Groupe de travail Réseau
Request for Comments : 3435
RFC rendue obsolète : 2705
Catégorie : Information

F. Andreasen, Cisco Systems
B. Foster, Cisco Systems
janvier 2003
Traduction Claude Brière de L'Isle

Protocole de contrôle de passerelle de supports(MGCP) version 1.0

Statut de ce mémoire
Le présent mémoire apporte des informations pour la communauté de l'Internet. Le présent mémoire ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Note de l'IESG :
Le présent document est publié pour l'information de la communauté. Il décrit un protocole qui est actuellement déployé dans un certain nombre de produits. Les utilisateurs devraient savoir que la RFC 3015, qui a été développée dans le groupe de travail Megaco de l'IETF et le groupe d'études 16 de l'UIT-T est considérée par l'IETF et l'UIT-T comme étant la façon normalisée (incluant des considérations de sécurité révisées) de satisfaire les besoins que MGCP vise à satisfaire.

Résumé
Le présent document décrit une interface de programmation d'application et un protocole correspondant (MGCP) qui est utilisé entre les éléments d'une passerelle multimédia décomposée. La passerelle multimédia décomposée consiste en un agent d'appel, qui contient "l'intelligence" de contrôle, et une passerelle de supports qui contient les fonctions des supports, par exemple, la conversion de voix TDM en voix sur IP.

Les passerelles de supports contiennent des points d'extrémité sur lesquels l'agent d'appel peut créer, modifier et supprimer les connexions afin d'établir et contrôler les sessions de supports avec d'autres points d'extrémité multimédia. Aussi, l'agent d'appel peut demander aux points d'extrémité de détecter certains événements et de générer des signaux. Les points d'extrémité communiquent automatiquement les changements de l'état de service à l'agent d'appel. De plus, l'agent d'appel peut inspecter les points d'extrémité ainsi que les connexions sur les points d'extrémité.

Le protocole MGCP de base et général est défini dans le présent document, cependant la plupart des passerelles de supports vont devoir mettre en œuvre un ou plusieurs paquetages MGCP, qui définissent des extensions au protocole convenables pour l'utilisation de types spécifiques de passerelles de supports. De tels paquetages sont définis dans des documents distincts

Table des matières

1. Introduction

Le présent document décrit une interface abstraite de programmation d'application (MGCI) et un protocole correspondant (MGCP) pour le contrôle des passerelles de supports à partir d'éléments externes de contrôle d'appel appelés contrôleurs de passerelle de supports ou agents d'appel. Une passerelle de supports est normalement un élément de réseau qui assure la conversion entre les signaux audio portés sur les circuits de téléphone et les paquets de données portés sur l'Internet ou sur d'autres réseaux de paquets. Des exemples de passerelles de supports sont :

* Les passerelles de jonctions, qui font l'interface entre le réseau téléphonique et un réseau de voix sur IP. De telles passerelles gèrent normalement un grand nombre de circuits numériques.

* Les passerelles de voix sur ATM, qui opèrent de façon assez semblable aux passerelles de jonction avec IP, sauf que leur interface est avec un réseau ATM.

* Les passerelles résidentielles, qui fournissent une interface analogique traditionnelle (RJ11) à un réseau de voix sur IP. Des exemples de passerelles résidentielles incluent des boîtiers de modem câble, des appareils xDSL, et des appareils sans fil haut débit.

* Les passerelles d'accès, qui fournissent une interface analogique traditionnelle (RJ11) ou de PBX numérique à un réseau de voix sur IP. Des exemples de passerelles d'accès incluent des passerelles de voix sur IP à petite échelle.

* Les passerelles d'affaires, qui fournissent une interface numérique de PBX traditionnel ou un interface intégrée de "PBX mou" à un réseau de voix sur IP.

* Des serveurs d'accès réseau, qui peuvent rattacher un "modem" à un circuit de téléphone et fournir un accès de données à l'Internet. On s'attend à ce qu'à l'avenir, les mêmes passerelles combinent des services de voix sur IP et des services d'accès réseau.

* Des commutateurs de circuits, ou des commutateurs de paquets, qui peuvent offrir une interface de contrôle à un élément externe de contrôle d'appel.

MGCP suppose une architecture de contrôle d'appel où "l'intelligence" du contrôle d'appel est en dehors des passerelles et est traitée par des éléments externes de contrôle d'appel appelés des agents d'appel. MGCP suppose que ces éléments de contrôle d'appel, ou agents d'appel, vont se synchroniser les uns avec les autres pour envoyer des commandes et réponses cohérentes aux passerelles qu'ils contrôlent. Si cette hypothèse ne tient pas, on devrait s'attendre à un comportement incohérent. MGCP ne définit pas de mécanisme pour synchroniser les agents d'appel. MGCP est, par nature, un protocole maître/esclave, où les passerelles sont supposées exécuter les commandes envoyées par les agents d'appel. Par conséquent, le présent document spécifie en grand détail le comportement attendu des passerelles, mais spécifie seulement les parties d'une mise en œuvre d'agent d'appel, comme la gestion des temporisateurs, qui sont obligatoires pour le bon fonctionnement du protocole.

MGCP suppose un modèle de connexion où les constructions de base sont les points d'extrémité et les connexions. Les points d'extrémité sont les sources et/ou collecteurs de données et peuvent être physiques ou virtuels. Des exemples de points d'extrémité physiques sont :

* Une interface sur une passerelle qui termine une jonction connectée à un commutateur RTPC (par exemple, de classe 5, de classe 4, etc.). Une passerelle qui termine des jonctions est appelée une passerelle de jonctions.

* Une interface sur une passerelle qui termine une connexion analogique à un téléphone, un système clé, un PBX, etc. Une passerelle qui termine des lignes résidentielles RTPC (sur des téléphones) est appelée une passerelle résidentielle.

Un exemple de point d'extrémité virtuel est une source audio dans un serveur de contenu audio. La création de points d'extrémité physiques exige une installation matérielle, tandis que la création de points d'extrémité virtuels peut être faite par un logiciel.

Les connexions peuvent être en point à point ou multi points. Une connexion point à point est une association entre deux points d'extrémité dans le but de transmettre des données entre ces points d'extrémité. Une fois cette association établie pour les deux points d'extrémité, le transfert des données entre ces points d'extrémité peut avoir lieu. Une connexion multi points est établie en connectant le point d'extrémité à une session multi points.

Des connexions peuvent être établies sur plusieurs types de réseaux porteurs, par exemple :

* transmission de paquets audio en utilisant RTP et UDP sur un réseau IP,
* transmission de paquets audio en utilisant AAL2, ou une autre couche d'adaptation, sur un réseau ATM,
* transmission de paquets sur une connexion interne, par exemple le plan arrière TDM ou le bus d'interconnexion d'une passerelle. Ceci est utilisé, en particulier, pour les connexions "en épingle à cheveux", connexions qui se terminent sur une passerelle mais sont immédiatement réacheminées sur le réseau téléphonique.

Pour les connexions en point à point, les points d'extrémité d'une connexion pourraient être dans des passerelles séparées ou dans les mêmes passerelle.

1.1 Relation avec la Recommandation H.323

MGCP est conçu comme un protocole interne au sein d'un système réparti qui apparaît à l'extérieur d'une seule passerelle VoIP. Ce système est composé d'un agent d'appel, qui peut ou non être réparti sur plusieurs plates-formes informatiques, et d'un ensemble de passerelles, incluant au moins une "passerelle de supports" qui effectue la conversion des signaux de supports entre circuits et paquets, et au moins une "passerelle de signalisation" quand on se connecte à un réseau contrôlé par le SS7. Dans une configuration normale, ce système de passerelles réparties fait l'interface d'un côté avec un ou plusieurs commutateurs de téléphonie (c'est-à-dire, de circuits) et de l'autre côté avec des systèmes conformes à H.323, comme indiqué dans le tableau suivant :

Plan fonctionnel	**Commutateur téléphonique**	**Entité de terminaison**	**Systèmes conforme à H.323**
Plan de signalisation	Échanges de signalisation par SS7/ISUP	Agent d'appel	Échanges de signalisation avec l'agent d'appel par H.225/RAS et H.225/Q.931. Possible négociation de canaux logiques et de paramètres de transmission par H.245 avec l'agent d'appel.
		Synchronisation interne par MGCP	
Plan de transport des données de porteur	Connexion par des groupes de circuits à haut débit	Passerelles de téléphonie	Transmission de données VoIP utilisant directement RTP entre la station H.323 et la passerelle.|

Dans le modèle MGCP, les passerelles se concentrent sur la fonction de traduction des signaux audio, tandis que l'agent d'appel traite les fonctions de signalisation et de traitement d'appel. Par conséquent, l'agent d'appel met en œuvre les couches de "signalisation" de la Recommandation H.323, et se présente comme un "portier H.323" ou comme un ou plusieurs "points d'extrémité H.323" aux systèmes H.323.

1.2 Relation avec les normes de l'IETF

Bien que H.323 soit la norme reconnue pour les terminaux VoIP, l'IETF a aussi produit des spécifications pour les autres types d'applications multi-média. Ces autres spécifications incluent :

* le protocole de description de session (SDP), RFC 2327
* le protocole d'annonce de session (SAP), RFC 2974
* le protocole d'initiation de session (SIP), RFC 3261
* le protocole de flux en temps réel (RTSP), RFC 2326.

Ces trois dernières spécifications sont en fait des normes de signalisation de remplacement qui permettent la transmission d'une description de session à une partie intéressée. SAP est utilisé par les gestionnaires de session en diffusion groupée pour distribuer une description de session en diffusion groupée à un grand groupe de receveurs, SIP est utilisé pour inviter un utilisateur individuel à prendre part à une session en point à point ou en envoi individuel, RTSP est utilisé pour faire l'interface d'un serveur qui fournit des données en temps réel. Dans ces trois cas, la description de session est faite selon SDP ; quand de l'audio est transmis, c'est par le protocole de transport en temps réel, RTP.

Les systèmes de passerelle réparties et MGCP vont permettre aux utilisateurs de téléphonie du RTPC d'accéder à des sessions établies en utilisant SAP, SIP ou RTSP. L'agent d'appel assure la conversion de la signalisation, selon le tableau suivant :

Plan fonctionnel	Commutateur téléphonique	Entité de terminaison	Systèmes conformes à l'IETF
Plan de signalisation	Échanges de signalisation SS7/ISUP	Agent d'appel	Échanges de signalisation avec l'agent d'appel par SAP, SIP ou RTSP. Négociation des paramètres de description de session avec SDP (passerelle de téléphonie terminée mais passée via l'agent d'appel de et vers le système conforme IETF)|
		Synchronisation interne par MGCP	
Plan de transport des données de porteur	Connexion par des groupes de circuits à haut débit| Passerelles de téléphonie	Transmission de données VoIP avec RTP, directement entre le système d'extrémité IP distant et la passerelle.	

La norme SDP a un statut pivot dans cette architecture. On va voir dans la description qui suit qu'on l'utilise aussi pour porter les descriptions de session dans MGCP.

1.3 Définitions

Circuit : un canal de communication entre deux systèmes de commutation, par exemple, un DS0 sur une ligne T1 ou E1.

1.4 Conventions utilisées dans le document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Interface de contrôle de passerelle de supports

Les fonctions d'interface assurent le contrôle de connexion et de point d'extrémité. Les deux utilisent les mêmes modèles de système et les mêmes conventions de dénomination.

2.1 Modèle et conventions de dénomination

MGCP suppose un modèle de connexion où les constructions de base sont les points d'extrémité et les connexions. Les connexions sont groupées en appels. Une ou plusieurs connexions peuvent appartenir à un appel. Les connexions et les appels sont établis à l'initiative d'un ou plusieurs agents d'appel.

2.1.1 Types de points d'extrémité

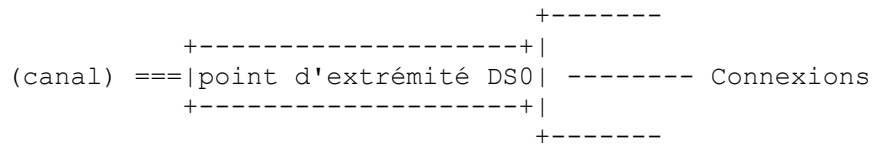
Dans l'introduction, on a présenté plusieurs classes de passerelles. De telles classifications peuvent cependant être trompeuses. Des fabricants peuvent arbitrairement décider de fournir plusieurs types de services dans un seul paquetage. Un seul produit pourrait bien, par exemple, fournir des connexions de circuits aux commutateurs téléphoniques, des connexions au débit primaire et des interfaces de ligne analogique, partageant donc les caractéristiques de ce qui est décrit dans l'introduction comme des passerelles de "jonctions", "d'accès" et "résidentielles". MGCP ne fait pas d'hypothèse sur de tels groupements. On suppose simplement que les passerelles de supports prennent en charge des collections de points d'extrémité. Le type du point d'extrémité détermine ses fonctions. Notre analyse nous a conduit, jusqu'à présent à isoler les types de point d'extrémité de base suivants :

* canal numérique (DS0),
* ligne analogique,
* point d'accès de serveur d'annonces,
* point d'accès de réponse vocale interactive,
* point d'accès de pont de conférence,
* relais de paquet,
* interface ATM "côté circuit".

Dans cette section, on va décrire le comportement attendu de ces points d'extrémité.
Cette liste n'est pas définitive. D'autres types de points d'extrémité pourront être définis à l'avenir, par exemple des points d'extrémité d'essai qui pourraient être utilisés pour vérifier la qualité du réseau, ou des points d'extrémité de relais de trame qui pourraient être utilisés pour gérer des canaux audio multiplexés sur un circuit virtuel en relais de trame.

2.1.1.1 Canaux numériques (DS0)

Les canaux numériques fournissent un service à 64 kbit/s. De tels canaux se trouvent dans les circuits et interfaces RNIS. Ils font normalement partie de multiplex numériques, comme les interfaces T1, E1, T3 ou E3. Les passerelles de supports qui prennent en charge de tels canaux sont capables de traduire les signaux numériques reçus sur le canal, qui peut être codé selon la loi A ou mu, en utilisant soit l'ensemble complet de 8 bits par échantillon ou seulement 7 de ces bits, dans les paquets audio. Quand la passerelle de supports prend aussi en charge un service de serveur d'accès réseau (NAS, *Network Access Server*) la passerelle devra être capable de recevoir des données codées audio (connexion de modem) ou des données binaires (connexion RNIS) et de les convertir en paquets de données.

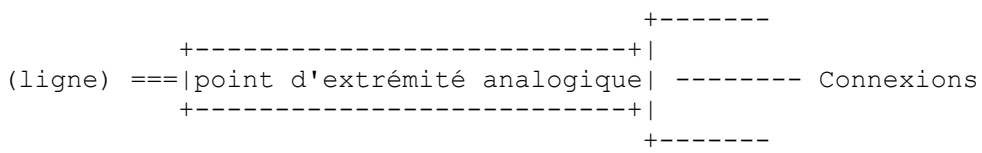


Les passerelles de supports devraient être capables d'établir plusieurs connexions entre le point d'extrémité et les réseaux de paquets, ou entre le point d'extrémité et d'autres points d'extrémité dans la même passerelle. Les signaux originaires de ces connexions devront être mixés en accord avec le "mode" de connexion, comme spécifié plus loin dans le présent document. Le nombre précis de connexions qu'un point d'extrémité prend en charge est une caractéristique de la passerelle, et peut en fait varier selon l'allocation des ressources au sein de la passerelle.

Dans certains cas, les canaux numériques sont utilisés pour porter la signalisation. C'est le cas par exemple pour les liaisons SS7 "F", ou les canaux "D" RNIS. Les passerelles de supports qui prennent en charge ces fonctions de signalisation devront être capables d'envoyer et recevoir les paquets de signalisation de et vers un agent d'appel, en utilisant les procédures de "transport arrière" *(backhaul)* définies par le groupe de travail SIGTRAN de l'IETF. Les canaux numériques sont parfois utilisés en conjonction avec la signalisation associée au canal, comme le "MF R2". Les passerelles de supports qui prennent en charge ces fonctions de signalisation devront être capables de détecter et produire les signaux correspondants, comme par exemple le "clignotement" *(wink)* ou "A", selon les procédures de signalisation et de rapport d'événement définies dans MGCP.

2.1.1.2 Ligne analogique

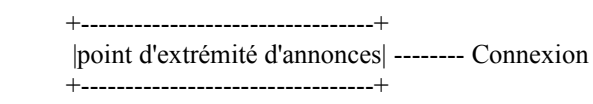
Les lignes analogiques peuvent être utilisées soit comme une interface de "client", fournissant le service à une unité de téléphone classique, soit comme une interface de "service", permettant à la passerelle d'envoyer et recevoir des appels analogiques. Quand la passerelle de supports prend aussi en charge un service de NAS, la passerelle devra être capable de recevoir des données codées en audio (connexion de modem) et de les convertir en paquets de données.



Les passerelles de supports devraient être capables d'établir plusieurs connexions entre le point d'extrémité et les réseaux de paquets, ou entre le point d'extrémité et d'autres points d'extrémité dans la même passerelle. Les signaux audio originaires de ces connexions devront être mixés en accord avec le "mode" de connexion, comme spécifié plus loin dans le présent document. Le nombre précis de connexions qu'un point d'extrémité prend en charge est une caractéristique de la passerelle, et peut en fait varier selon l'allocation de ressources au sein de la passerelle. Une passerelle normale devrait cependant être capable de prendre en charge deux ou trois connexions par point d'extrémité, afin de prendre en charge des services comme "l'appel en attente" ou "l'appel à trois".

2.1.1.3 Point d'accès de serveur d'annonces

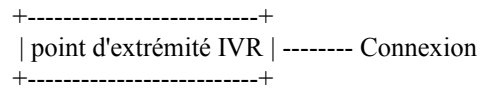
Un point d'extrémité serveur d'annonces donne accès à un service d'annonces. Sur demande de l'agent d'appel, le serveur d'annonces va "exécuter" l'annonce spécifiée. Les demandes de l'agent d'appel vont suivre les procédures de signalement et de rapport d'événement définies dans MGCP.



Un point d'extrémité d'annonces donné n'est pas supposé prendre en charge plus d'une connexion à la fois. Si plusieurs connexions ont été établies avec le même point d'extrémité, alors les mêmes annonces vont être exécutées simultanément sur toutes les connexions. Les connexions à un serveur d'annonces sont normalement unidirectionnelles, ou "semi duplex" -- le serveur d'annonces n'est pas supposé écouter les signaux audio provenant de la connexion.

