux acceptées ou refusées.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

ResourceID est un paramètre D-QoS retourné par la passerelle si elle exécute une réservation de ressource et obtient un nouvel identificateur ResourceID fourni par le routeur d'accès. Lorsqu'une réservation de ressource D-QoS réussie est exécutée, le paramètre ResourceID fournit l'identificateur pour les ressources réservées.

6.3.5 Commande DeleteConnection (lancée par l'agent d'appel)

Cette commande sert à mettre fin à une connexion. Elle recueille également des statistiques sur l'exécution de la connexion.

```
ReturnCode
, Connection-parameters
     ← DeleteConnection(CallId
                        , EndpointId
                        , ConnectionId
                        [, NotifiedEntity]
                        [, RequestedEvents]
                        [, RequestIdentifier]
                        [, DigitMap]
                        [, SignalRequests]
                        [, QuarantineHandling]
                        [, DetectEvents])
```

Dans cette forme de la commande DeleteConnection, l'identificateur de point d'extrémité DOIT être complet. Les conventions de remplacement par caractères génériques NE DOIVENT PAS être utilisées.

Dans le cas général où une connexion comporte deux extrémités, cette commande doit être envoyée aux deux passerelles impliquées dans la connexion. Après la suppression de la connexion, les flux médias du réseau de paquets qui étaient pris en charge par la connexion ne sont plus disponibles. Les éventuels paquets médias reçus pour l'ancienne connexion sont tout simplement éliminés et il n'en est pas envoyé de nouveaux pour le flux. Lorsqu'on avait procédé à une ou plusieurs réservations et/ou affectations de ressources D-QoS, la commande DeleteConnection aura pour effet de libérer les ressources réservées.

En réponse à la commande DeleteConnection, la passerelle renvoie une liste de paramètres qui décrivent l'état de la connexion. Ces paramètres sont:

- **le nombre de paquets envoyés**: nombre total de paquets de données RTP transmis par l'expéditeur depuis le début de la transmission sur la connexion. Le décompte n'est pas réinitialisé si l'expéditeur modifie son identificateur de source de synchronisation (SSRC, comme défini dans le protocole RTP) – par exemple, par suite d'une commande Modify. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "recevoir uniquement";
- **le nombre d'octets envoyés**: nombre total d'octets de charge utile (c'est-à-dire, en-tête ou bourrage non compris) transmis par l'expéditeur dans les paquets de données RTP depuis le début de la transmission sur la connexion. Le décompte n'est pas réinitialisé si l'expéditeur modifie son identificateur SSRC – par exemple, par suite d'une commande ModifyConnection. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "recevoir uniquement";

- **le nombre de paquets reçus**: nombre total de paquets de données RTP reçus par l'expéditeur depuis le début de la réception sur la connexion. Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "envoyer uniquement";
- **le nombre d'octets reçus**: nombre total d'octets de charge utile (c'est-à-dire, en-tête ou bourrage non compris) transmis par l'expéditeur dans les paquets de données RTP depuis le début de la transmission sur la connexion. Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "envoyer uniquement";
- **le nombre de paquets perdus**: nombre total de paquets de données RTP qui ont été perdus depuis le début de la réception. Ce nombre est défini comme étant le nombre de paquets attendus moins le nombre de paquets effectivement reçus (ce dernier comprend tous les paquets tardifs ou dupliqués éventuels). Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. Donc, les paquets qui arrivent en retard ne sont pas comptés comme perdus et la perte peut être négative s'il y a des doublons. Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. Le nombre de paquets attendus est défini comme étant le dernier numéro de séquence étendu reçu moins le numéro de séquence initial reçu. Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "envoyer uniquement";
- **gigue de réception intermédiaire**: estimation de la variance statistique du temps de réception intermédiaire des paquets de données RTP, mesurée en millisecondes et exprimée comme un entier non signé. La gigue de réception intermédiaire "J" est définie comme étant l'écart moyen (valeur absolue lissée) de la différence "D" d'espacement des paquets au niveau du récepteur comparé à celui au niveau de l'expéditeur, pour une paire de paquets donnée. On trouvera des algorithmes de calcul détaillés dans la RFC 1889. Le décompte inclut les paquets reçus d'un identificateur SSRC différent si l'expéditeur en a utilisé plusieurs valeurs. La valeur de ce paramètre est zéro si, par exemple, la connexion a toujours été établie dans le mode "envoyer uniquement";
- **temps moyen de transmission**: estimation de la latence dans le réseau, exprimée en millisecondes. Il s'agit de la valeur moyenne de la différence entre le marqueur temporel NTP indiqué par les expéditeurs des messages RTCP et celui indiqué par les récepteurs, mesurée au moment de la réception des messages. On obtient cette moyenne en calculant la somme de toutes les estimations divisée par le nombre des messages RTCP qui ont été reçus. Il convient de noter que le calcul correct de ce paramètre s'appuie sur des horloges synchronisées. En variante, les dispositifs de clients intégrés PEUVENT estimer le temps moyen de transmission en divisant par deux le temps aller-retour mesuré.

Se reporter à la RFC 1889 pour une définition plus détaillée de ces variables.

Les paramètres **NotifiedEntity**, **RequestedEvents**, **RequestIdentifier**, **DigitMap**, **SignalRequests**, **QuarantineHandling** et **DetectEvents** sont facultatifs. L'agent d'appel peut les utiliser pour transmettre une demande de notification qui est exécutée simultanément avec la suppression de la connexion à laquelle elle est liée. Cependant, en cas de présence de l'un ou plusieurs de ces paramètres, RequestIdentifier DOIT être l'un d'entre eux. Par exemple, lorsqu'un usager raccroche le combiné, il se pourrait que la passerelle reçoive l'instruction de supprimer la connexion et de commencer à rechercher un événement décrochage. Cela peut être réalisé dans une commande DeleteConnection unique en transmettant le paramètre RequestedEvents pour l'événement décrochage et un paramètre SignalRequests vide.

Lorsque ces paramètres sont présents, la suppression de connexion et la requête de notification DOIVENT être synchronisées, ce qui signifie qu'elles sont toutes deux acceptées ou refusées.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

6.3.6 DeleteConnection (lancée par le client intégré)

Dans un certain nombre de circonstances, il se peut qu'une passerelle doive libérer une connexion, par exemple, parce qu'elle a perdu les ressources associées à cette connexion. La passerelle peut mettre fin à la connexion en utilisant une variante de la commande DeleteConnection:

```
ReturnCode
     ← DeleteConnection(CallId,
                        EndpointId,
                        ConnectionId,
                        Reason-code,
                        Connection-parameters)
```

Dans cette forme de la commande DeleteConnection, l'identificateur **EndpointId** DOIT être complet. Les conventions de remplacement par caractères génériques NE DOIVENT PAS être utilisées.

Le paramètre **Reason-code** est une chaîne textuelle qui commence par un code de cause numérique, suivi éventuellement d'une chaîne textuelle descriptive. On trouvera une liste de codes de cause au 6.6.

Outre les paramètres **CallId**, **EndpointId** et **ConnectionId,** le client intégré envoie également les paramètres de connexion, qui auraient été renvoyés à l'agent d'appel en réponse à une commande DeleteConnection lancée par ce dernier. Le code de cause indique la cause de la suppression de connexion. Lorsqu'on avait procédé à une ou plusieurs réservations et/ou affectations de ressources D-QoS, le client intégré libérera les ressources réservées.

ReturnCode est un paramètre retourné par l'agent d'appel. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

6.3.7 DeleteConnection (plusieurs connexions depuis l'agent d'appel)

L'agent d'appel peut utiliser une variante de la fonction DeleteConnection pour supprimer plusieurs connexions à la fois. La commande peut servir à supprimer toutes les connexions qui se rapportent à un appel pour un point d'extrémité:

```
ReturnCode
     ← DeleteConnection(CallId,
                        EndpointId)
```

Dans cette forme de la commande DeleteConnection, le paramètre **EndpointId** NE DOIT PAS utiliser la structure générique "any of". Toutes les connexions pour le ou les points d'extrémité avec le paramètre CallId spécifié seront supprimées. La commande ne renvoie pas de statistique individuelle ou de paramètre d'appel.

La commande DeleteConnection peut également être utilisée par l'agent d'appel pour supprimer toutes les connexions qui se terminent en un point d'extrémité donné:

```
ReturnCode
     ← DeleteConnection(EndpointId)
```

Dans cette forme de la commande DeleteConnection, les agents d'appel peuvent tirer profit de la structure hiérarchique de nommage des points d'extrémité pour supprimer toutes les connexions qui appartiennent à un groupe de points d'extrémité. Dans ce cas, on peut spécifier une partie de la composante "local endpoint name" (*nom local de point d'extrémité*) du paramètre EndpointId à l'aide de la convention de remplacement par le caractère générique "all", comme spécifié au 6.1.1. La convention de remplacement par le caractère générique "any of" NE DOIT PAS être utilisée. La commande ne renvoie pas de statistique individuelle ou de paramètre d'appel.

Après la suppression de la connexion, les flux médias du réseau de paquets qui étaient pris en charge par la connexion ne sont plus disponibles. Les éventuels paquets médias reçus pour l'ancienne connexion sont tout simplement éliminés et il n'en est pas envoyé de nouveaux pour le flux. Lorsqu'on avait procédé à une ou plusieurs réservations et/ou affectations de ressources D-QoS, le client intégré libérera les ressources réservées.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

6.3.8 Auditing (Audit)

Le protocole MGCP est fondé sur une architecture de commande d'appel centralisée où un agent d'appel agit comme contrôleur distant de dispositifs clients qui assurent des interfaces vocales à des usagers et à des réseaux. Afin d'obtenir des niveaux de disponibilité égaux ou supérieurs à ceux du réseau RTPC actuel, un certain nombre de protocoles ont implémenté des mécanismes pour "sonder" périodiquement les abonnés afin de minimiser le temps avant la détection d'une panne individuelle. A cet effet, un mécanisme d'audit spécifique au protocole MGCP est prévu entre les clients intégrés et les agents d'appel dans un système IPCablecom; il permet à l'agent d'appel d'auditer l'état du point d'extrémité et de la connexion et de récupérer les capacités d'un point d'extrémité qui sont spécifiques du protocole.

Deux commandes d'audit sont définies pour les clients intégrés:

- **AuditEndPoint**: commande utilisée par l'agent d'appel pour déterminer l'état d'un point d'extrémité;
- **AuditConnection**: commande utilisée par l'agent d'appel pour obtenir des informatives concernant une connexion.

La gestion de réseau au-delà des capacités procurées par ces commandes est généralement souhaitable (par exemple, des informations relatives à l'état du client intégré par opposition aux points d'extrémité individuels). On s'attend à ce que de telles capacités soient prises en charge par l'utilisation du protocole SNMP (protocole simple de gestion de réseau) et par la définition d'une base de données MIB pour le client intégré, toutes deux ne s'inscrivant pas dans le cadre de la présente Recommandation.

6.3.8.1 AuditEndPoint

L'agent d'appel peut se servir de la commande AuditEndPoint pour déterminer l'état d'un point d'extrémité donné.

```
{ ReturnCode
          [, EndPointIdList]
          [, NumEndPoints] } |
{ ReturnCode
          [, RequestedEvents]
          [, DigitMap]
          [, SignalRequests]
          [, RequestIdentifier]
          [, NotifiedEntity]
          [, ConnectionIdentifiers]
          [, DetectEvents]
```

```
[, ObservedEvents]
[, EventStates]
[, Capabilities] }
     ← AuditEndPoint(EndpointId
                              [, RequestedInfo] |
                         [, SpecificEndPointID]
                                  [, MaxEndPointIDs] })
```

Le paramètre **EndpointId** identifie le point d'extrémité qui fait l'objet de l'audit. La convention de remplacement par le caractère générique "any of" NE DOIT PAS être utilisée.

La convention de remplacement par le caractère générique "all of" peut être utilisée pour auditer un groupe de points d'extrémité. Si cette convention est utilisée, la passerelle DOIT renvoyer la liste des identificateurs de point d'extrémité qui correspondent à la valeur générique dans le paramètre **EndPointIdList**, qui est simplement une liste de paramètres SpecificEndpointId – Le paramètre RequestedInfo NE DOIT PAS y être inclus dans ce cas. **MaxEndPointIDs** est une valeur numérique qui indique le nombre maximal d'identificateurs EndpointId devant être renvoyés. S'il y a d'autres points d'extrémité, le paramètre de retour **NumEndPoints** DOIT être présent et indiquer le nombre total de points d'extrémité qui correspondent au paramètre EndpointID spécifié. Afin de récupérer le prochain bloc d'identificateurs EndpointID, le paramètre **SpecificEndPointID** est réglé à la valeur du dernier point d'extrémité retourné dans le paramètre EndPointIDList précédent, et la commande est lancée.

Lorsque la convention de remplacement par caractères génériques n'est pas utilisée, le paramètre (éventuellement vide) **RequestedInfo** décrit les informations qui sont demandées pour l'identificateur EndpointId spécifié – Dans ce cas, les paramètres SpecificEndpointID et MaxEndpointID NE DOIVENT PAS être utilisés. Les informations suivantes, spécifiques à chaque point d'extrémité, peuvent alors être auditées avec cette commande:

> RequestedEvents, DigitMap, SignalRequests, RequestIdentifier, NotifiedEntity, ConnectionIdentifiers, DetectEvents, ObservedEvents, EventStates, VersionSupported, and Capabilities.

La réponse, à son tour, comprendra les informations concernant chacun des éléments pour lesquels des informations d'audit ont été demandées:

- **RequestedEvents** – Valeur actuelle des événements demandés que le point d'extrémité utilise, y compris l'action associée à chaque événement. Les événements persistants sont inclus dans la liste.
- **DigitMap** – Script de numérotation actuellement utilisé par le point d'extrémité.
- **SignalRequests** – Liste des signaux temporisés (Time-Out) qui sont actuellement actifs, des signaux commutés (On/Off) qui sont actuellement "on" pour le point d'extrémité (avec ou sans paramètre) ainsi que des éventuels signaux brefs (Brief) en attente[21]. Les signaux temporisés Time-Out qui ont expiré et les signaux brefs Brief qui sont actuellement lus ne sont pas inclus. Les signaux paramétrés sont communiqués avec les paramètres avec lesquels ils ont été appliqués.
- **RequestIdentifier** – Identificateur de demande (RequestIdentifier) pour la dernière demande de notification (NotificationRequest) reçue par le point d'extrémité (y est incluse la demande de notification intégrée dans les primitives de manipulation de la connexion).
- **NotifiedEntity** – "L'entité avisée" courante pour le point d'extrémité.
- **ConnectionIdentifiers** – Une liste d'identificateurs ConnectionIdentifiers (séparés par une virgule) pour toutes les connexions qui existent actuellement pour le point d'extrémité spécifié.

---

[21] Il convient que des signaux brefs (*Brief*) ne soient pas actuellement en attente.

- **DetectEvents** – Valeur actuelle des DetectEvents que le point d'extrémité utilise. Les événements persistants sont inclus dans la liste.
- **ObservedEvents** – Liste actuelle des événements observés pour le point d'extrémité.
- **EventStates** – Dans le cas d'événements qui comportent des états qu'il est possible d'auditer qui leur sont associés, il s'agit de l'événement correspondant à l'état dans lequel le point d'extrémité se trouve, par exemple, événement décrochage dans le paquet ligne exemple si le point d'extrémité est décroché. La définition des événements individuels indique si l'événement en question comporte un état qu'il est possible d'auditer qui lui est associé.
- **VersionSupported** – Liste des versions de protocole prises en charge par le point d'extrémité.
- **Capabilities** – Capacités pour le point d'extrémité qui sont similaires au paramètre LocalConnectionOptions et qui comprennent des paquets d'événements et des modes de connexion. S'il est nécessaire de spécifier qu'un certain nombre de paramètres (comme par exemple la suppression du silence) sont seulement compatibles avec quelques codecs, la passerelle renvoie plusieurs jeux de capacités. Si un point d'extrémité est interrogé sur une capacité qu'il ne comprend pas, il NE DOIT PAS générer une erreur; au contraire, le paramètre DOIT être omis dans la réponse:
- **Compression Algorithm** – Liste des codecs pris en charge. Le reste des paramètres s'applique à tous les codecs spécifiés dans cette liste.
- **Packetization Period** – On peut spécifier une valeur unique ou une plage.
- **Bandwidth** – On peut spécifier une valeur unique ou une plage correspondant à la plage pour les périodes de mise en paquets (en supposant qu'il n'y a pas de suppression du silence).
- **Echo Cancellation** – Indique si l'annulation d'écho est prise en charge ou non.
- **Silence Suppression** – Indique si la suppression du silence est prise en charge ou non.
- **Type of Service** – Indique si le type de service est pris en charge ou non.
- **Event Packages** – Liste des paquets d'événements pris en charge. Le premier paquet d'événement dans la liste est le paquet par défaut.
- **Modes** – Liste des modes de connexion pris en charge.
- **Dynamic Quality of Service** – Indique si la qualité de service dynamique est prise en charge ou non.
- **Security** – Indique si les services de sécurité IPCablecom sont pris en charge ou non. S'ils sont pris en charge, les paramètres ci-après peuvent être également présents:
- **RTP Ciphersuites** – Liste des algorithmes d'authentification et de chiffrement pris en charge pour le protocole RTP.
- **RTCP Ciphersuites** – Liste des algorithmes d'authentification et de chiffrement pris en charge pour le protocole RTCP.

L'agent d'appel peut alors décider d'utiliser la commande AuditConnection pour obtenir des informations supplémentaires sur les connexions.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

Si aucune information n'a été demandée et si le paramètre EndpointId se rapporte à un EndpointId complètement spécifié et valide, la passerelle renvoie simplement une réponse de succès (code de retour 200 – transaction exécutée normalement).

Il convient de noter que la totalité de l'information renvoyée est simplement un instantané. De nouveaux commentaires reçus, une activité locale, etc. peuvent modifier la majeure partie de ce qui précède. Par exemple, l'état du crochet peut changer avant que l'agent d'appel reçoive les informations ci-dessus.

6.3.8.2 AuditConnection

L'audit des connexions individuelles sur un point d'extrémité peut être réalisé à l'aide de la commande AuditConnection.

```
ReturnCode
[, CallId]
[, NotifiedEntity]
[, LocalConnectionOptions]
[, Mode]
[, RemoteConnectionDescriptor]
[, LocalConnectionDescriptor]
[, ConnectionParameters]
              ← AuditConnection(EndpointId
                        , ConnectionId
                        [, RequestedInfo])
```

Le paramètre **EndpointId** identifie le point d'extrémité qui fait actuellement l'objet de l'audit. Les caractères génériques NE DOIVENT PAS être utilisés. Le paramètre (éventuellement vide) **RequestedInfo** décrit les informations qui sont demandées pour le paramètre **ConnectionId** à l'intérieur du paramètre EndpointId spécifié. Les informations suivantes sur la connexion peuvent être auditées avec cette commande:

```
CallId, NotifiedEntity, LocalConnectionOptions,
Mode, ConnectionParameters, RemoteConnectionDescriptor,
LocalConnectionDescriptor.
```

La réponse, à son tour, comprendra les informations concernant chacun des éléments pour lesquels des informations d'audit ont été demandées:

- **CallId** – Identificateur d'appel pour l'appel auquel cette connexion appartient.
- **NotifiedEntity** – "L'entité avisée" courante pour le point d'extrémité.
- **LocalConnectionOptions** – Options de connexion locale prises en charge pour la connexion.
- **Mode** – Mode de connexion courante.
- **ConnectionParameters** – Paramètres de connexion actuels pour la connexion.
- **LocalConnectionDescriptor** – Descripteur de connexion locale que la passerelle a fourni pour la connexion.
- **RemoteConnectionDescriptor** – Descripteur de connexion distante qui a été fourni à la passerelle pour la connexion.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

Si aucune information n'a été demandée et si le paramètre EndpointId se rapporte à un point d'extrémité valide, la passerelle vérifie simplement que la connexion spécifiée existe et, dans l'affirmative, renvoie une réponse positive (code de retour 200 – transaction exécutée).

6.3.9 Restart in Progress (redémarrage en cours)

La commande RestartInProgress est utilisée par la passerelle pour signaler qu'un point d'extrémité ou un groupe de points d'extrémité est mis hors service ou en cours d'être remis en service.

```
ReturnCode
[, NotifiedEntity]
[, VersionSupported]
              ← RestartInProgress(EndpointId
                                  , RestartMethod
                                  [, RestartDelay])
```

Le paramètre **EndpointId** identifie les points d'extrémité qui sont mis en service ou hors service. La convention de remplacement par le caractère générique "all of" peut être utilisée pour appliquer la commande à un groupe de points d'extrémité (par exemple, à tous les points d'extrémité qui sont rattachés à une interface spécifiée ou même à tous ceux qui sont rattachés à une passerelle donnée). La convention de remplacement par le caractère générique "any of" NE DOIT PAS être utilisée.

Le paramètre RestartMethod spécifie le type de redémarrage:

- une méthode de redémarrage "graceful" indique que le ou les points d'extrémité seront mis hors service après le "délai de redémarrage" spécifié. Les connexions établies n'ont pas encore été touchées mais il convient que l'agent d'appel s'abstienne d'en établir de nouvelles et qu'il tente de supprimer progressivement toutes les connexions existantes;
- une méthode de redémarrage "forced" indique que les points d'extrémité spécifiés sont mis hors service de manière soudaine. Les éventuelles connexions qui ont été établies sont perdues;
- une méthode "restart" indique que le service sera restauré sur les points d'extrémité après le "délai de redémarrage" spécifié. Aucune connexion n'est actuellement établie sur les points d'extrémité;
- une méthode "disconnected" indique que le point d'extrémité a été déconnecté et essaye maintenant d'établir la connectivité. Le "délai de redémarrage" spécifie le nombre de secondes pendant lesquelles le point d'extrémité a été déconnecté. Les connexions établies ne sont pas touchées.

Le paramètre facultatif "restart delay" (*délai de redémarrage*) est exprimé en un nombre de secondes. Si le nombre est absent, il convient que la valeur du délai soit considérée comme "null" (*néant*). Dans le cas de la méthode "graceful", un délai "null" (*néant*) indique qu'il convient que l'agent d'appel attende simplement la fin naturelle des connexions existantes, sans en établir de nouvelles. Le délai de redémarrage est toujours considéré "null" (*néant*) dans le cas de la méthode "forced". Un délai de redémarrage "null" (*néant*) pour la méthode "restart" indique que le service a déjà été restauré. Cela se produit généralement après un démarrage/réamorçage de passerelle. Pour atténuer les effets du changement d'adresse IP d'un client, l'agent d'appel PEUT souhaiter résoudre le nom de domaine du client intégré en interrogeant le DNS, indépendamment de la durée de vie (TTL) d'un enregistrement de ressource courant pour le client intégré qui a été réinitialisé.

Lorsqu'ils sont mis hors service, par exemple, en étant complètement arrêtés, ou mis hors service par un système de gestion de réseau, il CONVIENT que les clients intégrés envoient un message RestartInProgress "graceful" ou "forced" comme une courtoisie envers l'agent d'appel, même si ce dernier ne peut pas compter sur le fait de toujours recevoir de tels messages. Lorsqu'ils sont remis en service, les clients intégrés DOIVENT envoyer un message RestartInProgress "restart" ayant un délai "null" (néant) à leur agent d'appel conformément à la procédure de redémarrage spécifiée au 6.4.3.5 – Les agents d'appel peuvent compter sur la réception de ce message. En outre, les clients intégrés DOIVENT envoyer un message RestartInProgress "disconnected" à leur "entité avisée" actuelle conformément à la procédure "disconnected" spécifiée au 6.4.3.6. Le paramètre "délai de redémarrage" NE DOIT PAS être utilisé avec la méthode de redémarrage "forced".

Le message RestartInProgress sera envoyé à "l'entité avisée" courante pour l'identificateur EndpointId en question. Un agent d'appel par défaut, à savoir "une entité avisée", est censé avoir été fourni pour chaque point d'extrémité de sorte qu'après un réamorçage, cet agent d'appel par défaut soit l'entité avisée pour chaque point d'extrémité. Les clients intégrés DOIVENT tirer pleinement profit du remplacement par caractères génériques afin de minimiser le nombre de messages RestartInProgress qui sont générés lorsque plusieurs points d'extrémité dans une passerelle redémarrent et que les points d'extrémité sont gérés par le même agent d'appel.

ReturnCode est un paramètre retourné par l'agent d'appel. Il indique le résultat de la commande et se compose d'un nombre entier (voir 6.5) éventuellement suivi d'un commentaire.

Un paramètre **NotifiedEntity** peut en outre être renvoyé avec la réponse de l'agent d'appel:

- si la réponse indique un succès (code de retour 200 – Transaction exécutée), la procédure de redémarrage s'est achevée, et le paramètre NotifiedEntity renvoyé est la nouvelle "entité avisée" pour le ou les points d'extrémité;
- si la réponse de l'agent d'appel indique une erreur, la procédure de redémarrage n'est pas encore achevée, et doit donc être lancée à nouveau. Si un paramètre NotifiedEntity a été renvoyé, il spécifie alors la nouvelle "entité avisée" pour le ou les points d'extrémité, qui doit donc être utilisée pour réessayer la procédure de redémarrage.

Enfin, un paramètre **VersionSupported** comportant une liste des versions prises en charge peut être renvoyé si la réponse indique une incompatibilité de versions (code d'erreur 528).

6.4 Etats, reprise sur défaillance et conditions de concurrence

Afin d'implémenter une signalisation d'appel correcte, l'agent d'appel doit garder la trace de l'état du point d'extrémité tandis que la passerelle doit s'assurer que les événements sont correctement notifiés à l'agent d'appel. Des états spéciaux peuvent exister lorsque la passerelle ou l'agent d'appel a redémarré: il se peut qu'il soit nécessaire de rediriger la passerelle vers un nouvel agent d'appel pendant les procédures de "reprise sur défaillance"; d'une manière similaire, il se peut qu'il soit nécessaire que l'agent d'appel prenne une mesure spéciale lorsque la passerelle est mise hors ligne ou réinitialisée.

6.4.1 Récapitulations et points essentiels

Comme il a été mentionné au 6.1.4, les agents d'appel sont identifiés par leur nom de domaine, et chaque point d'extrémité présente une et une seule "entité avisée" qui lui est associée à chaque instant. Le présent paragraphe récapitule et souligne les zones qui revêtent une importance spéciale quant à la fiabilité et à la reprise sur défaillance dans le protocole MGCP:

- un agent d'appel est identifié par son nom de domaine et non par ses adresses réseau, et plusieurs adresses réseau peuvent être associées à un nom de domaine;
- un point d'extrémité a un et un seul agent d'appel qui lui est associé à chaque instant. Cet agent d'appel associé à un point d'extrémité est la valeur actuelle de "l'entité avisée";
- "l'entité avisée" est initialement réglée à une valeur fournie. Lorsque des commandes avec un paramètre NotifiedEntity sont reçues pour le point d'extrémité, y compris des noms de points d'extrémité remplacés par des caractères génériques, cette entité avisée est réglée à la valeur spécifiée. Si " l'entité avisée " d'un point d'extrémité est vide ou n'a pas été définie explicitement[22], elle prend alors la valeur par défaut de l'adresse source de la dernière commande de gestion de connexion ou de la dernière requête de notification reçue concernant le point d'extrémité. Dans ce cas, l'agent d'appel sera donc identifié par son adresse réseau, ce qu'il CONVIENT de n'effectuer qu'à titre exceptionnel;

[22] Cela pourrait se produire par exemple en spécifiant un paramètre NotifiedEntity vide.

- les réponses à une commande sont toujours envoyées à l'adresse source de la commande, indépendamment de l'entité avisée actuelle. Lorsqu'il est nécessaire de superposer un message Notify à la réponse, le datagramme est toujours envoyé à l'adresse source de la nouvelle commande reçue, indépendamment du paramètre NotifiedEntity pour l'une quelconque des commandes;
- lorsque l'entité avisée se rapporte à un nom de domaine qui se décompose en plusieurs adresses IP, les points d'extrémité sont capables de commuter entre chacune de ces adresses mais ils ne peuvent pas eux-même modifier l'entité avisée pour la faire correspondre à un autre nom de domaine. Cependant, un agent d'appel peut leur donner la consigne de commuter en leur fournissant une nouvelle "entité avisée";
- si un agent d'appel n'est plus disponible, les points d'extrémité qu'il gère deviennent finalement "disconnected" (déconnectés). Leur unique façon de se connecter à nouveau est que l'agent d'appel défaillant redevienne disponible ou qu'un autre agent d'appel (de secours) contacte avec une nouvelle "entité avisée" les points d'extrémité touchés;
- lorsqu'un autre agent d'appel (de secours) a pris le contrôle d'un groupe de points d'extrémité, on suppose que l'agent d'appel défaillant communique et se synchronise avec cet agent d'appel de secours afin de rendre à l'agent d'appel de départ le contrôle des points d'extrémité touchés, si on le souhaite. En variante, l'agent d'appel défaillant pourrait simplement devenir l'agent d'appel de secours.

Il convient de noter qu'il n'est pas fourni de mécanisme de résolution des conflits de transfert entre des agents d'appel distincts – on s'appuie strictement sur le fait que les agents d'appel savent ce qu'ils font et communiquent les uns avec les autres (même si l'on peut utiliser AuditEndpoint pour obtenir des informations sur "l'entité avisée" courante).

6.4.2 Retransmission et détection d'associations perdues

Le protocole MGCP est organisé en un jeu de transactions dont chacune est composée d'une commande et d'une réponse. Les messages MGCP peuvent être sujets à des pertes car ils sont transportés sur le protocole UDP. En l'absence d'une réponse à brefs délais (voir 7.5), les commandes sont répétées. Les passerelles DOIVENT mémoriser une liste des réponses qu'elles ont envoyées à de récentes transactions ainsi qu'une liste des transactions qui sont en cours d'exécution. Le terme "récente" est défini ici par la valeur Tt_{hist} qui spécifie le nombre de secondes pendant lesquelles les réponses à d'anciennes transactions doivent être conservées. La valeur par défaut pour Tt_{hist} est de 30 secondes.