2.1.1.4 Point d'accès de réponse vocale interactive

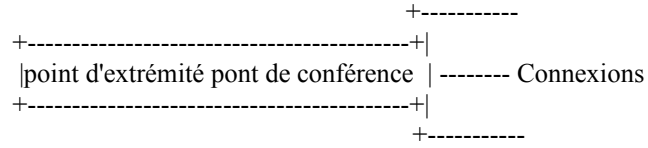
Un point d'extrémité de réponse vocale interactive (IVR, *Interactive Voice Response*) donne accès à un service IVR. Sur demande de l'agent d'appel, le serveur IVR va "exécuter" les annonces et tonalités, et va "écouter" les réponses, comme des entrées DTMF ou des messages vocaux, provenant de l'utilisateur. Les demandes de l'agent d'appel vont suivre les procédures de signalisation et rapport d'événement définies dans MGCP.



Un point d'extrémité IVR n'est pas supposé prendre en charge plus d'une connexion à la fois. Si plusieurs connexions étaient établies avec le même point d'extrémité, alors les mêmes tonalités et annonces seraient exécutées simultanément sur toutes les connexions.

2.1.1.5 Point d'accès de pont de conférence

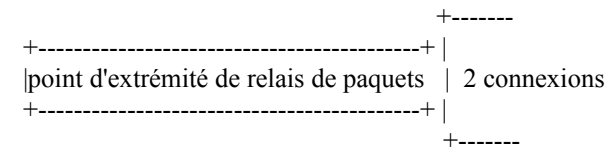
Un point d'extrémité pont de conférence est utilisé pour donner accès à une conférence spécifique.



Les passerelles de supports devraient être capables d'établir plusieurs connexions entre le point d'extrémité et les réseaux de paquets, ou entre le point d'extrémité et d'autres points d'extrémité dans la même passerelle. Les signaux originaires de ces connexions devront être mixés en accord avec le "mode" de connexion, comme spécifié plus loin dans le présent document. Le nombre précis de connexions qu'un point d'extrémité prend en charge est une caractéristique de la passerelle, et peut en fait varier selon l'allocation des ressources au sein de la passerelle.

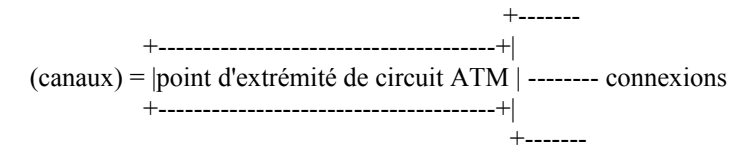
2.1.1.6 Relais de paquet

Un point d'extrémité de relais de paquets est une forme spécifique de pont de conférence, qui normalement ne supporte que deux connexions. Les relais de paquets peuvent se trouver dans des pare-feu entre un réseau protégé et un réseau ouvert, ou dans des serveurs de transcodage utilisés pour assurer l'interopération entre des passerelles incompatibles, par exemple des passerelles qui ne prennent pas en charge des algorithmes de compression compatibles, ou des passerelles qui opèrent sur des réseaux de transmission différents comme IP et ATM.



2.1.1.7 Interface ATM "côté circuit"

Les points d'extrémité ATM "côté circuit" se trouvent normalement quand un ou plusieurs circuits virtuels permanents ATM sont utilisés comme remplacement des commutateurs classiques qui relient des circuits "TDM". Quand ATM/AAL2 est utilisé, plusieurs circuits ou canaux sont multiplexés sur un seul circuit virtuel; chacun de ces circuits correspondant à un seul point d'extrémité.



Les passerelles de supports devraient être capables d'établir plusieurs connexions entre le point d'extrémité et les réseaux de paquets, ou entre le point d'extrémité et d'autres points d'extrémité dans la même passerelle. Les signaux générés par ces connexions devront être mixés en accord avec le "mode" de connexion, comme spécifié plus loin dans le présent document. Le nombre précis de connexions qu'un point d'extrémité supporte est une caractéristique de la passerelle, et peut en fait varier selon l'allocation des ressources au sein de la passerelle.

2.1.2 Identifiants de points d'extrémité

Les identifiants de point d'extrémité ont deux composants qui sont tous deux insensibles à la casse :
* le nom de domaine de la passerelle qui gère le point d'extrémité
* un nom local au sein de cette passerelle.

Les noms de point d'extrémité sont de la forme :

nom de point d'extrémité local@nom de domaine

où nom de domaine est un nom de domaine absolu comme défini dans la RFC 1034 et inclut une portion hôte, donc un exemple de nom de domaine pourrait être :

mapasserelle.quelconque.net

Aussi, le nom de domaine peut être une adresse IP de la forme définie pour un nom de domaine dans la RFC 821, donc un autre exemple pourrait être (voir les détails dans la RFC 821) :

[192.168.1.2]

Les adresses IPv4 et IPv6 peuvent être spécifiées, cependant l'utilisation d'adresses IP comme identifiant de point d'extrémité est généralement déconseillée.

Noter que comme la portion nom de domaine fait partie de l'identifiant de point d'extrémité, différentes formes ou différentes valeurs se référant à la même entité ne sont pas librement interchangeables. La forme et valeur les plus récemment fournies DOIVENT toujours être utilisées.

Le nom du point d'extrémité local est insensible à la casse. La syntaxe du nom du point d'extrémité local est hiérarchique, où le composant le moins spécifique du nom est le terme plus à gauche, et le composant le plus spécifique est le plus à droite. La syntaxe précise dépend du type de point d'extrémité désigné et PEUT commencer avec un terme qui identifie le type de point d'extrémité. En tous cas, le nom du point d'extrémité local DOIT respecter les règles de désignation suivantes :

1) Les termes individuels du chemin de désignation DOIVENT être séparés par une seule barre oblique ("/", ASCII 2F hexadécimal).

2) Les termes individuels sont des chaînes de caractères composées de lettres, chiffres ou autres caractères imprimables, à l'exception des caractères utilisés comme délimiteurs ("/", "@"), des caractères utilisés pour les caracères génériques ("*", "$") et des espaces.

3) Le caractère générique est représenté par un astérisque ("*") ou un signe dollar ("$") pour les termes du chemin de désignation qui sont remplacés par un caractère générique. Donc, si le nom local complet du point d'extrémité est de la forme : terme1/terme2/terme3, alors le champ de nom de l'entité ressemblera à ce qui suit selon les termes qui sont remplacés par un caractère générique :

   */terme2/terme3 si terme1 est remplacé par un caractère générique
   terme1/*/terme3 si terme2 est remplacé par un caractère générique
   terme1/terme2/* si terme3 est remplacé par un caractère générique
   terme1/*/* si terme2 et terme3 sont remplacés par un caractère générique, etc.

   Dans chacun de ces exemples un signe dollar aurait pu apparaître à la place d'un astérisque.

4) Un terme représenté par un astérisque ("*") est à interpréter comme "utiliser TOUTES les valeurs de ce terme connues dans la portée de la passerelle de supports". Sauf mention contraire, ceci se réfère à tous les points d'extrémité configurés pour le service, sans considération de leur état de service réel, c'est-à-dire, en service ou hors de service.

5) Un terme représenté par un signe dollar ("$") est à interpréter comme : "utiliser TOUTE valeur de ce terme connue au sein de la portée de la passerelle de supports". Sauf mention contraire, ceci se réfère seulement aux points d'extrémité qui sont en service.

De plus, il est RECOMMANDÉ que les agents d'appel respectent les règles suivantes :

* Le remplacement par un caractère générique devrait seulement être fait à partir de la droite, donc si un terme est remplacé par un caractère générique, alors tous les termes à la droite de ce terme devraient aussi être remplacés par un caractère générique.

* Dans les cas où des caractères génériques mixtes dollar et astérisque sont utilisés, les signes dollar devraient seulement être utilisés à partir de la droite, donc si un terme a un caractère générique de signe dollar, tous les termes à la droite de ce terme devraient aussi contenir des caractères génériques de signe dollar.

La description d'une commande spécifique peut ajouter d'autres critères pour le choix au sein des règles générales données ci-dessus.

Noter que les caractères génériques peuvent être appliqués à plus d'un terme et dans ce cas ils devront être évalués de gauche à droite. Par exemple, si on a les noms de point d'extrémité "a/1", "a/2", "b/1", et "b/2", alors "$/*" (qui n'est pas recommandé) va s'évaluer comme "a/1, a/2", ou "b/1, b/2". Cependant, "*/$" peut s'évaluer comme "a/1, b/1", "a/1, b/2", "a/2, b/1", ou "a/2, b/2". L'utilisation de caractères génériques mixtes dans une commande est considérée comme enclin à l'erreur et est par conséquent déconseillée.

Un nom local qui est composé seulement de caractères génériques se réfère soit à tous les points d'extrémité (*) soit à tout ($) point d'extrémité au sein de la passerelle de supports.

2.1.3 Appels et connexions

Les connexions sont créées sur l'agent d'appel sur chaque point d'extrémité qui va être impliqué dans "l'appel". Dans l'exemple classique d'une connexion entre deux points d'extrémité "DS0" (EP1 et EP2) les agents d'appel qui contrôlent les points d'extrémité vont établir deux connexions (C1 et C2) :

Chaque connexion va être désignée localement par un identifiant de connexion unique de point d'extrémité, et va être caractérisée par des attributs de connexion.

Quand les deux points d'extrémité sont situés sur des passerelles gérées par le même agent d'appel, la création est faite via les trois étapes suivantes :

1) L'agent d'appel demande à la première passerelle de "créer une connexion" sur le premier point d'extrémité. La passerelle alloue des ressources à cette connexion, et répond à la commande en fournissant une "description de session". La description de session contient les informations nécessaires pour qu'un tiers envoie des paquets sur la connexion

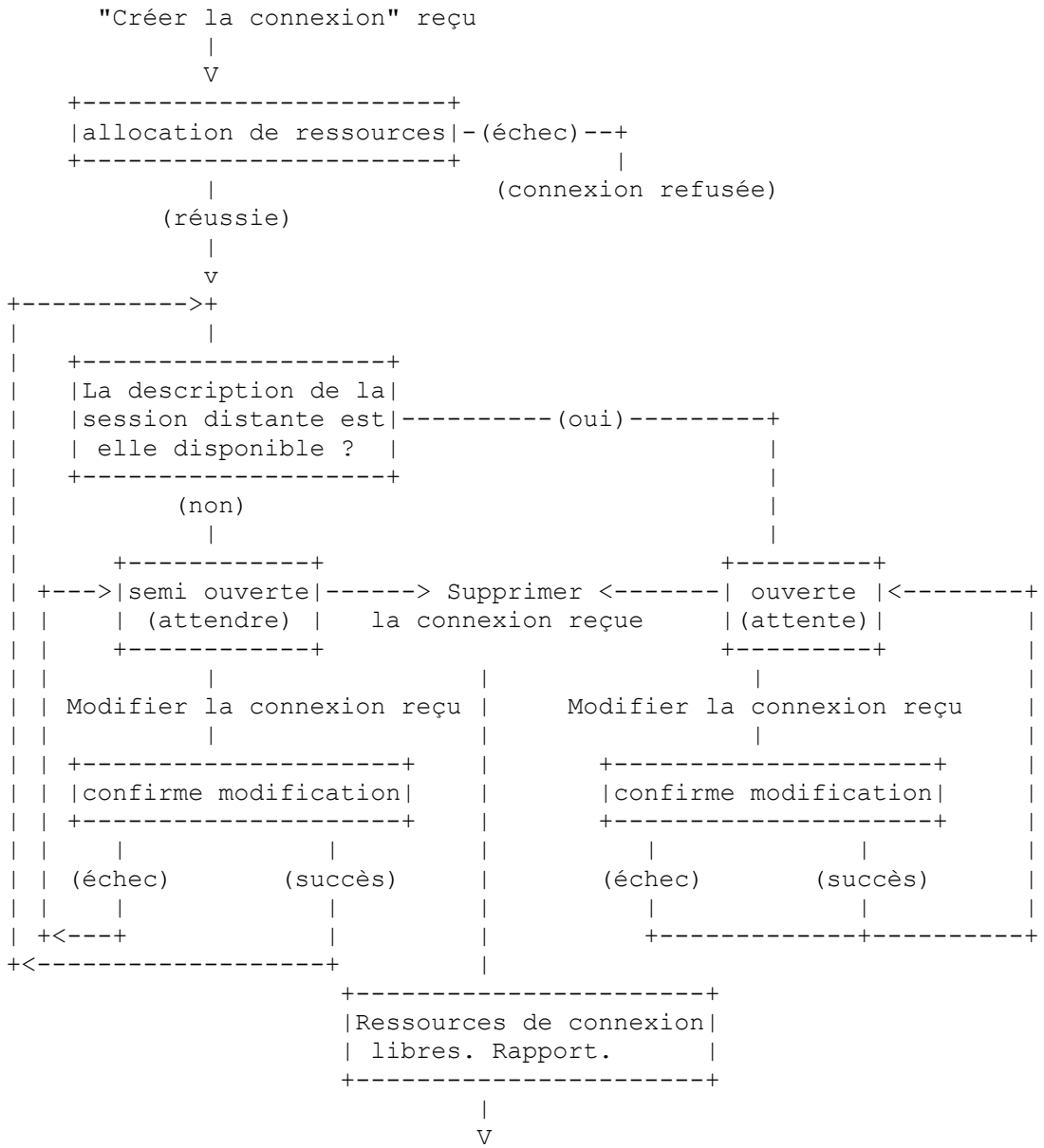
nouvellement créée, comme par exemple l'adresse IP, l'accès UDP, et les paramètres de codec.

2) L'agent d'appel demande alors à la seconde passerelle de "créer une connexion" sur le second point d'extrémité. La commande porte la "description de session" fournie par la première passerelle. La passerelle alloue des ressources à cette connexion, et répond à la commande en fournissant sa propre "description de session".

3) L'agent d'appel utilise alors une commande "modifier la connexion" pour fournir cette seconde "description de session" au premier point d'extrémité. Une fois que c'est fait, la communication peut s'effectuer dans les deux directions.

Quand les deux points d'extrémité sont situés sur des passerelles gérées par deux agents d'appel différents, les agents d'appel échangent les informations à travers un protocole de signalisation d'agent d'appel à agent d'appel, par exemple, SIP [RFC3261], afin de synchroniser la création de la connexion sur les deux points d'extrémité. Une fois qu'une connexion a été établie, les paramètres de connexion peuvent être modifiés à tout moment par une commande "modifier la connexion". L'agent d'appel peut par exemple donner pour instruction à la passerelle de changer le codec utilisé sur une connexion, ou de modifier l'adresse IP et l'accès UDP auxquels les données devraient être envoyées, si une connexion est "redirigée".
L'agent d'appel supprime une connexion en envoyant une commande "supprimer la connexion" à la passerelle. La passerelle peut aussi, dans certaines circonstances, informer une passerelle qu'une connexion ne pourra pas être maintenue.

Le diagramme suivant donne une vue des états d'une connexion, vue de la passerelle :



2.1.3.1 Noms des appels

Un des attributs de chaque connexion est "l'identifiant d'appel", qui, pour ce qui concerne le protocole MGCP a peu de signification sémantique, et est principalement conservé pour la rétro compatibilité.

Les appels sont identifiés par des identifiants univoques, indépendants des plates-formes ou agents sous-jacents. Les identifiants d'appel sont des chaînes hexadécimales, qui sont créées par l'agent d'appel. La longueur maximale des identifiants d'appel est de 32 caractères.

Les identifiants d'appel sont supposés être uniques au sein du système, ou au minimum, uniques au sein de la collection d'agents d'appel qui contrôlent les mêmes passerelles. Du point de vue de la passerelle, l'identifiant d'appel est donc unique. Quand un agent d'appel construit plusieurs connexions qui relèvent du même appel, soit sur la même passerelle, soit dans des passerelles différentes, ces connexions qui appartiennent au même appel devraient partager le même identifiant d'appel. Cet identifiant peut alors être utilisé par les procédures de comptabilité ou de gestion, qui sortent du domaine d'application de MGCP.

2.1.3.2 Noms des connexions

Les identifiants de connexion sont créés par la passerelle quand il lui est demandé de créer une connexion. Ils identifient la connexion au sein du contexte d'un point d'extrémité. Les identifiants de connexion sont traités dans MGCP comme des chaînes hexadécimales. La passerelle DOIT s'assurer qu'une période d'attente appropriée, d'au moins 3 minutes, s'écoule entre la fin d'une connexion qui a utilisé cet identifiant et son utilisation sur une nouvelle connexion pour le même point d'extrémité (les passerelles PEUVENT décider d'utiliser des identifiants qui sont uniques au sein du contexte de la passerelle). La longueur maximale d'un identifiant de connexion est de 32 caractères.

2.1.3.3 Gestion des ressources, attributs des connexions

De nombreux types de ressources vont être associées à une connexion, comme des fonctions spécifiques de traitement de signal ou de fonctions de mise en paquet. Généralement, ces ressources entrent dans deux catégories :

1) Ressources visibles en externe, qui affectent le format des "bits sur le réseau" et doivent être communiquées au second point d'extrémité impliqué dans la connexion.

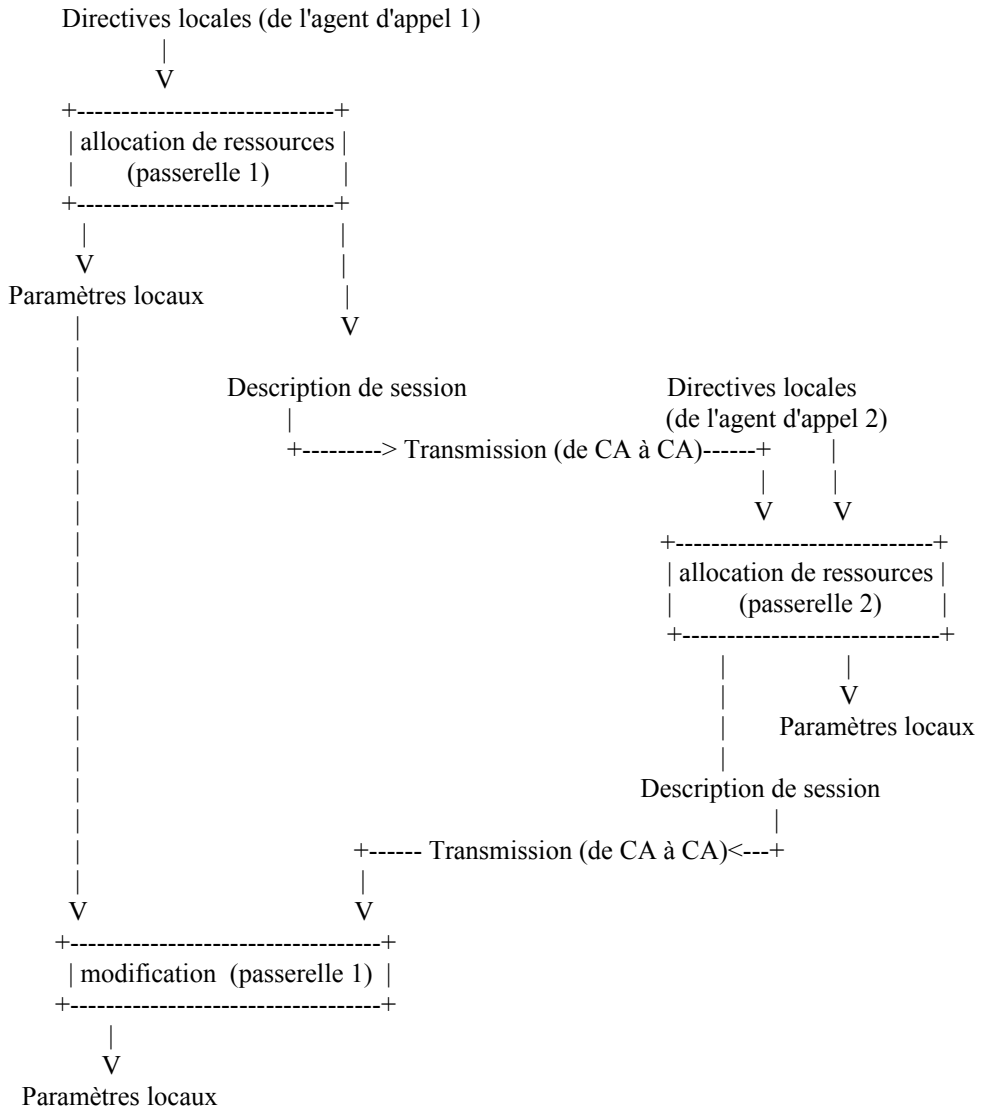
2) Ressources internes, qui déterminent quel signal est envoyé sur la connexion et comment les signaux reçus sont traités par le point d'extrémité.

Les ressources allouées à une connexion, et plus généralement le traitement de la connexion, sont choisies par la passerelle sur instructions de l'agent d'appel. L'agent d'appel va fournir ces instructions en envoyant deux ensembles de paramètres à la passerelle :
1) les directives locales guident la passerelle dans le choix des ressources qui devraient être utilisées pour une connexion,
2) quand elle est disponible, la "description de session" fournie par l'autre extrémité de la connexion (appelée la description de session distante).

Les directives locales spécifient des paramètres comme le mode de la connexion (par exemple, envoi seul, ou envoi-réception) le codage préféré ou les méthodes de mise en paquet, l'usage de l'annulation d'écho ou de la suppression de silence. (Une liste détaillée se trouve dans la spécification du paramètre "LocalConnectionOptions" de la commande CreateConnection.) Selon le paramètre, l'agent d'appel PEUT spécifier une valeur, une gamme de valeurs, ou pas de valeur du tout. Cela permet à diverses mises en œuvre d'utiliser divers niveaux de contrôle, depuis un contrôle très serré où l'agent d'appel spécifie les détails minutieux du traitement de connexion jusqu'à un contrôle très lâche où l'agent d'appel spécifie seulement de très vagues lignes directrices, comme la bande passante maximum, et laisse la passerelle choisir le détail des valeurs soumises aux lignes directrices.

Sur la base de la valeur des directives locales, la passerelle va déterminer les ressources à allouer à la connexion. Quand c'est possible, la passerelle va choisir les valeurs qui sont en ligne avec la description de session distante – mais il n'y a pas d'exigence absolue que les paramètres soient exactement les mêmes.

Une fois que les ressources ont été allouées, la passerelle va composer une "description de session" qui décrit la façon dont elle entend envoyer et recevoir les paquets. Noter que la description de session peut dans certains cas présenter une gamme de valeurs. Par exemple, si la passerelle est prête à accepter un algorithme de compression parmi plusieurs, elle peut fournir une liste des algorithmes acceptés.



-- Flux d'informations : directives locales & descriptions de session --

2.1.3.4 Cas particuliers des connexions locales

Les grandes passerelles incluent un grand nombre de points d'extrémité qui sont souvent de types différents. Dans certains réseaux, on peut souvent avoir à établir des connexions entre des points d'extrémité qui sont situés au sein de la même passerelle. Des exemples de telles connexions peuvent être :

* pour connecter un appel à une unité de réponse vocale interactive,
* pour connecter un appel à une unité de conférence,
* pour acheminer un appel d'un point d'extrémité à un autre, quelque chose qui est souvent décrit comme une connexion "en épingle à cheveux".

Les connexions locales sont beaucoup plus simples à établir que des connexions de réseau. Dans la plupart des cas, la connexion va être établie par un appareil d'interconnexion local, comme par exemple un bus TDM.

Quand deux points d'extrémité sont gérés par la même passerelle, il est possible de spécifier la connexion dans une seule commande qui porte les noms des deux points d'extrémité à connecter. La commande est essentiellement un "Créer la connexion" qui comporte le nom du second point d'extrémité au lieu de la "description de session distante".

2.1.4 Noms des agents d'appel et autres entités

Le protocole de contrôle de passerelle de supports a été conçu pour permettre la mise en œuvre d'agents d'appel redondants,

pour une fiabilité de réseau améliorée. Cela signifie qu'il n'y a pas de lien fixe entre les entités et les plates-formes de matériel ou interfaces réseau.

Les noms d'agent d'appel consistent en deux parties, similaires aux noms de point d'extrémité. Sémantiquement, la portion locale du nom ne présente aucune structure interne. Un exemple de nom d'agent d'appel est :

ca1@ca.quelconque.net

Noter que la partie locale et le nom de domaine doivent tous deux être fournis. Néanmoins, les mises en œuvre sont invitées à accepter des noms d'agent d'appel consistant seulement en un nom de domaine.

La fiabilité peut être améliorée en utilisant les procédures suivantes :

* Les entités comme des points d'extrémité ou des agents d'appel sont identifiées par leur nom de domaine, non par leur adresse réseau. Plusieurs adresses peuvent être associées à un nom de domaine. Si une commande ou réponse ne peut pas être transmise à une des adresses réseau, les mises en œuvre DOIT réessayer la transmission en utilisant une autre adresse.

* Les entités PEUVENT passer à une autre plate-forme. L'association entre un nom logique (nom de domaine) et la plate-forme réelle est conservée dans le service de noms de domaine. Les agents d'appel et les passerelles DOIVENT garder trace de la durée de vie de l'enregistement lu sur le DNS. Ils DOIVENT interroger le DNS pour rafraîchir les informations si la durée de vie est expirée.

En plus des indications fournies par l'utilisation des noms de domaines et du DNS, le concept d'une "entité notifiée" est central pour la fiabilité et la reprise sur défaillance dans MGCP. Une "entité notifiée" pour un point d'extrémité est l'agent d'appel qui contrôle actuellement ce point d'extrémité. À tout moment, un point d'extrémité a une, et seulement une, "entité notifiée" associée à lui. Une "entité notifiée" détermine où le point d'extrémité va envoyer les commandes ; quand le point d'extrémité a besoin d'envoyer une commande à l'agent d'appel, il DOIT envoyer la commande à son "entité notifiée" actuelle. Une "entité notifiée" ne détermine cependant pas d'où les commandes peuvent être reçues ; tout agent d'appel peut envoyer des commandes au point d'extrémité. Voir à la Section 5 les considérations de sécurité pertinentes.

Au démarrage, une "entité notifiée" DOIT être réglée à une valeur provisionnée. La plupart des commandes envoyées par l'agent d'appel incluent la capacité de désigner explicitement "l'entité notifiée" par l'utilisation d'un paramètre "NotifiedEntity". L'"entité notifiée" va rester la même jusqu'à ce que soit un nouveau paramètre "NotifiedEntity" soit reçu, soit que le point d'extrémité fasse un redémarrage à chaud ou à froid (cycle d'alimentation).

Si un paramètre "NotifiedEntity" est envoyé avec une valeur "vide", "l'entité notifiée" pour le point d'extrémité va être réglée à vide. Si "l'entité notifiée" pour un point d'extrémité est vide ou n'a pas été réglée explicitement (ni par une commande ni par provisionnement) "l'entité notifiée" va alors être par défaut l'adresse de source (c'est-à-dire, l'adresse IP et le numéro d'accès UDP) de la dernière commande réussie non d'audit reçue pour le point d'extrémité. L'audit ne va donc pas changer "l'entité notifiée". L'utilisation d'une valeur vide de paramètre "NotifiedEntity" est fortement déconseillée car c'est enclin à l'erreur et élimine les mécanismes de reprise sur défaillance et de fiabilité fondés sur le DNS.

2.1.5 Transpositions de chiffres

L'agent d'appel peut demander à la passerelle de collecter les chiffres composés par l'utilisateur. Cette facilité est destinée à être utilisée avec les passerelles résidentielles pour collecter les numéros composés par un utilisateur ; elle peut aussi être utilisée avec les passerelles de circuits et les passerelles d'accès, pour collecter les codes d'accès, les numéros de carte de crédit et autres numéros demandés par les services de contrôle d'accès.

Une procédure est que la passerelle notifie à l'agent d'appel chaque chiffre individuel composé, aussitôt qu'il est composé. Cependant, une telle procédure génère un grand nombre d'interactions. Il est préférable d'accumuler les numéros composés dans une mémoire tampon, et de les transmettre dans un seul message.

Le problème avec cette approche de l'accumulation est qu'il est cependant difficile à la passerelle de prédire combien de numéros elle a besoin d'accumuler avant la transmission. Par exemple, en utilisant le téléphone de son bureau, on peut composer les numéros suivants :

0	opérateur local
00	opérateur longue distance
xxxx	numéro d'extension locale

8xxxxxxx	numéro local
#xxxxxxx	raccourci du numéro local sur d'autres sites de l'entreprise
*xx	services confort
91xxxxxxxxxx	numéro longue distance
9011 + jusqu'à 15 chiffres	numéro international

La solution à ce problème est que l'agent d'appel charge la passerelle avec un script de numérotation qui peut correspondre au plan de numérotage. Ce script de numérotation est exprimé en utilisant une syntaxe dérivée de la commande egrep du système Unix. Par exemple, le plan de numérotage décrit ci-dessus résulte en 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 par la règle DigitMap *(transposition de chiffre)* dans la description de la syntaxe formelle du protocole (Appendice A). La prise en charge des lettres de base de script de numérotation est EXIGÉE, tandis que la prise en charge des lettres d'extension de script de numérotation est FACULTATIVE. Une passerelle qui reçoit un script de numérotation avec une lettre d'extension de script de numérotation non prise en charge DEVRAIT retourner le code d'erreur 537 (extension de script de numérotation inconnue).

Un script de numérotation, selon cette syntaxe, est défini par une "chaîne" (insensible à la casse) ou par une liste de chaînes. Chaque chaîne de la liste est un autre schéma de numérotation, spécifié comme un ensemble de chiffres ou temporisateurs, ou comme une expression sur laquelle la passerelle va tenter de trouver une plus courte correspondance possible. Les constructions suivantes peuvent être utilisées dans chaque schéma de numérotation :
* Digit : un chiffre de "0" à "9".
* Timer : le symbole "T" correpondant à une expiration de temporisateur.
* DTMF : un chiffre, un temporisateur, ou un des symboles "A", "B", "C", "D", "#", ou "*". Des extensions peuvent être définies.
* Wildcard : le symbole "x" qui correspond à tout chiffre ("0" à "9").
* Range : un ou plusieurs symboles DTMF enclos entre des crochets ("[" et "]").
* Subrange : deux chiffres séparés par un tiret ("-") qui correspond à tout chiffre entre les deux inclus. La construction de sous gamme ne peut être utilisée qu'à l'intérieur d'une construction de gamme, c'est-à-dire, entre "[" et "]".
* Position : un point (".") qui correspond à un nombre arbitraire, incluant zéro, d'occurrences de la construction précédente.

Une passerelle qui détecte les événements à confronter à un script de numérotation DOIT faire ce qui suit :

1) Ajouter le code d'événement comme jeton à la fin d'une variable d'état interne pour le point d'extrémité, appelée la "chaîne de numérotation courante".

2) Appliquer la chaîne de numérotation courante au tableau des scripts de numérotation, en tentant une correspondance pour chaque expression dans le script de numérotation.

3) Si le résultat est sous qualifié (correspondance partielle d'au moins une entrée dans le script de numérotation et pas de correspondance complète à une autre entrée) ne rien faire de plus.

Si le résultat correspond à une entrée, ou est sur-qualifié (c'est-à-dire, aucun autre chiffre de plus ne pourrait produire une correspondance) envoyer la liste des événements accumulés à l'agent d'appel. Une correspondance, dans la présente spécification, peut être soit une "correspondance parfaite," correspondant exactement à une des alternatives spécifiées, soit une correspondance impossible, ce qui se produit quand la chaîne de numérotage ne correspond à aucune des alternatives. Des temporisateurs inattendus, par exemple, peuvent causer des "correspondances impossibles". Les correspondances parfaites et les correspondances impossibles déclenchent la notification des chiffres accumulés (qui peut inclure d'autres événements – voir le paragraphe 2.3.3).

L'exemple suivant illustre cela. On suppose qu'on a le script de numérotation : (xxxxxxx|x11) et une chaîne de numérotation courante de "41". Avec l'entrée "1" la chaîne de numérotation courante devient "411". On a une correspondance partielle avec "xxxxxxx", mais une correspondance complète avec "x11", et donc on envoie "411" à l'agent d'appel.

L'exemple de script de numérotation suivant est plus subtil : (0[12].|00|1[12].1|2x.#)

Étant donnée l'entrée "0", une correspondance va se produire immédiatement car la position (".") permet zéro occurrence de la construction précédente. L'entrée "00" ne peut donc jamais être produite dans ce script de numérotation.

Étant donnée l'entrée "1", seule une correspondance partielle existe. L'entrée "12" est aussi seulement une correspondance partielle, cependant les deux "11" et "121" sont une correspondance.

Étant donnée l'entrée "2", une correspondance partielle existe. Une correspondance partielle existe aussi pour les entrées "23", "234", "2345", etc. Une correspondance complète ne se produit pas avant qu'un "#" soit généré, par exemple, "2345#". L'entrée "2#" serait aussi une correspondance.

Noter que les scripts de numérotation définissent simplement un moyen de confronter des séquences de codes d'événement à une grammaire. Bien que les scripts de numérotation tels que définis ici soient pour une entrée DTMF, des paquetages d'extension peuvent aussi être définis afin que des scripts de numérotation puissent être utilisés pour d'autres types d'entrées représentées par des codes d'événement qui respectent la syntaxe de script de numérotation déjà définie pour ces codes d'événement (par exemple, "1" ou "T"). Lorsque un tel usage est envisagé, la définition des événements particuliers DEVRAIT déclarer explicitement cela dans la définition du paquetage.

Comme les scripts de numérotation ne sont pas limités en taille, il est RECOMMANDÉ que les passerelles prennent en charge des scripts de numérotation jusqu'à au moins 2048 octets par point d'extrémité.

2.1.6 Paquetages

MGCP est un protocole modulaire et extensible, cependant avec l'extensibilité vient le besoin de gérer, identifier, et nommer les extensions individuelles. Ceci est réalisé par le concept de paquetages, qui sont simplement des groupements bien définis d'extensions. Par exemple, un paquetage peut supporter un certain groupe d'événements et signaux, par exemple, décroché et sonnerie, pour les lignes d'accès analogiques. Un autre paquetage peut supporter un autre groupe d'événements et signaux pour les lignes d'accès analogiques ou pour un autre type de point d'extrémité comme de vidéo. Un ou plusieurs paquetages peuvent être pris en charge par un certain point d'extrémité.

MGCP permet que les types d'extensions suivants soient définis dans un paquetage :
* BearerInformation *(informations de support)*
* LocalConnectionOptions *(options de connexion locale)*
* ExtensionParameters *(paramètres d'extension)*
* ConnectionModes *(modes de connexion)*
* Events *(événements)*
* Signals *(signaux)*
* Actions
* DigitMapLetters *(lettres de script de numérotation)*
* ConnectionParameters *(paramètres de connexion)*
* RestartMethods *(méthodes de redémarrage)*
* ReasonCodes *(codes de cause)*
* ReturnCodes *(codes de retour)*

Chacun d'eux va être expliqué en détails ci-dessous. Les règles pour définir chacune de ces extensions dans un paquetage sont décrites à la Section 6 ; le codage et la syntaxe sont définis dans la Section 3 et l'Appendice A.

À l'exception de DigitMapLetters, un paquetage définit un espace de noms séparé pour chaque type d'extension en ajoutant le nom de paquetage comme préfixe à l'extension, c'est-à-dire :

nom de paquetage/extension

Donc le nom de paquetage est suivi d'une barre oblique ("/") et du nom de l'extension.

Un point d'extrémité qui prend en charge un ou plusieurs paquetages peut définir un de ces paquetages comme paquetage par défaut pour le point d'extrémité. L'utilisation du nom de paquetage pour les événements et signaux dans le paquetage par défaut pour un point d'extrémité est FACULTATIVE, cependant il est RECOMMANDÉ de toujours inclure le nom de paquetage. Toutes les autres extensions, sauf DigitMapLetter, définies dans le paquetage DOIVENT inclure le nom de paquetage quand elles se réfèrent à l'extension.

Les noms de paquetage sont des chaînes de lettres, tirets et chiffres insensibles à la casse, avec la restriction que les tirets ne devront jamais être le premier ou le dernier caractère d'un nom. Des exemples de noms de paquetage sont "D", "T", et "XYZ". Les noms de paquetage ne sont pas sensibles à la casse – des noms comme "XYZ", "xyz", et "xYz" sont égaux.