Les identificateurs des commandes entrantes sont tout d'abord comparés aux identificateurs de transaction contenus dans les réponses récentes. Si une correspondance est trouvée, la passerelle n'exécute pas la transaction mais répète simplement l'ancienne réponse. S'il n'y a pas de correspondance avec une transaction à laquelle il a été répondu précédemment, l'identificateur de transaction de la commande entrante est comparé à la liste des transactions dont l'exécution n'est pas encore terminée. Si une correspondance est trouvée, la passerelle n'exécute pas la transaction, qui est simplement ignorée – une réponse sera envoyée lorsque l'exécution de la commande sera terminée.

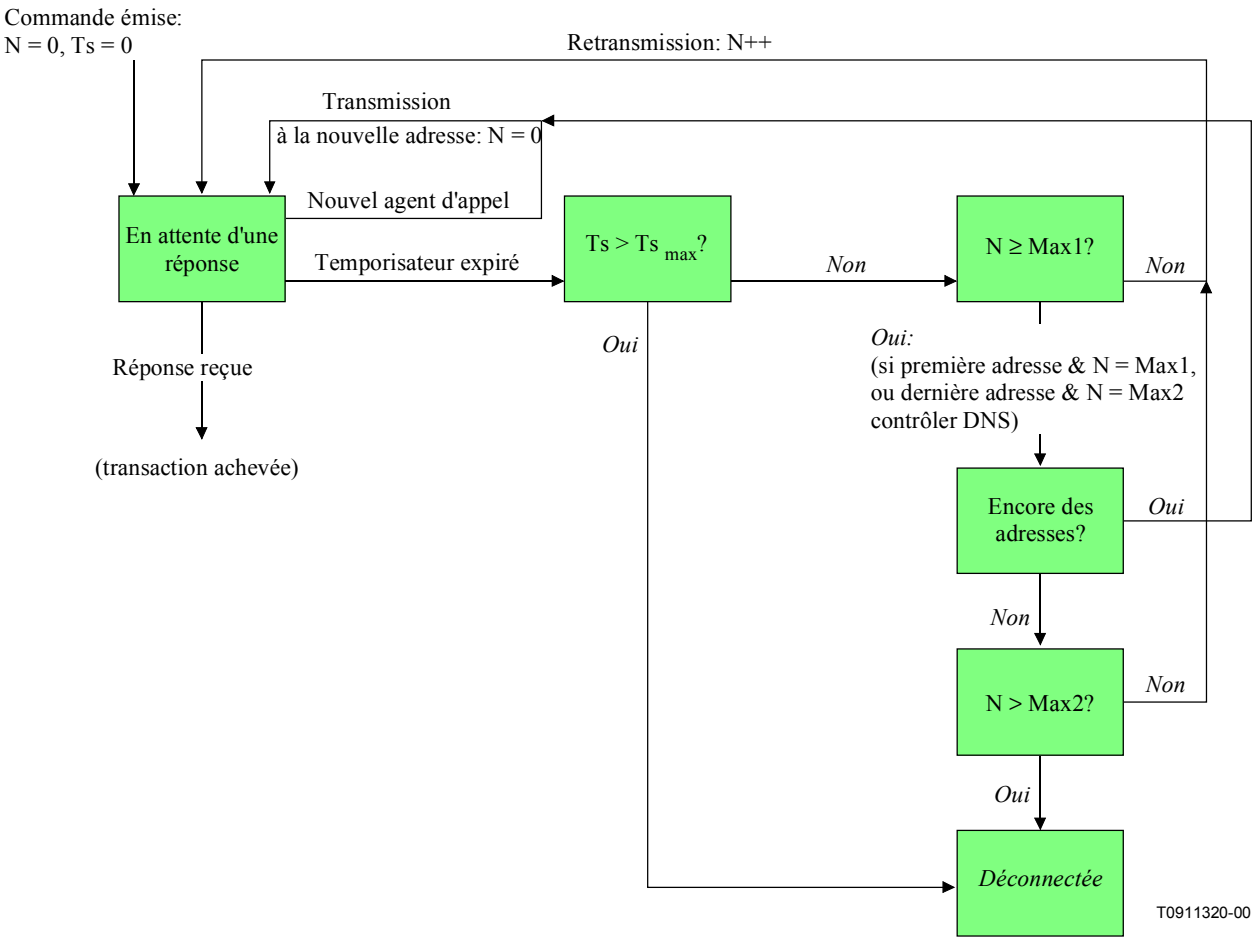
Ce mécanisme de répétition est utilisé pour éviter quatre types d'erreur possibles:

- les erreurs de transmission, lorsque par exemple un paquet est perdu à cause du bruit en ligne ou d'un encombrement dans une file d'attente;
- la défaillance d'un composant, lorsque par exemple une interface pour un agent d'appel n'est plus disponible;
- la défaillance d'un agent d'appel, lorsque par exemple toutes les interfaces pour un agent d'appel ne sont plus disponibles;
- la reprise sur défaillance, lorsqu'un nouvel agent d'appel "prend le contrôle" de manière transparente.

Il convient que les éléments soient en mesure de déduire, de l'historique enregistré, une estimation du taux de pertes de paquets. Dans un système correctement configuré, il convient que ce taux de pertes soit très faible, généralement inférieur à 1% en moyenne. Si un agent d'appel ou une passerelle doit répéter plusieurs fois un message, il est très légitime de supposer que quelque chose d'autre qu'une erreur de transmission est en train de se produire. Par exemple, pour un taux de pertes uniformément réparti de 1%, la probabilité que 5 tentatives de transmission consécutives échouent est de 1 sur 100 milliards, soit un événement qui devrait se produire moins souvent qu'une fois tous les 10 jours pour un agent d'appel traitant 1 000 transactions par seconde. (En fait, le nombre de répétitions qui est considéré comme excessif devrait être fonction du taux de pertes de paquets observé.) Lorsque les erreurs ne sont pas uniformément réparties, la probabilité de défaillance résultante peut devenir quelque peu plus élevée. Il convient de remarquer que le "seuil de suspicion", que l'on appellera "Max1", est normalement inférieur au "seuil de déconnexion", que l'on appellera "Max2", qui devrait être fixé à une valeur beaucoup plus grande.

Un algorithme classique de retransmission comptera simplement le nombre de répétitions successives et conclura que l'association est rompue après la réémission du même paquet un nombre de fois excessif (normalement entre 7 et 11 fois). Afin de tenir compte de la possibilité d'une "reprise sur défaillance" non détectée ou en cours, on modifie l'algorithme classique, comme suit:

- la passerelle DOIT vérifier la présence d'un nouvel agent d'appel. Cette présence peut être signalée par:
  - la réception d'une commande où NotifiedEntity pointe sur un nouvel agent d'appel;
  - la réception d'une réponse de redirection qui pointe vers un nouvel agent d'appel;
- si un nouvel agent d'appel a été détecté, la passerelle DOIT diriger les retransmissions de toutes les commandes en attente pour le ou les points d'extrémité qui ont été redirigées vers ce nouvel agent d'appel. Les réponses aux nouvelles ou anciennes commandes continueront d'être envoyées à l'adresse source de chaque commande;
- avant toute retransmission, on vérifie que le temps écoulé depuis l'envoi du datagramme initial n'est pas supérieur à Ts_{max}. Si un temps supérieur à Ts_{max} s'est écoulé, le point d'extrémité se déconnecte;
- si le nombre de retransmissions vers cet agent d'appel est égal à "Max1", la passerelle PEUT interroger activement le serveur de noms afin de détecter la possible modification des interfaces d'agent d'appel, indépendamment de la durée de vie (TTL, *time to live*) associée à l'enregistrement des DNS;
- la passerelle peut avoir pris connaissance de plusieurs adresses IP pour l'agent d'appel. Si le nombre de retransmissions pour cette adresse IP est supérieur à "Max1" et inférieur "Max2" alors qu'il y a plus d'adresses IP qui n'ont pas été essayées, la passerelle DOIT diriger les retransmissions vers les autres adresses restantes dans sa liste locale;
- s'il n'y a plus d'interfaces à essayer alors que le nombre de retransmissions est Max2, il CONVIENT que la passerelle contacte encore une fois le DNS pour voir si d'autres interfaces sont devenues disponibles. Dans le cas contraire, le ou les points d'extrémité gérés par cet agent d'appel sont maintenant déconnectés. Lorsqu'un point d'extrémité devient déconnecté, il DOIT lancer la procédure "disconnected" comme spécifié au 6.4.3.6.



Afin de s'adapter automatiquement à la charge du réseau, le protocole MGCP spécifie des temporisateurs qui augmentent de manière exponentielle (voir 7.5.2). Si la temporisation initiale est réglée à 200 millisecondes, la perte d'une cinquième retransmission sera détectée au bout d'environ 6 secondes. C'est sans doute un temps d'attente acceptable pour détecter une reprise sur défaillance. Il convient que les retransmissions se poursuivent après ce délai, non seulement pour résoudre un éventuel problème de connexité transitoire mais aussi pour accorder un peu plus de temps pour l'exécution d'une reprise sur défaillance – une attente totale de 30 secondes est sans doute acceptable.

Il convient de remarquer qu'il existe une relation intime entre Ts_{max}, Tt_{hist}, et le temps maximal de transit Tp_{max}. En particulier, la relation ci-après DOIT être satisfaite pour empêcher que les commandes retransmises ne soient exécutées plus d'une fois:

$$Tt_{hist} \geq Ts_{max} + Tp_{max}$$

La valeur par défaut de Ts_{max} est de 20 secondes. Donc, si on suppose que le retard maximal de propagation est de 10 secondes, les réponses aux anciennes transactions doivent être conservées pendant une période de 30 secondes au moins. L'importance du fait que l'expéditeur et le récepteur s'accordent sur ces valeurs ne peut pas être exagérée.

La valeur par défaut de Max1 est de 5 retransmissions tandis que celle de Max2 est de 7 retransmissions. Ces valeurs peuvent être modifiées par le processus de mise en service.

En outre, le processus de mise en service DOIT pouvoir désactiver l'une ou les deux interrogations de DNS sur Max1 et Max2.

6.4.3 Conditions de concurrence

Dans le présent paragraphe, on trouvera une description illustrant comment le protocole MGCP traite des conditions de concurrence.

En tout premier lieu, le protocole MGCP traite des conditions de concurrence par le biais du concept d'une "liste de quarantaine" ("quarantine list") qui place des événements en quarantaine et par le biais d'une détection explicite de désynchronisation, par exemple, dans le cas d'un état de crochet discordant du fait d'une collision d'appel pour un point d'extrémité.

Deuxièmement, le protocole MGCP ne suppose pas que le mécanisme de transport va conserver l'ordre des commandes et des réponses. Il peut en résulter des conditions de concurrence qui peuvent être évitées par le biais d'un comportement correct de l'agent d'appel s'appuyant sur un ordonnancement correct des commandes.

Enfin, dans un certain nombre de cas, de nombreuses passerelles peuvent décider de redémarrer l'exploitation simultanément. Ce qui peut se produire par exemple si une zone perd son alimentation électrique ou sa capacité de transmission pendant un séisme ou une tempête de pluie verglaçante. Lorsque l'alimentation électrique et la capacité de transmission sont rétablies, de nombreuses passerelles peuvent décider d'envoyer simultanément des commandes RestartInProgress; cette situation pourrait être à l'origine d'un fonctionnement très instable si elle n'est pas contrôlée avec beaucoup de soin.

6.4.3.1 Liste de quarantaine (Quarantine list)

Les passerelles commandées par le protocole MGCP reçoivent des demandes de notification qui les enjoignent d'effectuer une surveillance afin de détecter une liste d'événements. Les éléments de protocole qui déterminent la gestion de ces événements sont les listes "Requested Events" (*événements demandés*), "Digit Map" (*script de numérotation*) et "Detect Events" (*détecter les événements*).

Lorsque le point d'extrémité est initialisé, la liste des événements est constituée uniquement des événements persistants pour ce point d'extrémité et le script de numérotation est vide. Après la réception d'une commande, la passerelle commence à observer le point d'extrémité pour détecter les occurrences des événements mentionnés dans la liste, y compris les événements persistants.

Les évènements sont examinés au fur et à mesure de leur apparition. L'action qui suit est déterminée par le paramètre "action" associé à l'événement figurant dans la liste des événements demandés, et aussi par le script de numérotation. Les événements qui sont définis comme étant "accumulate" (*cumulé*) ou "accumulate according to digit map" (*accumulation selon le script de numérotation*) sont accumulés dans une liste d'événements observés. Les événements qui sont repérés comme "cumulés selon le script de numérotation" seront en plus accumulés dans la "chaîne de numérotation actuelle". Ce processus se poursuit jusqu'à ce qu'on rencontre un événement qui déclenche une commande Notify qui sera envoyée à "l'entité avisée".

A ce stade, la passerelle transmet la commande Notify et place le point d'extrémité dans un "état de notification". Tant que le point d'extrémité est dans cet état de notification, les événements qui sont détectés sur ce point d'extrémité sont mémorisés dans un tampon de "quarantaine" en vue d'un traitement ultérieur. Dans un sens, les événements sont "mis en quarantaine". Les événements détectés sont ceux qui sont spécifiés par l'union du paramètre RequestedEvents et du paramètre DetectEvents reçu le plus récemment, ou dans le cas où aucun paramètre n'aurait été reçu, ceux auxquels il est fait référence dans le paramètre RequestedEvents. Les événements persistants sont également détectés.

Le point d'extrémité sort de "l'état de notification" à la réception de la réponse à la commande Notify[23]. Il est possible de retransmettre la commande Notify dans "l'état de notification", comme il est spécifié au 6.4.2.

Lorsque le point d'extrémité sort de "l'état de notification", il réinitialise la liste des événements observés et sa propre "chaîne de numérotation actuelle" à la valeur "null" (*néant*).

Le profil NCS exige l'usage du mode "verrouillé", ce qui implique que la passerelle DOIT recevoir une nouvelle commande NotificationRequest après avoir envoyé une commande Notify. Dans cette attente, le point d'extrémité est dans un "état verrouillé", et les événements qui se produisent et doivent être détectés sont simplement mémorisés dans le tampon de quarantaine. Les événements devant être mis en quarantaine sont les mêmes que ceux dans "l'état de notification". Une fois que la nouvelle commande NotificationRequest est reçue et exécutée avec succès, le point d'extrémité quitte "l'état verrouillé".

Une passerelle peut recevoir à tout moment une nouvelle commande NotificationRequest pour le point d'extrémité, ce qui aura également pour effet de sortir le point d'extrémité de "l'état de notification" en supposant que la commande NotificationRequest s'exécute avec succès.

Lorsqu'une nouvelle commande NotificationRequest est reçue dans "l'état de notification", la passerelle doit s'assurer que la commande Notify en attente est bien reçue par l'agent d'appel avant une réponse positive à cette nouvelle commande NotificationRequest. Elle le fait en utilisant la fonctionnalité "superposition" du protocole et en plaçant les messages (commandes et réponses) à envoyer dans l'ordre, le message le plus ancien étant le premier. Les messages sont ensuite envoyés dans un seul paquet à la source de la nouvelle commande NotificationRequest, indépendamment de la source et de l'entité avisée pour l'ancienne et la nouvelle commande. Les étapes impliquées sont les suivantes:

1) la passerelle construit un message qui transporte dans un seul paquet une répétition de l'ancienne commande Notify en attente et la réponse à la nouvelle commande NotificationRequest;

2) puis le point d'extrémité est sorti de "l'état de notification" sans attendre la réponse à la commande Notify;

3) une copie de la commande Notify en attente est conservée en attendant la réception d'une réponse. Si une temporisation se produit, la commande Notify sera répétée dans un paquet qui transportera également une répétition de la commande NotificationRequest.

   - Si le paquet transportant la réponse à la commande NotificationRequest est perdu, l'agent d'appel transmet à nouveau la commande NotificationRequest. La passerelle répond à cette répétition en retransmettant dans un seul paquet la commande Notify en instance et la réponse à la commande NotificationRequest – Ce datagramme sera envoyé à la source de la commande NotificationRequest.
   - Si la passerelle doit transmettre une nouvelle commande Notify avant qu'une réponse à la commande Notify précédente soit reçue, elle construit un paquet qui superpose une répétition de l'ancienne commande, une répétition de la dernière commande NotificationRequest et la nouvelle commande Notify – Ce datagramme sera envoyé à "l'entité avisée" actuelle.

Après réception d'une commande NotificationRequest, les listes "requested events" (*événements demandés*) et "script de numérotation" (si une nouvelle liste a été fournie) sont remplacées par les paramètres nouvellement reçus tandis que la liste des "événements observés" et la "chaîne de numérotation actuelle" sont réinitialisées à une valeur "null" (néant). Le comportement subséquent

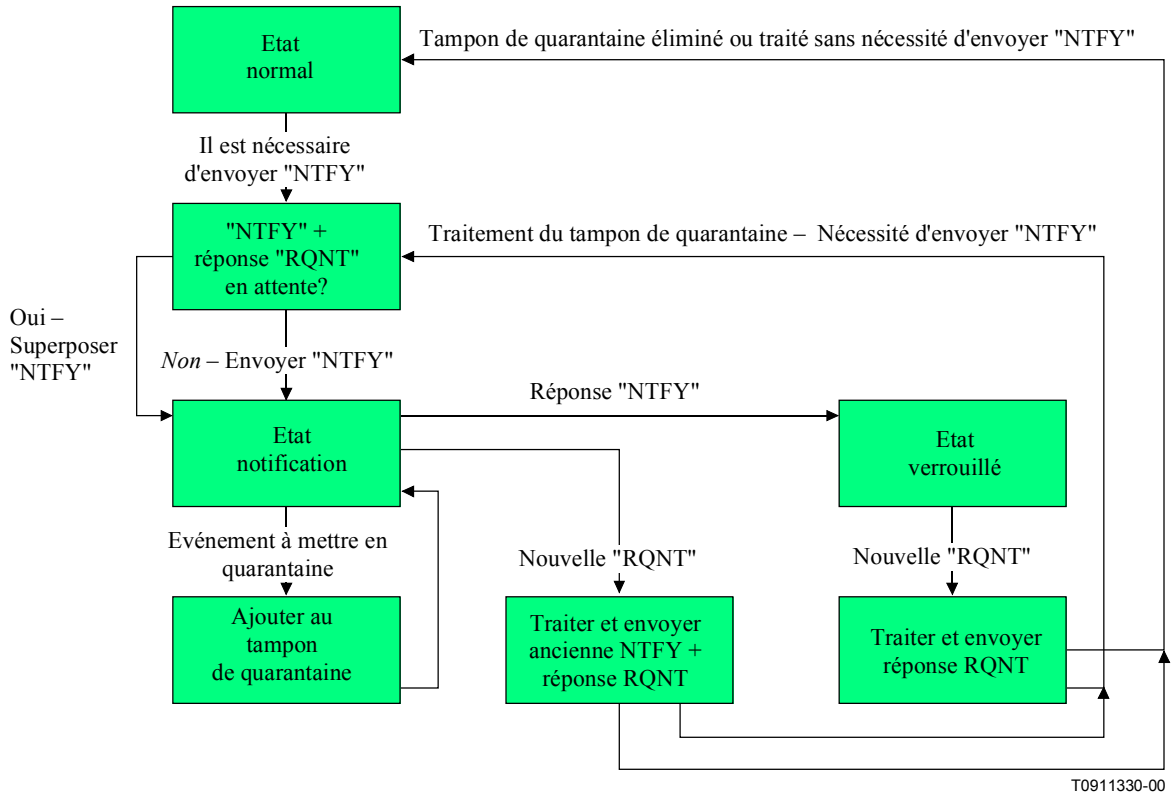
---

[23] Il convient de remarquer que l'action Notify ne peut pas être combinée avec une commande NotificationRequest intégrée.

est conditionné par la valeur du paramètre QuarantineHandling. Il se peut que ce paramètre spécifie que des événements mis en quarantaine doivent être éliminés, auquel cas ils le sont tous. Si le paramètre spécifie qu'il convient de traiter les événements mis en quarantaine, la passerelle commencera à traiter la liste des événements mis en quarantaine, en utilisant la liste des "événements demandés" nouvellement reçue et le "script de numérotation" s'il est fourni. Pendant le traitement de ces événements, la passerelle peut rencontrer un événement qui déclenche une commande Notify à envoyer. Si tel est le cas, la passerelle transmettra immédiatement une commande Notify qui fera rapport de tous les événements qui ont été accumulés dans la liste des "événements observés" jusqu'à l'événement déclenchant compris, laissant dans le tampon de quarantaine les événements non traités. Puis le point d'extrémité se met à nouveau dans "l'état de notification".

La procédure précédente s'applique à toutes les formes de demandes de notification, indépendamment du fait qu'elles font partie d'une commande de gestion de connexion ou sont fournies en tant que commande NotificationRequest. Les commandes de gestion qui ne comportent pas une demande de notification n'ont pas d'incidence sur la procédure précédente et réciproquement.

Le schéma ci-après illustre la procédure spécifiée ci-dessus, avec l'hypothèse que toutes les transactions ont été exécutées avec succès:



Il CONVIENT que les agents d'appel fournissent dans le même datagramme la réponse à un message Notify réussi et la nouvelle commande NotificationRequest, en se servant du mécanisme de superposition[24].

[24] Il convient que les fournisseurs qui choisissent de ne pas suivre la présente Recommandation examinent avec beaucoup de soin des scénarios de défaillance d'agent d'appel.

6.4.3.2 Détection explicite

Un élément clé de l'état d'un grand nombre de points d'extrémité est la position du crochet. Des conditions de concurrence et une discordance d'états peuvent apparaître, par exemple lorsque l'usager décide de raccrocher alors que l'agent d'appel est engagé dans le processus de demander à la passerelle de rechercher des événements de décrochage et peut-être de générer un signal de sonnerie (l'état de "collision d'appel" bien connu dans les capacités téléphoniques).

Pour prévenir cette condition de concurrence, la passerelle DOIT vérifier l'état du point d'extrémité avant de répondre à la commande NotificationRequest. D'une manière spécifique, elle DOIT renvoyer une erreur:

1) si l'on demande à la passerelle de notifier une transition "décrochage" alors que le combiné est déjà décroché (code d'erreur 401 – Combiné décroché);
2) si l'on demande à la passerelle de notifier un état de "raccrochage" ou de "rappel d'enregistreur" alors que le combiné est déjà raccroché (code d'erreur 402 – Combiné raccroché).

De plus, des définitions individuelles de signaux peuvent spécifier qu'un signal fonctionnera seulement dans certaines conditions, par exemple, la sonnerie peut seulement être possible si le combiné est déjà décroché. Si de telles conditions préalables existent pour un signal donné, la passerelle DOIT renvoyer l'erreur spécifiée dans la définition du signal dans le cas où la condition préalable ne serait pas satisfaite.

Il convient de remarquer que la vérification d'état est effectuée au moment de la réception de la demande de notification, alors que l'événement effectif à l'origine de l'état actuel peut avoir été rapporté ou ignoré plus tôt ou il peut être actuellement en quarantaine.

Les autres variables d'état de la passerelle, telles que la liste d'événements demandés ou la liste de signaux demandés, sont entièrement remplacées après chaque commande NotificationRequest réussie, ce qui empêche toute discordance à long terme entre l'agent d'appel et la passerelle.

Lorsqu'une commande NotificationRequest n'a pas réussi, qu'elle soit incluse dans une commande de gestion de connexion ou non, la passerelle continuera simplement comme si la commande n'avait jamais été reçue, bien qu'une erreur soit renvoyée. Comme toutes les autres transactions, la commande NotificationRequest DOIT fonctionner comme une transaction atomique; ainsi les éventuelles modifications initiées en tant que résultat de la commande DOIVENT être remises.

Une autre condition de concurrence peut se produire lorsqu'une commande Notify est émise très peu de temps avant la réception d'une commande NotificationRequest par la passerelle. Le paramètre RequestIdentifier est utilisé pour corréler des commandes Notify avec les commandes NotificationRequest, permettant ainsi à l'agent d'appel de déterminer si la commande Notify a été générée avant ou après la réception de la nouvelle commande NotificationRequest par la passerelle.

6.4.3.3 Sémantique transactionnelle

Au fur et à mesure que les temps d'achèvement de transaction potentielle augmentent, par exemple du fait de réservations de ressources externes, une définition soignée de la sémantique transactionnelle devient de plus en plus importante. En particulier, la question des conditions de concurrence, spécifiquement dans la manière dont elle se rapporte à l'état du crochet, doit être définie avec beaucoup de soin.

Un point important à considérer est qu'en fait l'état du crochet est susceptible de se modifier entre le moment où la transaction est déclenchée et le moment où elle s'achève. D'une manière plus générale, on peut dire qu'un achèvement réussi d'une transaction dépend d'une ou plusieurs conditions préalables dont une ou plusieurs sont susceptibles de dynamiquement changer pendant l'exécution de la transaction.

La sémantique la plus simple à cet effet est de simplement exiger que toutes les conditions préalables DOIVENT être satisfaites, à partir du moment où la transaction est lancée jusqu'au moment où elle s'achève. Ainsi, si l'une quelconque des conditions préalables se modifie pendant l'exécution de la transaction, la transaction DOIT échouer. En outre, dès que la transaction est déclenchée, tous les nouveaux événements sont mis en quarantaine. Lorsque le résultat de la transaction est connu, tous les événements mis en quarantaine sont alors traités.

A titre d'exemple, considérons une transaction qui comprend une demande pour un événement "décrochage". Lorsque la transaction est déclenchée, le combiné est "raccroché" et la condition préalable est donc satisfaite. Si l'état du crochet se modifie pour passer à l'état "décrochage" avant que la transaction ne s'achève, la condition préalable n'est plus satisfaite et, par conséquent, la transaction échoue immédiatement. L'événement "décrochage" sera maintenant mémorisé dans le tampon de "quarantaine" qui est alors traité.

6.4.3.4 Ordonnancement des commandes et traitement du désordre

Le protocole MGCP ne demande pas que le protocole de transport sous-jacent garantisse la mise en séquence des commandes envoyées à une passerelle ou à un point d'extrémité. Cette propriété a tendance à maximaliser l'actualité des actions, mais elle présente quelques inconvénients. Par exemple:

- il se peut que les commandes Notify soient retardées et arrivent à l'agent d'appel après la transmission d'une nouvelle commande Notification Request;
- si une nouvelle commande NotificationRequest est transmise avant qu'une précédente ait reçu une réponse, il n'est pas garanti que la commande précédente ne sera pas reçue en seconde position.

Les agents d'appel et les passerelles qui souhaitent garantir un fonctionnement cohérent des points d'extrémité peuvent utiliser les règles spécifiées:

1) lorsqu'une passerelle gère plusieurs points d'extrémité, les commandes relatives à ces différents points d'extrémité peuvent être envoyées en parallèle, par exemple selon un modèle où chaque point d'extrémité est commandé par son propre processus ou par son propre fil d'exécution individuelle;
2) lorsque plusieurs connexions sont créées sur le même point d'extrémité, les commandes relatives à différentes connexions peuvent être envoyées en parallèle;
3) sur une connexion donnée, il convient qu'il n'y ait qu'une seule commande en instance (create ou modify). Une commande DeleteConnection peut cependant être émise à tout moment. En conséquence, une passerelle peut parfois recevoir une commande ModifyConnection qui s'applique à une terminaison déjà supprimée. De telles commandes DOIVENT être ignorées, et une erreur renvoyée (code d'erreur 515 – connection-id incorrect);
4) sur un point d'extrémité donné, il convient qu'il n'y ait normalement qu'une seule commande NotificationRequest en instance à un moment donné. Le paramètre RequestId est utilisé pour corréler les commandes Notify avec la commande de déclenchement NotificationRequest;
5) dans un certain nombre de cas, une commande DeleteConnection implicitement ou explicitement remplacée par une structure générique, qui s'applique à un groupe de points d'extrémité, peut se placer devant une commande CreateConnection en instance. Il convient que l'agent d'appel supprime individuellement toutes les connexions dont l'exécution était en instance au moment de la commande DeleteConnection globale. De plus, il convient de ne pas envoyer les nouvelles commandes CreateConnection pour des points d'extrémité nommés par la convention de remplacement par des caractères génériques, jusqu'à la réception d'une réponse à la commande DeleteConnection remplacée par un caractère générique;

6) lorsque les commandes sont intégrées les unes dans les autres, les prescriptions relatives à l'ordonnancement DOIVENT être suivies pour toutes les commandes. Par exemple, une commande CreateConnection comprenant une demande de notification doit être conforme aux prescriptions portant sur l'ordonnancement pour les commandes CreateConnection et NotificationRequest simultanément;

7) les commandes AuditEndpoint et AuditConnection ne sont l'objet d'aucun ordonnancement;

8) la commande RestartInProgress doit toujours être la première commande envoyée par un point d'extrémité, comme il est défini dans la procédure de redémarrage (voir 6.4.3.5). Toute autre commande ou réponse doit être acheminée après cette commande RestartInProgress (la superposition est autorisée);

9) lorsque plusieurs messages sont superposés en un seul paquet, ils sont toujours traités dans l'ordre.

Celles des règles ci-dessus qui spécifient le comportement des passerelles DOIVENT être respectées par les clients intégrés, mais un client intégré NE DOIT PAS poser d'hypothèses concernant la question de savoir si les agents d'appel suivent ou non les règles. Par conséquent, les passerelles DOIVENT répondre aux commandes, qu'elles soient ou non conformes aux règles ci-dessus.