Les définitions de paquetages seront fournies par d'autres documents et avec les noms de paquetage et les noms

d'extensions enregistrés par l'IANA. Pour les détails, se reporter à la Section 6.

Les développeurs peuvent faire des expériences en utilisant des paquetages expérimentaux. Le nom d'un paquetage expérimental DOIT commencer par les deux caractères "x-" ; l'IANA NE DEVRA PAS enregistrer des noms de paquetage qui commencent par ces caractères, ou les caractères "x+", qui sont réservés. Une passerelle qui reçoit une commande se référant à un paquetage non pris en charge DOIT retourner une erreur (le code d'erreur 518 - paquetage non pris en charge, est RECOMMANDÉ).

2.1.7 Événements et signaux

Le concept d'événements et de signaux est central pour MGCP. Un agent d'appel peut demander à être notifié de certains événements se produisant à un point d'extrémité (par exemple, des événements de décroché) en incluant le nom de l'événement dans un paramètre RequestedEvents (dans une commande NotificationRequest - paragraphe 2.3.3).

Un agent d'appel peut aussi demander que certains signaux soient appliquées à un point d'extrémité (par exemple, des tonalités de numérotation) en fournissant le nom de l'événement dans un paramètre SignalRequests.

Les événements et signaux sont groupés en paquetages, au sein desquels il partagent le même espace de noms qu'on appellera des noms d'événement dans la suite du texte. Les noms d'événement sont des chaînes de lettres, tirets et chiffres, insensibles à la casse, avec cette restriction que les tirets NE DEVRONT PAS être le premier ou dernier caractère d'un nom. Certains codes d'événement peuvent devoir être paramétrés avec des données supplémentaires, ce qui est fait en ajoutant les paramètres entre des parenthèses. Les noms d'événement ne sont pas sensibles à la casse – des valeurs telles que "hu", "Hu", "HU" ou "hU" sont égales.

Des exemples de noms d'événement peuvent être "hu" (transition à décroché ou "hang-up") "hf" (hook-flash, *impulsion crochet*) ou "0" (le chiffre zéro).

Le nom de paquetage est FACULTATIF pour les événements dans le paquetage par défaut pour un point d'extrémité, cependant il est RECOMMANDÉ de toujours inclure le nom de paquetage. Si le nom de paquetage est exclu du nom d'événement, le nom du paquetage par défaut pour ce point d'extrémité DOIT être supposé. Par exemple, pour une ligne d'accès analogique qui a le paquetage de ligne ("L") comme défaut avec la tonalité de numérotation ("dl") comme un des événements de ce paquetage, les deux noms d'événement suivants sont égaux : "L/dl" et "dl".

Pour tout autre paquetage non par défaut associé à ce point d'extrémité, (comme par exemple le paquetage générique pour un point d'extrémité de type accès analogique) le nom de paquetage DOIT être inclus avec le nom d'événement. Là encore, l'inclusion inconditionnelle du nom de paquetage est RECOMMANDÉE.

Des chiffres, ou lettres, sont pris en charge dans certains paquetages, notamment "DTMF". Les chiffres et lettres sont définis par les règles "Digit" et "Letter" dans la définition des scripts de numérotation. Cette définition se réfère aux chiffres (0 à 9) à l'astérisque ou étoile ("*") et l'orthotrope, numéro ou signe dièse ("#") et aux lettres "A", "B", "C" et "D", ainsi qu'à l'indication de temporisateur "T". Ces lettres peuvent être combinées en une "chaîne numérique" qui représente les clés qu'un utilisateur compose sur un clavier. De plus, la lettre "X" peut être utilisée pour représenter tous les chiffres (0 à 9). Aussi, des extensions PEUVENT définir l'utilisation d'autres lettres. Le besoin d'exprimer facilement les chaînes numériques dans les versions antérieures du protocole a des conséquence sur la forme des noms d'événement : un nom d'événement qui ne note pas un chiffre DOIT toujours contenir au moins un caractère qui n'est ni un chiffre, ni une des lettres A, B, C, D, T ou X (ces noms aussi NE DOIVENT PAS contenir juste les signes spéciaux "*", ou "#"). Les noms d'événement consistant en plus d'un caractère peuvent cependant les utiliser.

Un agent d'appel peut souvent devoir demander à une passerelle de détecter un groupe d'événements. Deux conventions peuvent être utilisées pour noter de tels groupes :

* Les conventions de caractère générique "*" et "all" (voir ci-dessous) peuvent être utilisées pour détecter tout événement appartenant à un paquetage, ou à un certain événement dans de nombreux paquetages, ou tout événement dans tout paquetage pris en charge par la passerelle.

* L'expression régulière "Range notation" *(notation de gamme)* peut être utilisée pour détecter une gamme de chiffres.

Le signe étoile (*) peut être utilisé comme caractère générique à la place d'un nom de paquetage, et le mot clé "all" *(tous)* peut être utilisé comme caractère générique à la place d'un nom d'événement :
* un nom comme "foo/all" note tous les événements dans le paquetage "foo".
* un nom comme "*/bar" note l'événement "bar" dans tout paquetage pris en charge par la passerelle.

* le nom "*/all" note tous les événements pris en charge par le point d'extrémité.

La présente spécification ne définit délibérement pas de détails supplémentaires sur les chaînes génériques "all packages" et "all events". Elles procurent des avantages limités, mais introduisent une complexité significative avec un potentiel d'erreurs. Leur utilisation est par conséquent fortement déconseillée.

L'agent d'appel peut demander à une passerelle de détecter un ensemble de chiffres ou lettres soit en décrivant individuellement ces lettres, soit en utilisant la notation "range" *(gamme)* définie dans la syntaxe des chaînes de chiffres. Par exemple, l'agent d'appel peut :
* Utiliser la lettre "x" pour noter les chiffres de 0 à 9.
* Utiliser la notation "[0-9#]" pour noter les chiffres de 0 à 9 et le signe dièse.

Les codes d'événement individuels sont cependant quand même définis dans un paquetage (par exemple, le paquetage "DTMF").

Les événements peuvent par défaut être seulement générés et détectés sur des points d'extrémité, cependant des événements peuvent être aussi être définis afin d'être générés ou détectés sur des connexions plutôt que sur le point d'extrémité lui-même (paragraphe 6.6). Par exemple, il peut être demandé à des passerelles de fournir une tonalité de retour d'appel sur une connexion. Quand un événement est à appliquer sur une connexion, le nom de la connexion DOIT être ajouté au nom de l'événement, en utilisant un signe "@" comme délimiteur, comme dans : G/rt@0A3F58 où "G" est le nom du paquetage et "rt" est le nom de l'événement. Si la connexion devait être supprimée alors qu'un événement ou signal est détecté ou appliqué sur elle, cette détection d'événement ou génération de signal particulière s'arrête simplement. Selon le signal, cela peut générer une défaillance (voir ci-dessous).

Le caractère générique "*" (étoile) peut être utilisé pour noter "toutes les connexions". Quand cette convention est utilisée, la passerelle va générer ou détecter l'événement sur toutes les connexions qui sont reliées au point d'extrémité. Ceci s'applique aux connexions existantes aussi bien que futures créées sur le point d'extrémité. Un exemple de cette convention pourrait être : R/qa@* où "R" est le nom du paquetage et "qa" est le nom de l'événement.

Lors du traitement d'une commande qui utilise la convention générique "toutes les connexions", le caractère générique "*" s'applique à toutes les connexions courantes et futures sur le point d'extrémité, cependant il ne va pas être étendu. Si une commande suivante se réfère explicitement (par exemple, par examen) ou implicitement (par exemple, par persistance) à un tel événement, la valeur "*" va être utilisée. Cependant, quand l'événement est réellement observé, cette occurrence particulière de l'événement va inclure le nom de la connexion spécifique sur laquelle il s'est produit.

Le caractère générique "$" peut être utilisé pour noter "la connexion courante". Il peut seulement être utilisé par l'agent d'appel, quand la demande de notification d'événement est "encapsulée" au sein d'une commande de création ou modification de connexion. Quand cette convention est utilisée, la passerelle va générer ou détecter l'événement sur la connexion qui est actuellement créée ou modifiée. Un exemple de cette convention est : G/rt@$

Quand on traite une commande qui utilise la convention générique "la connexion courante", le caractère générique "$" va être étendu à la valeur de la connexion courante. Si une commande suivante se réfère explicitement (par exemple, par examen) ou implicitement (par exemple, par persistance) à un tel événement, la valeur étendue va être utilisée. En d'autres termes, la convention générique "la connexion courante" est étendue une fois, qui est au traitement initial de la commande dans laquelle elle a été explicitement incluse.

L'identifiant de connexion, ou un caractère générique de remplacement, peut être utilisé en conjonction avec les conventions "tous les paquetages" et "tous les événements". Par exemple, la notation */all@* peut être utilisée pour désigner tous les événements sur toutes les connexions courantes et futures sur le point d'extrémité. Cependant, comme mentionné précédemment, l'usage des conventions génériques "tous les paquetages" et "tous les événements" est fortement déconseillé.

Les signaux sont divisés en différents types selon leur comportement :

* Ouvert/fermé (OO, *On/off*) : une fois appliqués, ces signaux durent jusqu'à ce qu'ils soient arrêtés. Cela ne peut arriver que comme résultat d'un réamorçage/redémarrage ou de nouvelles SignalRequests *(demande de signaux)* où le signal est explicitement arrêté (voir plus loin). Les signaux de type OO sont définis comme idempotents, donc plusieurs demandes d'ouvrir (ou fermer) un signal OO sont parfaitement valides et NE DOIVENT PAS résulter en une erreur. Un signal OO pourrait être un indicateur visuel de message en attente (VMWI, *visual message-waiting indicator*). Une fois ouvert, il NE DOIT PAS être fermé sans instruction explicite de l'agent d'appel, ou par suite d'un redémarrage du point d'extrémité, c'est-à-dire, ces signaux ne sont pas arrêtés par suite de la détection d'un événement demandé.

* Fin de temporisation (TO, *Time-out*) : une fois appliqués, ces signaux durent jusqu'à ce qu'ils soient annulés (par l'occurrence d'un événement ou en n'étant pas inclus dans une liste suivante (éventuellement vide) de signaux) ou par l'écoulement d'une période spécifique du signal. Un signal TO qui arrive en fin de temporisation va générer un événement "opération achevée". Un signal TO pourrait être une fin de temporisation de "retour d'appel" après 180 secondes. Si un événement se produit avant les 180 secondes, le signal va, par défaut, être arrêté (l'action "Garder les signaux actifs" – paragraphe 2.3.3 – va outrepasser ce comportement). Si le signal n'est pas arrêté, le signal va arriver en fin de temporisation, s'arrêter et générer un événement "opération achevée", sur lequel l'agent d'appel peut ou non avoir demandé d'en être notifié. Si l'agent d'appel a demandé que l'événement "opération achevée" soit notifié, l'événement "opération achevée" envoyé à l'agent d'appel DEVRA inclure le ou les noms du ou des signaux qui sont arrivés en fin de temporisation (noter que si les paramètres ont été passés au signal, les paramètres ne seront pas rapportés). Si le signal a été généré sur une connexion, le nom de la connexion DEVRA être inclus comme décrit ci-dessus. La fin de temporisation des signaux a une valeur par défaut définie pour eux, qui PEUT être altérée par le processus de provisionnement. Aussi, la période de temporisation peut être fournie comme un paramètre du signal (voir au paragraphe 3.2.2.4). Une valeur de zéro indique que la période de temporisation est infinie. Un signal TO qui échoue après avoir commencé, mais avant d'avoir généré un événement "opération achevée" va générer un événement "échec d'opération" qui va inclure le nom du signal qui a échoué. La suppression d'une connexion avec un signal TO actif va résulter en une telle défaillance.

* Bref (BR) : la durée de ces signaux est normalement si courte qu'ils arrêtent d'eux-mêmes. Si un événement d'arrêt de signal se produit, ou si une nouvelle SignalRequests est appliquée, un signal BR actuellement actif ne va pas s'arrêter. Cependant, tous les signaux BR en instance non encore appliqués DOIVENT être annulés (un signal BR devient en instance si une NotificationRequest inclut un signal BR, et si il y a déjà un signal BR actif). Par exemple, une tonalité brève pourrait être un chiffre DTMF. Si le chiffre DTMF "1" est actuellement en cours d'éxécution, et si un événement d'arrêt de signal se produit, le "1" va s'exécuter jusqu'à achèvement. Si une demande d'exécution du chiffre DTMF "2" arrive avant la fin de l'exécution du chiffre DTMF "1", le chiffre DTMF "2" va devenir en instance.

Les signaux générés sur une connexion DOIVENT inclure le nom de cette connexion.

2.2 Usage de SDP

L'agent d'appel utilise le MGCP pour fournir au point d'extrémité la description des paramètres de connexion comme les adresses IP, l'accès UDP et les profils RTP. Ces descriptions vont suivre les conventions décrites dans le protocole de description de session qui est une proposition de norme de l'IETF, documentée dans la RFC 2327.

2.3 Commandes de contrôle de passerelle

2.3.1 Vue d'ensemble des commandes

Ce paragraphe décrit les commandes de MGCP. Le service consiste en commandes de traitement de connexion et en commandes de traitement de point d'extrémité. Il y a acturellement neuf commandes dans le protocole :

* L'agent d'appel peut produire une commande EndpointConfiguration *(configuration de point d'extrémité)* à une passerelle, donnant des instructions à la passerelle sur les caractéristiques de codage attendues par le "côté ligne" du point d'extrémité.

* L'agent d'appel peut produire une commande NotificationRequest *(demande de notification)* à une passerelle, lui donnant pour instruction de surveiller des événements spécifiques comme des actions de crochet ou de tonalités DTMF sur un point d'extrémité spécifié.

* La passerelle va alors utiliser la commande Notify *(notifier)* pour informer l'agent d'appel quand les événements demandés se produisent.

* L'agent d'appel peut utiliser la commande CreateConnection *(créer une connexion)* pour créer une connexion qui se termine sur un "point d'extrémité" derrière la passerelle.

* L'agent d'appel peut utiliser la commande ModifyConnection *(modifier la connexion)* pour changer les paramètres associés à une connexion déjà établie.

* L'agent d'appel peut utiliser la commande DeleteConnection *(supprimer la connexion)* pour supprimer une connexion existante. La commande DeleteConnection peut aussi être utilisée par une passerelle pour indiquer qu'une connexion ne peut plus être maintenue.

* L'agent d'appel peut utiliser les commandes AuditEndpoint *(examen du point d'extrémité)* et AuditConnection *(examen de la connexion)* pour examiner l'état d'un "point d'extrémité" et de toutes les connexions qui lui sont associées. Une gestion de réseau au delà des capacités fournies par ces commandes est généralement désirable. De telles capacités sont supposées être prises en charge par l'utilisation du protocole simple de gestion de réseau (SNMP, *Simple Network Management Protocol*) et la définition d'une MIB (base de données d'informations de gestion) qui sort du domaine d'application de la présente spécification.

* La passerelle peut utiliser la commande RestartInProgress *(redémarrage en cours)* pour notifier à l'agent d'appel qu'un groupe de points d'extrémité gérés par la passerelle est en train d'être mis hors service ou est remis en service.

Ces services permettent à un contrôleur (normalement, l'agent d'appel) de donner des instructions à une passerelle sur la création des connexions qui se terminent à un "point d'extrémité" rattaché à la passerelle, et d'être informé des événements qui se produisent au point d'extrémité. Un point d'extrémité peut être par exemple :
* un circuit spécifique, au sein d'un groupe de circuits se terminant à une passerelle,
* une annonce spécifique traitée par un serveur d'annonces.

Les connexions sont logiquement groupées en "appels" (le concept d'un "appel" a cependant peu de signification sémantique dans MGCP lui-même). Plusieurs connexions, qui peuvent ou non appartenir aux mêmes appels, peuvent se terminer au même point d'extrémité. Chaque connexion est qualifiée par un paramètre "mode", qui peut être réglé à "envoi seul" *(sendonly)*, "réception seule" *(recvonly)*, "envoi/réception" *(sendrecv)*, "conférence" *(confrnce)*, "inactive" *(inactive)*, "bouclage arrière" *(loopback)*, "essai de continuité" *(conttest)*, "bouclage arrière de réseau" *(netwloop)* ou "essai de continuité de réseau" *(netwtest)*.

Le support généré par le point d'extrémité est envoyé sur les connexions dont le mode est "envoi seul", "envoi/réception", ou "conférence", sauf si le point d'extrémité a une connexion en mode "bouclage arrière" ou "essai de continuité". Cependant, les supports générés en appliquant un signal à une connexion sont toujours envoyés sur la connexion, sans considération du mode.

Le traitement du flux de supports reçus sur les connexions est déterminé par les paramètres du mode :

* Les flux de supports reçus par des connexions en mode "réception", "conférence" ou "envoi/réception" sont mixés et envoyés au point d'extrémité, sauf si le point d'extrémité a une autre connexion en mode "bouclage arrière" ou "esssai de continuité".

* Les flux de supports originaires du point d'extrémité sont transmis sur toutes les connexions dont le mode est "envoi", "conférence" ou "envoi/réception", sauf si le point d'extrémité a une autre connexion en mode "bouclage arrière" ou "essai de continuité".

* En plus d'être envoyé au point d'extrémité, un flux de supports reçu sur une connexion en mode "conférence" est transmis à toutes les autres connexions dont le mode est "conférence". Ceci s'applique aussi quand le point d'extrémité a une connexion en mode "bouclage arrière" ou "essai de continuité". Les détails de cette transmission, par exemple, traducteur ou mixeur RTP, sortent du domaine d'application du présent document.

Noter qu'afin de détecter les événements sur une connexion, la connexion doit être par défaut dans un des modes "réception", "conférence", "envoi/réception", "bouclage arrière de réseau" ou "essai de continuité réseau". La detection d'événement s'applique seulement aux données entrantes. Les connexions en mode "envoi seul", "inactive", "bouclage arrière", ou "essai de continuité" ne vont donc normalement pas détecter d'événements, bien que demander de le faire ne soit pas considéré comme une erreur.

Les modes "bouclage arrière" et "essai de continuité" sont utilisés durant les opérations de maintenance et d'essai de continuité. Un point d'extrémité peut avoir plus d'une connexion en mode "bouclage arrière" ou "essai de continuité". Tant qu'il y a une connexion dans ce mode particulier, et qu'aucune autre connexion sur le point d'extrémité n'est placée dans un mode différent de maintenance ou d'essai, l'opération de maintenance ou d'essai devra se poursuivre sans perturbation. Il y a deux nuances d'essai de continuité, une spécifiée par l'UIT, et une utilisée aux USA. Dans le premier cas, l'essai est un bouclage arrière. Le commutateur d'origine va envoyer une tonalité (la tonalité d'envoi) sur le circuit porteur, et attend que le commutateur de terminaison renvoie la tonalité. Si le commutateur d'origine voit la même tonalité retournée (la tonalité de retour) l'essai de continuité (COT) est réussi. Sinon, le COT a échoué. Dans le second cas, la tonalité d'envoi et la

tonalité de retour sont différentes. Le commutateur d'origine envoie une certaine tonalité d'envoi. Le commutateur de terminaison détecte la tonalité d'envoi, il indique une tonalité de retour différente dans la direction inverse. Quand le commutateur d'origine détecte la tonalité de retour, le COT est réussi. Si le commutateur d'origine ne détecte pas de tonalité de retour, le COT a échoué.

Si le mode est réglé à "loopback", la passerelle est supposée retourner le signal entrant provenant du point d'extrémité au même point d'extrémité. Cette procédure va être utilisée, normalement, pour tester la continuité des circuits de jonction conformément aux spécifications de l'UIT. Si le mode est réglé à "essai de continuité", la passerelle est informée que l'autre extrémité du circuit a initié une procédure d'essai de continuité conformément à la spécification GR (voir [LSSGR]). La passerelle va placer le circuit en mode transpondeur requis pour les essais de continuité bi-tonalités.

Si le mode est réglé à "netwloop", les signaux audio reçus de la connexion vont être en écho sur la même connexion. Le support n'est pas transmis au point d'extrémité.

Si le mode est réglé à "netwtest", la passerelle va traiter les paquets reçus de la connexion en accord avec le mode de transpondeur requis pour l'essai de continuité en bi-tonalité, et renvoyer le signal traité sur la connexion. Le support n'est pas transmis au point d'extrémité. Le mode "essai de continuité du réseau" est seulement inclus pour la rétro compatibilité et son utilisation est déconseillée.

2.3.2 EndpointConfiguration

La commande EndpointConfiguration peut être utilisée pour spécifier le codage des signaux qui vont être reçus par le point d'extrémité. Par exemple, dans certaines configurations de la téléphonie internationale, certains appels vont porter des signaux audio codés en loi mu, alors que d'autres vont utiliser la loi A. L'agent d'appel peut utiliser la commande EndpointConfiguration pour passer cette information à la passerelle. La configuration peut varier appel par appel, mais peut aussi être utilisée en l'absence de toute connexion.

```
ReturnCode,
[PackageList]
<-- EndpointConfiguration(EndpointId, [BearerInformation])
```

EndpointId *(identifiant de point d'extrémité)* est le nom du point d'extrémité dans la passerelle où EndpointConfiguration s'exécute. La convention de caractère générique "any of" NE DOIT PAS être utilisée. Si la convention de caractère générique "all of" est utilisée, la commande s'applique à tous les points d'extrémité dont le nom correspond au caractère générique.

BearerInformation *(informations de porteuse)* est un paramètre qui définit le codage des données envoyées au et reçues du côté ligne. Les informations sont codées comme une liste de sous paramètres. Le seul sous paramètre défini dans la présente version de la spécification est le codage de porteuse, dont la valeur peut être réglée à "loi A" ou "loi mu". Le réglage des sous paramètres peut être étendu.

Afin de permettre l'extensibilité, tout en restant rétro compatible, le paramètre BearerInformation est conditionellement facultatif dans les conditions suivantes :

* si des paramètres d'extension (fabricant, paquetage ou autre) ne sont pas utilisés, le paramètre BearerInformation est EXIGÉ,
* autrement, le paramètre BearerInformation est FACULTATIF.

Quand il est omis, BearerInformation DOIT conserver sa valeur courante.

ReturnCode *(code de retour)* est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList *(liste de paquetages)* est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (paquetage non pris en charge).

2.3.3 NotificationRequest

La commande NotificationRequest est utilisée pour demander à la passerelle d'envoyer des notifications sur l'occurrence d'événements spécifiés dans un point d'extrémité. Par exemple, une notification peut être demandée pour quand une passerelle détecte qu'un point d'extrémité reçoit des tonalités associées à une communication de télécopie. L'entité qui

reçoit cette notification peut alors décider de spécifier l'utilisation d'un type différent de méthode de codage dans les connexions liées à ce point d'extrémité et pour donner des instructions en conséquence à la passerelle avec une commande ModifyConnection.

ReturnCode,
[PackageList]
<-- NotificationRequest(EndpointId,
[NotifiedEntity,]
[RequestedEvents,]
RequestIdentifier,
[DigitMap,]
[SignalRequests,]
[QuarantineHandling,]
[DetectEvents,]
[encapsulated EndpointConfiguration])

EndpointId est l'identifiant du point d'extrémité dans la passerelle où la NotificationRequest s'exécute. Le caractère générique "any of" NE DOIT PAS être utilisé.

NotifiedEntity *(entité notifiée)* est un paramètre facultatif qui spécifie une nouvelle "entité notifiée" pour le point d'extrémité.

RequestIdentifier *(identifiant de demande)* est utilisé pour corréler cette demande avec les notifications qu'elle déclenche. Il va être répété dans la commande Notify correspondante.

RequestedEvents *(événements demandés)* est une liste d'événements, éventuellement qualifiés par des paramètres d'événement (voir le paragraphe 3.2.2.4) qu'il est demandé à la passerelle de détecter et rapporter. Ces événements peuvent inclure, par exemple, des tonalités de télécopie, des tonalités de continuité, ou une transition à raccroché. Sauf mention contraire, les événements sont détectés sur le point d'extrémité, cependant certains événements peuvent être détectés sur une connexion. Un événement donné NE DOIT PAS apparaître plus d'une fois dans un RequestedEvents. Si le paramètre est omis, il est vide par défaut.

À chaque événement est associée une ou plusieurs actions, qui peuvent être :
* Notifier l'événement immédiatement, avec la liste des événements observés accumulés,
* Passer à l'audio,
* Accumuler les événements dans une mémoire tampon d'événements, mais ne pas les notifier tout de suite,
* Accumuler en accord avec le script de numérotation,
* Garder le ou les signaux actifs,
* Traiter la demande de notification incorporée,
* Ignorer l'événement.

La prise en charge de Notifier, Accumuler, Garder le ou les signalux actifs, Traiter la demande de notification incorporée, et Ignorer est EXIGÉE. La prise en charge de Accumuler en accord avec le script de numérotation est EXIGÉE sur tout point d'extrémité capable de détecter le DTMF. La prise en charge de toute autre action est FACULTATIVE. L'ensemble d'actions peut être étendu.

Une certaine action peut être par défaut spécifiée pour tout événement, bien que certaines actions n'aient pas de sens pour tous les événements. Par exemple, un événement de décroché avec le Accumuler en accord avec le script de numérotation action est valide, mais va bien sûr déclencher immédiatement une discordance de script de numérotation quand l'événement de décroché se produit. Il n'est pas besoin de dire qu'une telle pratique est déconseillée.

Certaines actions peuvent être combinées comme montré dans le tableau ci-dessous, où "O" signifie que les deux actions peuvent être combinées, et "N" signifie qu'elles ne le peuvent pas:

	Notif	Swap	Accum	AccDi	KeSiA	EmbNo	Ignore
Notif	N	O	N	N	O	O*	N
Swap	-	N	O	N	N	N	O
Accum	-	-	N	N	O	O	N
AccDi	-	-	-	N	O	N	N
KeSiA	-	-	-	-	N	O	O
EmbNo	-	-	-	-	-	N	N
Ignore	-	-	-	-	-	-	N

Note (*) : la "demande de notification incorporée" peut seulement être combinée avec "Notifier", si il est permis à la passerelle de produire plus d'une commande Notifier par demande de notification (voir ci-dessous et au paragraphe 4.4.1).

Si aucune action n'est spécifiée, l'action Notifier va être appliquée. Si une ou plusieurs actions sont spécifiées, seules ces actions s'appliquent. Quand deux actions ou plus sont spécifiées, chaque action DOIT être combinable avec toutes les autres actions comme défini dans le tableau ci-dessus - les actions individuelles sont supposées se produire simultanément.

Si un client reçoit une demande avec une action invalide ou non prise en charge ou une combination illégale d'actions, il DOIT retourner une erreur à l'agent d'appel (le code d'erreur 523 - combination d'actions inconnue ou illégale, est RECOMMANDÉ).

En plus du paramètre RequestedEvents spécifié dans la commande, certains paquetages MGCP peuvent contenir des "événements persistants" (ceci est cependant généralement déconseillé - voir l'Appendice B pour une solution de remplacement). Les événements persistants dans un certain paquetage sont toujours détectés sur un point d'extrémité qui met en œuvre ce paquetage. Si un événement persistant n'est pas inclus dans la liste des RequestedEvents, et si l'événement se produit, l'événement va être détecté de toutes façons et traité comme tous les autres événements, comme si l'événement persistant avait été demandé avec une action Notifier. Une NotificationRequest DOIT cependant quand même être en place pour qu'un événement déclenche un Notifier. Donc, de façon informelle, les événements persistants peuvent être vus comme étant toujours implicitement inclus dans la liste des RequestedEvents avec une action de Notifier, bien qu'aucune détection ne soit effectuée.

Les événements non persistants sont les événements qui ont besoin d'être explicitement inclus dans la liste des RequestedEvents. La liste (éventuellement vide) des événements demandés remplace complètement la liste précédente d'événements demandés. En plus des événements persistants, seuls les événements spécifiés dans la liste des événements demandés vont être détectés par le point d'extrémité. Si un événement persistant est inclus dans la liste de RequestedEvents, l'action spécifiée va remplacer l'action par défaut associée à l'événement pour la durée de vie de la liste RequestedEvents, après quoi l'action par défaut va être restaurée. Par exemple, si "décroché" était un événement persistant, si l'action "Ignorer le décroché" était spécifiée, et si une nouvelle demande sans instruction de décroché a été reçue, l'opération "Notifier le décroché" par défaut va être restaurée.

La passerelle va détecter l'union des événements persistants et des événements demandés. Si un événement n'est inclus dans aucune des listes, il va être ignoré

L'agent d'appel peut envoyer une NotificationRequest avec une liste vide (ou omise) de RequestedEvents à la passerelle. L'agent d'appel peut le faire, par exemple, à une passerelle quand il ne veut plus collecter de chiffres DTMF. Cependant, les événements persistants vont encore être détectés et notifiés.

L'action Swap Audio peut être utilisée quand une passerelle traite plus d'une connexion sur un point d'extrémité. Cela va être le cas pour l'appel en attente, et éventuellement d'autres scénarios. Afin d'éviter l'aller-retour avec l'agent d'appel quand c'est juste le changement de la connexion de rattachement aux fonctions audio du point d'extrémité, la NotificationRequest peut se transposer en un événement (généralement une impulsion crochet, mais pourrait être un autre événement) à une fonction d'échange audio locale, qui choisit la "prochaine" connexion à la façon d'un round robin. Si il y a seulement une connexion, cette action est effectivement une non opération. Si il y a plus de deux connexions, l'ordre est indéfini. Si le point d'extrémité a exactement deux connexions, dont une est "inactive", et l'autre est dans le mode "envoi/réception", alors le changement audio va tenter de rendre la connexion "envoi/réception" "inactive", et vice versa. La présente spécification ne fournit intentionnellement pas de détail supplémentaire sur l'action de changement audio.

Si il est désiré qu'un ou des signaux démarrent quand un événement recherché se produit, l'action "NotificationRequest" incorporée peut être utilisée. La NotificationRequest incorporée peut aussi inclure une nouvelle liste de RequestedEvents, SignalRequests et un nouveau script de numérotation. La sémantique de la NotificationRequest incorporée est comme si une nouvelle NotificationRequest était juste reçue avec les mêmes NotifiedEntity, RequestIdentifier, QuarantineHandling et

DetectEvents. Quand la "NotificationRequest" incorporée est activée, la "chaîne de numérotation courante" va être supprimée ; cependant la liste des événements observés et la mémoire tampon de quarantaine ne seront pas affectées (si elles sont combinées avec un Notifier, le Notifier va cependant supprimer la liste des événements observés - voir le paragraphe 4.4.1). Noter que l'action NotificationRequest incorporée n'accumule pas d'événement déclenchant, cependant elle peut être combinée avec l'action Accumulate pour réaliser cela. Si la NotificationRequest incorporée échoue, un événement Échec de demande de notification incorporée DEVRAIT être généré (voir l'Appendice B).

Les mises en œuvre de MGCP DEVRONT être capables de prendre en charge au moins un niveau d'incorporation. Une NotificationRequest incorporée qui respecte cette limitation NE DOIT PAS contenir une autre NotificationRequest incorporée.

DigitMap est un paramètre facultatif qui permet à l'agent d'appel de provisionner le point d'extrémité avec un script de numérotation conformément auquel les chiffres vont être accumulés. Si ce paramètre facultatif est absent, la valeur définie précédemment est conservée. Ce paramètre DOIT être défini, explicitement ou par une commande précédente, si le paramètre RequestedEvents contient une demande de "accumuler en accord avec le script de numérotation". La collection de ces chiffres va résulter en une chaîne de chiffres. La chaîne de chiffres est initialisée à une chaîne nulle à réception de la NotificationRequest, de sorte qu'une notification suivante retourne seulement les chiffres qui ont été collectés après cette demande. Les chiffres qui ont été accumulés en accord avec le script de numérotation sont rapportés comme tout autre événement accumulé, dans l'ordre dans lequel ils arrivent. Il est donc possible que d'autres événements accumulés se trouvent entre la liste des chiffres. Si il est demandé à la passerelle "d'accumuler en accord avec le script de numérotation" et si la passerelle n'a pas actuellement de script de numérotation pour le point d'extrémité en question, la passerelle DOIT retourner une erreur (le code d'erreur 519 - le point d'extrémité n'a pas de script de numérotation, est RECOMMANDÉ).

SignalRequests est un paramètre facultatif qui contient l'ensemble des signaux qu'il est demandé à la passerelle d'appliquer. Quand il est omis, il est vide par défaut. Quand plusieurs signaux sont spécifiés, les signaux DOIVENT être appliqués en parallèle. Sauf mention contraire, les signaux sont appliqués au point d'extrémité. Cependant certains signaux peuvent être appliqué à une connexion. Les signaux sont identifiés par leur nom, qui est un nom d'événement, et peuvent être qualifiés par des paramètres de signal (voir le paragraphe 3.2.2.4). Les signaux suivants sont des exemples:
* Sonnerie
* tonalité d'occupation,
* tonalité d'appel en attente,
* tonalité d'annonce de décroché,
* tonalités de retour d'appel sur une connexion.

Les noms et les descriptions de signaux sont définis dans le paquetage approprié.

Les signaux sont, par défaut, appliqués aux points d'extrémité. Si un signal appliqué à un point d'extrémité résulte en la génération d'un flux de supports (audio, vidéo, etc.) alors par défaut le flux de supports NE DOIT PAS être transmis sur une connexion associée à ce point d'extrémité, sans considération du mode de la connexion. Par exemple, si une tonalité d'appel en instance est appliquée à un point d'extrémité impliqué dans un appel actif, seule la partie utilisant le point d'extrémité en question va entendre la tonalité d'appel en instance. Cependant, des signaux individuels peuvent définir un comportement différent.

Quand un signal est appliqué à une connexion qui a reçu un RemoteConnectionDescriptor *(descripteur de connexion distante)* le flux de supports généré par ce signal va être transmis sur la connexion sans considération du mode actuel de la connexion (incluant les essais de bouclage arrière et de continuité). Si un RemoteConnectionDescriptor n'a pas été reçu, la passerelle DOIT retourner un code d'erreur (le code d'erreur 527 - RemoteConnectionDescriptor manquant, est RECOMMANDÉ). Noter que cette restriction ne s'applique pas à la détection d'événements sur une connexion.

Quand une liste (éventuellement vide) de signaux est fournie, cette liste remplace complètement la liste courante des signaux actifs de fin de temporisation. Les signaux actuellement actifs de fin de temporisation qui ne sont pas fournis dans la nouvelle liste DOIVENT être arrêtés et les nouveaux signaux fournis vont alors devenir actifs. Les signaux actuellement actifs de fin de temporisation qui sont fournis dans la nouvelle liste de signaux DOIVENT rester actifs sans interruption, et donc le temporisateur pour de tels signaux de fin de temporisation ne sera pas affecté. Par conséquent, il n'y a actuellement aucun moyen de redémarrer le temporisateur pour un signal de fin de temporisation actuellement actif sans arrêter d'abord le signal. Si le signal de fin de temporisation est paramétré, l'ensemble original de paramètres DOIT rester en effet, sans considération des valeurs qui seront fournies ensuite. Un signal donné NE DOIT PAS apparaître plus d'une fois dans une SignalRequests. Noter qu'appliquer un signal S à un point d'extrémité, à une connexion C1 et à une connexion C2, constitue trois signaux différents et indépendants.

L'action déclenchée par SignalRequests est synchronisée avec la collection des événements spécifiée dans le paramètre

RequestedEvents. Par exemple, si la NotificationRequest rend obligatoire "sonnerie" et si les RequestedEvents demandent de chercher un événement "décroché", la sonnerie DEVRA s'arrêter aussitôt que la passerelle détecte un événement de décroché. La définition formelle est que la génération de tous les signaux "Fin de temporisation" DEVRA s'arrêter aussitôt qu'un des événements demandés est détecté, sauf si l'action "Garder les signaux actifs" est associée à l'événement détecté. Les RequestedEvents et SignalRequests peuvent se référer aux mêmes définitions d'événement. Dans un cas, il est demandé à la passerelle de détecter l'occurrence de l'événement, et dans l'autre cas, il lui est demandé de le générer. Les événements et signaux spécifiques qu'un certain point d'extrémité peut détecter ou effectuer sont déterminés par la liste des paquetages qui sont supportés par ce point d'extrémité. Chaque paquetage spécifie une liste d'événements et signaux qui peuvent être détectés ou effectués. Une passerelle à qui il est demandé de détecter ou effectuer un événement appartenant à un paquetage qui n'est pas pris en charge par le point d'extrémité spécifié DOIT retourner une erreur (le code d'erreur 518 - Paquetage non pris en charge ou inconnu, est RECOMMANDÉ). Quand le nom d'événement n'est pas qualifié par un nom de paquetage, le nom de paquetage par défaut pour le point d'extrémité est supposé. Si le nom d'événement n'est pas enregistré dans ce paquetage par défaut, la passerelle DOIT retourner une erreur (le code d'erreur 522 - Pas de tel événement ou signal, est RECOMMANDÉ).