6.4.3.5 Combattre l'avalanche de redémarrage

Supposons qu'un grand nombre de passerelles soit mis sous tension simultanément. Si elles doivent toutes lancer une transaction RestartInProgress, l'agent d'appel sera très probablement submergé, ce qui se traduira par des pertes de messages et par des encombrements dans le réseau au cours de la période critique du rétablissement des services. Afin d'éviter de telles avalanches, le comportement suivant DOIT être adopté:

1) lorsqu'une passerelle est mise sous tension, elle déclenche un temporisateur de redémarrage à une valeur aléatoire, uniformément réparti entre 0 et un temps d'attente maximale (MWD, *maximum waiting delay*) fourni, par exemple, 360 secondes (voir ci-dessous). On DOIT veiller à éviter le synchronisme de la production des nombres aléatoires entre plusieurs passerelles utilisant le même algorithme;

2) la passerelle doit ensuite attendre l'expiration de cette temporisation, la réception d'une commande de l'agent d'appel ou la détection d'une activité d'utilisateur local, comme par exemple une transition de décrochage sur une passerelle résidentielle. Une condition de décrochage préexistante aboutit à la création d'un événement de décrochage;

3) lorsque le temporisateur de redémarrage expire, quand une commande est reçue ou quand une activité ou une condition de décrochage préexistante est détectée, la passerelle déclenche la procédure de redémarrage.

La procédure de redémarrage établit simplement que le point d'extrémité DOIT envoyer une commande RestartInProgress à l'agent d'appel l'informant sur le redémarrage et, en outre, garantir que le premier message (commande ou réponse) que l'agent d'appel voit depuis ce point d'extrémité DOIT être la présente commande RestartInProgress. A cet effet, le point d'extrémité DOIT tirer pleinement profit de la superposition. Par exemple, si une activité de décrochage a lieu avant l'expiration du temporisateur de redémarrage, il sera créé un paquet contenant la commande RestartInProgress et comportant une commande Notify superposé pour l'événement de décrochage. Dans le cas où le temporisateur de redémarrage expirerait sans une quelconque autre activité, la passerelle envoie simplement un message RestartInProgress.

Si la passerelle entrait dans un état "disconnected" pendant l'exécution de la procédure de redémarrage, la procédure "disconnected" spécifiée au 6.4.3.6 DOIT être exécutée, sauf qu'un message "restart" (redémarrage) est envoyé à la place d'un message "disconnected" (déconnecté) pendant l'exécution de la procédure.

Chaque point d'extrémité dans la passerelle est censé avoir un agent d'appel (que l'on peut mettre à disposition), à savoir une "entité avisée", vers lequel diriger le message de redémarrage initial. Lorsque l'ensemble de points d'extrémité présents dans une passerelle est géré par deux agents d'appel ou plus, la procédure ci-dessus doit être exécutée pour chaque ensemble de points d'extrémité gérés par un agent d'appel donné. La passerelle DOIT tirer pleinement profit du remplacement par caractères génériques afin de minimiser le nombre de messages RestartInProgress qui sont générés lorsque plusieurs points d'extrémité dans une passerelle redémarrent et que les points d'extrémité sont gérés par le même agent d'appel.

La valeur du temps MWD est un paramètre de configuration qui dépend du type de passerelle. Le raisonnement suivant peut être utilisé afin de déterminer la valeur de ce temps dans les passerelles résidentielles.

Les agents d'appel sont normalement dimensionnés de façon à absorber la charge de trafic à l'heure de pointe, pendant laquelle 10% des lignes en moyenne seront occupées à faire aboutir des communications dont la durée moyenne est normalement de 3 minutes. Le traitement d'un appel implique typiquement 5 ou 6 transactions entre chaque point d'extrémité et l'agent d'appel. Ce simple calcul montre que l'agent d'appel est censé traiter 5 ou 6 transactions pour chaque point d'extrémité, toutes les 30 minutes en moyenne. En d'autres termes, 1 transaction environ sera traitée toutes les 5 ou 6 minutes en moyenne dans chaque point d'extrémité, ce qui permet d'estimer qu'une valeur raisonnable du temps MWD pour une passerelle résidentielle sera de 10 à 12 minutes. En l'absence de configuration explicite, les clients intégrés DOIVENT adopter une valeur par défaut de 600 secondes pour le temps MWD.

6.4.3.6 Points d'extrémité déconnectés

Outre la procédure de redémarrage, les clients intégrés ont également une procédure "disconnected" (déconnectée), qui est déclenchée lorsqu'un point d'extrémité devient "déconnecté" comme il est décrit au 6.4.2. Il convient de noter ici que les points d'extrémité peuvent devenir déconnectés seulement lorsqu'ils essayent de communiquer avec l'agent d'appel. Les étapes ci-après sont suivies par un point d'extrémité qui devient "déconnecté":

1) un temporisateur "déconnecté" est initialisé à une valeur aléatoire, uniformément réparti entre 0 et un délai d'attente initial "déconnecté" (Td_{init}) fourni, de 15 secondes par exemple. On DOIT veiller à éviter le synchronisme de production de nombres aléatoires entre plusieurs passerelles et points d'extrémité utilisant le même algorithme;
2) la passerelle doit ensuite attendre l'expiration de cette temporisation, la réception d'une commande de l'agent d'appel ou la détection d'une activité d'utilisateur local pour le point d'extrémité, comme par exemple une transition de décrochage;
3) lorsque le temporisateur "déconnecté" expire, quand une commande est reçue ou quand une activité d'utilisateur local est détectée, la passerelle déclenche la procédure "disconnected" (déconnectée) pour le point d'extrémité. Dans le cas de l'activité d'utilisateur local, un temps d'attente minimale "déconnecté" (Td_{min}) fourni doit avoir expiré depuis le moment où la passerelle est devenue déconnectée ou depuis la dernière fois qu'elle a déclenché la procédure "Disconnected" (déconnectée) afin de limiter la vitesse d'exécution de la procédure;
4) si la procédure "disconnected" (déconnectée) laisse encore déconnecté le point d'extrémité, le temporisateur "déconnecté" est doublé, susceptible de subir un temps d'attente maximale "déconnecté" (Td_{max}) fourni, de 600 secondes par exemple, et la passerelle continue à nouveau avec l'étape 2).

La procédure "disconnected" (déconnectée) est similaire à la procédure de redémarrage en ce qu'elle établit simplement que le point d'extrémité DOIT envoyer une commande RestartInProgress à l'agent d'appel pour lui indiquer si un point d'extrémité a été déconnecté et, en outre, garantir que le premier message (commande ou réponse) que l'agent d'appel voit maintenant depuis ce point d'extrémité DOIT être une commande RestartInProgress. A cet effet, le point d'extrémité DOIT tirer pleinement profit de la superposition. L'agent d'appel peut alors décider par exemple d'auditer le point d'extrémité ou simplement de supprimer toutes les connexions pour le point d'extrémité.

A dessein, la présente Recommandation ne spécifie pas de comportement supplémentaire pour un point d'extrémité déconnecté. Des fournisseurs PEUVENT par exemple choisir de fournir un silence, de jouer une tonalité "tous circuits occupés" ou même de permettre qu'un fichier wav téléchargé soit joué sur des points d'extrémité touchés.

La valeur par défaut de Td_{init} est de 15 secondes, celle de Td_{min} de 15 secondes et celle de Td_{max} de 600 secondes.

6.5 Codes de retour et codes d'erreur

Toutes les commandes du protocole MGCP reçoivent une réponse. La réponse contient un code de retour qui indique l'état de la commande. Le code de retour est un nombre entier, pour lequel trois plages de valeurs ont été définies:

- la valeur 000 indique un acquittement de réponse[25];
- les valeurs comprises entre 100 et 199 indiquent une réponse provisoire;
- les valeurs comprises entre 200 et 299 indiquent une fin réussie;
- les valeurs comprises entre 400 et 499 indiquent une erreur transitoire;
- les valeurs comprises entre 500 et 599 indiquent une erreur permanente.

Les valeurs qui ont été définies sont énumérées dans le tableau ci-après:

Code	Signification
000	Acquittement de réponse.
100	La transaction est actuellement en cours d'exécution. Un message d'achèvement effectif suivra.
200	La transaction demandée a été exécutée normalement.
250	La ou les connexions ont été supprimées.
400	La transaction n'a pas pu être exécutée, à cause d'une erreur transitoire.
401	Le combiné est déjà décroché.
402	Le combiné est déjà raccroché.
500	La transaction n'a pas pu être exécutée car le point d'extrémité est inconnu.
501	La transaction n'a pas pu être exécutée car le point d'extrémité n'est pas prêt.
502	La transaction n'a pas pu être exécutée car le point d'extrémité ne dispose pas de ressources suffisantes.
510	La transaction n'a pas pu être exécutée car une erreur de protocole a été détectée.
511	La transaction n'a pas pu être exécutée car la commande contenait une extension non reconnue.
512	La transaction n'a pas pu être exécutée car la passerelle n'est pas équipée pour détecter un des événements demandés.

[25] L'acquittement de réponse est utilisé pour les réponses provisoires (voir 7.8).

Code	Signification
513	La transaction n'a pas pu être exécutée car la passerelle n'est pas équipée pour générer un des signaux demandés.
514	La transaction n'a pas pu être exécutée car la passerelle ne peut pas envoyer l'annonce spécifiée.
515	La transaction fait référence à un paramètre connection-id incorrect (qui a peut-être été déjà supprimé).
516	La transaction fait référence à un paramètre call-id inconnu.
517	Mode non pris en charge ou valide.
518	Paquet non pris en charge ou inconnu.
519	Le point d'extrémité n'a pas de script de numérotation.
520	La transaction n'a pas pu être exécutée car le point d'extrémité "redémarre".
521	Point d'extrémité redirigé vers un autre agent d'appel.
522	Cet événement ou ce signal n'existe pas.
523	Action inconnue ou combinaison illégale d'actions.
524	Incohérence interne dans le paramètre LocalConnectionOptions.
525	Extension inconnue dans le paramètre LocalConnectionOptions.
526	Largeur de bande insuffisante.
527	Paramètre RemoteConnectionDescriptor manquant.
528	Version de protocole incompatible.
529	Panne matérielle interne.
532	Valeur(s) non prise(s) en charge dans le paramètre LocalConnectionOptions.
533	Réponse trop grande.

6.6 Codes de cause

Les codes de cause sont utilisés par la passerelle pendant la suppression d'une connexion pour indiquer à l'agent d'appel la cause de la suppression de la connexion. Le code de cause est un nombre entier, et les valeurs suivantes ont été définies:

Code	Signification
900	Mauvais fonctionnement de point d'extrémité.
901	Point d'extrémité mis hors service.
902	Perte de connectivité de la couche inférieure (par exemple, synchronisation descendante).
903	La réservation de ressources de QS a été perdue.

7 Protocole de commande de passerelle de média

Le protocole MGCP implémente l'interface de commande de la passerelle de média en tant que jeu de transactions. Les transactions sont composées d'une commande et d'une réponse obligatoire. Il y a huit types de commandes:

- CreateConnection;
- ModifyConnection;
- DeleteConnection;
- NotificationRequest;

- Notify;
- AuditEndpoint;
- AuditConnection;
- RestartInProgress.

Les quatre premières commandes sont envoyées à la passerelle par l'agent d'appel. La commande Notify est envoyée à l'agent d'appel par la passerelle. La passerelle peut également envoyer une commande DeleteConnection comme il est défini au 6.3.6. L'agent d'appel peut envoyer l'une des commandes Audit à la passerelle et, enfin, la passerelle peut envoyer une commande RestartInProgress à l'agent d'appel.

7.1 Description générale

Toutes les commandes sont composées d'un en-tête Commande qui, pour un certain nombre de commandes, peut être suivi d'une description de session.

Toutes les réponses sont composées d'un en-tête Réponse qui, pour un certain nombre de commandes, peut être suivi d'une description de session.

En-têtes et descriptions de session sont codés comme un jeu de lignes de texte, séparées par un retour de chariot et un caractère interligne (ou, en option, par un seul caractère interligne). Les en-têtes sont séparés de la description de session par une ligne vide.

Le protocole MGCP utilise un identificateur de transaction ayant une valeur comprise entre 1 et 999999999 pour corréler les commandes et les réponses. L'identificateur de transaction est codé comme une composante de l'en-tête commande et est répété comme une composante de l'en-tête réponse.

7.2 En-tête Commande (*command header*)

L'en-tête Commande est composé:

- d'une ligne de commande qui identifie l'action ou le verbe demandé, l'identificateur de transaction, le point d'extrémité vers lequel l'action est demandée, et la version du protocole MGCP;
- d'un jeu de lignes de paramètres composées d'un nom de paramètre suivi d'une valeur de paramètre.

Sauf notification ou ordre contraire émanant d'autres normes référencées, chaque composante de l'en-tête commande est insensible à la casse. Il en est ainsi pour les verbes et pour les paramètres et valeurs; toutes les comparaisons DOIVENT traiter les majuscules et les minuscules (ainsi que leurs combinaisons) comme étant égaux.

7.2.1 Ligne de commande (*command line*)

La ligne de commande est composée:

- du nom du verbe demandé;
- de l'identification de la transaction;
- du nom du ou des points d'extrémité qui doivent exécuter la commande [dans les notifications ou redémarrages, il s'agit du nom du ou des points d'extrémité qui émet (émettent) la commande];
- de la version du protocole.

Ces quatre éléments sont codés comme des chaînes de caractères ASCII imprimables séparées par des espaces, c'est-à-dire les caractères espace ASCII (0x20) ou tabulation (0x09). Il CONVIENT que

les clients intégrés utilisent exactement un seul séparateur espace ASCII, mais ils DOIVENT être capables d'interpréter des messages comportant d'autres caractères espace.

7.2.1.1 Codage des verbes demandés

Les verbes demandés sont codés sous la forme de codes ASCII de quatre lettres majuscules et/ou minuscules (les comparaisons DOIVENT être insensibles à la casse) comme il est défini dans le tableau ci-après:

Verbe	Code
CreateConnection	CRCX
ModifyConnection	MDCX
DeleteConnection	DLCX
NotificationRequest	RQNT
Notify	NTFY
AuditEndpoint	AUEP
AuditConnection	AUCX
RestartInProgress	RSIP

De nouveaux verbes peuvent être définis dans les futures versions de la présente Recommandation. Il peut être nécessaire, pour des besoins expérimentaux, d'utiliser de nouveaux verbes avant qu'ils ne soient approuvés dans une version publiée de la présente Recommandation. Il convient d'identifier les verbes expérimentaux par un code de quatre lettres commençant par la lettre X (XPER, par exemple).

Lorsqu'il reçoit une commande comportant un verbe expérimental qu'il ne prend pas en charge, un client intégré DOIT renvoyer une erreur (code d'erreur 511 – Extension non reconnue).

7.2.1.2 Identificateurs de transaction

Les identificateurs de transaction sont utilisés pour corréler commandes et réponses.

Un client intégré prend en charge deux espaces nominatifs distincts pour les identificateurs de transaction:

- un espace nominatif d'identificateur de transaction pour l'envoi de transactions;
- un espace nominatif d'identificateur de transaction pour la réception de transactions.

A tout le moins, les identificateurs de transaction pour des commandes envoyées à un client intégré donné DOIVENT être uniques pour la durée de vie maximale des transactions à l'intérieur de l'ensemble des agents d'appel qui contrôlent ce client intégré (voir 7.5). Ainsi, quel que soit l'agent d'appel expéditeur, les clients intégrés peuvent toujours détecter des transactions doubles en examinant simplement l'identificateur de transaction. Toutefois, la coordination de ces identificateurs de transaction entre les agents d'appel ne s'inscrit pas dans le cadre de la présente Recommandation.

Les identificateurs de transaction pour toutes les commandes envoyées à partir d'un client intégré donné DOIVENT être uniques pour la durée de vie maximale des transactions (voir 7.5), quel que soit l'agent d'appel auquel la commande est envoyée. Ainsi, un agent d'appel peut toujours détecter une transaction dupliquée provenant d'un client intégré, à l'aide de la combinaison du nom de domaine du point d'extrémité et de l'identificateur de transaction. Le client intégré à son tour peut toujours détecter un acquittement de réponse dupliqué en examinant l'identificateur ou les identificateurs de transaction.

L'identificateur de transaction est codé comme une chaîne de chiffres arabes pouvant atteindre neuf au total. Dans les lignes de commande, il suit immédiatement le codage du verbe.

Les identificateurs de transaction ont des valeurs comprises entre 1 et 999999999. Une entité selon le protocole MGCP NE DOIT PAS réutiliser un identificateur de transaction moins de trois minutes après l'achèvement de la précédente commande ayant utilisé cet identificateur.

7.2.1.3 Codage de nom de point d'extrémité, d'agent d'appel et d'entité avisée

Les noms de points d'extrémité et ceux d'agents d'appel sont codés comme des adresses de courrier électronique (mél), comme défini dans la RFC 821. Dans ces adresses, le nom de domaine identifie le système auquel le point d'extrémité est rattaché tandis que la partie gauche identifie un point d'extrémité spécifique de ce système. Ces deux composantes DOIVENT être insensibles à la casse.

Des exemples de ces noms sont:

aaln/1@ncs2.quelconque.net	La ligne d'accès analogique 1 dans le client intégré ncs2 dans le réseau "quelconque".
Call-agent@ca.quelconque.net	Call Agent (agent d'appel) pour le réseau "quelconque".

Le nom des entités notifiées est exprimé avec la même syntaxe, avec la possibilité d'addition d'un numéro de port, comme dans:

```
Call-agent@ca.quelconque.net:5234
```

Si le numéro de port est omis, le port MGCP par défaut (2427) est utilisé. On trouvera au 6.1.1 des détails supplémentaires relatifs aux noms de points d'extrémité.

7.2.1.4 Codage de la version de protocole

La version du protocole est codée sous la forme du mot clé "MGCP" suivi d'un espace et du numéro de version, qui est à nouveau suivi du nom de profil "NCS" et d'un numéro de version de profil. Les numéros de version sont composés d'un numéro de version majeure, d'un point et d'un numéro de version mineure. Les numéros de version majeure et mineure sont codés comme des nombres décimaux. Le numéro de version de profil défini par la présente Recommandation est 1.0.

La version de protocole pour la présente Recommandation DOIT être codée sous la forme:

```
MGCP 1.0 NCS 1.0
```

La partie "NCS 1.0" signale qu'il s'agit du profil NCS 1.0 du protocole MGCP 1.0.

Une entité qui reçoit une commande avec une version de protocole qu'elle ne prend pas en charge DOIT répondre par une erreur (code d'erreur 528 – Version de protocole incompatible).

7.2.2 Lignes de paramètres

Les lignes de paramètres sont composées d'un nom de paramètre composé dans la plupart des cas d'un seul caractère majuscule, suivi de deux points, d'un espace, et de la valeur du paramètre. Toutefois, les noms et les valeurs des paramètres sont toujours insensibles à la casse. Les paramètres que l'on peut trouver dans des commandes sont définis dans le tableau ci-après:

Nom du paramètre	Code	Valeur de paramètre
ResponseAck[26]	K	Voir la description.
CallId	C	Chaîne hexadécimale dont la longueur NE DOIT PAS dépasser 32 caractères.
ConnectionId	I	Chaîne hexadécimale dont la longueur NE DOIT PAS dépasser 32 caractères.
NotifiedEntity	N	Identificateur, au format RFC 821, composé d'une chaîne arbitraire et du nom de domaine de l'entité demandeuse, éventuellement complétée par un numéro de port, comme dans: Call-agent@ca.quelconque.net:5234 .
RequestIdentifier	X	Chaîne hexadécimale dont la longueur NE DOIT PAS dépasser 32 caractères.
LocalConnectionOptions	L	Voir la description.
Connection Mode (Mode de connexion)	M	Voir la description.
RequestedEvents	R	Voir la description.
SignalRequests	S	Voir la description.
DigitMap	D	Codage textuel d'un script de numérotation.
ObservedEvents	O	Voir la description.
ConnectionParameters	P	Voir la description.
ReasonCode	E	Voir la description.
SpecificEndPointId	Z	Identificateur, au format RFC 821, composé d'une chaîne arbitraire, facultativement suivie d'une arobase "@" suivie du nom de domaine du client intégré auquel ce point d'extrémité est rattaché.
MaxEndPointIds	ZM	Chaîne décimale dont la longueur NE DOIT PAS dépasser16 caractères.
NumEndPoints	ZN	Chaîne décimale dont la longueur NE DOIT PAS dépasser 16 caractères.
RequestedInfo	F	Voir la description.
QuarantineHandling	Q	Voir la description.
DetectEvents	T	Voir la description.
EventStates	ES	Voir la description.
ResourceID	DQ-RI	Voir la description.
RestartMethod	RM	Voir la description.
RestartDelay	RD	Nombre de seconde codé comme un nombre décimal.
Capabilities	A	Voir la description.
VersionSupported	VS	Voir la description.

[26] Le paramètre ResponseAck n'a pas été montré au 6.3 car les identificateurs de transaction n'apparaissent pas dans notre exemple d'API. Des réalisateurs peuvent choisir une approche différente.

Les paramètres ne sont pas nécessairement présents dans toutes les commandes. Le tableau suivant présente l'association entre paramètres et commandes. L'abréviation M signifie obligatoire, O signifie facultatif(ve) et F signifie interdit(e):

Nom du paramètre	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
ResponseAck[26]	O	O	O	O	O	O	O	O
CallId	M	M	O	F	F	F	F	F
ConnectionId	F	M	O	F	F	F	M	F
RequestIdentifier	O	O	O	M	M	F	F	F
LocalConnectionOptions	M	O	F	F	F	F	F	F
Connection Mode	M	O	F	F	F	F	F	F
RequestedEvents	O*	O*	O*	O*	F	F	F	F
SignalRequests	O*	O*	O*	O*	F	F	F	F
NotifiedEntity	O	O	O	O	O	F	F	F
ReasonCode	F	F	O	F	F	F	F	F
ObservedEvents	F	F	F	F	M	F	F	F
DigitMap	O	O	O	O	F	F	F	F
Connection parameters	F	F	O	F	F	F	F	F
Specific Endpoint Id	F	F	F	F	F	O	F	F
MaxEndPointIds	F	F	F	F	F	O	F	F
NumEndPoints	F	F	F	F	F	F	F	F
RequestedInfo	F	F	F	F	F	O	O	F
QuarantineHandling	O	O	O	O	F	F	F	F
DetectEvents	O	O	O	O	F	F	F	F
EventStates	F	F	F	F	F	F	F	F
ResourceID	F	F	F	F	F	F	F	F
RestartMethod	F	F	F	F	F	F	F	M
RestartDelay	F	F	F	F	F	F	F	O
Capabilities	F	F	F	F	F	F	F	F
VersionSupported	F	F	F	F	F	F	F	F
RemoteConnectionDescriptor	O	O	F	F	F	F	F	F

* Les paramètres RequestedEvents et SignalRequests sont facultatifs dans la commande NotificationRequest. Si ces paramètres sont omis, les listes correspondantes seront considérées comme étant vides. Pour les commandes de gestion de connexion, cela s'applique également lorsqu'un paramètre RequestIdentifier est inclus.

Il CONVIENT que les clients intégrés et les agents d'appel fournissent toujours les paramètres obligatoires avant les paramètres facultatifs; toutefois, les clients intégrés NE DOIVENT PAS échouer si cette recommandation n'est pas suivie.

Si des réalisateurs ont besoin d'expérimenter de nouveaux paramètres (par exemple, pour développer une nouvelle application MGCP), il convient qu'ils identifient ces paramètres par des noms qui commencent par la chaîne "X−" ou "X+", comme par exemple:

```
X-FlowerOfTheDay: Daisy
```

Les noms de paramètre qui commencent par "X+" sont des extensions de paramètre obligatoires. Une passerelle qui reçoit une extension de paramètre obligatoire qu'elle ne comprend pas DOIT répondre par une erreur (code d'erreur 511 – Extension non reconnue).

Les noms de paramètre qui commencent par "X−" sont des extensions de paramètre non critiques. Une passerelle qui reçoit une extension de paramètre non critique qu'elle ne peut pas comprendre peut ignorer ce paramètre, en toute sécurité.

Il convient de remarquer que les verbes expérimentaux ont le format X*ABC* tandis que les paramètres expérimentaux ont le format X-ABC.

Si une ligne de paramètres est reçue avec un paramètre interdit, ou avec toute autre erreur de formatage, il convient que l'entité réceptrice réponde avec le code d'erreur le plus spécifique pour l'erreur en question. Le code d'erreur le moins spécifique est 510 – Erreur de protocole. Un texte de commentaires peut toujours être fourni.

7.2.2.1 Acquittement de réponse

Le paramètre d'acquittement de réponse[26] est utilisé pour prendre en charge le dialogue à trois décrit au 7.7. Il contient une liste de "plages d'identificateurs de transaction confirmés" séparées par une virgule.

Chaque "plage d'identificateurs de transaction confirmés" est composée d'un nombre décimal lorsque la plage contient exactement une transaction ou de deux nombres décimaux séparés par un trait d'union, décrivant les identificateurs de transaction le plus faible et le plus élevé qui sont compris dans la plage.

Un exemple d'acquittement de réponse est:

```
K: 6234-6255, 6257, 19030-19044
```

7.2.2.2 RequestIdentifier

L'identificateur de demande corrèle une commande Notify à la commande NotificationRequest qui l'a déclenchée. Un paramètre RequestIdentifier est une chaîne hexadécimale dont la longueur NE DOIT PAS dépasser 32 caractères. La chaîne "0" est réservée pour rapporter des événements persistants dans le cas où aucune commande NotificationRequest n'aurait encore été reçue (voir 6.3.2).

7.2.2.3 Options de connexion locale

Les options de connexion locale décrivent les paramètres opérationnels que les agents d'appel chargent la passerelle d'utiliser pour une connexion. Ces paramètres sont:

- la période de mise en paquet, exprimée en millisecondes et codée sous la forme du mot clé "p" suivi de deux points et d'un nombre décimal;
- le nom littéral de l'algorithme de compression, codé sous la forme du mot clé "a" suivi de deux points et d'une chaîne de caractères;
- le paramètre d'annulation de l'écho, codé sous la forme du mot clé "e" suivi de deux points et de la valeur "on" ou "off";
- le paramètre de type de service, codé sous la forme du mot clé "t" suivi de deux points et de la valeur codée sous la forme de deux chiffres hexadécimaux;
- le paramètre de suppression du silence, codé sous la forme du mot clé "s" suivi de deux points et de la valeur "on" ou "off".

Les paramètres LocalConnectionOptions utilisés pour la Qualité de service dynamique sont:

- le paramètre D-QoS GateID, codé sous la forme du mot clé "dq-gi" suivi de deux points et d'une chaîne hexadécimale (qui peut contenir jusqu'à huit caractères) correspondant à un identificateur de 32 bits pour le paramètre GateID;

- le paramètre Réservation de ressources D-QoS, codé sous la forme du mot clé "dq-rr" suivi de deux points et d'une chaîne de caractères. Il est possible de spécifier une liste de valeurs séparées par un point virgule. Les valeurs possibles sont:

Mode	Signification
sendresv	Réserver uniquement dans le sens envoyer.
recvresv	Réserver uniquement dans le sens recevoir.
snrcresv	Réserver dans le sens envoyer et recevoir.
sendcomt	Affecter uniquement dans le sens envoyer.
recvcomt	Affecter uniquement dans le sens recevoir.
snrccomt	Affecter dans le sens envoi et réception.

- le paramètre ResourceID, codé sous la forme du mot clé "dq-ri" suivi de deux points et d'une chaîne hexadécimale (qui peut contenir jusqu'à huit caractères) correspondant à un identificateur de 32 bits pour le paramètre ResourceID;
- le paramètre ReserveDestination, codé sous la forme du mot clé "dq-rd" suivi de deux points et d'une adresse IP codée de manière similaire à l'adresse IP pour la partie nom de domaine du nom d'un point d'extrémité. Le paramètre ReserveDestination peut éventuellement être suivi de deux points et de caractères décimaux (jusqu'à 5) pour le numéro de port UDP à utiliser.

Les paramètres LocalConnectionOptions utilisés pour le paramètre Security (sécurité) sont codés comme suit:

- le secret est codé sous la forme du mot clé "sc-st" suivi de deux points, d'une méthode, de deux points et du secret effectif. La méthode est la chaîne "clear" (*en clair*) si le secret est codé en texte clair ou la chaîne "base64" s'il est codé en utilisant base64;
- le système cryptographique RTP est codé sous la forme du mot clé "sc-rtp" suivi de deux points et d'une chaîne de systèmes cryptographiques RTP comme défini ci-dessous. Il est possible de spécifier une liste de valeurs séparées par un point virgule;
- le système cryptographique RTCP est codé sous la forme du mot clé "sc-rtcp" suivi de deux points et d'une chaîne de systèmes cryptographiques RTCP comme défini ci-dessous. Il est possible de spécifier une liste de valeurs séparées par un point virgule.

Les chaînes de systèmes cryptographiques RTP et RTCP obéissent à la grammaire suivante:

SystemeCryptographique = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]

AuthenticationAlgorithm = 1*(ALPHA / DIGIT / "-"/ "_")

EncryptionAlgorithm = 1*(ALPHA / DIGIT | "-" / "_")

où ALPHA et DIGIT sont définis dans la RFC 2234. Les espaces ne sont pas autorisés à l'intérieur d'un système cryptographique. L'exemple ci-après illustre l'utilisation d'un système cryptographique:

62/51

La liste effective de l'IPCablecom prenait en charge des systèmes cryptographiques devant être fournis dans la spécification de sécurité de l'IPCablecom UIT-T J.170.

Lorsque plusieurs paramètres sont présents, leurs valeurs sont séparées par une virgule. On DOIT considérer une erreur le fait d'inclure un paramètre n'ayant pas une valeur (code d'erreur 524 – Incohérence dans le paramètre LocalConnectionOptions).

Des exemples d'options de connexion locale sont:

```
L: p:10, a:PCMU
L: p:10, a:PCMU, e:off, t:20, s:on
L: p:30, a:G729A, e:on, t:A0, s:off
```

La valeur hexadécimale de type de service égale à "20" implique une préséance IP de 1 tandis qu'une valeur hexadécimale de type de service égale à "A0" implique une préséance IP de 5.