L'agent d'appel peut envoyer une NotificationRequest dont la liste des signaux demandés est vide. Il va faire cela par exemple quand un ou des signaux de fin de temporisation devraient s'arrêter.

Si il est désiré que le ou les signaux commencent aussitôt que l'événement "recherché" se produit, l'action "NotificationRequest incorporée" peut être utilisée. La demande de notification incorporée peut inclure aussi une nouvelle liste de RequestedEvents, SignalRequests et un nouveau script de numérotation. L'action de demande de notification incorporée permet à l'agent d'appel d'établir un "mini-script" à traiter par la passerelle immédiatement à la suite de la détection de l'événement associé. Toute SignalRequests spécifiée dans la demande de notification incorporée va commencer immédiatement. Un soin particulier doit être apporté à empêcher des discordances entre l'agent d'appel et la passerelle. Cependant, des discordances à long terme ne devraient pas se produire car une nouvelle SignalRequests remplace complètement l'ancienne liste de signaux actifs de fin de temporisation, et les signaux de type BR s'arrêtent toujours d'eux-mêmes. Limiter le nombre de signaux de type Ouvert/Fermé est encouragé. Il est considéré de bonne pratique pour un agent d'appel d'ouvrir occasionnellement les signaux Ouvert/Fermé qui devraient être ouverts, et de fermer tous les signaux Ouvert/Fermé qui devraient être fermés.

L'action Ignorer peut être utilisée pour ignorer un événement, par exemple, pour empêcher un événement persistant d'être notifié. Cependant, la synchronisation entre l'événement et un signal actif de temporisation va quand même se produire par défaut (par exemple, un signal de temporisation de tonalité de numérotation va s'arrêter quand un événement de décroché se produit, même si le décroché était un événement demandé avec l'action "Ignorer"). Pour empêcher cette synchronisation de se produire, l'action "Garder le signal actif" va aussi être spécifiée.

Le paramètre facultatif QuarantineHandling *(traitement de quarantaine)* spécifie le traitement des événements de "quarantaine", c'est-à-dire, des événements qui ont été détectés par la passerelle avant l'arrivée de cette commande NotificationRequest, mais n'ont pas encore été notifiés à l'agent d'appel. Le paramètre fournit un ensemble d'options de traitement (voir les détails au paragraphe 4.4.1) :

* si les événements en quarantaine devraient être traités ou éliminés (il sont traités par défaut) ;
* si la passerelle est supposée générer au plus une notification (étape par étape) ou plusieurs notifications (boucle) en réponse à cette demande (au plus une par défaut).

Quand le paramètre est absent, la valeur par défaut est supposée.

On devrait noter que le paramètre Traitement de quarantaine gouverne aussi le traitement des événements qui ont été détectés et traités mais non encore notifiés quand la commande est reçue.

DetectEvents *(détecter les événements)* est un paramètre facultatif, éventuellement qualifié par des paramètres d'événement qui spécifient une liste d'événements qu'il est demandé à la passerelle de détecter durant la période de quarantaine. Quand ce paramètre est absent, les événements à détecter dans la période de quarantaine sont ceux de la dernière liste d'événements détectés reçue. De plus, la passerelle va aussi détecter les événements persistants et les événements spécifiés dans la liste RequestedEvents *(événements demandés)* incluant ceux pour lesquels l'action "ignore" est spécifiée.

Certains événements et signaux, comme le retour d'appel en ligne ou l'alerte sur la qualité, sont effectués ou détectés sur des connexions qui se terminent au point d'extrémité plutôt que sur le point d'extrémité lui-même. La structure des noms d'événement (voir au paragraphe 2.1.7) permet à l'agent d'appel de spécifier la ou les connexions sur lesquelles les événements devraient être effectués ou détectés.

La commande NotificationRequest peut porter une commande EndpointConfiguration *(configuration de point d'extrémité)*

encapsulée, qui va s'appliquer aux mêmes points d'extrémité. Quand cette commande est présente, les paramètres de la commande EndpointConfiguration sont inclus avec les paramètres normaux de la demande de notification, à l'exception de EndpointId *(identifiant de point d'extrémité)* qui n'est pas reproduit.

La commande EndpointConfiguration encapsulée partage le sort de la commande NotificationRequest. Si la demande de notification est rejetée, la configuration de point d'extrémité n'est pas exécutée.

ReturnCode *(code de retour)* est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.4 Notify

Les notifications avec les événements observés sont envoyées par la passerelle via la commande Notify quand un événement déclenchant se produit.

```
ReturnCode,
[PackageList]
<-- Notify(EndpointId,
        [NotifiedEntity,]
        RequestIdentifier,
        ObservedEvents)
```

EndpointId est le nom du point d'extrémité dans la passerelle qui produit la commande Notify. L'identifiant DOIT être un identifiant de point d'extrémité pleinement qualifié, incluant le nom de domaine de la passerelle. La partie locale du nom NE DOIT PAS utiliser de convention de caractères génériques.

NotifiedEntity est un paramètre qui identifie l'entité qui a demandé la notification. Ce paramètre est égal au paramètre NotifiedEntity de la demande de notification qui a déclenché cette notification. Le paramètre est absent si il n'y a pas un tel paramètre dans la demande déclenchante. Sans considération de la valeur du paramètre NotifiedEntity, la notification DOIT être envoyée à la "entité notifiée" en cours pour le point d'extrémité.

RequestIdentifier est un paramètre qui répète le paramètre RequestIdentifier de la demande de notification qui a déclenché cette notification. Il est utilisé pour corréler cette notification avec la demande qui l'a déclenchée. Les événements persistants vont être vus comme si ils avaient été inclus dans la dernière demande de notification. Une NotificationRequest implicite PEUT être en place juste après le redémarrage - le RequestIdentifier utilisé pour elle va être zero ("0") - voir les détails au paragraphe 4.4.1.

ObservedEvents *(événements observés)* est une liste des événements que la passerelle a détectés et accumulés. Une seule notification peut rapporter une liste des événements qui vont être rapportés dans l'ordre dans lequel ils ont été détectés (FIFO).

La liste va seulement contenir l'identification des événements qui étaient demandés dans le paramètre RequestedEvents de la demande de notification déclencheuse. Elle va contenir les événements qui ont été accumulés (mais non notifiés) ou traités conformément au script de numérotation (mais pas encore confrontés) et l'événement final qui a déclenché la notification ou fourni une confrontation finale au script de numérotation. On devrait noter que les chiffres DOIVENT être ajoutés à la liste des événements observés comme ils sont accumulés, sans considération de si ils sont accumulés en accord ou non au script de numérotation. Par exemple, si un utilisateur entre les chiffres "1234" et si un événement E est accumulé entre les chiffres "3" et "4" qui sont entrés, la liste des événements observés va être "1, 2, 3, E, 4". Les événements qui ont été détectés sur une connexion DEVRONT inclure le nom de cette connexion comme dans "R/qa@0A3F58" (paragraphe 2.1.7).

Si la liste des événements observés dépasse la capacité du point d'extrémité, un événement ObservedEvents pleins (voir l'Appendice B) DEVRAIT être généré (le point d'extrémité devra s'assurer qu'il a la capacité d'inclure cet événement dans la liste des ObservedEvents). Si l'événement ObservedEvents pleins n'est pas utilisé pour déclencher un Notify, le traitement d'événement continue comme avant (incluant la confrontation au script de numérotation) ; cependant, les événements suivants ne vont pas être inclus dans la liste des événements observés.

ReturnCode est un paramètre retourné par l'agent d'appel. Il indique le résultat de la commande et consiste en un nombre

entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.5 CreateConnection

Cette commande est utilisée pour créer une connexion entre deux points d'extrémité.

```
ReturnCode,
[ConnectionId,]
[SpecificEndPointId,]
[LocalConnectionDescriptor,]
[SecondEndPointId,]
[SecondConnectionId,]
[PackageList]
<-- CreateConnection(CallId,
            EndpointId,
            [NotifiedEntity,]
            [LocalConnectionOptions,]
            Mode,
            [{RemoteConnectionDescriptor |
            SecondEndpointId}, ]
            [Encapsulated NotificationRequest,]
            [Encapsulated EndpointConfiguration])
```

Une connexion est définie par ses points d'extrémité. Les paramètres d'entrée dans CreateConnection fournissent les données nécessaires pour construire la "vue" d'une connexion par une passerelle.

CallId (*identifiant d'appel)* est un paramètre qui identifie l'appel (ou session) auquel appartient cette connexion. Ce paramètre DEVRAIT, au minimum, être unique au sein de la collection des agents d'appel qui contrôlent les mêmes passerelles. Les connexions qui appartiennent au même appel DEVRAIENT partager le même identifiant d'appel *(call-id)*. L'identifiant d'appel a peu de signification sémantique dans le protocole ; cependant il peut être utilisé pour identifier les appels pour les besoins de rapport et de comptabilité. Il n'affecte pas le traitement des connexions par la passerelle.

EndpointId est l'identifiant pour le point d'extrémité de la connexion dans la passerelle où CreateConnection s'exécute. Le EndpointId peut être pleinement spécifié en allouant une valeur au paramètre EndpointId dans l'invocation de fonction, ou il peut être sous spécifié en utilisant la convention de caractère générique "any of" *(n'importe lequel de)*. Si le point d'extrémité est sous spécifié, l'identifiant de point d'extrémité DEVRA être alloué par la passerelle et sa valeur complète retournée dans le paramètre SpecificEndPointId *(identifiant de point d'extrémité spécifique)* de la réponse. Quand le caractère générique "any of" est utilisé, le point d'extrémité alloué DOIT être en service et NE DOIT PAS avoir déjà de connexions sur lui. Si aucun point d'extrémité de ce type n'est disponible, le code d'erreur 410 (Pas de point d'extrémité disponible) DEVRAIT être retourné. Le caractère générique "all of" *(tous les)* NE DOIT PAS être utilisé.

NotifiedEntity *(entité notifiée)* est un paramètre facultatif qui spécifie une nouvelle "entité notifiée" pour le point d'extrémité.

LocalConnectionOptions *(options de connexion locale)* est une structure facultative utilisée par l'agent d'appel pour diriger le traitement de la connexion par la passerelle. Les champs contenus dans une structure LocalConnectionOptions peuvent inclure un ou plusieurs de ceux qui suivent (aucun champ NE DOIT être fourni plus d'une fois) :

* Algorithme de compression de codec : un ou plusieurs codecs, dans l'ordre de préférence. Pour l'interopérabilité, il est RECOMMANDÉ de prendre en charge le codage de loi μ ("PCMU") de la Recommandataion UIT-T G.711. Voir au paragraphe 2.6 les détails du processus de choix du codec.

* Période de mise en paquets : une seule valeur en millisecondes ou une gamme peut être spécifiée. La période de mise en paquets NE DEVRAIT PAS contredire la spécification de l'algorithme de compression de codec. Si un codec est spécifié avec une taille de trame incohérente avec la période de mise en paquets, et si ce codec est choisi, la passerelle est autorisée à utiliser une période de mise en paquets cohérente avec la taille de trame même si elle est différente de celle spécifiée. Ce faisant, la passerelle DEVRAIT choisir une période de mise en paquet non zéro aussi proche que possible de celle spécifiée. Si aucune période de mise en paquet n'est spécifiée, le point d'extrémité DEVRAIT utiliser

la ou les périodes de mise en paquet par défaut pour le ou les codecs choisis.

* Bande passante : la bande passante admissible, c'est-à-dire, la charge utile plus tous frais généraux d'en-tête de la couche transport et au dessus, par exemple, IP, UDP, et RTP. La spécification de bande passante NE DEVRAIT PAS contredire la spécification de l'algorithme de compression du codec ou de la période de mise en paquets. Si un codec est spécifié, alors la passerelle est autorisée à l'utiliser, même si il en résulte l'utilisation d'une plus grande bande passante que ce qui est spécifié. Une discordance entre la bande passante et la spécification du codec ne sera pas rapportée comme une erreur.

* Type de service : cela indique la classe du service à utiliser pour cette connexion. Quand le type de service n'est pas spécifié, la passerelle DEVRA utiliser une valeur par défaut de zéro sauf si elle est autrement provisionnée.

* Usage de l'annulation d'écho : par défaut, les passerelles de téléphonie effectuent toujours l'annulation d'écho sur le point d'extrémité. Cependant, il peut être nécessaire, pour certains appels, de désactiver ces opérations. Le paramètre Annulation d'écho peut avoir deux valeurs, "activé" (quand l'annulation d'écho est demandée) et "désactivé" (quand il est désactivé). Le paramètre est facultatif. Si le paramètre est omis à la création d'une connexion et si il n'y a pas d'autre connexion sur le point d'extrémité, le point d'extrémité DEVRA appliquer initialement l'annulation d'écho. Si le paramètre est omis à la création d'une connexion et si il y a des connexions existantes sur le point d'extrémité, l'annulation d'écho est inchangée. Le point d'extrémité DEVRAIT ensuite activer ou désactiver l'annulation d'écho quand des données de bande vocale sont détectées - voir par exemple, les Recommandations UIT-T V.8, V.25, et G.168. Suite à la fin des données de bande vocale, le traitement de l'annulation d'écho DEVRA alors revenir à la valeur courante du paramètre d'annulation d'écho. Il est RECOMMANDÉ que le traitement de l'annulation d'écho soit laissé à la passerelle plutôt que d'avoir ce paramètre spécifié par l'agent d'appel.

* Suppression de silence : les passerelles de téléphonie peuvent effectuer la détection d'activité vocale, et éviter d'envoyer des paquets durant les périodes de silence. Cependant, il est nécessaire, par exemple pour les appels par modem, de désactiver cette détection. Le paramètre Suppression de silence peut avoir deux valeurs, "on" (quand la détection est demandée) et "off" (quand elle n'est pas demandée). La valeur par défaut est "off" (sauf provisionnement différent). À la détection de données de bande vocale, le point d'extrémité DEVRAIT désactiver la suppression de silence. À la suite de la terminaison des données de bande vocale, le traitement de la suppression de silence DEVRA alors revenir à la valeur courante du paramètre Suppression de silence.

* Contrôle de gain : les passerelles de téléphonie peuvent effectuer le contrôle de gain sur le point d'extrémité, afin d'adapter le niveau du signal. Cependant, il est nécessaire, par exemple pour certains appels de modem, de désactiver cette fonction. Le paramètre Contrôle de gain peut soit être spécifié comme "automatique", soit comme un nombre explicite de décibels de gain. Le gain spécifié va être ajouté au support envoyé sur le point d'extrémité (par opposition à sur la connexion) et soustrait des supports reçus sur le point d'extrémité. Le paramètre est facultatif. Quand il n'y a pas d'autre connexion sur le point d'extrémité, et que le paramètre est omis, la valeur par défaut est de ne pas effectuer de contrôle de gain (sauf provisionné autrement) ce qui est équivalent à spécifier un gain de 0 décibel. Si il y a d'autres connexions sur le point d'extrémité, et si le paramètre est omis, le contrôle de gain est inchangé. À la détection de données de bande vocale, le point d'extrémité DEVRAIT désactiver le contrôle de gain si nécessaire. À la suite de la terminaison des données de bande vocale, le traitement du contrôle de gain DEVRA alors revenir à la valeur courante du paramètre Contrôle de gain. Ont devrait noter que le traitement du contrôle de gain est normalement laissé à la passerelle et donc l'utilisation de ce paramètre n'est PAS RECOMMANDÉE.

* Sécurité RTP : l'agent d'appel peut demander à la passerelle d'activer le chiffrement des paquets audio. Il le fait en fournissant une spécification de clé, comme spécifié dans la RFC 2327. Par défaut, le chiffrement n'est pas effectué.

* Type de réseau : l'agent d'appel peut donner pour instruction à la passerelle de préparer la connexion sur un type de réseau spécifié. Si il est absent, la valeur est fondée sur le type de réseau de la passerelle utilisée.

* Réservation de ressource : l'agent d'appel peut donner pour instruction à la passerelle d'utiliser la réservation de ressources du réseau pour la connexion. Voir les détails au paragraphe 2.7.

L'agent d'appel spécifie les champs pertinents dont il se soucie dans la commande et laisse le reste à la discrétion de la passerelle. Pour ceux des paramètres ci-dessus qui ne sont pas explicitement inclus, la passerelle DEVRAIT utiliser les valeurs par défaut si possible. Pour une liste détaillée des options de connexion locale incluses dans la présente spécification, se reporter au paragraphe 3.2.2.10. L'ensemble des options de connexion locale peut être étendu.

Le mode indique le mode de fonctionnement pour ce côté de la connexion. Les modes de base sont "send" *(envoi)* "receive" *(réception)* "send/receive", "conference", "inactive", "loopback" *(rebouclage)* "continuity test" *(essai de continuité)*

"network loop back" *(rebouclage réseau)* et "network continuity test" *(essai de continuité réseau)*. Le traitement attendu de ces modes est spécifié dans l'introduction de "Commandes de contrôle de passerelle" au paragraphe 2.3. Noter que les signaux appliqués à une connexion ne suivent pas le mode de connexion. Certains points d'extrémité peuvent n'être pas capables de prendre en charge tous les modes. Si la commande spécifie un mode que le point d'extrémité ne prend pas en charge, une erreur DEVRA être retournée (l'erreur 517 - mode non pris en charge, est RECOMMANDÉE). Aussi, si une connexion n'a pas encore reçu un RemoteConnectionDescriptor *(descripteur de connexion distante)* une erreur DOIT être retournée si on tente de placer la connexion dans un des modes "send only", "send/receive", "conference", "network loopback", "network continuity test", ou si un signal (par opposition à la détection d'un événement) doit être appliqué à la connexion (le code d'erreur 527 - Descripteur de connexion distante manquant, est RECOMMANDÉ). L'ensemble des modes peut être étendu.

La passerelle retourne un identifiant de connexion, qui identifie de façon univoque la connexion au sein du point d'extrémité, et un LocalConnectionDescriptor *(descripteur de connexion locale)* qui est une description de session contenant des informations sur la connexion, par exemple, adresse IP et accès pour le support, comme défini dans SDP.

SpecificEndPointId *(identifiant spécifique de point d'extrémité)* est un paramètre facultatif qui identifie le point d'extrémité qui répond. Il est retourné quand l'argument EndpointId se réfère à un nom avec un caractère générique de "any of" et que la commande a réussi. Quand un SpecificEndPointId est retourné, l'agent d'appel DEVRA l'utiliser comme valeur de EndpointId dans les commandes successives qui se réfèrent à cette connexion.

SecondEndpointId *(second identifiant de point d'extrémité)* peut être utilisé à la place du RemoteConnectionDescriptor pour établir une connexion entre deux points d'extrémité situés sur la même passerelle. La connexion est par définition une connexion locale. Le SecondEndpointId peut être pleinement spécifié en allouant une valeur au paramètre SecondEndpointId dans l'invocation de la fonction, ou il peut être sous spécifié en utilisant la convention de caractère générique "any of". Si le SecondEndpointId est sous spécifié, le second identifiant de point d'extrémité va être alloué par la passerelle et sa valeur complète sera retournée dans le paramètre SecondEndPointId de la réponse.

Quand un SecondEndpointId est spécifié, la commande crée en réalité deux connexions qui peuvent être manipulées séparément par les commandes ModifyConnection et DeleteConnection. En plus du ConnectionId et LocalConnectionDescriptor pour la première connexion, la réponse à la création fournit un paramètre SecondConnectionId qui identifie la seconde connexion. La seconde connexion est établie en mode "send/receive".

Après la réception d'une demande "CreateConnection" qui n'incluait pas de paramètre RemoteConnectionDescriptor, une passerelle est dans une situation ambiguë. Parce qu'elle a exporté unparamètre LocalConnectionDescriptor, elle peut recevoir des paquets. Parce qu'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 par l'agent d'appel. Elle doit donc naviguer entre deux risques, c'est-à-dire, couper certaines annonces importantes ou écouter des données non sûres. Le comportement de la passerelle est déterminé par la valeur du paramètre Mode :

* Si le mode était réglé à ReceiveOnly *(réception seule)* la passerelle DOIT accepter les supports et les transmettre à travers le point d'extrémité.
* Si le mode était réglé à Inactive, Loopback, ou Continuity Test, la passerelle NE DOIT PAS transmettre les supports à travers le point d'extrémité.

Noter que les valeurs de mode SendReceive, Conference, SendOnly, Network Loopback et Network Continuity Test n'ont pas de sens dans cette situation. Elles DOIVENT être traitées comme des erreurs, et la commande DOIT être rejetée (le code d'erreur 527 - RemoteConnectionDescriptor manquant, est RECOMMANDÉ).

La commande peut facultativement contenir une commande Demande de notification encapsulée, qui s'applique à l'identifiant de point d'extrémité, auquel cas un paramètre RequestIdentifier *(identifiant de demande)* DOIT être présent, ainsi que, facultativement, d'autres paramètres de la Demande de notification à l'exception de l'identifiant de point d'extrémité, qui n'est pas dupliqué. La demande de notification encapsulée est exécutée simultanément à la création de la connexion. Par exemple, quand l'agent d'appel veut initier un appel à une passerelle résidentielle, it pourrait :

* demander à la passerelle résidentielle de préparer une connexion, afin d'être sûr que l'utilisateur peut commencer à parler aussitôt que le téléphone est décroché,
* demander à la passerelle résidentielle de commencer à sonner,
* demander à la passerelle résidentielle de notifier à l'agent d'appel quand le téléphone est décroché.

Ceci peut être accompli dans une seule commande CreateConnection, en transmettant aussi les paramètres RequestedEvents pour l'événement de décroché, et le paramètre SignalRequests pour le signal de sonnerie.

Quand ces paramètres sont présents, la création et la demande de notification DOIVENT être synchronisées, ce qui signifie

que les deux DOIVENT être acceptées, ou les deux DOIVENT être refusées. Dans notre exemple, le CreateConnection peut être refusé si la passerelle n'a pas de ressources suffisantes, ou ne peut pas obtenir les ressources adéquates de l'accès réseau local, et la demande de notification de décroché peut être refusée dans la condition de double établissement, si l'utilisateur est déjà décroché. 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 est déjà décroché.

Le paramètre NotifiedEntity, si il est présent, définit la nouvelle "entité notifiée" pour le point d'extrémité.

La commande peut porter une commande encapsulée EndpointConfiguration *(configuration de point d'extrémité)* qui s'applique à l'identifiant de point d'extrémité. Quand cette commande est présente, les paramètres de la commande EndpointConfiguration sont incluses avec les paramètres normaux de CreateConnection à l'exception de l'identifiant de point d'extrémité, qui n'est pas dupliqué. La commande EndpointConfiguration peut être encapsulée avec une commande NotificationRequest encapsulée. Noter que les deux ne s'appliquent qu'au EndpointId.

La commande encapsulée EndpointConfiguration partage le sort de la commande CreateConnection. Si CreateConnection est rejetée, la configuration de point d'extrémité n'est pas exécutée.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.6 ModifyConnection

Cette commande est utilisée pour modifier les caractéristiques de la "vue" d'une connexion depuis une passerelle. Cette "vue" de l'appel inclut le descripteur de connexion locale ainsi que le descripteur de connexion distante.

```
ReturnCode,
[LocalConnectionDescriptor,]
[PackageList]
<-- ModifyConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     [NotifiedEntity,]
                     [LocalConnectionOptions,]
                     [Mode,]
                     [RemoteConnectionDescriptor,]
                     [Encapsulated NotificationRequest,]
                     [Encapsulated EndpointConfiguration])
```

Les paramètres utilisés sont les mêmes que dans la commande CreateConnection, avec l'ajout d'un ConnectionId qui identifie la connexion au sein du point d'extrémité. Ce paramètre a été retourné par la commande CreateConnection, en plus du descripteur de connexion locale. Il identifie de façon univoque la connexion au sein du contexte du point d'extrémité. Le CallId utilisé quand la connexion a été créée DOIT aussi être inclus.

Le EndpointId DOIT être un identifiant de point d'extrémité pleinement qualifié. Le nom local NE DOIT PAS utiliser de convention de caractère générique.

La commande ModifyConnection peut être utilisée pour affecter les paramètres d'une connexion de la façon suivante :

* Fournir des informations sur l'autre extrémité de la connexion, par le descripteur de connexion distante. Si le paramètre est omis, il conserve sa valeur courante.

* Activer ou désactiver la connexion, en changeant la valeur du paramètre Mode. Cela peut arriver à tout moment durant la connexion, avec des valeurs de paramètre arbitraires. Si le paramètre est omis, il conserve sa valeur courante.

* Changer les paramètres de la connexion avec les options de connexion locale, par exemple en passant à un schéma de codage différent, en changeant la période de mise en paquets, ou en modifiant le traitement de l'annulation d'écho. Si un ou plusieurs paramètres LocalConnectionOptions sont omis, la passerelle DEVRAIT alors s'abstenir de changer la valeur courante de ce paramètre, sauf si un autre paramètre nécessitant un tel changement est explicitement fourni. Par

exemple, un changement de codec pourrait exiger un changement de la suppression de silence. Noter que si un RemoteConnectionDescriptor est fourni, alors seul les LocalConnectionOptions actuellement founies avec la commande ModifyConnection vont affecter la négociation de codec (comme décrit au paragraphe 2.6).

Les connexions ne peuvent être pleinement activées que si le descripteur de connexion distante a été fourni à la passerelle. Le mode réception seule, peut cependant être activé sans la fourniture de ce descripteur.

La commande va seulement retourner un descripteur de connexion locale si les paramètres de connexion locale, comme les accès RTP, ont été modifiés. Donc, si, par exemple, seul le mode de la connexion est changé, un descripteur de connexion locale ne va pas être retourné. Noter cependant, que l'inclusion des options de connexion locale dans la commande n'est pas un prérequis pour que se produise un changement des paramètres de connexion locale. Si un paramètre de connexion est omis, par exemple, la suppression de silence, l'ancienne valeur de ce paramètre va être conservée si possible. Si un changement de paramètre nécessite un changement d'un ou plusieurs paramètres non spécifiés, la passerelle est libre de choisir des valeurs convenables pour les paramètres non spécifiés qui doivent changer. Cela peut arriver par exemple si la période de mise en paquets n'était pas spécifiée. Si le nouveau codec prend en charge l'ancienne période de mise en paquets, la valeur de ce paramètre ne va pas changer, car un changement ne serait pas nécessaire. Cependant, si il ne prend pas en charge l'ancienne période de mise en paquets, il va choisir une valeur convenable.

La commande peut facultativement contenir une commande Demande de notification encapsulée, auquel cas un paramètre RequestIdentifier DOIT être présent, ainsi que, facultativement, d'autres paramètres de la demande de notification à l'exception de l'identifiant de point d'extrémité, qui n'est pas dupliqué. La demande de notification encapsulée est exécutée simultanément à la modification de la connexion. Par exemple, quand une connexion est acceptée, la passerelle appelante devrait recevoir pour instruction de placer le circuit en mode envoi-réception et d'arrêter de fournir des tonalités de sonnerie. Cela peut être fait dans une seule commande ModifyConnection, en transmettant aussi le paramètre RequestedEvents, pour l'événement de raccroché, et un paramètre SignalRequests vide, pour arrêter la fourniture des tonnalités de sonnerie.

Quand ces paramètres sont présents, la modification et la demande de notification DOIVENT être synchronisées, ce qui signifie que les deux DOIVENT être acceptées, ou les deux DOIVENT être refusées.

Le paramètre NotifiedEntity, si il est présent, définit la nouvelle "entité notifiée" pour le point d'extrémité.

La commande peut porter une commande encapsulée EndpointConfiguration, qui va s'appliquer au même point d'extrémité. Quand cette commande est présente, les paramètres de la commande EndpointConfiguration sont incluses avec les paramètres normaux de ModifyConnection à l'exception du EndpointId, qui n'est pas dupliqué. La commande EndpointConfiguration peut être encapsulée avec une commande Demande de notification encapsulée.

La commande encapsulée EndpointConfiguration partage le sort de la commande ModifyConnection. Si ModifyConnection est rejetée, EndpointConfiguration n'est pas exécuté.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.7 DeleteConnection (provenant de l'agent d'appel)

Cette commande est utilisée pour terminer une connexion. Elle collecte de plus des statistiques sur l'exécution de la connexion.

```
ReturnCode,
ConnectionParameters,
[PackageList]
<-- DeleteConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     [NotifiedEntity,]
                     [Encapsulated NotificationRequest,]
                     [Encapsulated EndpointConfiguration])
```

L'identifiant de point d'extrémité, dans cette forme de la commande DeleteConnection, DEVRA être pleinement qualifié. Les conventions de caractères généraux NE DEVRONT PAS être utilisées.

ConnectionId identifie la connexion à supprimer. Le CallId utilisé quand la connexion a été créée est aussi inclus.

Le paramètre NotifiedEntity, si il est présent, définit la nouvelle "entité notifiée" pour le point d'extrémité.

Dans le cas de diffusion groupée IP, les connexions peuvent être supprimées individuellement et indépendamment. Cependant, dans le cas d'envoi individuel où une connexion a deux extrémités, une commande DeleteConnection doit être envoyée aux deux passerelles impliquées dans la connexion. Après la supression de la connexion, les flux de supports précédemment pris en charge par la connexion ne sont plus disponibles. Tous les paquets de supports reçus pour l'ancienne connexion sont simplement éliminés et aucun nouveau paquet de supports pour le flux n'est envoyé.

Après que la connexion a été supprimée, tout rebouclage qui avait été demandé pour la connexion doit être annulé (sauf si le point d'extrémité a une autre connexion qui demande le rebouclage).

En réponse à la commande DeleteConnection, la passerelle retourne une liste de paramètres de connexion qui décrivent les statistiques de la connexion.

Quand la connexion était pour un flux de supports Internet, ces paramètres sont :

Nombre de paquets envoyés : nombre total de paquets de supports transmis par l'envoyeur depuis le début de la transmission sur cette connexion. Dans le cas de RTP, le compte n'est pas réinitialidé si l'envoyeur change son identifiant de source de synchorisation (SSRC, *synchronization source,* comme définie dans RTP) par exemple, par suite d'une commande ModifyConnection. La valeur est zéro si la connexion a toujours été réglée en mode "réception seule" et si aucun signaux n'ont été appliqués à la connexion.

Nombre d'octets envoyés : nombre total d'octets de charge utile (c'est-à-dire, non inclus d'en-tête ou de bourrage) transmis dans les paquets de supports par l'envoyeur depuis le début de la transmission sur cette connexion. Dans le cas de RTP, le compte n'est pas réinitialisé si l'envoyeur change son identifiant de SSRC, par exemple par suite d'une commande ModifyConnection. La valeur est zéro si la connexion a toujours été réglée en mode "réception seule" et si aucun signaux n'ont été appliqués à la connexion.

Nombre de paquets reçus : nombre total de paquets de supports reçus par l'envoyeur depuis le début de la réception sur cette connexion. Dans le cas de RTP, le compte inclut les paquets reçus de SSRC différentes, si l'envoyeur a utilisé plusieurs valeurs. La valeur est zéro si la connexion a toujours été réglée en mode "envoi seul".

Nombre d'octets reçus : nombre total d'octets de charge utile (c'est-à-dire, non inclus d'en-tête, par exemple, RTP, ou de bourrage) transmis dans les paquets de supports par l'envoyeur depuis le début de la transmission sur cette connexion. Dans le cas de RTP, le compte inclut les paquets reçus de SSRC différentes, si l'envoyeur a utilisé plusieurs valeurs. La valeur est zéro si la connexion a toujours été réglée au mode "envoi seul".

Nombre de paquets perdus : nombre total de paquets de supports qui ont été perdus depuis le début de la réception. Ce nombre est défini comme le nombre de paquets attendus moins le nombre de paquets réellement reçus, où le nombre de paquets reçus inclut tous ceux qui sont en retard ou dupliqués. Pour RTP, le compte inclut les paquets reçus de SSRC différentes, si l'envoyeur a utilisé plusieurs valeurs. Donc, les paquets qui arrivent en retard ne sont pas comptés comme perdus, et la perte peut être négative si il y a des dupliqués. Le compte inclut les paquets reçus de SSRC différentes, si l'envoyeur a utilisé plusieurs valeurs. Le nombre de paquets attendus est défini comme étant le dernier numéro de séquence étendu reçu, comme défini plus loin, moins le numéro de séquence initial reçu. Le compte inclut les paquets reçus de différentes SSRC, si l'envoyeur a utilisé plusieurs valeurs. La valeur est zéro si la connexion a toujours été réglée en mode "envoi seul".

Gigue inter arrivées : estimation de la variance statistique du temps inter arrivées des paquets de supports mesuré en millisecondes et exprimée comme un entier non signé. Pour RTP, la gigue inter arrivées J est définie comme étant la déviation moyenne (valeur absolue lissée) de la différence D de l'espacement de paquets chez le receveur comparé à l'envoyeur pour une paire de paquets. On trouvera les algorithmes de calcul détaillés dans la RFC 1889. Le compte inclut les paquets reçus de différentes SSRC, si l'envoyeur a utilisé plusieurs valeurs. La valeur est zéro si la connexion a toujours été en mode "envoi seul".

Délai de transmission moyen : estimation de la latence du réseau, exprimée en millisecondes. Pour RTP, c'est la valeur moyenne de la différence entre l'horodatage NTP indiqué par les envoyeurs des messages RTCP et l'horodatage NTP

des receveurs, mesurée quand les messages sont reçus. La moyenne est obtenue en aditionnant toutes les estimations, puis en divisant par le nombre de messages RTCP reçus. Quand l'horloge de la passerelle n'est pas synchronisée par NTP, la valeur de latence peut être calculée comme la moitié du délai d'aller-retour, comme mesuré par RTCP. Quand la passerelle ne peut pas calculer le délai unidirectionnel ou le délai d'aller-retour, le paramètre porte une valeur nulle.

Une définition détaillée de ces variables figure dans la RFC 1889.

Quand la connexion a été établie sur une interconnexion LOCAL, la signification de ces paramètres est définie comme suit :

Nombre de paquets envoyés : non significatif - PEUT être omis.

Nombre d'octets envoyés : le nombre total d'octets de charge utile transmis sur la connexion locale.

Nombre de paquets reçus : non significatif - PEUT être omis.

Nombre d'octets reçus : nombre total d'octets de charge utile reçus sur la connexion.

Nombre de paquets perdus : non significatif - PEUT être omis. Une valeur de zéro est supposée.

Gigue inter arrivées : non significatif - PEUT être omis. Une valeur de zéro est supposée.

Délai moyen de transmission : non significatif - PEUT être omis. Une valeur de zéro est supposée.

L'ensemble des paramètres de connexion peut être étendu. Aussi, la signification peut être plus définie par d'autres types de réseaux qui PEUVENT de plus choisir de ne pas retourner tous les paramètres spécifiés ci-dessus, ou même aucun.

La commande peut facultativement contenir une commande Demande de notification encapsulée, auquel cas un paramètre RequestIdentifier DOIT être présent, ainsi que, facultativement, d'autres paramètres de la demande de notification à l'exception de l'identifiant de point d'extrémité, qui n'est pas dupliqué. La demande de notification encapsulée est exécutée simultanément avec la suppression de la connexion. Par exemple, quand un décroché d'utilisateur est notifié, la passerelle devrait recevoir pour instruction de supprimer la connexion et de commencer à chercher un événement de raccroché.

Cela peut être accompli dans une seule commande DeleteConnection, en transmettant aussi les paramètres RequestedEvents, pour l'événement de décroché, et un paramètre SignalRequests vide.

Quand ces paramètres sont présents, la DeleteConnection et la NotificationRequest doivent être synchronisées, ce qui signifie que les deux DOIVENT être acceptées, ou les deux DOIVENT être refusées.