Ce jeu d'attributs peut être étendu par des attributs d'extension. Ces attributs d'extension sont composés d'un nom d'attribut, suivi de deux points, et d'une liste de valeurs d'attribut séparées par un point virgule. Le nom d'attribut DOIT commencer par les deux caractères "x+" pour une extension obligatoire ou par "x−" pour une extension non obligatoire. Si une passerelle reçoit un attribut d'extension obligatoire qu'elle ne reconnaît pas, elle DOIT rejeter la commande avec une erreur (code d'erreur 525 – Extension inconnue dans le paramètre LocalConnectionOptions).

7.2.2.4 **Capabilities** (*capacités*)

Le paramètre Capabilities (*capacités*) informe l'agent d'appel sur ses capacités lors d'un audit. Le codage des capacités est fondé sur le codage des options de connexion locale pour ce qui concerne les paramètres qui leur sont communs. En outre, le paramètre capabilities peut contenir une liste des paquets pris en charge et une liste des modes pris en charge.

Les paramètres utilisés sont:

- la période de mise en paquets, exprimée en millisecondes et codée sous la forme du mot clé "p" suivi de deux points et d'un nombre décimal. il est possible de spécifier une plage sous la forme de deux nombres décimaux séparés par un trait d'union;
- le nom littéral de l'algorithme de compression, codé sous la forme du mot clé "a" suivi de deux points et d'une chaîne de caractères. Il est possible de spécifier une liste de valeurs séparées par un point virgule;
- la largeur de bande, exprimée en kilobits par seconde (1 000 bits par seconde) et codée sous la forme du mot clé "b" suivi par deux points et d'un nombre décimal. Il est possible de spécifier une plage sous la forme de deux nombres décimaux séparés par un trait d'union;
- le paramètre annulation de l'écho, codé sous la forme du mot clé "e" suivi de deux points et de la valeur "on" si l'annulation de l'écho est prise en charge; ou de la valeur "off" dans le cas contraire;
- le paramètre type de service, codé sous la forme du mot clé "t" suivi de deux points et de la valeur "0" si le type de service n'est pas pris en charge; toute autre valeur indiquant la prise en charge du type de service;
- le paramètre suppression du silence, codé sous la forme du mot clé "s" suivi de deux points et de la valeur "on" si la suppression du silence est prise en charge; ou de la valeur "off" dans le cas contraire;
- les paquets d'événements pris en charge par ce point d'extrémité, codés sous la forme du mot clé "v" suivi de deux puis d'une liste des noms de paquets pris en charge séparés par un point virgule. La première valeur spécifiée est le paquet par défaut pour le point d'extrémité;
- les modes de connexion pris en charge par ce point d'extrémité, codés sous la forme du mot clé "m" suivi de deux points et d'une liste des modes de connexion pris en charge séparés par un point virgule, comme défini au 7.2.2.7;
- le mot clé "dq-gi" si la qualité de service dynamique est prise en charge;

- le mot clé "sc-st" si la sécurité IPCablecom est prise en charge. Dans ce cas, les mots clés ci-après indiquent les systèmes cryptographiques pris en charge:
  - le mot clé "sc-rtp" suivi de deux points et d'une liste de paramètres AuthenticationAlgorithms RTP séparés par un point virgule, d'une barre oblique et d'une liste des EncryptionAlgorithms pris en charge séparés par un point virgule;
  - le mot clé "sc-rtcp" suivi de deux points et d'une liste de paramètres AuthenticationAlgorithms RTCP séparés par un point virgule, d'une barre oblique et d'une liste des EncryptionAlgorithms pris en charge séparés par un point virgule.

Lorsque plusieurs paramètres sont présents, leurs valeurs sont séparées par une virgule.

Des exemples de capacités sont:

```
A:   a:PCMU;G729A, p:10-100, e:on, s:off, v:L;S,
     m:sendonly;recvonly;sendrecv;inactive
A:   a:G729A; p:30-90, e:on, s:on, v:L;S,
     m:sendonly;recvonly;sendrecv;inactive;confrnce,
     dq-gi, sc-st, sc-rtp: 00/51;03
```

Il convient de noter que les codecs et les algorithmes de sécurité sont simplement des exemples – Des spécifications IPCablecom séparées détaillent les codecs et algorithmes réels qui sont pris en charge, ainsi que le codage utilisé.

7.2.2.5 Paramètres de connexion

Les paramètres de connexion sont codés sous la forme d'une chaîne de paires type et valeur, où le type est un identificateur du paramètre comportant deux lettres et où la valeur est un nombre entier décimal. Les types sont séparés des valeurs par le signe "=". Les paramètres sont séparés les uns des autres par une virgule.

Les types des paramètres de connexion sont spécifiés dans le tableau ci-après:

Nom du paramètre de connexion	Code	Valeur du paramètre de connexion
Packets sent (*paquets envoyés*)	PS	Le nombre de paquets qui ont été envoyés dans la connexion
Octets sent (*octets envoyés*)	OS	Le nombre d'octets qui ont été envoyés dans la connexion
Packets received (*paquets reçus*)	PR	Le nombre de paquets qui ont été reçus dans la connexion
Octets received (*octets reçus*)	OR	Le nombre d'octets qui ont été reçus dans la connexion
Packets lost (*paquets perdus*)	PL	Le nombre de paquets qui n'ont pas été reçus dans la connexion, déduit à partir de trous dans le nombre de séquences
Jitter (*gigue*)	JI	La gigue moyenne de réception intermédiaire des paquets, exprimée en millisecondes et comme un nombre entier
Latency (*latence*)	LA	Latence moyenne, exprimée en millisecondes et comme un nombre entier

Les noms des paramètres de connexion pour les extensions sont composés de la chaîne "X−" suivi d'un nom de paramètre d'extension comportant deux lettres. Les agents d'appel qui reçoivent des extensions non reconnues DOIVENT les ignorer en silence.

Un exemple de codage de paramètre de connexion est:

```
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

7.2.2.6 Codes de cause

Les codes de cause sont des valeurs numériques à trois chiffres. Un code de cause est éventuellement suivi d'un espace et d'un commentaire, comme par exemple:

```
900 Endpoint malfunctioning
```

On trouvera une liste de codes de cause au 6.6.

7.2.2.7 Mode de connexion

Le mode de connexion décrit le mode de fonctionnement de la connexion. Les valeurs possibles sont:

Mode	Signification
M: sendonly	La passerelle ne devrait qu'envoyer des paquets
M: recvonly	La passerelle ne devrait que recevoir des paquets
M: sendrecv	Il convient que la passerelle envoie et reçoive des paquets
M: confrnce	Il convient que la passerelle envoie et reçoive des paquets selon le mode "conference" (*conférence*)
M: inactive	Il convient que la passerelle n'envoie ni ne reçoive de paquets
M: replcate	La passerelle ne devrait qu'envoyer des paquets selon le mode "replicate" (*répliquer*)
M: netwloop	Il convient que la passerelle place le point d'extrémité dans le mode "Network Loopback" (*boucle réseau*)
M: netwtest	Il convient que la passerelle place le point d'extrémité dans le mode "Network Continuity Test" (*test de continuité réseau*)

7.2.2.8 Codage de nom d'événement/de signal

Les noms d'événement/de signal sont composés d'un nom de paquet facultatif, séparé par une barre oblique (/) du nom de l'événement effectif. Le nom de l'événement peut éventuellement être suivi du signe arobase (@) et de l'identificateur d'une connexion sur laquelle il convient d'observer l'événement. Les noms d'événement sont utilisés dans les paramètres RequestedEvents, SignalRequests, DetectEvents, ObservedEvents et EventStates. Chaque événement est identifié par un code d'événement. Ces codages ASCII sont insensibles à la casse: il convient de considérer que des valeurs telles que "hu", "Hu", "HU" ou "hU" sont égales.

Des exemples de noms d'événement sont:

X/hu	Transition raccrochage, dans le paquet ligne exemple
X/0	Chiffre 0 dans le paquet ligne exemple
hf	Rappel d'enregistreur, en supposant que le paquet ligne exemple soit le paquet par défaut pour le point d'extrémité
X/rt@0A3F58	Retour d'appel sur la connexion "0A3F58"

En outre, à la place des noms individuels, il est possible d'utiliser la plage et la notation de remplacement par caractères génériques dans les paramètres RequestedEvents et DetectEvents (et non dans les paramètres SignalRequests ObservedEvents, ou EventStates):

X/[0-9]	Les chiffres 0 à 9 dans le paquet ligne exemple
X/X	Les chiffres 0 à 9 dans le paquet ligne exemple
[0-9*#A-D]	Tous les chiffres et toutes les lettres dans le paquet ligne exemple (valeur par défaut pour le point d'extrémité)
X/all	Tous les événements dans le paquet ligne exemple

Enfin, on peut utiliser le signe étoile pour indiquer "toutes les connexions" et le signe dollar pour indiquer la connexion "actuelle". Des exemples de ces notations sont:

X/rt@*	Retour d'appel sur toutes les connexions pour le point d'extrémité
X/rt@$	Retour d'appel sur la connexion courante

On trouvera dans l'Annexe A un jeu initial de paquets d'événements pour des clients intégrés.

7.2.2.9 RequestedEvents

Le paramètre RequestedEvents fournit la liste des événements qui ont été demandés. Les codes d'événement actuellement définis sont décrits dans l'Annexe A.

Chaque événement peut être qualifié par une action demandée ou par une liste d'actions. Ce ne sont pas toutes les actions qui peuvent être combinées – se reporter au 6.3.1 pour connaître les combinaisons valides. Lorsqu'elles sont spécifiées, les actions sont codées sous la forme d'une liste de mots clés mis entre parenthèses et séparés par des virgules. Les codes pour les différentes actions sont:

Action	Code
Aviser immédiatement	N
Cumuler	A
Cumuler en fonction du script de numérotation	D
Ignorer	I
Garder le ou les signaux actifs	K
Commande NotificationRequest intégrée	E
Commande ModifyConnection intégrée	C

Si un script de numérotation n'est pas fourni alors que l'action "accumulation en fonction du script de numérotation" est spécifiée, le point d'extrémité utilise simplement son script de numérotation actuel. S'il n'en a pas, une erreur doit être renvoyée (code d'erreur 519 – Aucun script de numérotation).

Lorsque aucune action n'est spécifiée, c'est l'action par défaut qui doit aviser de l'événement. Cela signifie que, par exemple, "oc" et "oc(N)" sont équivalents. A l'exception des événements persistants, tous les autres événements qui ne sont pas énumérés sont éliminés.

L'action script de numérotation ne peut être spécifiée que pour les chiffres, les lettres et les temporisateurs.

La liste des événements demandés est codée en une seule ligne, les groupes d'événements/d'actions y étant séparés par des virgules. Des exemples de codages de RequestedEvents sont (en utilisant le paquet ligne exemple):

```
R: hu(N), hf(N)          Notify on-hook, notify hook-flash. (aviser du
                         raccrochage, aviser du retour d'enregistreur)
R: hu(N), [0-9#T](D)     Notify on-hook, accumulate digits according to digit
                         map. (aviser du raccrochage, accumuler les chiffres
                         en fonction du script de numérotation)
```

La commande NotificationRequest intégrée respecte le format suivant:

```
E ( R( <RequestedEvents>), D( <Digit Map>), S( <SignalRequests>) )
```

où chacun des R, D et S est facultatif et éventuellement fourni dans un ordre différent. L'exemple suivant illustre l'utilisation de la commande NotificationRequest intégrée, en utilisant le paquet ligne exemple:

```
R: hd(A, E(S(dl), R( B/oc(N), [0-9#T](D) ), D((1xxxxxxxxxx|9011x.T)) ) )
```

Sur décrochage, cumuler l'événement, fournir une tonalité de numérotation et commencer à accumuler les chiffres selon le script de numérotation fourni. Arrêter la tonalité de numérotation lorsque le premier chiffre a été entré ou si aucun chiffre n'a été entré avant l'expiration de la tonalité de numérotation, Notifier que l'opération est terminée. Sinon, aviser du décrochage et recueillir les chiffres lorsqu'il s'est produit une correspondance, une discordance ou une temporisation entre les chiffres. Etant donné que le raccrochage est un événement persistant, il convient de remarquer qu'il sera encore détecté et notifié bien qu'il n'ait pas été spécifié ici.

L'action ModifyConnection intégrée respecte le format suivant:

```
C(M(<ConnectionMode1>( <ConnectionID1> )) , … ,
                                  M(<ConnectionModen>(ConnectionIDn )))
```

L'exemple suivant illustre l'utilisation de la commande ModifyConnection intégrée, en utilisant le paquet ligne exemple:

```
R: hf(A, C(M(inactive(X43DC)), M(sendrecv($)))), B/oc(N), B/of(N)
```

Sur raccrochage-rappel d'enregistreur, modifier le mode de connexion de "X43DC" à "inactive", puis modifier le mode de connexion de "current connection" (*connexion courante*) à "send/receive" (*envoyer/recevoir*). Notifier les événements sur "operation complete" (*opération terminée*) et "operation failure" (*échec de l'opération*).

7.2.2.10 SignalRequests

Le paramètre SignalRequests fournit le nom des signaux qui ont été demandés. Les signaux actuellement définis se trouvent dans l'Annexe A. Un même signal ne peut apparaître qu'une seule fois dans la liste et, par définition, tous les signaux seront appliqués simultanément.

On peut qualifier un certain nombre de signaux par des paramètres de signal. Lorsqu'un signal est qualifié par plusieurs paramètres de signal, ceux-ci sont séparés par des virgules. Chaque paramètre de signal DOIT respecter le format spécifié ci-après (les espaces sont autorisés):

```
paramètre-de-signal              =   valeur-de-paramètre-de-signal |
                                     nom-de-paramètre-de-signal "="
                                     valeur-de-paramètre-de-signal |
                                     nom-de-paramètre-de-signal "("
                                     liste-de-paramètres-de-signal ")"
liste-de-paramètres-de-signal    =   valeur-de-paramètre-de-signal 0*( ","
                                     valeur-de-paramètre-de-signal)
```

où valeur-de-paramètre-de-signal peut être soit une chaîne soit une chaîne entre guillemets, c'est-à-dire une chaîne entourée de deux doubles guillemets. Deux doubles guillemets consécutifs constituent un échappement pour un double guillemet inclus dans la chaîne entre guillemets. Par exemple, `"ab""c"` fournit la chaîne `ab"c`.

Chaque signal comporte un des types de signal suivants qui lui sont associés (voir 6.3.1):

- On/Off (OO);
- Time-out (TO);
- Brief (BR).

Les signaux On/Off peuvent être paramétrés avec un "+" pour activer le signal ou un "−" pour le désactiver. Si un signal on/off n'est pas paramétré, il est actif. Chacun des deux termes ci-après active le signal vmwi provenant du paquet ligne exemple:

vmwi(+), vmwi

Les signaux temporisés Time-out peuvent être paramétrés avec le paramètre de signal "TO" et une valeur de temporisation qui écrase la valeur de temporisation par défaut. Si un signal de temporisation n'est pas paramétré avec une valeur de temporisation, c'est la valeur de temporisation par défaut qui sera utilisé. Chacun des termes suivants appliquera un signal de sonnerie provenant du paquet ligne exemple pendant 6 secondes:

rg(to=6000)

rg(to(6000))

Chaque signal individuel peut définir des paramètres de signal supplémentaires.

Les paramètres de signal seront mis entre parenthèses comme illustré précédemment.

Lorsque plusieurs signaux sont demandés, leurs codes sont simplement séparés par une virgule.

7.2.2.11 ObservedEvents

Les paramètres d'événements observés fournissent la liste d'événements qui ont été observés. Les codes d'événement sont les mêmes que ceux utilisés dans la commande NotificationRequest. Lorsqu'un événement est détecté dans une connexion, l'événement observé identifie la connexion où l'événement a été détecté, à l'aide de la syntaxe "@<connection>". Des exemples d'événements observés qui utilisent le paquet ligne exemple sont:

```
O: hu


O: 8,2,9,5,5,5,5,T
O: hf,hf,hu
```

Les événements qui ont été cumulés conformément au script de numérotation sont rapportés comme des événements individuels dans l'ordre de leur détection. Il est possible de mélanger d'autres événements en les insérant parmi eux. Il convient de noter que si la "chaîne de numérotation courante" n'est pas vide et contient une correspondance partielle alors qu'un autre événement se produit avec pour résultat la création d'un message Notify, la "chaîne de numérotation courante" dont la correspondance est partielle sera incluse dans la liste des événements observés et la "chaîne de numérotation courante" sera alors supprimée – Voir 6.4.3.1 pour les détails.

7.2.2.12 RequestedInfo

Le paramètre RequestedInfo contient une liste de codes de paramètre séparés par une virgule, comme défini dans la section "Lignes de paramètres" – Le 6.3.8 énumère les paramètres qui peuvent être audités. Les valeurs suivantes sont également prises en charge:

Paramètre RequestedInfo	Code
LocalConnectionDescriptor	LC
RemoteConnectionDescriptor	RC

Par exemple, si l'on souhaite auditer la valeur des paramètres NotifiedEntity, RequestIdentifier, RequestedEvents, SignalRequests, DigitMap, DetectEvents, EventStates, LocalConnectionDescriptor, et RemoteConnectionDescriptor, la valeur du paramètre RequestedInfo sera:

```
F: N,X,R,S,D,T,ES,LC,RC
```

La demande de capacités, pour la commande AuditEndPoint, est codée par le code de paramètre "A", comme dans:

```
F: A
```

7.2.2.13 QuarantineHandling

Le paramètre de gestion de quarantaine contient le mot clé "process" (*traiter*) ou "discard" (*ignorer*) pour indiquer le traitement des événements mis en quarantaine, par exemple:

```
Q: process
```

7.2.2.14 DetectEvents

Le paramètre DetectEvents est codé sous la forme d'une liste d'événements séparés par une virgule, comme par exemple:

```
T: hu,hd,hf,[0-9#*]
```

Il convient de noter qu'aucune action ne peut être associée aux événements.

7.2.2.15 EventStates

Le paramètre EventStates est codé sous la forme d'une liste d'événements séparés par une virgule, comme par exemple:

```
ES: hu
```

Il convient de noter qu'aucune action ne peut être associée aux événements.

7.2.2.16 ResourceID

Le paramètre ResourceID est un paramètre de retour utilisé pour la qualité de service dynamique afin de signaler l'identité de la ressource affectée à la passerelle en question. Le paramètre ResourceID est codé sous la forme d'une chaîne pouvant contenir jusqu'à 8 caractères hexadécimaux, comme par exemple:

```
DQ-RI: AB345DC
```

7.2.2.17 RestartMethod

Le paramètre RestartMethod est codé sous la forme d'un des mots clés "graceful", "forced", "restart" ou "disconnected", comme par exemple:

```
RM: restart
```

7.2.2.18 VersionSupported

Le paramètre VersionSupported est codé sous la forme d'une liste de versions prises en charge séparées par une virgule, comme par exemple:

```
VS: MGCP 1.0, MGCP 1.0 NCS 1.0
```

7.3 Formats d'en-tête réponse

L'en-tête réponse est composé d'une ligne de réponse éventuellement suivi d'en-têtes qui codent les paramètres de réponse.

La ligne de réponse commence avec un code de réponse, qui est une valeur numérique à trois chiffres. Le code est suivi d'un espace, de l'identificateur de transaction et éventuellement d'un commentaire précédé d'un espace, par exemple:

```
200 1201 OK
```

Le tableau ci-après résume les paramètres de réponse dont la présence est obligatoire ou facultative dans l'en-tête réponse, en fonction de la commande qui a déclenché la réponse, dans le cas d'une commande réussie. Cependant, il convient que le lecteur étudie tout de même les définitions individuelles des commandes car ce tableau ne fournit que des informations sommaires. L'abréviation M signifie obligatoire, O signifie facultatif(ve) et F signifie interdit(e):

Nom du paramètre	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
ResponseAck[26]	O*	O*	O*	O*	O*	O*	O*	O*
CallId	F	F	F	F	F	F	O	F
ConnectionId	M	F	F	F	F	O	F	F
RequestIdentifier	F	F	F	F	F	O	F	F
LocalConnectionOptions	F	F	F	F	F	O	O	F
Connection Mode	F	F	F	F	F	F	O	F
RequestedEvents	F	F	F	F	F	O	F	F
SignalRequests	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	O	O	O
ReasonCode	F	F	F	F	F	F	F	F
ObservedEvents	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	O	F	F
ConnectionParameters	F	F	O	F	F	F	O	F
Specific Endpoint ID	O	F	F	F	F	O	F	F
MaxEndPointIds	F	F	F	F	F	F	F	F
NumEndPoints	F	F	F	F	F	O	F	F
RequestedInfo	F	F	F	F	F	F	F	F
QuarantineHandling	F	F	F	F	F	F	F	F
DetectEvents	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	O	F	F
ResourceID	O	O	F	F	F	F	F	F
RestartMethod	F	F	F	F	F	F	F	F
RestartDelay	F	F	F	F	F	F	F	F
Capabilities	F	F	F	F	F	O	F	F

Nom du paramètre	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
VersionSupported	F	F	F	F	F	O	F	O
LocalConnection Descriptor	M	O	F	F	F	F	O	F
RemoteConnection Descriptor	F	F	F	F	F	F	O	F
* Le paramètre ResponseAck NE DOIT PAS être utilisé avec n'importe quelles autres réponses que la réponse finale émise après une réponse provisoire pour la transaction en question. Dans ce cas, la présence du paramètre ResponseAck DOIT déclencher un message d'acquittement de réponse – Toute valeur fournie pour ResponseAck sera ignorée.								

Les paramètres de réponse sont décrits pour chacune des commandes dans ce qui suit.

7.3.1 CreateConnection

Dans le cas du message CreateConnection, la ligne de réponse est suivie d'un paramètre Connection-Id avec une réponse réussie (code 200). Un paramètre LocalConnectionDescriptor est en outre transmis avec une réponse positive. Le paramètre LocalConnectionDescriptor est codé comme une "description de session", telle que définie au 7.4. Il est séparé de l'en-tête réponse par une ligne vide, par exemple:

```
200 1204 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

Lorsqu'une réponse provisoire a été émise précédemment, la réponse finale peut en outre contenir le paramètre acquittement de réponse et, lorsqu'une qualité de service dynamique est utilisée, la réponse finale peut également contenir un paramètre ResourceID, comme dans:

```
200 1204 OK
K:
I: FDE234C8
DQ-RI: 23DB4A43

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

La réponse finale est acquittée par un acquittement de réponse:

```
000 1204
```

7.3.2 ModifyConnection

Dans le cas d'un message ModifyConnection réussi, la ligne de réponse est suivie d'un paramètre LocalConnectionDescriptor, si la modification a abouti à celle des paramètres de session (par exemple: le fait de modifier uniquement le mode d'une connexion ne modifie pas les paramètres de session). Le paramètre LocalConnectionDescriptor est codé comme une "description de session", telle que définie au 7.4. Il est séparé de l'en-tête réponse par une ligne vide.

```
200 1207 OK

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

La réponse peut également contenir un paramètre ResourceID lorsqu'une qualité de service dynamique est utilisée comme dans:

```
200 1207 OK
DQ-RI: 12345
```

Lorsqu'une réponse provisoire a été émise précédemment, la réponse finale peut en outre contenir le paramètre acquittement de réponse comme dans:

```
526 1207 No bandwidth (pas de largeur de bande)
K:
```

La réponse finale est acquittée par un acquittement de réponse:

```
000 1207 OK
```

7.3.3 DeleteConnection

En fonction de la variante du message DeleteConnection, la ligne réponse peut être suivie d'une ligne paramètre paramètres de connexion, telle que définie au 7.2.2.5.

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

7.3.4 NotificationRequest

Une réponse NotificationRequest ne comporte aucun autre paramètre de réponse.

7.3.5 Notify

Une réponse Notify ne comporte aucun autre paramètre de réponse.

7.3.6 AuditEndpoint

Dans le cas d'un paramètre AuditEndPoint, la ligne de réponse peut être suivie d'informations pour chacun des paramètres demandés – Chaque paramètre apparaîtra sur une ligne distincte. Les paramètres pour lesquels il n'existe actuellement pas de valeur (par exemple, un script de numérotation) seront tout de même fournis. Chaque nom de point d'extrémité local "allongé" par un caractère générique de remplacement apparaîtra sur une ligne distincte en utilisant le code de paramètre "SpecificEndPointId", par exemple:

```
200 1200 OK
Z: aaln/1@rgw.quelconque.net
Z: aaln/2@rgw.quelconque.net
```

ou:

```
200 1200 OK
A: a:PCMU;G728, p:10-100, e:on, s:off, t:1, v:X;B,
     m:sendonly;recvonly;sendrecv;inactive
A: a:G729A; p:30-90, e:on, s:on, t:1, v:X;B,
     m:sendonly;recvonly;sendrecv;inactive;confrnce
```

7.3.7 AuditConnection

Dans le cas du paramètre AuditConnection, la réponse peut être suivie d'informations pour chacun des paramètres demandés. Les paramètres pour lesquels il n'existe actuellement pas de valeur seront tout de même fournis. Les descripteurs de connexion apparaîtront toujours en dernier, chacun étant précédé d'une ligne vide, comme par exemple:

```
200 1203 OK
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:PCMU;G728
M: sendrecv
P: PS=622, OS=31172, PR=390, OR=22561, PL=5, JI=29, LA=50

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

Si un descripteur de connexion locale et un descripteur de connexion distante sont tous deux fournis, le descripteur de connexion locale sera le premier des deux. Si un descripteur de connexion est demandé alors qu'il n'en existe pas pour la connexion auditée, il n'apparaîtra qu'avec le champ version de protocole SDP.

7.3.8 RestartInProgress

La réponse à une commande RestartInProgress peut inclure le nom d'un autre agent d'appel à contacter, par exemple lorsque l'agent d'appel redirige le point d'extrémité vers un autre agent d'appel, comme dans:

```
521 1204 Redirect
N: CA-1@quelconque.net
```

7.4 Codage de description de session

La description de session est codée conformément au protocole de description de session (SDP, *session description protocol*) mais les clients intégrés peuvent poser un certain nombre d'hypothèses simplificatrices relatives à la description de session telle que spécifiée dans ce qui suit. Il convient de noter que les descriptions de session sont sensibles à la casse conformément à la RFC 2327.

L'utilisation du protocole SDP dépend du type de session, comme spécifié dans le paramètre "media":

- si le paramètre média est mis à "audio", la description de session est pour un service audio;
- si le paramètre média est mis à "video", la description de session est pour un service vidéo.

Pour un service audio, la passerelle considérera que les informations fournies dans le protocole SDP sont pour le média "audio" tandis que, pour le service vidéo, elle les considéra fournies pour le média "video".

7.4.1 Utilisation du service audio du protocole SDP

Dans une passerelle uniquement vocale, il faut seulement décrire les sessions qui utilisent un support et un seul, l'audio. Les paramètres du protocole SDP qui sont pertinents pour l'application vocale sont spécifiés ci-dessous. Les clients intégrés DOIVENT prendre en charge des descriptions de session qui sont conformes à ces règles et dans l'ordre suivant:

1) le profil SDP présenté ci-après;

2) la RFC 2327 (SDP).

Le profil SDP fourni décrit l'utilisation du protocole de description de session dans la signalisation NCS. La description générale et l'explication des paramètres individuels peuvent être trouvées dans la RFC 2327. Toutefois, nous détaillons ci-après quelles valeurs il est nécessaire que les points d'extrémité NCS fournissent pour ces champs (send) et ce qu'il convient qu'ils fassent des valeurs fournies ou non pour ces champs (receive).

Tout paramètre non spécifié ci-après ne DEVRAIT PAS être fourni par un point d'extrémité NCS quelconque et, si un tel paramètre est reçu, il CONVIENT de l'ignorer.

7.4.1.1 Version de protocole (v=)

```
v= <version>
v= 0
```

Send: DOIT être fourni conformément à la RFC 2327 (c'est-à-dire que v= 0).

Receive: DOIT être fourni conformément à la RFC 2327.

7.4.1.2 Origine (o=)

Le champ Origine (o=) est constitué de 6 sous-champs dans la RFC 2327:

```
o= <username> <session-ID> <version> <network-type> <address-type> <address>
o= - 2987933615 2987933615  IN   IP4     A3C47F2146789F0
```

Username (*nom d'utilisateur*)

Send: le trait d'union DOIT être utilisé comme nom d'utilisateur si la confidentialité est demandée.

Sinon, il CONVIENT d'utiliser un trait d'union[27].

Receive: il CONVIENT d'ignorer ce champ.

Session-ID

Send: ce champ DOIT être conforme à la RFC 2327 à des fins d'interopérabilité avec clients non-IPCablecom.

Receive: il CONVIENT d'ignorer ce champ.

Version

Send: conformément à la RFC 2327.

Receive: il CONVIENT d'ignorer ce champ.

Network Type (*type de réseau*)

Send: le type "IN" DOIT être utilisé.

Receive: il CONVIENT d'ignorer ce champ.

Address Type (*type d'adresse*)

Send: le type "IP4" DOIT être utilisé.

Receive: il CONVIENT d'ignorer ce champ.

---

[27] Etant donné que les points d'extrémité NCS ne savent pas quand la confidentialité est demandée, il CONVIENT qu'ils utilisent toujours un trait d'union.

Address (*adresse*):

Send: ce champ DOIT être conforme à la RFC 2327 à des fins d'interopérabilité avec clients non-IPCablecom.

Receive: ce champ DOIT être ignoré.

7.4.1.3 Session Name (s=) (*nom de session*)

```
s= <session-name>
s= -
```

Send: un trait d'union DOIT être utilisé comme Nom de session.

Receive: ce champ DOIT être ignoré.

7.4.1.4 Session and Media Information (i=) (*informations sur la session et le support*)

```
i= <session-description>
```

Send: pour la signalisation NCS, le champ NE DOIT PAS être utilisé.

Receive: ce champ DOIT être ignoré.

7.4.1.5 URI (u=)

```
u= <URI>
```

Send: pour la signalisation NCS, le champ NE DOIT PAS être utilisé.

Receive: ce champ DOIT être ignoré.

7.4.1.6 E-Mail address and phone number (e=, p=) (*adresse mél et numéro de téléphone*)

```
e= <e-mail-address>
p= <phone-number>
```

Send: pour la signalisation NCS, le champ NE DOIT PAS être utilisé.

Receive: ce champ DOIT être ignoré.

7.4.1.7 Connection Data (c=) (*données de connexion*)

Les données de connexion consistent en 3 sous-champs:

```
c= <network-type> <address-type> <connection-address>
c= IN          IP4 10.10.111.11
```

Network Type (*type de réseau*):

Send: le type "IN" DOIT être utilisé.

Receive: le type "IN" DOIT être présent.

Address Type (*type d'adresse*):

Send: le type "IP4" DOIT être utilisé.

Receive: le type "IP4" DOIT être présent.

Connection Address (*adresse de connexion*):

Send: ce champ DOIT être rempli avec une adresse IP de monodiffusion à laquelle l'application recevra le flux média. Donc une valeur de durée de vie (TTL) NE DOIT PAS être présente et une valeur "nombre d'adresses" NON PLUS. Le champ NE DOIT PAS être rempli avec un nom de domaine complet à la place d'une adresse IP. Une adresse différente de zéro spécifie à la fois **l'adresse d'envoi et celle de réception pour le ou les flux média qu'elle couvre**.

Receive: Une adresse IP de monodiffusion ou un nom de domaine complet DOIT être présent(e). Une adresse différente de zéro spécifie à la fois l'adresse d'envoi et celle de réception pour le ou les flux média qu'elle couvre.

7.4.1.8 Bandwidth (b=) (*largeur de bande*)

```
b= <modifier> : <bandwidth-value>
b= AS : 64
```

Send: l'information sur la largeur de bande est facultative dans le protocole SDP mais il CONVIENT de toujours l'inclure[28]. Lorsqu'un codec rtpmap ou pas bien connu[29] est utilisé, l'information sur la largeur de bande DOIT être utilisée.

Receive: il CONVIENT d'inclure l'information de largeur de bande. Si un modificateur de largeur de bande n'est pas inclus, le récepteur DOIT supposer des valeurs raisonnables de largeur de bande par défaut pour les codecs bien connus.

Modifier (*modificateur*):

Send: le type "AS" DOIT être utilisé.

Receive: le type "AS" DOIT être présent.

Bandwidth Value (*valeur de la largeur de bande*):

Send: Ce champ DOIT être rempli avec l'exigence portant sur la largeur de bande maximale pour le flux média, en kilobits par seconde.

Receive: L'exigence relative à la largeur de bande maximale pour le flux média, exprimée en kilobits par seconde, DOIT être présente.

7.4.1.9 Time, repeat times and time zones (t=, r=, z=) (*heure, heures de répétition et fuseaux horaires*)

```
t= <start-time> <stop-time>
t= 36124033 0
r= <repeat-interval> <active-duration> <list-of-offsets-from-start-time>
z= <adjustment-time> <offset>
```

Send: Time (heure) DOIT être présent; l'heure de début (start time) PEUT être zéro mais il CONVIENT que ce soit l'heure courante, et il CONVIENT que stop time (l'heure de fin) soit zéro. il CONVIENT DE NE PAS utiliser Repeat Times (heures de répétitions) et Time Zones (fuseaux horaires) mais s'ils sont utilisés, il convient que ce soit conformément à la RFC 2327.

Receive: si l'un quelconque de ces champs est présent, il CONVIENT de l'ignorer.

---

[28] Si ce champ n'est pas utilisé, le contrôleur de passerelle pourrait ne pas autoriser la longueur de bande appropriée.

[29] Un codec pas bien connu est un codec qui n'est pas défini dans la spécification de codecs J.161.

7.4.1.10 Encryption Keys (*clés de chiffrement*)

```
k= <method>
k= <method> : <encryption-keys>
```

Les services de sécurité pour IPCablecom UIT-T J.170 doivent être définis par la spécification de sécurité de l'IPCablecom. Les services de sécurité spécifiés pour les protocoles RTP et RTCP ne sont pas conformes à ceux de RFC 1889, RFC 1890 et RFC 2327. Dans l'intérêt de l'interopérabilité avec les dispositifs non-IPCablecom, le paramètre "k" ne sera donc pas utilisé pour acheminer les paramètres de sécurité.

Send: ce champ NE DOIT PAS être utilisé.

Receive: il CONVIENT d'ignorer ce champ.

7.4.1.11 Attributes (a=) (*attributs*)

```
a= <attribute> : <value>
a= rtpmap : <payload type> <encoding name>/<clock rate> [/<encoding parameters>]
a= rtpmap : 0 PCMU / 8000
a= X-pc-codecs: <alternative 1> <alternative 2> …
a= X-pc-secret: <method>:<encryption key>
a= X-pc-csuites-rtp: <alternative 1> <alternative 2> …
a= X-pc-csuites-rtcp: <alternative 1> <alternative 2> …
a= X-pc-spi-rtcp: <value>
a= X-pc-bridge: <number-ports>
a= <attribute>
a= recvonly
a= sendrecv
a= sendonly
a= ptime
```

Send: on PEUT inclure une ou plusieurs des lignes d'attribut "a" spécifiées ci-dessous. il CONVIENT DE NE PAS utiliser une ligne d'attribut non spécifiée ci-dessous.

Receive: on PEUT inclure une ou plusieurs des lignes d'attribut "a" spécifiées ci-dessous et on DOIT les manipuler en conséquence. Des lignes d'attribut "a" qui ne sont pas spécifiées ci-dessous peuvent être présentes mais DOIVENT être ignorées.

rtpmap:

Send: lorsqu'il est utilisé, ce champ DOIT l'être conformément à la RFC 2327. Il PEUT être utilisé tant pour les codecs bien connus que pour ceux qui ne le sont pas. Les noms de codage utilisés sont fournis dans une spécification IPCablecom distincte.

Receive: le champ DOIT être utilisé conformément à la RFC 2327.

X-pc-codecs:

Send: le champ contient la liste d'autres codecs que le point d'extrémité est capable d'utiliser pour cette connexion. La liste est ordonnée par degré de préférence décroissant, c'est-à-dire que le codec en variante le plus préféré est le premier de la liste. Un codec est codé de manière similaire à un "encoding name" (nom de codage) dans le champ rtpmap.

Receive: achemine une liste de codecs que le point d'extrémité distant est capable d'utiliser pour cette connexion. Les codecs NE DOIVENT PAS être utilisés jusqu'au moment de leur signalisation par le biais d'une ligne média (m=).

X-pc-secret:

Send: ce champ contient un secret de bout en bout devant être utilisé pour la sécurité RTP et RTCP. Le secret est codé de manière similaire au paramètre clé de chiffrement (k=) de la RFC 2327 avec les contraintes suivantes:

- la clé de chiffrement NE DOIT PAS contenir un système cryptographique, elle contient une phrase de passe uniquement;
- la <method> (méthode) spécifiant le codage de la phrase de passe DOIT être soit "clear" (*en clair*) ou "base64" comme défini dans la RFC 2045, à l'exception de la longueur maximale de ligne qui n'est pas spécifiée ici. La méthode "clear" (*en clair*) NE DOIT PAS être utilisée si le secret contient des caractères qui sont interdits dans le protocole SDP.

Receive: ce champ achemine un secret de bout en bout devant être utilisé pour la sécurité RTP et RTCP.

X-pc-csuites-rtp:

X-pc-csuites-rtcp:

Send: ce champ contient une liste de systèmes cryptographiques que le point d'extrémité est capable d'utiliser pour cette connexion (RTP et RTCP respectivement). Le premier système cryptographique énuméré est ce que le point d'extrémité prévoit actuellement d'utiliser. Tous les systèmes cryptographiques éventuels restant dans la liste représentent des variantes classées par ordre de préférence décroissant, c'est-à-dire que le système cryptographique en variante le plus préféré est le deuxième dans la liste. Un système cryptographique est codé comme il est spécifié ci-après:

Système cryptographique = [AuthenticationAlgorithm] "/" [EncryptionAlgorithm]

AuthenticationAlgorithm = 1*(ALPHA / DIGIT / "-" / "_")

EncryptionAlgorithm = 1*(ALPHA / DIGIT / "-" / "_")

où ALPHA et DIGIT sont définis dans la RFC 2234. Les espaces ne sont pas autorisés à l'intérieur d'un système cryptographique. L'exemple ci-après illustre l'utilisation d'un système cryptographique:

62/51

La liste effective des systèmes cryptographiques doit être fournie dans la spécification de sécurité de l'IPCablecom UIT-T J.170.

Receive: achemine une liste de systèmes cryptographiques que le point d'extrémité distant est capable d'utiliser pour cette connexion. Tout autre système cryptographique que le premier de la liste ne peut être utilisé sans avoir été signalisé par le biais d'une nouvelle ligne de système cryptographique où le système cryptographique souhaité est énuméré le premier.

X-pc-spi-rtcp:

Send: ce champ contient l'index des paramètres de sécurité IPSEC (SPI, *security parameter index*) devant être utilisé pour envoyer des paquets RTCP vers le point d'extrémité pour le flux média en question. L'index SPI est un identificateur de 32 bits codé sous la forme d'une chaîne pouvant contenir jusqu'à 8 caractères hexadécimaux. Le champ DOIT être fourni lorsque la sécurité RTCP est utilisée.

Receive: achemine l'index IPSEC SPI à utiliser pour envoyer des paquets RTCP sur IPSEC. Le champ DOIT être présent lorsque la sécurité RTCP est utilisée.

X-pc-bridge:

Send: les points d'extrémité NCS NE DOIVENT PAS utiliser cet attribut.

Receive: s'ils reçoivent cet attribut, les points d'extrémité NCS DOIVENT l'ignorer.

recvonly:

Send: le champ DOIT être utilisé conformément à la RFC 2543.

Receive: le champ DOIT être utilisé conformément à la RFC 2543.

sendrecv:

Send: le champ DOIT être utilisé conformément à la RFC 2543.

Receive: le champ DOIT être utilisé conformément à la RFC 2543.

sendonly:

Send: le champ DOIT être utilisé conformément à la RFC 2543, sauf que l'adresse IP et le numéro de port NE DOIVENT PAS être mis à zéro.

Receive: le champ DOIT être utilisé conformément à la RFC 2543.

ptime:

Send: il CONVIENT que le champ ptime soit toujours fourni et, lorsqu'on l'utilise, il DOIT l'être conformément à la RFC 2327. Lorsqu'un champ rtpmap ou un codec non bien connu est utilisé, le champ ptime DOIT être fourni.

Receive: le champ DOIT être utilisé conformément à la RFC 2327. Lorsque le champ "ptime" est présent, l'adaptateur MTA DOIT l'utiliser dans le calcul des réservations de QS. Si le champ "ptime" n'est pas présent, l'adaptateur MTA DOIT admettre des valeurs par défaut raisonnables pour les codecs bien connus.

7.4.1.12 Media Announcements (m=) (*annonces de médias*)

Les annonces de médias (m=) consistent en 3 sous-champs:

```
M= <media> <port> <transport> <format>
M= audio 3456 RTP/AVP 0
```

Media:

Send: le type de média "audio" DOIT être utilisé.

Receive: le type reçu DOIT être "audio".

Port:

Send: ce champ DOIT être rempli conformément à la RFC 2327. Le port spécifié est le port de réception, que le flux soit unidirectionnel ou bidirectionnel. Le port d'envoi peut être différent.

Receive: ce champ DOIT être utilisé conformément à la RFC 2327. Le port spécifié est le port de réception. Le port d'envoi peut être différent.

Transport:

Send: le protocole de transport "RTP/AVP" DOIT être utilisé.

Receive: le protocole de transport DOIT être "RTP/AVP".

Media Formats (*formats de médias*):

Send: un type de média approprié tel que défini dans la RFC 2327 DOIT être utilisé.

Receive: conformément à la RFC 2327.

7.4.2 Utilisation du service vidéo SDP

Les détails relatifs à l'utilisation du protocole SDP pour les services vidéo appellent un complément d'étude.

7.5 Transmission sur UDP

7.5.1 Fourniture de messages fiable

Les messages MGCP sont transmis sur le protocole UDP. Les commandes sont envoyées à une des adresses IP définies dans le DNS (système de nom de domaine) pour le point d'extrémité ou l'agent d'appel spécifié. Les réponses sont renvoyées à l'adresse source de la commande. Toutefois, il convient de noter que la réponse peut, dans les faits, provenir d'une adresse IP autre que celle à laquelle la commande a été envoyée.

Lorsque aucun port n'a été fourni pour le point d'extrémité[30], il convient que les commandes soient envoyées au port MGCP par défaut, à savoir le port 2427.

Les messages transportés sur le protocole UDP peuvent subir des pertes. En l'absence de réponse opportune, les commandes sont répétées. Les entités MGCP sont censées mémoriser une liste des réponses qu'elles ont envoyées à des transactions récentes, c'est-à-dire une liste de toutes les réponses qu'elles ont envoyées au cours des dernières Tt_{hist} secondes, ainsi qu'une liste des transactions qui sont en cours d'exécution. Les identificateurs des commandes entrantes sont comparés aux identificateurs de transaction contenus dans les réponses récentes. Si une correspondance est trouvée, l'entité MGCP n'exécute pas la transaction mais répète simplement la réponse. Si aucune correspondance n'est trouvée, l'entité MGCP examine la liste des transactions en cours d'exécution. Si une correspondance est trouvée, l'entité MGCP n'exécutera pas la transaction, qui est simplement ignorée.

Il appartient à l'entité émettrice de la demande de fournir des temporisations appropriées pour toutes les transactions en cours et de réessayer les commandes dont la temporisation a été dépassée. Une stratégie de retransmission est spécifiée au 7.5.2.

En outre, si les commandes répétées ne reçoivent pas de réponse, l'entité de destination est supposée ne pas être disponible. Il appartient à l'entité émettrice de la demande de rechercher des services redondants et/ou de libérer des connexions actives ou en instance comme il est spécifié au 6.4.

7.5.2 Stratégie de retransmission

La présente Recommandation évite de spécifier de quelconques valeurs statiques pour les temporisateurs de retransmission car il s'agit de valeurs typiquement dépendantes de chaque réseau. Normalement, il y a lieu que les temporisateurs de retransmission estiment leur temporisateur en mesurant la durée écoulée entre l'envoi d'une commande et le retour d'une réponse. Les clients intégrés DOIVENT implémenter une stratégie de retransmission en utilisant un algorithme de ralentissement exponentiel avec des valeurs minimales et maximales de temporisateur de retransmission.

---

[30] Chaque point d'extrémité peut être fourni avec une adresse d'agent d'appel et un port distincts.

Il CONVIENT que les clients intégrés utilisent l'algorithme implémenté dans le protocole TCP-IP, qui fait appel à deux variables comme suit:

- le temps d'acquittement moyen (AAD, *average acknowledgement delay*), qui est estimé par un calcul de moyenne lissée exponentiellement des temps observés;
- l'écart moyen (ADEV, *average deviation*), qui est estimé par un calcul de moyenne lissée exponentiellement de la valeur absolue de la différence entre le temps observé et la moyenne actuelle.

Le temporisateur de retransmission (RTO, *retransmission timer*), dans le protocole TCP, est réglé à la valeur de la somme du temps moyen d'acquittement plus N fois l'écart moyen, N étant une constante.

Après toute retransmission, il convient que l'entité MGCP effectue les opérations suivantes:

- doubler la valeur estimée du temps d'acquittement moyen (AAD);
- calculer une valeur aléatoire, répartie uniformément entre 0,5 AAD et AAD;
- régler le temporisateur de retransmission (RTO) à la valeur du minimum entre la somme de cette valeur aléatoire plus N fois l'écart moyen;
- RTO_{max} (la valeur par défaut de RTO_{max} est de 4 secondes).

Cette procédure a deux effets: étant donné qu'elle comporte une composante décroissant de manière exponentielle, elle ralentit automatiquement le flux de messages en cas d'un encombrement assujetti aux besoins de la communication en temps réel. D'autre part, étant donné qu'elle comporte une composante aléatoire, elle casse la synchronisation potentielle entre les notifications déclenchées par le même événement extérieur.

Pour le temporisateur de retransmission, la valeur initiale utilisée est par défaut de 200 millisecondes tandis que sa valeur maximale est par défaut de 4 secondes. Ces valeurs par défaut peuvent être modifiées par le processus de mise en service.

7.6 Superposition

Dans certains cas, un agent d'appel voudra envoyer plusieurs messages en même temps à un ou plusieurs points d'extrémité dans une passerelle et inversement. Lorsque plusieurs messages doivent être envoyés dans les mêmes paquets UDP, ils sont séparés par une ligne de texte qui contient un point unique, comme par exemple dans:

```
200 2005 OK

DLCX 1244 aaln/2@rgw.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
```

Les messages superposés DOIVENT être traités comme s'ils avaient été reçus dans des datagrammes distincts; toutefois, s'il est nécessaire de retransmettre un message (commande ou réponse), c'est le datagramme en entier qui DOIT être retransmis, pas seulement le message manquant. Les messages individuels contenus dans le datagramme DOIVENT être traités dans l'ordre, en commençant par le premier d'entre eux.

Les erreurs rencontrées dans un message qui a été superposé NE DOIVENT PAS affecter l'un quelconque des autres messages reçus dans ce paquet – Chaque message est traité séparément.

7.7 Identificateurs de transaction et dialogue à trois

Les identificateurs de transaction sont des nombres entiers dans la plage de 1 à 999 999 999. Les agents d'appel peuvent décider d'utiliser un espace numérique spécifique pour chacune des passerelles qu'ils gèrent ou d'utiliser le même espace numérique pour toutes les passerelles qui appartiennent à un certain groupe arbitraire. Les agents d'appel peuvent décider de partager entre plusieurs processus indépendants la charge de gérer une grande passerelle. Ces processus partageront le même espace numérique de transaction. Il existe de multiples implémentations possibles de cette mise en commun, par exemple une attribution centralisée des identificateurs de transaction ou une préaffectation à différents processus de plages d'identificateurs ne se chevauchant pas. Les implémentations DOIVENT garantir que des identificateurs de transaction uniques sont affectés à toutes les transactions provenant d'un quelconque agent d'appel qui a été envoyé à une passerelle particulière dans la période de Tt_{hist} secondes. Les passerelles peuvent simplement détecter les transactions en double en verrouillant au niveau du seul identificateur de transaction.

Le paramètre d'acquittement de réponse peut se trouver dans une commande quelconque. Il transporte un jeu de "plages d'identificateurs de transaction confirmés" pour les réponses finales reçues – les réponses provisoires NE DOIVENT PAS être confirmées.

Les passerelles MGCP peuvent choisir de supprimer les copies des réponses à des transactions dont l'identificateur est inclus dans des "plages d'identificateurs de transaction confirmés" reçues dans un message; toutefois, le fait que la transaction ait été exécutée DOIT être conservé pendant Tt_{hist} secondes. En outre, lorsqu'un message acquittement de réponse[31] est reçu, la réponse qu'il est en train d'acquitter peut être supprimée. Il convient que les passerelles éliminent en silence les commandes supplémentaires provenant de cet agent d'appel lorsque l'identificateur de transaction tombe dans ces plages et que la réponse avait été émise depuis moins de Tt_{hist} secondes.

Posons les termes $terme_{nouveau}$ et $terme_{ancien}$ comme étant le nom de point d'extrémité respectivement dans une nouvelle commande, $cmd_{nouveau}$, et une certaine ancienne commande, cmd_{ancien}. Les identificateurs de transaction à confirmer dans la commande $cmd_{nouveau}$ DEVRAIENT être déterminés comme suit:

1) si $terme_{nouveau}$ ne contient aucun caractère de remplacement générique:

   a) réponses non confirmées à d'anciennes commandes où $terme_{ancien}$ équivaut à $terme_{nouveau}$;

   b) éventuellement, une ou plusieurs réponses non confirmées où $terme_{ancien}$ contenait un caractère de remplacement générique "any-of" et où le nom de point d'extrémité renvoyé dans la réponse était $terme_{nouveau}$;

   c) éventuellement, une ou plusieurs réponses non confirmées où $terme_{ancien}$ contenait le caractère de remplacement générique "all" et où $terme_{nouveau}$ est couvert par le caractère de remplacement contenu dans $terme_{ancien}$;

   d) éventuellement, une ou plusieurs réponses non confirmées où $terme_{ancien}$ contenait un caractère de remplacement générique "any-of", où aucun nom de point d'extrémité n'avait été renvoyé et où $terme_{nouveau}$ est couvert par le caractère de remplacement générique contenu dans $terme_{ancien}$;

[31] Contrairement à une commande ayant un paramètre acquittement de réponse.

2) si $terme_{nouveau}$ contient le caractère de remplacement générique "all":

   a) éventuellement, une ou plusieurs réponses non confirmées où $terme_{ancien}$ contenait le caractère de remplacement générique "all" et où $terme_{nouveau}$ est couvert par le caractère de remplacement contenu dans le $terme_{ancien}$;

3) si le $terme_{nouveau}$ contient le caractère de remplacement générique "any of":

   a) éventuellement, une ou plusieurs réponses non confirmées où le $terme_{ancien}$ contenait un caractère de remplacement générique "all" et le $terme_{nouveau}$ est couvert par le caractère de remplacement générique contenu dans le $terme_{ancien}$ si le caractère de remplacement générique "any of" contenu dans le $terme_{nouveau}$ était remplacé par le caractère de remplacement générique "all".

Il CONVIENT DE NE PAS confirmer une même réponse dans deux messages distincts.

Les exemples suivants illustrent l'emploi de ces règles:

- si $terme_{nouveau}$ est "aaln/1" tandis que $terme_{ancien}$ est aussi "aaln/1", l'ancienne réponse peut être confirmée selon la règle 1a;
- si $terme_{nouveau}$ est "aaln/1" tandis que $terme_{ancien}$ est "*", l'ancienne réponse peut être confirmée selon la règle 1c;
- si $terme_{nouveau}$ est "aaln/*" tandis que $terme_{ancien}$ est "*", l'ancienne réponse peut être confirmée selon la règle 2a;
- si $terme_{nouveau}$ est "aaln/$" tandis que $terme_{ancien}$ est "aaln/*", l'ancienne réponse peut être confirmée selon la règle 3a.

Il CONVIENT DE NE PAS utiliser les valeurs "plages d'identificateurs de transaction confirmés" si plus de Tt_{hist} secondes se sont écoulées depuis que la passerelle a émis sa dernière réponse à cet agent d'appel ou lorsqu'une passerelle reprend son fonctionnement. En l'occurrence, il convient que les commandes soient acceptées et traitées, sans un test quelconque sur l'identificateur de transaction.

En outre, il CONVIENT DE NE PAS confirmer une réponse si elle a été reçue depuis plus de Tt_{hist} secondes.

Les messages de confirmation de réponses peuvent être émis et reçus dans un ordre quelconque. La passerelle doit conserver l'union des identificateurs de transaction qui ont été confirmés dans des commandes récentes.

7.8 Réponses provisoires

Dans un certain nombre de cas, les durées d'achèvement de transaction peuvent être significativement plus longues que d'ordinaire[32]. La signalisation NCS utilise le protocole UDP comme protocole de transport et la fiabilité est assurée par des transmissions sélectives basées sur une temporisation, elle-même basée sur une estimation de la somme des temps aller-retour dans le réseau plus la durée d'achèvement des transactions. La variance significative de la durée d'achèvement des transactions est donc problématique lorsqu'on souhaite la détection rapide des pertes de messages sans surcharge excessive.

Afin de surmonter ce problème, une réponse provisoire DOIT donc être émise si et seulement si la durée d'achèvement de transaction dépasse une certaine durée. La réponse provisoire acquitte la réception de la commande même si on peut ne pas encore connaître l'issue de la commande, par exemple du fait d'une réservation de ressources en attente. En règle générale, il convient qu'une

[32] Par exemple, lorsque des ressources sont réservées et affectées en externe comme partie d'une transaction.

transaction qui nécessite une communication extérieure pour s'achever (comme par exemple une réservation de ressources réseau) émette une réponse provisoire. En outre, si un doublon de commande CreateConnection ou ModifyConnection est reçu alors que la transaction n'a pas fini d'être exécutée, une réponse provisoire DOIT être renvoyée.

Une sémantique transactionnelle pure impliquerait que les réponses provisoires ne renvoient aucune autre information que le fait que la transaction soit en cours d'exécution. Par contre, une approche optimiste autorisant qu'une certaine information soit renvoyée permet une réduction du délai qui autrement aurait été encouru dans le système.

Les réponses provisoires DOIVENT être seulement envoyées en réponse à une commande CreateConnection ou ModifyConnection. Afin de diminuer le délai dans le système, un identificateur de connexion et une description de session DOIVENT être inclus dans la réponse provisoire à la commande CreateConnection. Si une description de session doit être retournée par la commande ModifyConnection, elle DOIT être incluse dans cette réponse provisoire également. Si la transaction s'achève avec succès, l'information renvoyée dans la réponse provisoire DOIT être répétée dans la réponse finale. Il s'agit d'une erreur de protocole que de ne pas répéter cette information ou de modifier l'une quelconque des informations précédemment fournies dans une réponse couronnée de succès. Si la transaction échoue, un code d'erreur est renvoyé – L'information renvoyée précédemment n'est plus valide.

Si une commande DeleteConnection est reçue pour le point d'extrémité, une transaction CreateConnection ou ModifyConnection en cours d'exécution DOIT être annulée. En l'occurrence, il CONVIENT de renvoyer automatiquement une réponse pour la transaction annulée si une retransmission de celle-ci est détectée.

Lorsqu'une réponse provisoire est reçue, la durée de temporisation pour la transaction en question DOIT être réglée à une valeur significativement supérieure ($Tt_{longtran}$). Le but de ce temporisateur est principalement de détecter une défaillance de point d'extrémité. La valeur par défaut de $Tt_{longtran}$ est 5 secondes mais un processus de mise en service peut la modifier.

Lorsque la transaction achève de s'exécuter, la réponse finale est envoyée et la réponse provisoire désormais obsolète est supprimée. Afin d'assurer la rapide détection d'une perte de réponse finale, la réponse finale émise après des réponses provisoires pour une transaction DOIT être acquittée. Par conséquent, le point d'extrémité DOIT inclure un paramètre "ResponseAck" vide dans ces réponses finales et celles-ci seulement. La présence du paramètre "ResponseAck" dans la réponse finale déclenchera une réponse "acquittement de réponse" à renvoyer au point d'extrémité. La réponse "acquittement de réponse" inclut l'identificateur de transaction de la réponse qu'elle acquitte dans l'en-tête réponse. La réception de cette réponse "acquittement de réponse" est soumise aux mêmes stratégies et procédures de temporisation et de retransmission que les réponses à des commandes (voir 6.4), c'est-à-dire que l'expéditeur de la réponse finale la transmettra si "l'acquittement de réponse" n'est pas reçu à temps. La réponse "'acquittement de réponse" n'est jamais acquittée.

8 Sécurité

Si des entités non autorisées pouvaient utiliser le protocole MGCP, elles pourraient établir des appels non autorisés ou perturber des appels autorisés. La sécurité n'est pas fournie en tant que partie intégrante du protocole MGCP. Au contraire, le protocole MGCP suppose l'existence d'une couche inférieure qui assure la sécurité réelle.

Les prescriptions et solutions pour la sécurité dans la signalisation NCS sont fournies dans la spécification de la sécurité IPCablecom UIT-T J.170, qu'il convient de consulter pour obtenir des informations supplémentaires.

ANNEXE A

Paquets d'événements

La présente annexe définit un jeu initial de paquets d'événements pour les différents types de points d'extrémité actuellement définis par IPCablecom pour les clients intégrés. Les paquets ci-après sont définis pour les types de points d'extrémité de clients intégrés qui sont énumérés:

Type de point d'extrémité	Paquet	Nom de paquet	Paquet par défaut
Analogue Access Line (ligne d'accès analogique)	Base	B	Néant
Vidéo	*A étudier*	*A étudier*	*A étudier*
Interface BRI du réseau RNIS	*A étudier*	*A étudier*	*A étudier*

Chaque paquet définit un nom de paquet pour le paquet ainsi que des codes et définitions d'événements pour chacun des événements qu'il contient. Les tableaux d'événements/signaux pour chaque paquet comportent cinq colonnes:

Code code d'événements unique pour le paquet, utilisé pour l'événement/le signal;

Description brève description de l'événement/signal;

Evénement une coche apparaît dans cette colonne si l'événement peut être demandé par le contrôleur de passerelle média. En variante, un ou plusieurs des symboles suivants peuvent apparaître:

"**P**" indique que l'événement est persistant;

"**S**" indique que l'événement est un état d'événement qui peut être audité;

"**C**" indique que l'événement/signal peut être détecté/appliqué sur une connexion;

Signal si rien n'apparaît dans cette colonne, l'événement ne peut pas être signalé sur commande par le contrôleur de passerelle média. Sinon, les symboles suivants identifient le type d'événement;

"**OO**" signal commuté (On/Off). Le signal est activé jusqu'à la commande par le contrôleur de passerelle média de le désactiver et inversement;

"**TO**" signal temporisé. Le signal a une durée donnée jusqu'à ce qu'il soit remplacé par un nouveau signal. Les valeurs de temporisation par défaut sont fournies. Une valeur zéro indique que le délai d'expiration est infini. Ces valeurs peuvent être modifiées par le processus de mise en service;

"**BR**" signal bref. L'événement a une durée courte et connue.

Autres informations: cette colonne fournit des informations supplémentaires sur l'événement/signal (par exemple la durée par défaut des signaux TO).

Sauf indication contraire, tous les événements/signaux sont détectés/appliqués sur des points d'extrémité; le signal audio qu'ils génèrent n'est transmis à aucune connexion que le point d'extrémité peut avoir. Toutefois, le signal audio généré par des événements/signaux détectés/appliqués sur une connexion sont transmis dans la connexion associée, quel que soit le mode de connexion.

Paquets du protocole de base

Les paquets suivants sont actuellement définis dans le protocole de base. Ils s'appliquent à tous les points d'extrémité:

- base

Paquet de base

Nom du paquet: B

Les codes suivants sont utilisés pour identifier les événements et signaux pour le paquet de "base", pour tous les types de point d'extrémité:

Code	Description	Evénement	Signal	Autres informations
oc	Opération terminée	√	–	
of	Echec de l'opération	√	–	

Opération terminée (oc, *operation complete*): l'événement opération terminée est généré lorsqu'on a demandé à la passerelle d'appliquer un ou plusieurs signaux du type TO au point d'extrémité alors qu'un ou plusieurs signaux se sont achevés sans avoir été arrêtés par la détection d'un événement demandé (comme par exemple une transition de décrochage ou la composition d'un chiffre). Le rapport d'achèvement peut comporter comme paramètre le nom du signal qui est parvenu à la fin de sa vie, comme dans

```
O: B/oc(monpaquet/monsignal)
```

Lorsque le signal rapporté a été appliqué à une connexion, le paramètre fourni inclut également le nom de la connexion, comme dans:

```
O: B/oc(monpaquet/monsignal@0A3F58)
```

Lorsque l'événement opération terminée est demandé, il ne peut pas être paramétré avec un quelconque paramètre d'événement. Lorsque le nom de paquet est omis, c'est le nom de paquet par défaut qui est utilisé.

L'événement opération terminée peut en plus être généré comme défini dans le protocole de base, par exemple lorsqu'une commande ModifyConnection intégrée s'achève avec succès, comme dans:

```
O: B/oc(B/C)
```

Echec de l'opération (of, *operation failure*): en général, l'événement échec de l'opération peut être généré lorsqu'on avait demandé à la passerelle d'appliquer un ou plusieurs signaux du type TO au point d'extrémité et qu'un ou plusieurs de ces signaux ont échoué avant d'être temporisés. Le rapport d'achèvement peut comporter comme paramètre le nom du signal qui a échoué, comme dans:

```
O: B/of(monpaquet/monsignal)
```

Lorsque le signal rapporté a été appliqué à une connexion, le paramètre fourni inclut également le nom de la connexion, comme dans:

```
O: B/of(monpaquet/monsignal@0A3F58)
```

Lorsque l'événement échec de l'opération est demandé, des paramètres d'événement ne peuvent pas être spécifiés. Lorsque le nom de paquet est omis, c'est le nom de paquet par défaut qui est utilisé.

L'événement échec de l'opération peut en plus être généré comme spécifié dans le protocole de base, par exemple lorsqu'une commande ModifyConnection imbriquée échoue, comme dans:

```
O: B/of(B/C(M(sendrecv(AB2354))))
```

Audio

Les paquets d'événements pour l'audio appellent un complément d'étude.

Vidéo

Les paquets d'événements pour la vidéo appellent un complément d'étude.

RNIS

Les paquets d'événements pour le réseau RNIS d'accès de base appellent un complément d'étude.

ANNEXE B

Qualité de service dynamique

Dans la présente annexe, on trouvera des détails supplémentaires sur l'utilisation de la qualité de service dynamique (D-QoS, *dynamic quality of service*) dans la signalisation NCS. Nous décrivons avec plus de détails l'adaptateur MTA escompté et incluons une machine à états que l'adaptateur MTA peut implémenter pour prendre en charge le comportement D-QoS décrit. Pour plus de détails, il convient de consulter la spécification de qualité de service dynamique (UIT-T J.163) de l'IPCablecom.

Introduction

La prise en charge par l'adaptateur MTA de la réalisation de la qualité de service dynamique doit mémoriser et entretenir l'état de D-QoS, connexion par connexion. A chaque fois que la D-QoS a été utilisée pour une connexion, le point d'extrémité gardera les informations de D-QoS associées à la connexion jusqu'à ce qu'elle soit supprimée:

- **GateID**: L'identificateur de passerelle courant utilisé pour la connexion;
- **ResourceID**: L'identificateur de ressource courant utilisé pour la connexion;
- **Last reservation** (dernière réservation): Paramètres relatifs à la plus récente réservation pour la connexion. Y sont compris classificateurs et paramètres de média tant dans le sens envoyer que dans le sens recevoir.
- **Last commit** (dernière affectation): Paramètres relatifs à plus récente affectation pour la connexion. Y sont compris classificateurs et paramètres de média tant dans le sens envoyer que dans le sens recevoir.
- **Reserve Destination** (destination de recherche): Adresse IP et port que l'on peut utiliser pour activer les réservations de ressources lorsque l'information sur l'adresse distante n'est pas encore connue, comme il est expliqué ci-après.
- **Gate Location** (emplacement de la passerelle): Adresse IP et port où il convient d'envoyer le message d'affectation de D-QoS lorsqu'on utilise le protocole RSVP. L'adaptateur MTA apprend cette adresse par le biais des messages QS du protocole RSVP.

L'identificateur de passerelle (GateID) est la clé à la réservation de ressource. Une fois un identificateur GateID fourni pour une connexion, une machine à états de D-QoS est créée pour la connexion et toutes les informations ci-dessus seront conservées pour la connexion jusqu'à ce qu'elle soit supprimée ou jusqu'à ce qu'un nouvel identificateur GateID soit fourni. Dans ce dernier cas, la machine à états D-QoS et les informations ci-dessus sont réinitialisées tandis que l'ancienne réservation est supprimée[33].

---

[33] Noter que, si un identificateur de ressource ResourceID est inclus et qu'il corresponde à l'ancien identificateur ResourceID, il convient de ne pas supprimer l'ancienne réservation avant que la nouvelle ne soit faite.

Les ressources peuvent être réservées et affectées de manière indépendante par l'adaptateur MTA tant dans le sens envoyer que dans le sens recevoir. L'adresse IP de destination dans le sens envoyer et le port ainsi que l'adresse IP source sont tirés du paramètre RemoteConnectionDescriptor, s'il a été fourni. Dans ce cas, l'adaptateur MTA DOIT utiliser les classificateurs suivant pour la réservation et l'affectation de ressource:

	MTA-o (J.112/RSVP)
Aval/recevoir	
Source IP	IP(SDP-t)
Source Port	*
Destination IP	IP(SDP-o)
Destination Port	Port(SDP-o)
Amont/envoyer	
Source IP	IP(SDP-o)
Source Port	Port(o)
Destination IP	IP(SDP-t)
Destination Port	Port(SDP-t)

où

- **IP(SDP-o)** se rapporte à l'adresse IP du média dans le paramètre LocalConnectionDescriptor de l'adaptateur MTA-o.
- **IP(SDP-t)** se rapporte à l'adresse IP du média dans le paramètre RemoteConnectionDescriptor de l'adaptateur MTA-o.
- **Port(SDP-o)** se rapporte au port du média dans le paramètre LocalConnectionDescriptor de l'adaptateur MTA-o.
- **Port(o)** se rapporte au port source que l'adaptateur MTA-o utilise pour envoyer un média sur cette connexion. Il convient de remarquer qu'il peut être ou non le même que Port(SDP-o).

Lorsqu'un paramètre RemoteConnectionDescriptor n'a pas encore été fourni, l'adresse IP réelle de destination dans le sens envoyer et le port sont inconnus et donc c'est l'adresse ReserveDestination qui est utilisée à la place. Dans le sens recevoir, l'adresse IP source et le port sont remplacés par une structure générique. Cela permet une réservation et une affectation dans le sens recevoir pour la ressource dans la liaison d'accès. Les classificateurs suivants DOIVENT être utilisés:

	MTA-o (J.112/RSVP)
Aval/recevoir	
Source IP	*
Source Port	*
Destination IP	IP(SDP-o)
Destination Port	Port(SDP-o)
Amont/envoyer	
Source IP	IP(SDP-o)
Source Port	Port(o)
Destination IP	IP(RD-o)
Destination Port	Port(RD-o)

où:

- **IP(RD-o)** se rapporte à l'adresse IP dans le paramètre ReserveDestination fourni;
- **IP(Port-o)** se rapporte au numéro de port dans le paramètre ReserveDestination fourni. Si aucun numéro de port n'est spécifié, la valeur par défaut de 9 s'applique;
- une fois que, pour les médias, les adresses et le port de destination dans le sens envoyer et sources dans le sens recevoir sont connus, les réservations sont mises à jour avec les classificateurs appropriés;
- lorsque le protocole RSVP est utilisé comme protocole de réservation de ressource, l'adresse de destination utilisée dans le message RSVP PATH (*chemin RSVP*) sera l'adresse IP de ReserveDestination fournie, jusqu'à ce qu'un paramètre RemoteConnectionDescriptor soit fourni.

Machine à états NCS/D-QoS

Comme il a été expliqué précédemment, l'adaptateur MTA conserve l'état pour la qualité de service dynamique utilisée sur une connexion. L'état est tiré d'une machine à états qui est piloté par les éléments suivants:

- **Current state** (*état courant*) qui consiste en la paire (SendQoSState, ReceiveQoSState), où chaque état de QS peut être l'un des suivants:
  - **N** – Aucune réservation de ressource n'existe pour ce sens;
  - **R** – Une réservation de ressource existe pour ce sens mais aucune ressource n'est actuellement affectée;
  - **C** – Une réservation de ressource existe pour ce sens et un certain nombre de ressources sont actuellement affectées;
  - **Connection mode** (*mode de connexion*) qui est le mode de connexion NCS. Les modes de connexion "Conference" (*conférence*), "Network Loopback" (*boucle réseau*), et "Network Continuity Test" (*test de continuité réseau*) n'apparaissent pas de manière explicite dans la machine à états car ils sont similaires à "SendReceive"(*envoyer et recevoir*). De même, le mode de connexion "Replicate" (*répliquer*) n'apparaît pas car il est similaire à "SendOnly" (*envoyer uniquement*).
- **Resource Change** (*modification de ressources*) qui est l'un ou plusieurs des éléments suivants:
  - modifications de l'adresse IP ou du port RemoteConnectionDescriptor (il est nécessaire de mettre à jour le classificateur). Y compris le cas où il arrive pour la première fois;
  - modifications de codec;
  - modification du paramètre Ptime;
  - etc.
- Les **règles D-QoS** fournies au 6.3.3.

Comme il a été expliqué précédemment, la machine à états est réinitialisé lorsqu'un nouvel identificateur de passerelle (GateID) est reçu. Si un identificateur de ressource ResourceID est également fourni et qu'il est le même que l'ancien ResourceID, on DOIT effectuer la ou les réservations pour la nouvelle machine à états avant de libérer celles pour l'ancienne.

L'ensemble des *états* possibles est:

- (N, N) les ressources dans le sens envoyer ne sont pas réservées, les ressources dans le sens recevoir ne sont pas réservées.
- (R, R) les ressources dans le sens envoyer sont réservées, les ressources dans le sens recevoir sont réservées.

- (C, R) les ressources dans le sens envoyer sont réservées et affectées, les ressources dans le sens recevoir sont réservées.
- (R, C) les ressources dans le sens envoyer sont réservées, les ressources dans le sens recevoir sont réservées et affectées.
- (C, C) les ressources dans le sens envoyer sont réservées et affectées, les ressources dans le sens recevoir sont réservées et affectées.
- (R, N) les ressources dans le sens envoyer sont réservées, les ressources dans le sens recevoir ne sont pas réservées.
- (C, N) les ressources dans le sens envoyer sont réservées et affectées, les ressources dans le sens recevoir ne sont pas réservées.
- (N, R) les ressources dans le sens envoyer ne sont pas réservées, les ressources dans le sens recevoir sont réservées.
- (N, C) les ressources dans le sens envoyer ne sont pas réservées, les ressources dans le sens recevoir sont réservées et affectées.

Une fois que des ressources ont été réservées et/ou affectées pour un sens, une réservation pour ce sens existe pour la durée de vie de la connexion. Le tableau ci-après montre la relation entre états et mode de connexion ou paramètres de réservation D-QoS:

	SendState	**RecvState**
Pas de paramètre réserver/affecter fourni – mode de connexion:		
inactive	R	R
sendonly, replcate	C	R
recvonly	R	C
sendrecv, confrnce, netwloop, netwtest	C	C
Paramètre réserver/affecter fourni:		
sendresv	R	N, R*
recvresv	N, R*	R
snrcresv	R	R
sendcomt	C	N, R*
recvcomt	N, R*	C
snrccomt	C	C
* Si des ressources ont été précédemment réservées ou affectées pour le sens, l'état sera R sinon il sera N.		

Le diagramme de transition d'état réelle est décrit dans la Figure B.1:

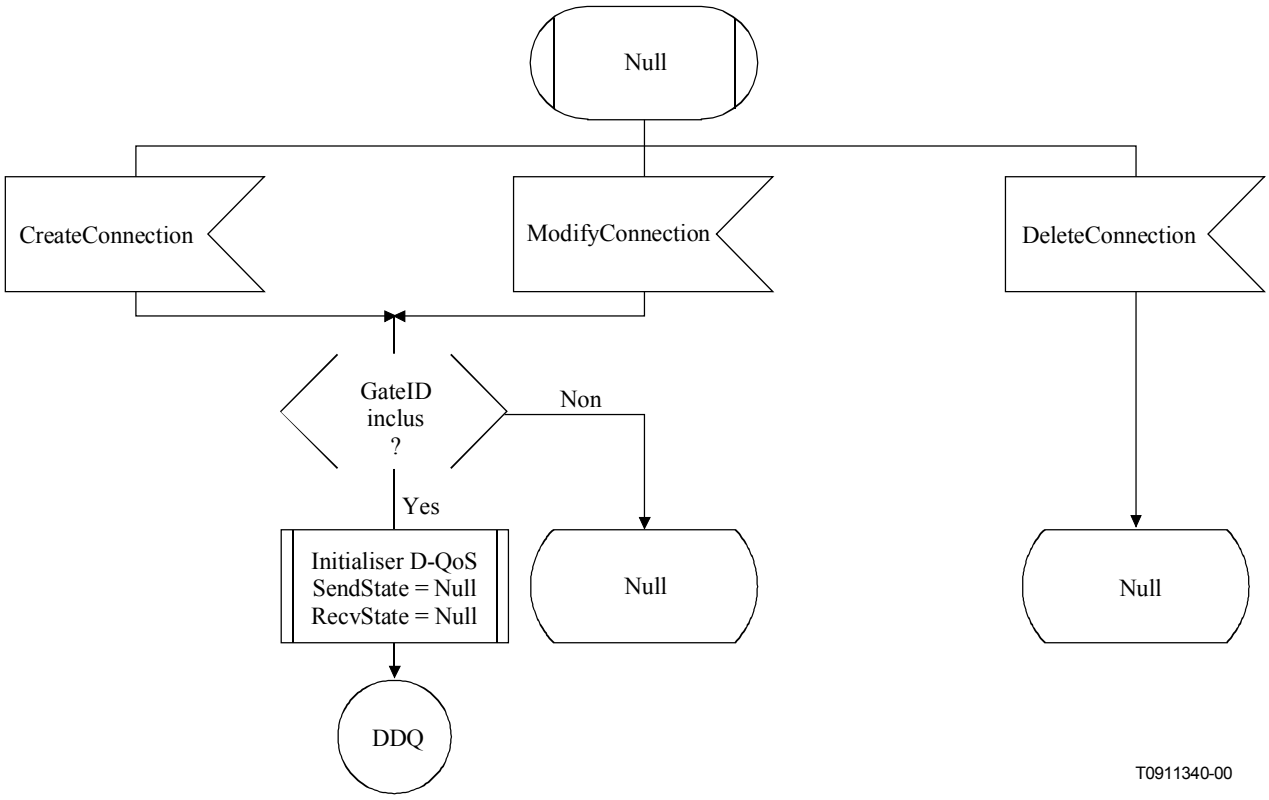


Figure B.1/J.162 – Diagramme d'état NCS/D-QoS *(feuillet 1 de 2)*

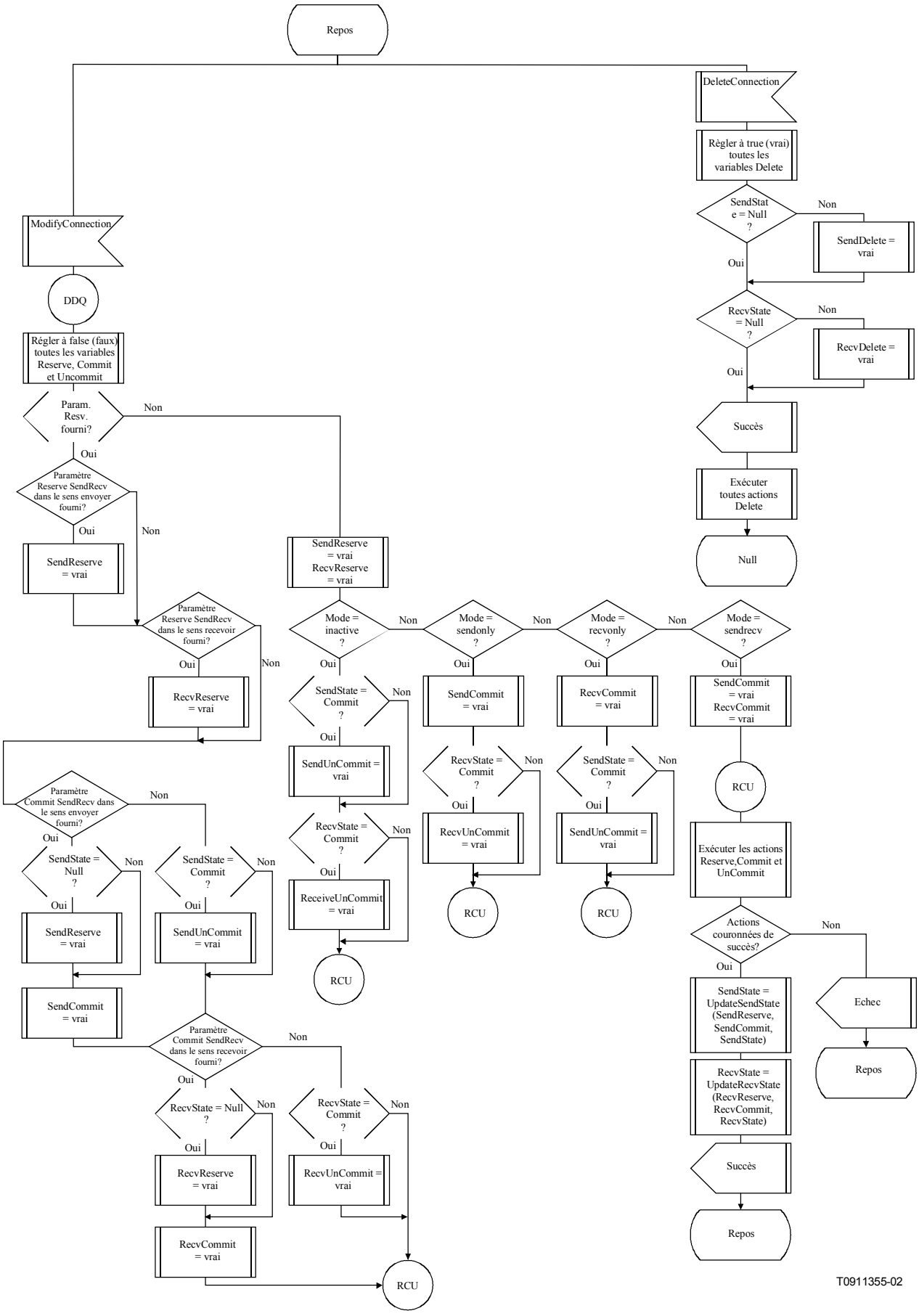


Figure B.2/J.162 – Diagramme d'état NCS/D-QoS *(feuillet 2 de 2)*

Pour exécuter la machine à états, des variables booléennes seront positionnées afin indiquer s'il faut effectuer des opérations reserve (*réserver*), unreserve (*annuler la réservation*), commit (*affecter*), et uncommit (*annuler l'affectation*). Le pseudo-code ci-après fournit des détails sur chacune des procédures D-QoS qui doivent être exécutées comme l'indiquent ces booléens. Les *actions* suivantes spécifient les actions D-QoS qui doivent être menées dans chacune de ces procédures:

- **SR** signifie qu'une réservation dans le sens envoyer de la D-QoS sera exécutée;
- **RR** signifie qu'une réservation dans le sens recevoir de la D-QoS sera exécutée,
- **SC** signifie qu'une affectation dans le sens envoyer de la D-QoS sera exécutée;
- **RC** signifie qu'une affectation dans le sens recevoir de la D-QoS sera exécutée,
- **SD** signifie qu'une suppression de réservation dans le sens envoyer de la D-QoS sera exécutée;
- **RD** signifie qu'une suppression de réservation dans le sens recevoir de la D-QoS sera exécutée;
- **SU** signifie qu'une annulation de l'affectation dans le sens envoyer pour la D-QoS (c'est-à-dire un abaissement des ressources affectées dans le sens envoyer jusqu'à zéro) sera exécutée.
- **RU** signifie qu'une annulation de l'affectation dans le sens recevoir pour la D-QoS (c'est-à-dire un abaissement des ressources affectées dans le sens envoyer jusqu'à zéro) sera exécutée.

SendReserve()

```
Si <ressources réservées actuelles ≠ ressources à réserver> alors {
                    -- sauter la réservation si réservation existante OK
          Si <RemoteConnectionDescriptor fourni> alors
               SR(RemoteConnectionDescriptor)
                    -- utiliser le classificateur de RemoteConnectionDescriptor
          sinon si <ReserveDestination fourni> alors
               SR(ReserveDestination)
                    -- utiliser le classificateur de ReserveDestination
                    -- classifier, envoyer vers ReserveDestination si RSVP
          sinon ERREUR
}
```

ReceiveReserve()

```
Si <ressources réservées actuelles ≠ ressources à réserver> alors {
                    -- sauter la réservation si réservation existante OK
          Si <RemoteConnectionDescriptor fourni> alors
               RR(RemoteConnectionDescriptor)
                    -- utiliser le classificateur de RemoteConnectionDescriptor
          sinon si <(J.112 QoS) ou (RSVP et ReserveDestination fourni)> alors
               RR(*)
                    -- utiliser le classificateur de caractère de remplacement
                    -- générique, envoyer vers ReserveDestination si RSVP
          sinon ERREUR
}
```

SendCommit()

```
Si <ressources affectées actuelles ≠ ressources à affecter> alors {
                    -- sauter l'affectation si l'existante OK
          Si <RemoteConnectionDescriptor fourni> alors {
               Si non <ressources à affecter ⊂ ressources réservées> alors {
                    -- l'ancienne réservation ne satisfait pas à ce qui est sur le
                    -- point d'être affecté, donc mettre à jour la réservation
               SR(RemoteConnectionDescriptor)
          }
          si <(J.112 QoS) ou (RSVP et ReserveDestination fourni)> alors {
               SC(RemoteConnectionDescriptor)
                    -- envoyer vers ReserveDestination si RSVP
               } sinon ERREUR
          } sinon ERREUR.     -- impossible d'affecter dans le sens envoyer sans
                              -- le paramètre RemoteConnectionDescriptor
}
```

ReceiveCommit()

```
Si <ressources affectées actuelles ≠ ressources à affecter> alors {
                    -- sauter l'affectation si l'existante OK
          Si non <ressources à affecter ⊂ ressources réservées> alors {
               Si <RemoteConnectionDescriptor fourni> alors
                    RR (RemoteConnectionDescriptor)
          sinon si <(J.112 QoS) ou (RSVP et ReserveDestination fourni)> alors
               RR (*)    -- utiliser le classificateur de caractère de
                         -- remplacement générique, envoyer vers
                         -- ReserveDestination si RSVP
          sinon ERREUR
     }
          Si <RemoteConnectionDescriptor fourni> alors
               RC (RemoteConnectionDescriptor)
          sinon si <(J.112 QoS) ou (RSVP et ReserveDestination fourni)> alors
               RC (*)    -- utiliser le classificateur de caractère de
                         -- remplacement générique, envoyer vers
                         -- ReserveDestination si RSVP
          sinon ERREUR
}
```

SendReserveDelete()

```
Si <ressources réservées dans le sens envoyer> alors
          SD ()     -- supprimer la réservation
```

ReceiveReserveDelete()

```
Si < ressources réservées dans le sens recevoir> alors
          RD ()     -- supprimer la réservation
```

SendUnCommit()

```
Si < ressources affectées dans le sens envoyer > alors
          SU ()     -- annuler l'affectation des ressources affectées
```

ReceiveUnCommit()

```
Si < ressources affectées dans le sens recevoir> alors
          RU ()     -- annuler l'affectation des ressources affectées
```

```
State UpdateState(DoCommit, DoReserve, OldState)

Si < DoCommit = vrai> alors
          retourner Commit
sinon si < DoReserve = vrai> alors
          retourner Reserve
sinon
          retourner OldState
```

APPENDICE I

Paquet d'événements exemple

Le présent appendice fournit un paquet d'événements exemple pour les lignes d'accès analogique. Le paquet est simplement présenté ici à titre illustratif et pour faciliter l'inclusion d'exemples informatifs dans la partie principale de la Recommandation. Il ne constitue en aucune manière une définition complète de paquet et il convient de ne pas considérer le nom de paquet illustré comme affecté. Etant donné que le paquet n'est simplement qu'un exemple, les détails des événements et signaux individuels seront omis et ne sont présentés qu'en tant que descriptions de haut niveau à titre illustratif.

Paquet ligne exemple

Nom du paquet: X

Les codes ci-après servent à identifier les événements et signaux pour le paquet "ligne exemple" pour "les lignes d'accès analogique":

Code	Description	Evénement	Signal	Autres informations
0-9,*,#,A, B,C,D	Tonalités DTMF	√	BR	
bz	Tonalité d'occupation	–	TO	
dl	Tonalité de numérotation	–	TO	
hd	Transition de décrochage	P, S	–	
hf	Rappel d'enregistreur	P	–	
hu	Transition de raccrochage	P, S	–	
rg	Sonnerie	–	TO	
rt	Tonalité de retour d'appel	–	C, TO	
t	Temporisateur	√	–	
vmwi	Indicateur visuel de message en attente	–	OO	
X	Joker pour tonalités DTMF	√	–	correspond à n'importe quel chiffre "0 à 9"

Etant donné que le paquet précédent n'est simplement qu'un exemple, la définition des événements et signaux individuels indiquée ci-après n'est présentée qu'en tant que description de haut niveau. Un paquet réel et implémentable aura à spécifier les détails de chaque événement et signal. Ces détails peuvent différer d'un prestataire de service du RTPC analogique à l'autre:

tonalités DTMF (0-9,*,#,A, B,C,D): définissent toutes les tonalités DTMF;

tonalité d'occupation (bz, *busy tone*): elle indique à l'appelant que l'appelé est déjà engagé dans un appel;

tonalité de numérotation (dl, *dial-tone*): elle indique à l'appelant qu'il peut placer un appel;

transition de décrochage (hd, *off-hook transition*): l'événement décrochage indique que le combiné associé au point d'extrémité a été décroché;

rappel d'enregistreur (hf, *flash hook*): l'événement rappel d'enregistreur indique qu'un rappel d'enregistreur s'est produit sur le combiné associé au point d'extrémité;

transition de raccrochage (hu, *on-hook transition*): l'événement raccrochage indique que le combiné associé au point d'extrémité a été raccroché;

sonnerie (rg, *ringing*): la sonnerie indique qu'il convient de faire sonner le téléphone de l'appelé;

tonalité de retour d'appel (rt, *ring back tone*): le signal de retour d'appel indique à l'appelant que l'appelé est en train d'être alerté;

temporisateur (t): comme il est décrit au 6.1.5, le temporisateur T est un temporisateur que l'on peut fournir et qui ne peut être annulé que par l'entrée DTMF;

indicateur visuel de message en attente (vmwi, *visual message waiting indicator*): le signal Indicateur visuel de message en attente active ou désactive une indication visuelle d'une messagerie vocale en attente;

joker de tonalités DTMF (X, *DTMF tones wildcard*): le joker pour les tonalités DTMF correspond à n'importe quel chiffre DTMF compris entre 0 et 9.

APPENDICE II

Exemple de codages de commande

Le présent appendice présente des exemples de commandes et de réponses accompagnées du codage réel utilisé en supposant que le paquet ligne exemple est utilisé. Des exemples sont fournis pour chaque commande. Tous les commentaires présentés dans les commandes et les réponses sont facultatifs.

NotificationRequest

Le premier exemple illustre une demande de notification (NotificationRequest) qui fait sonner un téléphone et recherche un événement décrochage:

```
RQNT 1201 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AC
R: hd(N)
S: rg
```

La réponse indique que la transaction a été couronnée de succès:

```
200 1201 OK
```

Le deuxième exemple illustre une demande de notification (NotificationRequest) qui recherchera et accumulera un événement décrochage, puis fournira une tonalité de numérotation et accumulera les chiffres conformément au script de numérotation fourni. "L'entité avisée" est réglée à "`ca@ca1.quelconque.net:5678`" et, étant donné que le paramètre SignalRequests est vide[34], tous les signaux TO actuellement actifs seront coupés. Tous les événements contenus dans le tampon de quarantaine seront traités, et la liste des événements à détecter dans l'état "notification" et

[34] Il pourrait aussi bien avoir été omis.

"verrouillé" sera incluse dans les tonalités de télécopie en plus des "événements demandés" et événements persistants:

```
RQNT 1202 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AC
R: hd(A, E(S(dl), R(B/oc, hu, [0-9#*T](D))))
D: (0T|00T|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
S:
Q: process
T: ft
```

La réponse indique que la transaction a été couronnée de succès:

```
200 1202 OK
```

Notify

L'exemple ci-dessous illustre un message Notify qui avise d'un événement décrochage suivi d'un nombre de 12 chiffres commençant par "91". Y est inclus un identificateur de transaction qui corrèle le message Notify avec la commande NotificationRequest dont il résulte. La commande est envoyée à "l'entité avisée" courante, qui est normalement la valeur effective fournie dans le paramètre NotifiedEntity, c'est-à-dire "`ca@ca1.quelconque.net:5678`" – Une situation de reprise sur défaillance pourrait l'avoir modifiée:

```
NTFY 2002 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AC
O: hd,9,1,2,0,1,8,2,9,4,2,6,6
```

La réponse Notify indique que la transaction a été couronnée de succès:

```
200 2002 OK
```

CreateConnection

Le premier exemple illustre une commande CreateConnection à créer une connexion sur le point d'extrémité spécifié. La connexion fera partie de l'identificateur CallId spécifié. Les options de connexion locale (LocalConnectionOptions) spécifient que G.711 μ-law sera le codec utilisé et que la période de mise en paquets sera de 10 ms. Le mode de connexion est "receive only" (*recevoir uniquement*):

```
CRCX 1204 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
```

La réponse indique que la transaction a été couronnée de succès, et donc un identificateur de connexion pour la connexion nouvellement créée est inclus. Une description de session pour cette nouvelle connexion est également incluse – Remarquer qu'elle est précédée d'une ligne vide.

```
200 1204 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

Le deuxième exemple illustre une commande CreateConnection contenant une demande de notification et un descripteur RemoteConnectionDescriptor:

```
CRCX 1205 aaln/1@rgw-2569.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: sendrecv
X: 0123456789AD
R: hd
S: rg

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

La réponse indique que la transaction a échoué car le combiné était déjà décroché. Par conséquent, il n'est pas renvoyé d'identificateur de connexion (connection-id) ou de description de session:

```
401 1205 Phone off-hook
```

Le troisième exemple illustre l'utilisation d'une réponse provisoire et d'un dialogue à trois. Nous créons cette fois une autre connexion utilisant la qualité de service dynamique et acquittant la réponse précédente reçue:

```
CRCX 1206 aaln/1@rgw-2569.quelconque.net MGCP 1.0 NCS 1.0
K: 1205
C: A3C47F21456789F0
L: p:10, a:PCMU, dq-gi:A735C2
M: inactive

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

Une réponse provisoire est renvoyée initialement:

```
100 1206 Pending
I: DFE233D1

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 0
```

Un peu plus tard, la réponse finale est reçue:

```
200 1206 OK
K:
DQ-RI: A12D5F1
I: DFE233D1

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
```

```
m=audio 3456 RTP/AVP 0
```

L'agent d'appel acquitte la réponse finale comme il est demandé:

```
000 1206
```

et la transaction est terminée.

ModifyConnection

Le premier exemple montre une commande ModifyConnection qui positionne simplement le mode de connexion d'une connexion sur "send/receive" (*envoyer/recevoir*) – "L'Entité avisée" est également positionnée:

```
MDCX 1209 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
N: ca@ca1.quelconque.net
M: sendrecv
```

La réponse indique que la transaction a été couronnée de succès:

```
200 1209 OK
```

Dans le deuxième exemple, nous passons une description de session et incluons une demande de notification avec la commande ModifyConnection. Le point d'extrémité commencera à jouer des tonalités de retour d'appel à l'attention de l'usager:

```
MDCX 1210 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: hu
S: rt

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 3456 RTP/AVP 0
```

La réponse indique que la transaction a été couronnée de succès:

```
200 1206 OK
```

DeleteConnection (lancée par l'agent d'appel)

Dans cet exemple, l'agent d'appel charge simplement le client intégré de supprimer la connexion FDE234C8 sur le point d'extrémité spécifié:

```
DLCX 1210 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
```

La réponse indique qu'il s'agit d'un succès et que la connexion a été supprimée. Donc, pour cette connexion, les paramètres de connexion sont également inclus:

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

DeleteConnection (lancée par le client intégré)

Dans cet exemple, le client intégré envoie à l'agent d'appel une commande DeleteConnection pour l'informer qu'une connexion sur le point d'extrémité spécifié a été supprimée. Le paramètre ReasonCode (code de cause) spécifie la cause de la suppression, et les paramètres de connexion sont également fournis pour la connexion:

```
DLCX 1210 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
E: 900 - Hardware error
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

L'agent d'appel envoie une réponse de succès à la passerelle:

```
200 1210 OK
```

DeleteConnection (plusieurs connexions depuis l'agent d'appel)

Dans le premier exemple, l'agent d'appel charge le client intégré de supprimer toutes les connexions liées à l'appel "A3C47F21456789F0" sur le point d'extrémité spécifié:

```
DLCX 1210 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
```

La réponse indique qu'il s'agit d'un succès et que cette ou ces connexions ont été supprimées:

```
250 1210 OK
```

Dans le deuxième exemple, l'agent d'appel charge le client intégré de supprimer toutes les connexions liées à tous les points d'extrémité spécifiés:

```
DLCX 1210 aaln/*@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
```

La réponse indique qu'il s'agit d'un succès:

```
250 1210 OK
```

AuditEndpoint

Dans le premier exemple, l'agent d'appel souhaite savoir quels points d'extrémité sont présents sur le client intégré spécifié, d'où l'utilisation du joker "all of" pour la partie locale du nom de point d'extrémité:

```
AUEP 1200 *@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
```

Le client intégré indique un succès et inclut une liste de noms de points d'extrémité:

```
200 1200 OK
Z: aaln/1@rgw-2567.quelconque.net
Z: aaln/2@rgw-2567.quelconque.net
```

Dans le deuxième exemple, les capacités d'un des points d'extrémité sont demandées:

```
AUEP 1201 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0 F: A
```

La réponse indique un succès ainsi que les capacités. Deux codecs sont pris en charge mais avec des capacités différentes. Par conséquent, deux jeux de capacités distincts sont renvoyés:

```
200 1201 OK

A: a:PCMU, p:10-100, e:on, s:off, v:X;B, m:sendonly;
                   recvonly;sendrecv;inactive;netwloop;netwtest
A: a:G729A, p:30-90, e:on, s:on, v:X;B, m:sendonly;
                   recvonly;sendrecv;inactive;confrnce;netwloop
```

Dans le troisième exemple, l'agent d'appel audite toutes les informations possibles pour le point d'extrémité:

```
AUEP 2002 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
F: R,D,S,X,N,I,T,O,ES
```

La réponse indique qu'il s'agit d'un succès:

```
200 2002 OK
R: X/hu,oc(N),[0-9](N)
D:
S: vmwi(+)
X: 0123456789B1
N: [128.96.41.12]
I: 32F345E2
T:
O: hd,9,1,2
ES: hd
```

La liste d'événements demandés contient trois événements. Lorsque le nom de paquet n'est pas spécifié, c'est le nom de paquet par défaut qui est utilisé. Il en va de même pour les actions et on doit donc prendre l'action par défaut – Notify – pour l'événement "X/hu". L'omission d'une valeur pour "script de numérotation" signifie que le point d'extrémité n'a pas actuellement de script de numérotation. Il n'y a pas actuellement de signaux temporisés actifs mais le signal OO "vmvi" est actuellement actif et est donc inclus – dans le cas présent, il a été paramétré mais le paramètre aurait pu être exclu. "L'entité avisée" actuelle se rapporte à une adresse IP et une seule connexion existe pour le point d'extrémité. La valeur actuelle de DetectEvents est vide, et la liste des événements observés (ObservedEvents) contient les quatre événements spécifiés. Enfin, les états d'événements audités révèlent que le combiné était décroché au moment où la transaction était traitée.

AuditConnection

Le premier exemple montre la commande AuditConnection lorsque nous auditons CallId, NotifiedEntity, LocalConnectionOptions, Connection Mode, LocalConnectionDescriptor, et les paramètres de connexion:

```
AUCX 2003 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
I: 32F345E2
F: C,N,L,M,LC,P
```

La réponse indique qu'il s'agit d'un succès et inclut les informations pour la commande

RequestedInfo:

```
200 2003 OK
C: A3C47F21456789F0
N: ca@ca1.quelconque.net
L: p:10, a:PCMU
M: sendrecv
P: PS=395, OS=22850, PR=615, OR=30937, PL=7, JI=26, LA=47

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0
```

Dans le deuxième exemple, nous demandons l'audit de RemoteConnectionDescriptor et de LocalConnectionDescriptor:

```
AUCX 1203 aaln/2@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
I: FDE234C8
F: RC,LC
```

La réponse indique qu'il s'agit d'un succès et inclut les informations pour la commande RequestedInfo: dans le cas présent, aucun RemoteConnectionDescriptor n'existe, donc le seul champ version de protocole est inclus pour le RemoteConnectionDescriptor:

```
200 1203 OK

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0

v=0
```

RestartInProgress

Le premier exemple illustre un message RestartInProgress envoyé par un client intégré pour informer l'agent d'appel que le point d'extrémité spécifié sera mis hors service dans 300 secondes:

```
RSIP 1200 aaln/1@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
RM: graceful
RD: 300
```

La réponse de l'agent d'appel indique que la transaction a été couronnée de succès:

```
200 1200 OK
```

Dans le deuxième exemple, le message RestartInProgress envoyé par le client intégré informe l'agent d'appel que tous les points d'extrémité du client intégré vont être mis en service dans 0 seconde, c'est-à-dire qu'ils sont remis en service. Le retard aurait pu être également omis:

```
RSIP 1204 *@rgw-2567.quelconque.net MGCP 1.0 NCS 1.0
RM: restart
RD: 0
```

La réponse de l'agent d'appel indique un succès et fournit en outre une nouvelle "entité avisée" aux points d'extrémité en question:

```
200 1204 OK
N: CA-1@quelconque.net
```

En variante, la commande aurait pu avoir échoué avec une nouvelle "entité avisée" comme dans:

```
521 1204 OK
N: CA-1@quelconque.net
```

Dans ce cas, la commande aurait alors été réessayée afin de satisfaire à la "procédure de redémarrage" (voir 6.4.3.5), allant cette fois vers l'agent d'appel "`CA-1@quelconque.net`".

APPENDICE III

Exemple de flux d'appels

Dans le présent appendice, un exemple de flux d'appels entre deux clients est présenté, EC-1 et EC-2. Il convient de noter que ce flux d'appels, même s'il est valide, est simplement un exemple qui peut ou non être utilisé dans la pratique. En outre, le flux d'appels utilise le paquet ligne exemple.

Dans le flux d'appels ci-après, l'abréviation CA se rapporte à l'agent d'appel, l'abréviation CDB à une base de données de configurations tandis que l'abréviation ACC se rapporte à une base de données de comptabilisation.

Usr-1	EC-1	CA	CDB	ACC	EC-2	Usr-2
	←	Notification Request (demande de notification)				
	Ack	→				
Décroché	Notify	→				
	←	Ack				
(Tonalité de numérotation)	←	Create Connection + Notification Request				
	Ack(SDP1)	→				
Chiffres	Notify	→				
	←	Ack				
(en cours)	←	Notification Request				
	Ack	→				
		Query(E.164)	→			
		←	IP			
		Create Connection(SDP1) + Notification Request	– – –	– – –	→	
		←	– – –	– – –	P-Ack(SDP2)	
		←	– – –	– – –	Ack(SDP2)	(sonnerie)
		Ack	– – –	– – –	→	
(retour d'appel)	←	Modify Connection(SDP2) + Notification Request				
	Ack	→				
		←	– – –	– – –	Notify	décroché
		Ack	– – –	– – –	→	
	←	ModifyConnection + Notification Request				
	Ack	→				
	(mis en circuit)	Début d'appel	– – –	→		
		Notification Request	– – –	– – –	→	
		←	– – –	– – –	Ack	
		(Appel établi)				
		←	– – –	– – –	Notify	raccroché
		Ack	– – –	– – –	→	
	←	Delete Connection				

Usr-1	EC-1	CA	CDB	ACC	EC-2	Usr-2
		Delete Connection	– – –	– – –	→	
	Ack (Perf Data)	→				
		←	– – –	– – –	Ack(Perf data)	
		Fin d'appel	– – –	→		
		Notification Request	– – –	– – –	→	
		←	– – –	– – –	Ack	
Raccroché	Notify	→				
	←	Ack				
	←	Notification Request				
	Ack	→				

Pendant ces échanges, l'agent d'appel se sert du profil NCS du protocole MGCP pour commander les deux clients intégrés. L'échange se produit sur deux côtés.

La première commande est une commande NotificationRequest, envoyée par l'agent d'appel au client imbriqué d'entrée. La requête est constituée des lignes suivantes:

```
RQNT 1201 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AB
R: hd
```

A ce stade, le client intégré est chargé de rechercher un événement décrochage, et d'en faire rapport. Il envoie d'abord une réponse à la commande, en répétant dans sa réponse l'identificateur de transaction que l'agent d'appel avait joint à la requête et en fournissant un code de retour indiquant un succès:

```
200 1201 OK
```

Lorsque l'événement décrochage a été noté, le client intégré envoie un message Notify à l'agent d'appel:

```
NTFY 2001 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AB
O: hd
```

L'agent d'appel acquitte immédiatement la notification:

```
200 2001 OK
```

L'agent d'appel examine les services associés à un événement décrochage pour ce point d'extrémité (il pourrait entreprendre des actions spéciales dans le cas d'une ligne directe, d'absence d'abonnement actuel, etc.). Dans la plupart des cas, il envoie une commande combinée CreateConnection et NotificationRequest pour créer une connexion, fournir une tonalité de numérotation et recueillir les chiffres DTMF[35]:

[35] Le script de numérotation effectif dépend du plan de numérotation dans la zone locale ainsi que des services auxquels on s'est abonné. Il convient de considérer le script de numérotation présenté comme seulement un exemple de script de numérotation.

```
CRCX 1202 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
N: ca@ca1.quelconque.net:5678
X: 0123456789AC
R: hu, [0-9#*T](D)
D: (0T | 00T | [2-9]xxxxxx | 1[2-9]xxxxxxxxx | 011xx.T)
S: dl
```

Le client intégré acquitte la transaction, en renvoyant l'identification de la connexion nouvellement créée et la description de session utilisée pour recevoir les données audio:

```
200 1202 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

La spécification du protocole SDP, dans notre exemple, spécifie l'adresse à laquelle le client intégré est prêt à recevoir les données audio (128.96.41.1), le protocole de transport (RTP), le port RTP (3456) et le profil audio (AVP). Le profil audio se rapporte à la RFC 1890, qui définit que le type de charge utile 0 a été affectée pour la transmission selon G.711 μ-law.

Le client intégré commence à accumuler les chiffres conformément au script de numérotation. Lorsque la correspondance avec le script de numérotation est obtenue, le client intégré avisera l'agent d'appel des événements observés:

```
NTFY 2002 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
N: ca@ca1.quelconque.net:5678
X: 0123456789AC
O: 1,2,0,1,8,2,9,4,2,6,6
```

L'agent d'appel acquitte immédiatement cette notification:

```
200 2002 OK
```

A ce stade, l'agent d'appel envoie une commande NotificationRequest pour arrêter la collecte de chiffres mais en continuant la surveillance pour détecter une transition de raccrochage. En outre, l'agent d'appel décide d'accuser réception des réponses pour la transaction 1202:

```
RQNT 1203 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
K: 1202
X: 0123456789AD
R: hu
```

Le client intégré acquitte immédiatement cette commande.

```
200 1203 OK
```

L'agent d'appel doit maintenant créer une connexion sur le client intégré de sortie, EC-2, et également faire sonner le combiné rattaché au client intégré. Il le fait en envoyant au client intégré une commande combinée CreateConnection et NotificationRequest:

```
CRCX 2001 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: sendrecv
X: 0123456789B0
R: hd
S: rg
```

```
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

A ce stade, le client intégré de sortie est chargé de faire sonner le combiné et de rechercher un événement décrochage et d'en faire rapport. L'événement décrochage et le signal de sonnerie sont synchronisés, de sorte que la sonnerie s'arrête si l'événement décrochage se produit. La partie créer une connexion de la commande a les mêmes paramètres que la commande envoyée au client intégré d'entrée, à deux différences près:

- l'identificateur de point d'extrémité pointe vers le circuit de départ;
- le message transporte la description de session renvoyée par le client intégré d'entrée;
- étant donné que la description de session est présente, le paramètre "mode" est mis à "send/receive" (*envoyer/recevoir*).

Nous observons que l'identificateur d'appel est identique pour les deux connexions, ce qui est normal car les deux connexions appartiennent au même appel.

Nous supposons que cette commande ne finit pas de s'exécuter immédiatement[36], et que donc une réponse provisoire est renvoyée par le client intégré de sortie pour acquitter la commande, envoyant dans la description de session ses propres paramètres tels qu'adresse, ports et profil RTP ainsi que l'identificateur de la nouvelle connexion:

```
100 2001 Pending
I: 32F345E2

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
```

Une fois que la transaction finit de s'exécuter, le client intégré envoie la réponse finale à l'agent d'appel, répétant l'information qu'il a fournie dans la réponse provisoire:

```
200 2001 OK
K:
I: 32F345E2

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
```

Lorsque l'agent d'appel reçoit la réponse finale, il remarque la présence de l'attribut vide acquittement de réponse et émet donc un acquittement de réponse pour la transaction:

```
000 2001
```

[36] Cela pourrait par exemple être dû à une réservation de ressource extérieure, même si ce n'est pas inclus dans notre exemple.

L'agent d'appel relaye l'information au client intégré d'entrée et le charge de générer des tonalités de retour d'appel locales, en utilisant une commande combinée ModifyConnection et NotificationRequest:

```
MDCX 1204 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: hu
S: rt@FDE234C8

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1297 RTP/AVP 0
```

Le client intégré acquitte immédiatement la modification:

```
200 1204 OK
```

A ce stade, l'agent d'appel a établi un chemin de transmission semi duplex. Le combiné rattaché au client intégré d'entrée sera capable de recevoir les signaux (comme par exemple des tonalités ou des annonces) qui peuvent être générés en cas d'erreurs quelconques, ainsi que les paroles initiales qui seront très vraisemblablement émises lorsque l'usager de sortie répondra au téléphone.

Lorsque l'événement décrochage sera observé, le client intégré de sortie enverra à l'agent d'appel un message Notify:

```
NTFY 3001 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789B0
O: hd
```

L'agent d'appel acquitte immédiatement cette notification.

```
200 3001 OK
```

L'agent d'appel envoie maintenant au client intégré d'entrée une commande combinée ModifyConnection et NotificationRequest, afin de placer la connexion dans le mode "send/receive" (*envoyer/recevoir*) et arrêter les tonalités de retour d'appel:

```
MDCX 1206 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8
M: sendrecv
X: 0123456789AF
R: hu
```

Le client intégré répond immédiatement à la commande:

```
200 1206 OK
```

En parallèle, l'agent d'appel demande au client intégré de sortie d'aviser de l'occurrence d'un événement raccrochage, ce qu'il fait en envoyant une NotificationRequest au client intégré[37]:

```
RQNT 2002 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789B1
R: hu
```

Le client intégré répond immédiatement à la commande:

```
200 2002 OK
```

A ce stade, l'appel est complètement établi.

A un certain moment ultérieur, le combiné rattaché au client intégré de sortie, dans notre scénario, raccroche. Cet événement est notifié à l'agent d'appel, conformément à la politique reçue dans la dernière NotificationRequest en envoyant une commande Notify:

```
NTFY 2003 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789B1
O: hu
```

L'agent d'appel répond immédiatement à la commande:

```
200 2003 OK
```

L'agent d'appel détermine alors que l'appel se termine et, donc, il envoie une commande DeleteConnection aux deux clients intégrés:

```
DLCX 1207 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: FDE234C8

DLCX 2004 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
C: A3C47F21456789F0
I: 32F345E2
```

Les clients intégrés répondent par des acquittements qui comprennent les paramètres de connexion pour la connexion:

```
250 1207 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
250 2004 OK
P: PS=790, OS=45700, PR=1230, OR=61875, PL=15, JI=27, LA=48
```

L'agent d'appel envoie une nouvelle commande NotificationRequest au client intégré de sortie, afin d'être prêt à recevoir le prochain événement décrochage détecté par le client intégré:

```
RQNT 2005 aaln/1@ec-2.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789B2
R: hd
```

Le client intégré acquitte ce message:

```
200 2005 OK
```

[37] Il convient de noter que, bien que le raccrochage soit un événement persistant, le mode "lockstep" (verrouillé) nécessite que l'agent d'appel envoie une nouvelle commande NotificationRequest au client intégré.

Enfin, le client intégré d'entrée raccroche le combiné, créant ainsi un message Notify envoyé à l'agent d'appel:

```
NTFY 1208 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789AF
O: hu
```

L'agent d'appel répond immédiatement à la commande:

```
200 1208 OK
```

L'agent d'appel envoie alors une nouvelle commande NotificationRequest au client intégré d'entrée, afin d'être prêt à recevoir le prochain événement décrochage détecté par le client intégré:

```
RQNT 1209 aaln/1@ec-1.quelconque.net MGCP 1.0 NCS 1.0
X: 0123456789B3
R: hd
```

Le client intégré acquitte ce message:

```
200 1209 OK
```

A ce stade, les deux clients intégrés sont prêts pour le prochain appel.

APPENDICE IV

Interactions de modes

Une connexion MGCP peut établir un ou plusieurs flux médias. Ces flux sont soit entrants (depuis un point d'extrémité distant) ou sortants (créés au niveau du microphone du combiné). Le paramètre "connection mode" (*mode de connexion*) établit le sens et la création de ces flux. Lorsqu'il n'y a qu'une seule connexion à un point d'extrémité, le mappage de ces flux est simple; le combiné lit le flux entrant sur le haut-parleur et crée le flux sortant du signal du microphone du combiné, en fonction du paramètre mode.

Par contre, lorsque plusieurs connexions sont établies à un point d'extrémité, il peut y avoir de nombreux flux entrants et sortants. En fonction du mode de connexion utilisé, ces flux peuvent interagir différemment les uns avec les autres et avec les flux partant du combiné/y arrivant.

Le tableau ci-dessous décrit comment il convient de mélanger des connexions différentes lorsqu'une ou plusieurs connexions sont "actives" simultanément. Une connexion active est définie ici comme une connexion qui se trouve dans l'un des modes suivants:

- "send/receive" (*envoyer/recevoir*);
- "send only" (*envoyer uniquement*);
- "receive only" (*recevoir uniquement*);
- "replicate" (*répliquer*);
- "conference" (*conférence*).

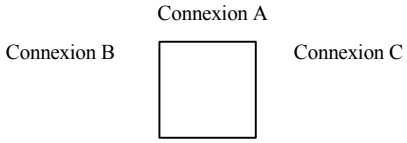
Les connexions se trouvant dans les modes "network loopback" (*boucle réseau*), "network continuity test" (*test de continuité réseau*) ou "inactive" (*inactive*) ne sont pas touchées par les connexions se trouvant dans les modes "actives". Le tableau utilise les conventions suivantes:

- A_{in} est le flux média entrant en provenance de la connexion A;
- B_{in} est le flux média entrant en provenance de la connexion B;
- H_{in} est le flux média entrant en provenance du microphone du combiné;
- A_{out} est le flux média sortant vers la connexion A;
- B_{out} est le flux média sortant vers la connexion B;
- H_{out} est le flux média sortant vers l'écouteur du combiné;
- NA indique l'absence de tout flux.

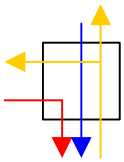
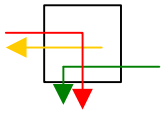
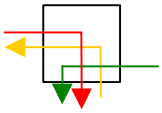
		Mode de connexion A						
		sendonly	**recvonly**	**sendrecv**	**confrnce**	**inactive**	**netwloop/ netwtest**	**replcate**
Mode de connexion B	**sendonly**	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=A_{in}$	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}$	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=NA$	$A_{out}=A_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$
	recvonly		$A_{out}=NA$ $B_{out}=NA$ $H_{out}=A_{in}+B_{in}$	$A_{out}=H_{in}$ $B_{out}=NA$ $H_{out}=A_{in}+B_{in}$	$A_{out}=H_{in}$ $B_{out}=NA$ $H_{out}=A_{in}+B_{in}$	$A_{out}=NA$ $B_{out}=NA$ $H_{out}=B_{in}$	$A_{out}=A_{in}$ $B_{out}=NA$ $H_{out}=B_{in}$	$A_{out}=H_{in}+B_{in}$ $B_{out}=NA$ $H_{out}=B_{in}$
	sendrecv			$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}+B_{in}$	$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=A_{in}+B_{in}$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=B_{in}$	$A_{out}=A_{in}$ $B_{out}=H_{in}$ $H_{out}=B_{in}$	$A_{out}=H_{in}+B_{in}$ $B_{out}=H_{in}$ $H_{out}=B_{in}$
	confrnce				$A_{out}=H_{in}+B_{in}$ $B_{out}=H_{in}+A_{in}$ $H_{out}=A_{in}+B_{in}$	$A_{out}=NA$ $B_{out}=H_{in}$ $H_{out}=B_{in}$	$A_{out}=A_{in}$ $B_{out}=H_{in}$ $H_{out}=B_{in}$	$A_{out}=H_{in}+B_{in}$ $B_{out}=H_{in}$ $H_{out}=B_{in}$
	inactive					$A_{out}=NA$ $B_{out}=NA$ $H_{out}=NA$	$A_{out}=A_{in}$ $B_{out}=NA$ $H_{out}=NA$	$A_{out}=H_{in}$ $B_{out}=NA$ $H_{out}=NA$
	netwloop/ netwtest						$A_{out}=A_{in}$ $B_{out}=B_{in}$ $H_{out}=NA$	$A_{out}=H_{in}$ $B_{out}=B_{in}$ $H_{out}=NA$
	replcate							$A_{out}=H_{in}$ $B_{out}=H_{in}$ $H_{out}=NA$

S'il y a au moins trois voies "actives", elles vont encore interagir comme défini dans le tableau ci-dessus avec les flux médias sortants mélangés pour chaque interaction. (Union de tous les flux) si les ressources internes ont été épuisées alors que les flux ne peuvent pas être mélangés, il convient que la passerelle renvoie une erreur ressources non disponibles.

Ces connexions peuvent être représentées graphiquement comme suit:



Par exemple, si la connexion A est dans le mode "Sendrecv", la connexion B dans le mode "confrnce" et la connexion C dans le mode "recvonly", les valeurs de sortie de chaque mode seront, d'après le tableau précédent:

Interaction A à B:	$B_{out}=H_{in}$	$A_{out}=H_{in}$	$H_{out}=A_{in}+B_{in}$	
Interaction A à C:	$A_{out}=H_{in}$	$C_{out}=NA$	$H_{out}=A_{in}+C_{in}$	
Interaction B à C:	$B_{out}=H_{in}$	$C_{out}=NA$	$H_{out}=B_{in}+C_{in}$	

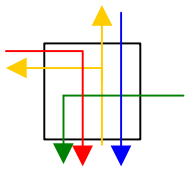
En prenant l'union de tous les flux dans chaque valeur de sortie, nous obtenons:

$A_{out}=H_{in}$

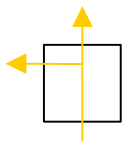
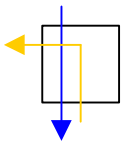
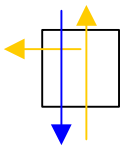
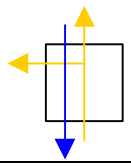
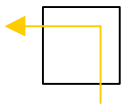
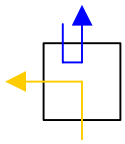
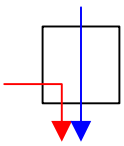
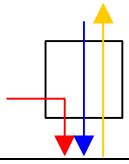
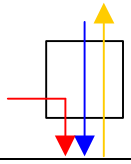
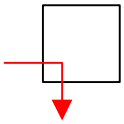
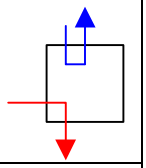
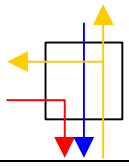
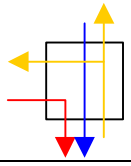
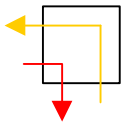
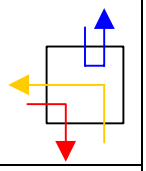
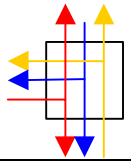
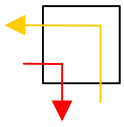
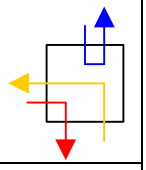
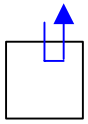
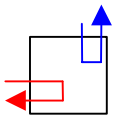
$B_{out}=H_{in}$

$C_{out}=NA$

$H_{out}=B_{in}+A_{in}+C_{in}$



Par souci de clarté, le tableau décrit ci-dessus est repris ci-dessous sous une forme graphique (en excluant le mode "replicate"):

		Mode de connexion A (en haut)					
		sendonly	**recvonly**	**sendrecv**	**confrnce**	**inactive**	**netwloop/ netwtest**
Mode de connexion B (à gauche)	**sendonly**						
	recvonly						
	sendrecv						
	confrnce						
	inactive						
	netwloop/ netwtest						

APPENDICE V

Informations de compatibilité

Le présent appendice fournit des informations de compatibilité au protocole de signalisation NCS.

Compatibilité avec le protocole MGCP

La signalisation NCS est un profil du protocole MGCP 1.0, mais elle a également introduit quelques ajouts. La liste suivante énumère les ajouts par la signalisation NCS qui ne sont pas actuellement inclus dans le protocole MGCP:

- **Endpoint Naming Scheme** (*système de nommage des points d'extrémité*) – Les règles de remplacement par caractères génériques sont plus restrictives que dans le protocole MGCP.
- **Embedded ModifyConnection** (*ModifyConnection intégrée*) – Une nouvelle action ModifyConnection intégrée a été introduite.

- **Dynamic Quality of Service** (*qualité de service dynamique*) – Les services sécurité de l'IPCablecom sont disponibles dans la signalisation NCS. Cela affecte les paramètres LocalConnectionOptions, Capabilities, et le protocole SDP. De plus, un nouveau paramètre ResourceID est ajouté pour les commandes CreateConnection et ModifyConnection.
- **Security** (*sécurité*) – Les services de sécurité IPCablecom sont pris en charge dans la signalisation NCS. Cela affecte les paramètres LocalConnectionOptions, Capabilities, et le protocole SDP.
- **Endpoint Name Retrieval** (*récupération du nom de point d'extrémité*) – La commande AuditEndpoint a été étendue avec une capacité permettant de renvoyer le nombre de points d'extrémité qui correspondent à un joker ainsi qu'avec un mécanisme permettant de récupérer, sous forme de blocs, ces noms de points d'extrémité. En plus de cette extension de la commande AuditEndpoint, il en découle l'introduction de deux noms de paramètres: MaxEndPointIds et NumEndPoints.
- **Supported Versions** (*versions prises en charge*) – La réponse RestartInProgress et la commande AuditEndpoint ont été étendues avec un paramètre VersionSupported afin de permettre aux agents d'appel et aux passerelles de déterminer quelles versions de protocole ils prennent en charge.
- **Error Codes** (*codes d'erreur*) – Deux nouveaux codes d'erreur ont été introduits: 532 et 533.
- **Usage of SDP** (*utilisation du protocole SDP*) – Un nouveau profil d'utilisation du protocole SDP est inclus dans la signalisation NCS. De manière plus remarquable, l'utilisation de profil et de tous les exemples exige spécifiquement une conformité stricte au protocole SDP, quelle que soit l'utilité des champs inclus. En outre, des extensions spécifiques à l'IPCablecom ont été ajoutées au protocole SDP.
- **Provisional Response** (*réponse provisoire*) – Des détails et une spécification supplémentaires du mécanisme de réponse provisoire ont été inclus dans la signalisation NCS. Une réponse acquittement de réponse (000) a été introduite, un paramètre ResponseAck vide a été autorisé dans les réponses finales qui suivent les réponses provisoires, et une procédure pour le mécanisme a été spécifiée.
- **Signal Parameters** (*paramètres de signal*) – La syntaxe des paramètres de signal a été étendue pour prévoir l'utilisation de parenthèses équilibrées dans les paramètres de signaux. La valeur de temporisation de tous les signaux temporisés peut être modifiée par un paramètre de signal.
- **Event Packages** (*paquets d'événements*) – La signalisation NCS introduit un jeu de nouveaux paquets d'événements.

Enfin, il convient de noter que la signalisation NCS fournit des interprétations et, dans un certain nombre de cas, une spécification ou clarification supplémentaire du comportement du protocole MGCP de base qui peut ou non refléter le comportement MGCP voulu.

APPENDICE VI

Autres exemples de paquets d'événements

Le présent appendice définit des paquets d'événements exemples supplémentaires pour les différents types de points d'extrémité actuellement définis pour les clients intégrés.

Lignes d'accès analogique

Le paquet suivant est actuellement défini pour les points d'extrémité des lignes d'accès analogique:

- ligne japonaise;
- ADSI.

Paquets ligne japonaise

Nom du paquet: J

Les codes ci-après servent à identifier des événements et signaux pour le paquet "ligne japonaise" pour les "lignes d'accès analogique":

1) *types de signalisation des lignes d'abonnés*

   Les signaux de lignes d'abonné (signaux) peuvent être classés en signaux liés à la commande de la connexion (signal de surveillance, *supervisory signal*), en signaux liés à la commande de sélection (signaux de sélection) et en signaux de tonalités à fréquence acoustique (tonalités à fréquence acoustique);

2) *signaux de surveillance*

Code	Nom de signal	Evénement	Signal	Autres informations
cs	Signal appelant (*Calling signal*)	P, S	–	Notification de l'appel d'origine (=transition de décrochage)
ir	Signal de sonnerie	–	TO	Notification d'appel entrant temporisation = infinie Voir l'Article 31, élément 2 dans les "*Carriers Telecommunication Facilities Regulations*"
as1	Signal de réponse 1	P, S	–	Notification que le terminal appelé a répondu (terminal vers réseau) (=transition de décrochage)
as2	Signal de réponse 2	–	TO	Notification que le terminal appelé a répondu (réseau vers terminal) temporisation = infinie
ds1	Signal de déconnexion 1	P, S	–	Notification que la communication s'est achevée (terminal vers réseau) (=transition de raccrochage)
ds2	Signal de déconnexion 2	–	TO	Notification que le terminal d'origine a terminé la communication (réseau vers terminal) temporisation = infinie

Code	Nom de signal	Evénement	Signal	Autres informations
cbs	Signal de raccrochage (*Clear back signal*)	P, S	–	Notification que le terminal appelé a terminé la communication (=transition de raccrochage)
hs	Signal de crochet (*Hooking signal*)	P	–	Pour "appel en attente" et "service à trois"
sir	Signal d'appel d'extension (*Extension call signal*)	–	TO	Emis par le système d'extensions centralisé (CES, *centralized extension system*). Temporisation = infinie
tir	Signal avertissement de renvoi d'appel (*Callforward warning signal*)	–	TO	Pour service "Warp vocal" temporisé = 2 à 3 secondes
car	Signal d'activation du terminal récepteur de données	–	TO	Notification par signal MODEM temporisation = infinie
pas	Signal de réponse primaire	P, S	–	Pour l'affichage du numéro (=transition de décrochage)
iss	Signal réussi entrant	P, S	–	Pour l'affichage du numéro (=transition de raccrochage)
cei1 (nu)	Callee ID (PB tone) *id de l'appelé (tonalité PB)*	–	BR	"nu" désigne un numéro
cei2 (nu)	Callee ID (Modem tone) *id de l'appelé (tonalité modem*)	–	BR	"nu" désigne un numéro
ci	Identité de l'appelant (*caller ID*)	–	BR	"nu" désigne un numéro
aw	Tonalité de réponse (*answer tone*)	✓	–	
ft	Tonalité de télécopieur (*fax tone*)	✓	–	
mt	Tonalité de modem (*modem tone*)	✓	–	
ma	Début de média (*media start*)	C	–	
oc	Opération terminée	✓	–	
of	Echec de l'opération	✓	–	
t	Temporisateur	✓	–	
l	DTMF longue durée	✓	–	
ld	Connexion de longue durée	C	–	

3) *Signal de sélection*

Code	Nom de signal	Evénement	Signal	Autres informations
ssn	Signal de sélection (0-9,*,#)	✓	BR	Temporisation de numérotation partielle: = 20-30 secondes Temporisation entre les chiffres = 4-6 secondes
ssw	Joker de tonalités PB	✓	–	Correspond à n'importe quel chiffre de "0-9"

4) *Tonalités à fréquence acoustique*

Code	Nom de signal	Evénement	Signal	Autres informations
dt	Tonalité de numérotation	–	TO	Prêt à recevoir le signal de sélection Temporisation = 20-30 secondes
sdt	Deuxième tonalité de numérotation	–	TO	Pour les services du type enregistrement tels que "Renvoi d'appel", "service de répondeur téléphonique automatique" Temporisation = 20-30 secondes
rbt	Tonalité de retour d'appel	–	C,TO	Temporisation = infinie
bt	Tonalité d'occupation	–	TO	Temporisation = 60-70 secondes
cpt	Tonalité de réception	–	BR	Pour les services du type enregistrement tels que "Renvoi d'appel", "Service de répondeur téléphonique automatique"
hst	Tonalité de service mise en garde	–	TO	Temporisation = infinie
iit	Tonalité d'identification entrante	–	C, BR	Pour le "service de répondeur téléphonique automatique"
siit	Tonalité d'identification entrante spécifique	–	C, BR	En cas de double contrat avec le "service de répondeur téléphonique automatique" et le "service NARIWAKE"
nft	Tonalité de notification	–	TO	Uniquement pour le "service réception de l'identification de message" Temporisation = 3-4 secondes
how1	Tonalité hurleur 1	–	TO	Temporisation = 10-22 secondes
how2	Tonalité hurleur 2	–	TO	Temporisation = infinie

La définition des événements et signaux individuels est comme suit:

Calling signal (cs) (*signal appelant*): avise le réseau d'un appel de départ.

Ringing signal (ir) (*signal de sonnerie*): se reporter Article 31, élément 2 dans les "*Carriers Telecommunication Facilities Regulations*". Le processus de mise en service peut définir une cadence de sonnerie. Le signal de sonnerie peut être paramétré avec le paramètre de signal "rep" qui spécifie le nombre maximal de cycles de sonnerie (répétitions) à appliquer. La ligne ci-après applique un signal de sonnerie pendant un nombre de cycles de sonnerie pouvant atteindre 6:

```
S: ir(rep=6)
```

Le fait d'essayer de faire sonner un téléphone qui est décroché est considéré comme étant une erreur et par conséquent il convient qu'une erreur soit renvoyée si des telles tentatives sont effectuées.

Answer signal (as) (*signal de réponse*): indique au réseau que le terminal appelé a répondu (as1). Dans le sens inverse, le réseau indique au terminal de départ que le terminal appelé a répondu (as2).

Disconnect signal (ds) (*signal de déconnexion*): le terminal de départ indique au réseau que la communication est achevée (ds1). Dans le sens inverse, le réseau indique au terminal appelé que le terminal de départ appelé a terminé la communication (ds2).

Clear back signal (cbs) (*signal de raccrochage*): indique au réseau que le terminal appelé a terminé la communication.

Hooking signal (hs) (*signal de crochet*): le terminal avise le réseau d'une affectation ou lui indique qu'un service a été modifié pendant la communication. Ce signal est utilisé pour "l'appel en attente" et "le service à trois".

Extension call signal (sir) (*signal d'appel d'extension*): avec un téléphone CES (*centralized extension system*), le réseau indique au terminal qu'un appel entrant est en cours de transfert. En outre, pour le "service NARIWAKE", le réseau informe le terminal qu'il y a un appel entrant provenant d'un correspondant qui souhaite être identifié.

Call forward warning signal (tir) (*signal avertissement de renvoi d'appel*): au cours du démarrage du service "Transfert de téléphone" ou pendant le mode de transfert inconditionnel dans le service "Warp vocal", le réseau indique au terminal qu'il y a un appel entrant vers l'abonné et que le transfert d'appel a été activé.

Data receiving terminal activation signal (car) (*signal d'activation du terminal récepteur de données*): le réseau indique à un terminal récepteur de données qu'il y a un appel entrant comportant des informations communiquées par un signal de modem.

Primary answer signal (pas) (*signal de réponse primaire*): le terminal appelé indique au réseau que le combiné téléphonique est décroché. Cette fonction est utilisée pour l'affichage du numéro.

Incoming successful signal (iss) (*signal réussi entrant*): le réseau indique au terminal de départ qu'un signal entrant est reçu avec succès. Cette fonction est utilisée pour l'affichage du numéro.

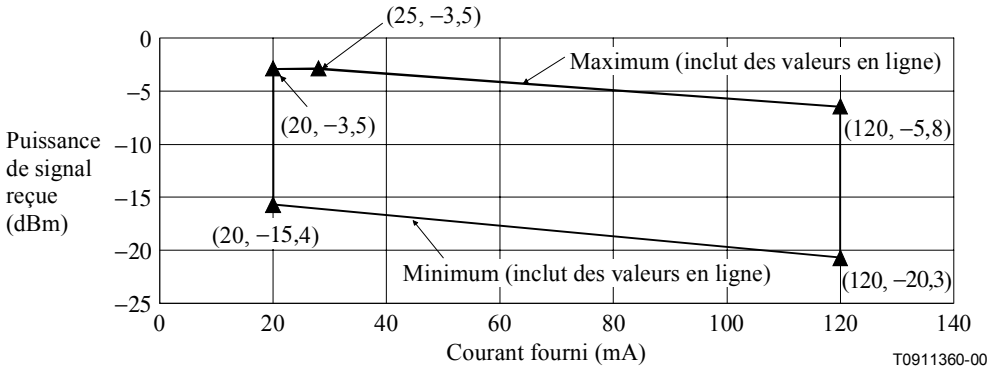
Selection signal (ss) (*signal de sélection*): le terminal de départ avise le réseau du type de service et du numéro de l'autre correspondant. Un code est affecté au Signal de sélection (0-9, *, #) sous la forme ssn et, pour le joker de tonalités PB, sous la forme ssw. Les tableaux et figures ci-après indiquent des fréquences et niveaux de réception pour les signaux de numérotation par boutons-poussoirs (PB, *push button*).

1) *Fréquence*

Hautes fréquences de groupe / Basses fréquences de groupe	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

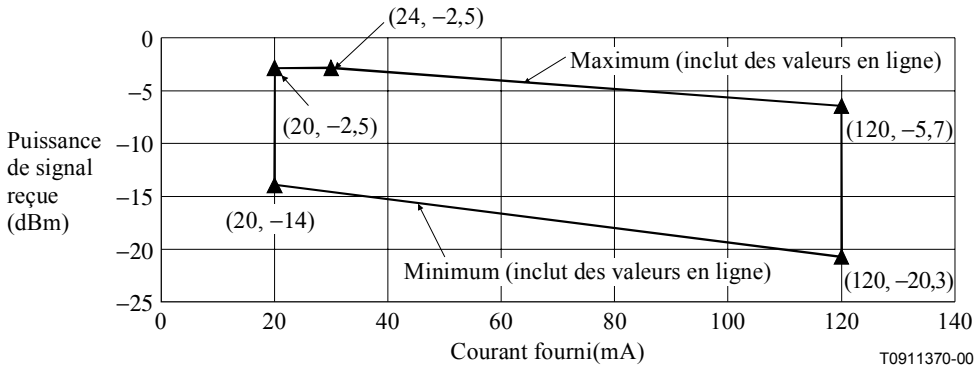
2) *Norme d'écoute*

Elément		Norme
Ecart de la fréquence du signal		A ±1,5 % près
Plage de tolérance pour la puissance de réception du signal	Basses fréquences de groupe	Voir la Figure VI.1
	Hautes fréquences de groupe	Voir la Figure VI.2
	Ecart de puissance électrique entre deux fréquences	A 5 dB près, mais il convient que la puissance électrique pour la basse fréquence de groupe soit inférieure à celle pour la haute fréquence de groupe.
Durée de sortie du signal		50 ms ou plus
Pause minimale		30 ms ou plus
Cycle		120 ms ou plus
NOTE 1 – La pause minimale est le temps mort le plus court entre des signaux adjacents. NOTE 2 – Un cycle est égal à la somme du temps d'envoi du signal et de la pause minimale.		



NOTE – Il convient que la puissance de signal reçue soit comprise entre −15,4 dBm et −3,5 dBm lorsque le courant fourni est inférieur à 20 mA , et comprise entre −20,3 dBm et −5,8 dBm lorsqu'il est supérieur à 120 mA.

Figure VI.1/J.162 – Plage de tolérance pour la puissance de réception du signal (basse fréquence de groupe)



NOTE – Il convient que la puissance de signal reçue soit comprise entre –14 dBm et –2,5 dBm lorsque le courant fourni est inférieur à 20 mA , et comprise entre –20,3 dBm et –5,7 dBm lorsqu'il est supérieur à 120 mA.

Figure VI.2/J.162 – Plage de tolérance pour la puissance de réception du signal (basse fréquence de groupe)

D'autres conditions sont stipulées dans l'Ordonnance 13 du Ministère des postes et télécommunications, 1998.

Dial tone (dt) (*tonalité de numérotation*): le réseau informe le terminal de départ qu'il est prêt à recevoir le signal de sélection. Dans un appel hors réseau à partir d'un combiné téléphonique réseau d'un membre, le réseau informe le terminal de départ qu'il est prêt à recevoir le signal de sélection. La tonalité de numérotation est une tonalité AC avec une fréquence de 400 Hz et des niveaux compris entre (–22–L) et –19 dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz.

Second dial tone (sdt) (*deuxième tonalité de numérotation*): le réseau informe le terminal de départ qu'il est prêt à recevoir un deuxième signal de sélection. Dans un appel hors réseau à partir d'un combiné téléphonique réseau d'un membre, le réseau informe le terminal de départ qu'il est prêt à recevoir le signal de sélection. La deuxième tonalité de numérotation est une tonalité AC avec une fréquence de 400 Hz et des niveaux compris entre (–22–L) et –19 dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz. Le taux de coupure-établissement et le taux d'établissement sont respectivement dans la limite de 240 IPM et 50%.

Ringing back tone (rbt) (*tonalité de retour d'appel*): le réseau indique au terminal de départ qu'il est en train d'appeler le terminal récepteur. La tonalité s'arrête lorsqu'un signal de réponse est reçu du terminal appelé. La tonalité à fréquence acoustique de retour d'appel est une combinaison de deux tonalités AC avec des fréquences de 400 et 15-20 Hertz et des niveaux compris entre –4 et (–29–L) dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz. Le taux de coupure-établissement et le taux d'établissement sont respectivement dans la limite de 20 IPM ± 20% et 33 ± 10% (taux de modulation: dans la limite de 85 ± 15%).

Busy tone (bt) (*tonalité d'occupation*): le réseau informe le terminal d'origine que le terminal récepteur est dans l'état communication et que donc il ne peut pas assurer le service ou la connexion que le terminal d'origine a demandé(e). La tonalité d'occupation est une tonalité AC avec une fréquence de 400 Hz et des niveaux compris entre (–29–L) et –4 dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz. Le taux de coupure-établissement et le taux d'établissement sont respectivement dans la limite de 60 IPM ± 20% et 50 ± 10%.

Acceptance tone (cpt) (*tonalité de réception*): le réseau indique au terminal de départ qu'il a reçu le service demandé. La tonalité de réception est une tonalité AC avec une fréquence de 400 Hz et des niveaux compris entre (–26–L) et –16 dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz.

Hold service tone (hst) (*tonalité de service mise en garde*): le réseau informe le terminal en attente que l'état d'attente se poursuit. La tonalité à fréquence acoustique de service mise en garde est une combinaison de deux tonalités AC avec des fréquences de 400 et 16 Hertz et des niveaux compris entre –14 et (–22–L) dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz (taux de modulation: dans la limite de 85%).

Incoming identification tone (iit) (*tonalité d'identification entrante*): le réseau informe le terminal appelé concerné qu'il a reçu un appel entrant émis par un tiers au cours de la conversation avec un deuxième correspondant. La tonalité à fréquence acoustique d'identification entrante est une combinaison de deux tonalités AC avec des fréquences de 400 et 16 Hertz et des niveaux compris entre –14 et (–25–L) dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz (taux de modulation: dans la limite de 85%).

Specific incoming identification tone (siit) (*tonalité d'identification entrante spécifique*): le réseau informe le terminal appelé concerné qu'il a reçu un appel entrant émis par un tiers qui a été identifié. La tonalité à fréquence acoustique d'identification entrante spécifique est une combinaison de deux tonalités AC avec des fréquences de 400 et 16 Hertz et des niveaux compris entre –14 et (–25–L) dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz (taux de modulation: dans la limite de 85%).

Notification tone (nft) (*tonalité de notification*): le réseau informe le terminal d'un abonné au service "réception d'identification de message" qu'il a reçu une identification de message. La tonalité de notification est une tonalité AC avec une fréquence de 400 Hz et des niveaux compris entre (–26–L) et –16 dBm où L est la perte de transmission dans une boucle d'abonné à 400 Hz.

Howler tone (how) (*tonalité hurleur*): le réseau informe un terminal qu'un récepteur téléphonique non utilisé a été décroché depuis un certain temps afin d'ordonner que le combiné soit raccroché. Deux tonalités hurleur sont fournies. La tonalité hurleur1 (how1) est une tonalité AC avec une fréquence de 400 Hz et des niveaux inférieurs ou égaux à +35 dBm. La tonalité hurleur1 correspond à un son dont le niveau augmente progressivement pendant 3 à 15 secondes puis à un signal temporisé de 10 à 22 secondes. La tonalité Hurleur2 (how2) est créée en combinant trois tonalités à des fréquences de 1600 Hz , 1000 Hz, 2000 Hz à la cadence de 0,5 seconde de 1600 Hz, en répétant deux fois 0,125 seconde de 1000 Hz et 2000 Hz. Le niveau de la tonalité combinée est inférieur ou égal à –1 dBm. Entre ces tonalités à fréquence acoustique, des indications vocales (comme par exemple "Le récepteur est décroché") sont insérées. Le fait d'essayer de jouer une tonalité hurleur sur un téléphone qui est raccroché est considéré comme étant une erreur et par conséquent il convient qu'une erreur soit renvoyée si de telles tentatives sont effectuées. La tonalité hurleur2 (how2) a une temporisation infinie.

Callee ID [cei1(nu)]: l'accès direct à un poste nécessite l'identificateur de l'appelé (callee ID) dans le système de signalisation PB (par boutons-poussoirs).

Callee ID [cei2(nu)]: l'accès direct à un poste nécessite l'identificateur de l'appelé (callee ID) dans le système de signalisation par modem.

Caller Id [ci(time, number, name)]: chacun des trois champs est facultatif mais chacune des virgules doit toujours être utilisée.

- Le paramètre **time** (*temps*) est codé sous la forme "MM/JJ/HH/MM", où MM est une valeur à deux chiffres pour le Mois comprise entre 01 et 12, JJ une valeur à deux chiffres pour le Jour comprise entre 1 et 31, tandis que Heure et Minute sont des valeurs à deux chiffres codées selon l'heure locale militaire, par exemple: 00 désigne minuit, 01 indique 1 h du matin et 13 désigne 1h de l'après-midi.
- Le paramètre **number** (*numéro*) est codé sous la forme d'une chaîne de caractères ASCII de chiffres décimaux qui identifient le numéro de la ligne appelante. Les espaces, autorisés si la chaîne est placée entre guillemets, sont toutefois ignorés.

- Le paramètre **name** (*nom*) est codé sous la forme d'une chaîne de caractères ASCII qui identifient le nom de la ligne appelante. Des espaces sont autorisés si la chaîne est placée entre guillemets.

Un "P" figurant dans le champ numéro ou nom sert à indiquer un nom ou numéro privé tandis qu'un "O" sert à indiquer un nom ou numéro indisponible. L'exemple suivant illustre l'utilisation du signal identification de l'appelant:

```
S: ci(02/20/19/47, "5273 4671", JCTEA)
```

Answer tone (aw) (*tonalité de réponse*): la tonalité de réponse est une tonalité qui peut être fournie par un modem ou un télécopieur (fax) qui répond à un appel entrant. La tonalité consiste en un signal sinusoïdal 2 100 Hz – voir la Rec. UIT-T V.8.

Fax tone (ft) (*tonalité de fax*): la tonalité de fax est générée à chaque fois qu'un appel par fax est détecté – Se reporter par exemple à la Rec. UIT-T T.30 ou UIT-T V.21.

Media start (ma) (*début de média*): l'événement début de média se produit à la connexion lorsque le premier paquet média RTP valide[38] est reçu sur la connexion. Cet événement peut être utilisé pour synchroniser un signal local (par exemple un retour d'appel) avec l'arrivée d'un média provenant de l'autre correspondant.

L'événement peut être détecté sur une connexion. Lorsque aucune connexion n'est spécifiée, l'événement s'applique à toutes les connexions pour le point d'extrémité, indépendamment du moment où les connexions sont créées.

Modem tones (mt) (*tonalités de modem*): la tonalité de modem est générée à chaque fois qu'un appel par modem est détecté – Se reporter par exemple à la Rec. UIT-T V.8.

Operation complete (oc) (*opération terminée*): l'événement opération terminée est généré lorsqu'on a demandé à la passerelle d'appliquer un ou plusieurs signaux du type TO au point d'extrémité alors qu'un ou plusieurs signaux se sont achevés sans avoir été arrêtés par la détection d'un événement demandé (comme par exemple une transition de décrochage ou la composition d'un chiffre). Le rapport d'achèvement peut comporter comme paramètre le nom du signal qui est parvenu à la fin de sa vie, comme dans

```
O: L/oc(L/dt)
```

Lorsque le signal rapporté a été appliqué à une connexion, le paramètre fourni inclut également le nom de la connexion, comme dans:

```
O: L/oc(L/rbt@0A3F58)
```

Lorsque l'événement opération terminée est demandé, il ne peut pas être paramétré avec un quelconque paramètre d'événement. Lorsque le nom de paquet est omis, c'est le nom de paquet par défaut qui est utilisé.

L'événement opération terminée peut en plus être généré comme défini dans le protocole de base, par exemple lorsqu'une commande Modify Connection intégrée s'achève avec succès, comme dans[39]:

```
O: L/oc(B/C)
```

---

[38] Lorsque des services de sécurité d'autentification et d'intégrité sont utilisés, un paquet RTP n'est pas considéré valide jusqu'à ce qu'il passe avec succès les vérifications de sécurité.

[39] Remarquer l'utilisation de "B" ici comme préfixe pour le paramètre rapporté.

Operation failure (of) (*échec de l'opération*): en général, l'événement échec de l'opération peut être généré lorsqu'on avait demandé à la passerelle d'appliquer un ou plusieurs signaux du type TO au point d'extrémité et qu'un ou plusieurs de ces signaux ont échoué avant d'être temporisés. Le rapport d'achèvement peut comporter comme paramètre le nom du signal qui a échoué, comme dans:

```
O: L/of(L/ir)
```

Lorsque le signal rapporté a été appliqué à une connexion, le paramètre fourni inclut également le nom de la connexion, comme dans:

```
O: L/of(L/rbt@0A3F58)
```

Lorsque l'événement échec de l'opération est demandé, des paramètres d'événement ne peuvent pas être spécifiés. Lorsque le nom de paquet est omis, c'est le nom de paquet par défaut qui est utilisé.

L'événement échec de l'opération peut en plus être généré comme spécifié dans le protocole de base, par exemple lorsqu'une commande Modify Connection intégrée échoue, comme dans[39]:

```
O: L/of(B/C(M(sendrecv(AB2354))))
```

Timer (t) (*temporisateur*): le temporisateur T est un temporisateur provisoire qui ne peut être annulé que par une donnée d'entrée DTMF. Lorsque timer T est utilisée avec l'action "accumulation en fonction du script de numérotation", la temporisation n'est démarrée que lorsque le premier chiffre est saisi, puis elle est redémarrée après chaque saisie d'un chiffre, jusqu'à l'obtention d'une correspondance ou une discordance avec le script de numérotation. Dans ce cas, le temporisateur T fonctionne comme une temporisation entre les chiffres et prend l'une des valeurs T_{par} ou T_{crit}. Lorsque au moins un chiffre de plus est requis pour que la chaîne de chiffres corresponde à l'un des modèles du script de numérotation, le temporisateur T prend la valeur T_{par}, ce qui correspond à une temporisation de numérotation partielle. S'il suffit d'une temporisation pour produire une correspondance, le temporisateur T prend la valeur T_{crit} correspondant à la temporisation critique. Un exemple d'utilisation est:

```
S:   dt
R:   [0-9T](D)
```

Si le temporisateur T est utilisé sans l'action "accumulation en fonction du script de numérotation", le temporisateur T prend la valeur T_{crit}, et la temporisation est immédiatement déclenchée et simplement annulée (et non redémarrée) dès qu'un chiffre est saisi. Dans ce cas, le temporisateur T peut être utilisé comme une temporisation entre les chiffres lors de l'utilisation de l'envoi avec chevauchement, par exemple:

```
R:   [0-9](N), T(N)
```

Il convient de noter que seule l'une des deux formes peut être utilisée à la fois car un événement donné ne peut être spécifié qu'une seule fois.

La valeur par défaut de T_{par} est de 16 secondes, tandis que la valeur par défaut de T_{crit} est de 4 secondes. Le processus de mise en service peut modifier ces deux valeurs.

DTMF Long duration (l) (*DTMF longue durée*): l'événement "DTMF longue durée" est observé lorsqu'un signal DTMF est produit pendant une durée supérieure à deux secondes. Dans ce cas, la passerelle détecte deux événements successifs: en premier lieu, le signal DTMF lorsque le signal a été reconnu et puis, deux secondes après, le signal de longue durée.

Long duration connection (ld) (*connexion de longue durée*): l'événement "connexion de longue durée" est détectée lorsqu'une connexion a été établie pendant une période supérieure à une certaine durée. La valeur par défaut est de 1 heure mais elle peut être modifiée par le processus de mise en service.

L'événement peut être détecté sur une connexion. Lorsque aucune connexion n'est spécifiée, l'événement s'applique à toutes les connexions pour le point d'extrémité, indépendamment du moment où les connexions sont créées.

PB tones wildcard (x) (*joker de tonalités PB*): le joker de tonalités PB correspond à n'importe quel chiffre PB compris entre 0 et 9.

Paquet ADSI

Nom de paquet: JS

Code	Nom de signal	Evénement	Signal	Autres informations
adsi(string)	ADSI display (*affichage ADSI*)	–	BR	

ADSI display [adsi(string)]: l'interface de services d'affichage analogique (ADSI, *analogue display services interface*) est principalement utilisée pour afficher le numéro de téléphone de l'appelant. Voir 4.2, Fonctions de réception pour le numéro de téléphone de l'appelant (Affichage de numéro), Référence technique des interfaces de services téléphoniques (*Receiving Functions for the Originator's Telephone Number (Number Display), Technical Reference of Telephone Service Interfaces*)

Video (Vidéo)

Les paquets d'événements pour la vidéo seront fournis dans une future version de la présente Recommandation.

SÉRIES DES RECOMMANDATIONS UIT-T

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

Imprimé en Suisse
Genève, 2002