La commande peut porter une commande encapsulée EndpointConfiguration, qui va s'appliquer au même point d'extrémité. Quand cette commande est présente, les paramètres de la commande EndpointConfiguration sont incluses avec les paramètres normaux de DeleteConnection à l'exception de l'identifiant de point d'extrémité, qui n'est pas dupliqué. La commande EndpointConfiguration peut être encapsulée avec une commande Demande de notification encapsulée.

La commande encapsulée EndpointConfiguration partage le sort de la commande DeleteConnection. Si DeleteConnection est rejetée, EndpointConfiguration n'est pas exécutée.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste de paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.8 DeleteConnection (provenant de la passerelle)

Dans certaines circonstances rares une passerelle peut avoir à supprimer une connexion, par exemple parce qu'elle a perdu la ressource associée à la connexion, ou parce qu'elle a détecté que le point d'extrémité n'est plus capable ou ne veut plus envoyer ou recevoir de supports. La passerelle peut alors terminer la connexion en utilisant une variante de la commande DeleteConnection :

ReturnCode,

[PackageList]
<-- DeleteConnection(CallId,
EndpointId,
ConnectionId,
ReasonCode,
Connection-paramètres)

Le EndpointId, dans cette forme de la commande DeleteConnection, DOIT être pleinement qualifié. Les conventions de caractères génériques NE DOIVENT PAS être utilisées.

Le ReasonCode *(code de cause)* est une chaîne de texte commençant par un code de cause numérique et facultativement suivi par une chaîne de texte descriptif. Le code de cause indique la cause du DeleteConnection. Une liste des codes de cause se trouve au paragraphe 2.5.

En plus de l'appel, des identifiants de point d'extrémité et de connexion, la passerelle va aussi envoyer les paramètres de connexion qui auraient été retournés à l'agent d'appel en réponse à une commande DeleteConnection.

ReturnCode est un paramètre retourné par l'agent d'appel. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

Noter que l'utilisation de cette commande est généralement déconseillée et ne devrait être faite qu'en dernier ressort. Si une connexion peut être conservée, sa suppression devrait être laissée à la discrétion de l'agent d'appel qui est dans une bien meilleure position pour prendre des décisions intelligentes dans ce domaine.

2.3.9 DeleteConnection (plusieurs connexions provenant de l'agent d'appel)

Une variante de la fonction DeleteConnection peut être utilisée par l'agent d'appel pour supprimer plusieurs connexions en même temps. Noter qu'encapsuler d'autres commandes avec cette variante de la commande DeleteConnection n'est pas permis. La commande peut être utilisée pour supprimer toutes les connexions qui se rapportent à un appel pour un point d'extrémité :

ReturnCode,
[PackageList]
<-- DeleteConnection(CallId, EndpointId)

Le EndpointId, dans cette forme de la commande DeleteConnection, NE DOIT PAS utiliser le caractère générique "any of". Toutes les connexions pour le ou les points d'extrémité avec le CallId spécifié vont être supprimées. Noter que la commande va encore réussir si il n'y a pas de connexion avec le CallId spécifié, tant que le EndpointId est valide. Cependant, si le EndpointId est invalide, la commande va échouer. La commande ne retourne aucune statistique individuelle ni paramètre d'appel.

Il peut aussi être utilisé pour supprimer toutes les connexions qui se terminente à un point d'extrémité donné :

ReturnCode,
[PackageList]
<-- DeleteConnection(EndpointId)

Le EndpointId, dans cette forme de la commande DeleteConnection, NE DOIT PAS utiliser le caractère générique "any of". Là encore, la commande réussit même si il n'y avait pas de connexion sur le point d'extrémité.

Finalement, les agents d'appel peuvent tirer parti de la structure hiérarchique des noms de point d'extrémité pour supprimer toutes les connexions qui appartiennent à un groupe de points d'extrémité. Dans ce cas, le composant "nom local" du EndpointId va être spécifié en utilisant la convention de caractère générique "all of". La convention "any of" NE DEVRA PAS être utilisée. Par exemple, si les noms de point d'extrémité sont structurés comme la combinaison d'un nom d'interface physique et d'un numéro de circuit, comme dans "X35V3+A4/13", l'agent d'appel peut remplacer le numéro de circuit par le caractère générique "all of" de "*", comme dans "X35V3+A4/*". Cette commande "wildcard" donne pour instruction à la passerelle de supprimer toutes les connexions qui étaient attachées aux circuits connectés à l'interface physique "X35V3+A4".

Après la suppression de toutes les connexions, tout bouclage arrière qui a été demandé pour les connexions DOIT être annulée par la passerelle.

Cette commande ne retourne aucune statistique ou paramètre d'appel individuel.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste de paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.10 AuditEndpoint

La commande AuditEndPoint peut être utilisée par l'agent d'appel pour trouver l'état d'un certain point d'extrémité.

ReturnCode,
EndPointIdList,|{
[RequestedEvents,]
[QuarantineHandling,]
[DigitMap,]
[SignalRequests,]
[RequestIdentifier,]
[NotifiedEntity,]
[ConnectionIdentifiers,]
[DetectEvents,]
[ObservedEvents,]
[EventStates,]
[BearerInformation,]
[RestartMethod,]
[RestartDelay,]
[ReasonCode,]
[MaxMGCPDatagram,]
[Capabilities]}
[PackageList]
<-- AuditEndPoint(EndpointId, [RequestedInfo])

Le EndpointId identifie le ou les points d'extrémité qui sont examinés. La convention de caractère générique "any of" NE DOIT PAS être utilisée.

Le EndpointId identifie le ou les points d'extrémité examinés. La convention de caractère générique "all of" peut être utilisée pour commencer l'examen d'un groupe de points d'extrémité (sans considération de leur état de service). Si cette convention est utilisée, la passerelle DEVRA retourner la liste des identifiants de point d'extrémité qui correspondent au caractère générique dans le paramètre EndPointIdList, qui est simplement un ou plusieurs SpecificEndpointId (chacun fourni séparément). Dans le cas où le caractère générique "all of" est utilisé, RequestedInfo NE DEVRAIT PAS être inclus (si il est inclus, il DOIT être ignoré). Noter que l'utilisation du caractère générique "all of" peut générer une EndPointIdList potentiellement grande. Si la EndPointIdList résultante est considérée comme trop grande, la passerelle retourne une erreur (le code d'erreur 533 - Réponse trop grande, est RECOMMANDÉ).

Quand un EndpointId sans caractère générique est spécifié, le paramètre (éventuellement vide) RequestedInfo décrit les informations qui sont demandées pour le EndpointId spécifié. Les informations de point d'extrémité suivantes peuvent être examinées avec cette commande : RequestedEvents, DigitMap, SignalRequests, RequestIdentifier, QuarantineHandling, NotifiedEntity, ConnectionIdentifiers, DetectEvents, ObservedEvents, EventStates, BearerInformation, RestartMethod, RestartDelay, ReasonCode, PackageList, MaxMGCPDatagram, et Capabilities.

La liste peut être étendue par des paramètres d'extension. La réponse va à son tour inclure des informations sur chaque élément pour lequel les informations d'examen ont été demandées. Les paramètres pris en charge avec des valeurs vides DOIVENT toujours être retournés. Cependant, si un point d'extrémité est interrogé sur un paramètre qu'il ne comprend pas, il NE DOIT PAS générer une erreur ; le paramètre DOIT plutôt être omis de la réponse :

* RequestedEvents *(événements demandés)* : valeur actuelle des RequestedEvents que le point d'extrémité utilise, incluant

les actions et paramètres d'événement associés à chaque événement - si aucune action n'est incluse, l'action par défaut est supposée. Les événements persistants sont inclus dans la liste. Si une demande de notification incorporée est active, le RequestedEvents va refléter les événements demandés dans la demande de notification incorporée, et pas les événements demandés environnants (qu'ils soient incorporés ou non).

* DigitMap *(transposition de chiffres)* : script de numérotation que le point d'extrémité utilise actuellement. Le paramètre va être vide si le point d'extrémité n'a pas de script de numérotation.

* SignalRequests *(demande de signaux)* : liste des signaux de temporisation qui sont actuellement actifs, des signaux On/Off qui sont actuellement "on" pour le point d'extrémité (avec ou sans paramètre) et tous signaux brefs en instance. Les signaux de temporisation qui ont expiré, et les signaux brefs en cours d'exécution ne sont pas inclus. Tous les paramètres de signaux inclus dans la SignalRequests d'origine vont être inclus.

* RequestIdentifier *(identifiant de demande)* : celui de la dernière demande de notification reçue par ce point d'extrémité (inclut les demandes de notification encapsulées dans d'autres commandes). Si aucune NotificationRequest n'a été créée depuis le réamorçage/redémarrage, la valeur zéro va être retournée.

* QuarantineHandling *(traitement de quarantaine)* : celui de la dernière demande de notification reçue par ce point d'extrémité. Si QuarantineHandling n'était pas inclus, ou si aucune demande de notification n'a été créée, les valeurs par défaut vont être retournées.

* DetectEvents *(détecter les événements)* : valeur du paramètre DetectEvents le plus récemment reçu plus tous les événements persistants mis en œuvre par le point d'extrémité. Si aucun paramètre DetectEvents n'a été créé, la liste (éventuellement vide) n'inclut que les événements persistants.

* NotifiedEntity : "entité notifiée" actuelle pour le point d'extrémité.

* ConnectionIdentifiers *(identifiants de connexion)* : liste des identifiants de connexion pour toutes les connexions qui existent actuellement pour le point d'extrémité spécifié.

* ObservedEvents *(événements observés)* : liste actuelle des événements observés pour le point d'extrémité.

* EventStates *(états d'événement)* : pour les événements qui ont des états auditables associés, c'est l'événement correspondant à l'état du point d'extrémité, par exemple, décroché si le point d'extrémité est décroché. Noter que la définition des événements individuels va déclarer si l'événement en question a un état auditable qui lui est associé.

* BearerInformation *(informations de porteuse)* : valeur du dernier paramètre BearerInformation reçu pour ce point d'extrémité (cela inclut le cas où BearerInformation a été provisionné). Le paramètre va être vide si le point d'extrémité n'a pas reçu de paramètre BearerInformation et si une valeur n'a pas été provisionnée.

* RestartMethod *(méthode de redémarrage)* : "redémarrage" si le point d'extrémité est en service et si le fonctionnement est normal, ou si le point d'extrémité est engagé dans le processus de passage en service (un RestartDelay non à zéro va indiquer ce dernier cas). Autrement, c'est la valeur du paramètre Méthode de redémarrage dans la dernière commande RestartInProgress produite (ou devrait avoir été produite) par le point d'extrémité. Noter qu'un point d'extrémité "déconnecté" va donc seulement rapporter "déconnecté" tant qu'il est effectivement déconnecté, et "redémarrage" sera rapporté une fois qu'il n'est plus déconnecté. De même, "annuler en douceur" ne sera pas rapporté, mais "en douceur" le pourrait (voir les détails au paragraphe 4.4.5).

* RestartDelay *(délai de redémarrage)* : valeur du paramètre Délai de redémarrage si une commande RestartInProgress va être produite par le point d'extrémité au moment de cette réponse, ou zéro si la commande ne va pas inclure ce paramètre.

* ReasonCode *(code de cause)* : valeur du paramètre ReasonCode dans la dernière commande RestartInProgress ou DeleteConnection produite par la passerelle pour le point d'extrémité, ou la valeur spéciale 000 si l'état du point d'extrémité est normal.

* PackageList *(liste des paquetages)* : paquetages pris en charge par le point d'extrémité incluant les numéros de version de paquetage. Pour la rétro compatibilité, la prise en charge du paramètre est FACULTATIVE mais les mises en œuvre avec des versions de paquetage supérieures à zéro DEVRAIT le prendre en charge.

* MaxMGCPDatagram : taille maximum d'un datagramme MGCP en octets qui peut être reçu par le point d'extrémité

(paragraphe 3.5.4). La valeur exclut tous frais généraux de couche inférieure. Pour la rétro compatibilité, la prise en charge de ce paramètre est FACULTATIVE. La taille maximum de datagramme MGCP par défaut DEVRAIT être supposée si une valeur n'est pas retournée.

* Capabilities *(capacités)* : capacités du point d'extrémité similaires au paramètre LocalConnectionOptions et incluant les paquetages et modes de connexion. Des extensions PEUVENT aussi être incluses. Si des capacités inconnues sont rapportées, elles DOIVENT être simplement ignorées. Si il est besoin de spécifier que certains paramètres, comme par exemple la suppression de silence, ne sont compatibles qu'avec certains codecs, la passerelle DOIT alors retourner plusieurs ensembles de capacités, dont chacun peut inclure :
  - l'algorithme de compression : une liste des codecs pris en charge. Le reste des paramètres dans l'ensemble de capacités va s'appliquer à tous les codecs spécifiés dans cette liste.
  - Période de mise en paquets : une seule valeur ou une gamme peut être spécifiée.
  - Bande passante : une seule valeur ou une gamme correspondant aux périodes de mise en paquets peut être spécifiée (en supposant qu'il n'y a pas de suppression de silence).
  - Annulation d'écho : si l'annulation d'écho est ou non prise en charge pour le point d'extrémité.
  - Suppression de silence : si la suppression de silence est prise en charge ou non.
  - Contrôle de gain : si le contrôle de gain est pris en charge ou non.
  - Type de service: si le type de service est pris en charge ou non.
  - Réservation de ressources : si la réservation de ressources est prise en charge ou non.
  - Sécurité : si le chiffrement des supports est pris en charge ou non.
  - Type de réseau : le ou les types de réseau pris en charge.
  - Paquetages : liste des paquetages pris en charge. Le premier paquetage de la liste va être le paquetage par défaut.
  - Modes : liste des modes de connexion pris en charge.

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

Si aucune information n'a été demandée et si l'identifiant de point d'extrémité se réfère à un point d'extrémité valide (en service ou non) la passerelle returne simplement un accusé de réception positif.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

Noter que PackageList PEUT aussi être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.11 AuditConnection

La commande AuditConnection peut être utilisée par l'agent d'appel pour restituer les paramètres attachés à une connexion.

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

Le paramètre EndpointId spécifie le point d'extrémité qui traite la connexion. Les conventions de caractère générique NE DEVRONT PAS être utilisées.

Le paramètre ConnectionId est l'identifiant de la connexion examinée, au sein du contexte du point d'extrémité spécifié.

Le paramètre (éventuellement vide) RequestedInfo décrit les informations demandées pour l'identifiant de connexion au sein du point d'extrémité spécifié. Les informations de connexion suivantes peuvent être examinées avec cette commande : CallId, NotifiedEntity, LocalConnectionOptions, Mode, RemoteConnectionDescriptor, LocalConnectionDescriptor, ConnectionParameters.

La réponse à AuditConnection va inclure des informations sur chaque élément d'informations d'audit demandé pour :

* CallId : l'identifiant d'appel de l'appel auquel la connexion appartient.

* NotifiedEntity : "entité notifiée" actuelle pour la connexion. Noter que c'est la même que l'"entité notifiée" pour le point d'extrémité (incluse ici pour la rétro compatibilité).

* LocalConnectionOptions, les plus récents paramètres LocalConnectionOptions qui ont été actuellement fournis pour la connexion (en omettant les LocalConnectionOptions provenant d'une commande qui ne change pas cette valeur). Noter que les paramètres par défaut omis de la plus récente LocalConnectionOptions ne vont pas être inclus. Les LocalConnectionOptions qui conservent leur valeur à travers les commandes ModifyConnection et qui ont été incluses dans une précédente commande pour la connexion sont aussi incluses, sans considération de si elles ont été fournies dans la plus récente LocalConnectionOptions ou non.

* Mode : le mode actuel de la connexion.

* RemoteConnectionDescriptor : descripteur de connexion distante qui a été fourni à la passerelle pour la connexion.

* LocalConnectionDescriptor : descripteur de connexion locale que la passerelle a fourni pour la connexion.

* ConnectionParameters : valeurs actuelles des paramètres de connexion pour la connexion.

Si aucune information n'était demandée et si l'identifiant de point d'extrémité est valide, la passerelle vérifie simplement que la connexion existe, et si il en est ainsi, elle retourne un accusé de réception positif. Noter que par définition, le point d'extrémité doit être en service pour que cela arrive, car les points d'extrémité hors service n'ont pas de connexions.

ReturnCode est un paramètre retourné par la passerelle. Il indique le résultat de la commande et consiste en un nombre entier facultativement suivi par un commentaire.

PackageList est une liste des paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.3.12 RestartInProgress

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

```
ReturnCode,
[NotifiedEntity,]
[PackageList]
<-- RestartInProgress(EndPointId, RestartMethod, [RestartDelay,] [ReasonCode])
```

Le EndPointId identifie le ou les points d'extrémité qui sont mis en service ou hors service. La convention de caractère générique "all of" peut être utilisée pour s'appliquer à la commande pour un groupe de points d'extrémité gérés par le même agent d'appel, comme par exemple tous les points d'extrémité qui sont rattachés à une interface spécifiée, ou même tous les points d'extrémité qui sont rattachés à une certaine passerelle. La convention de caractère générique "any of" NE DEVRA PAS être utilisé.

Le paramètre RestartMethod spéeifie le type de redémarrage. Les valeurs suivantes sont définies :

* Une méthode de redémarrage "graceful" *(en douceur)* indique que les points d'extrémité spécifiés vont être mis hors service après un délai spécifié. Les connexions établies ne sont pas encore affectées, mais l'agent d'appel DEVRAIT s'abstenir d'établir de nouvelles connexions, et DEVRAIT essayer de supprimer en douceur les connexions existantes.

* Une méthode de redémarrage "forcée" indique que les points d'extrémité spécifiés sont mis abruptement hors service. Les connexions établies, si il en est, sont perdues.

* Une méthode "restart" *(redémarrage)* indique que le service va être restauré sur les points d'extrémité après le "délai de redémarrage" spécifié, c'est-à-dire, les points d'extrémité vont être en service. Les points d'extrémité sont dans leur état par défaut parfait et il n'y a pas de connexions actuellement établies sur les points d'extrémité.

* Une méthode "déconnectée" indique que le point d'extrémité est devenu déconnecté et essaye maintenant d'établir la

connexité (voir au paragraphe 4.4.7). Le "délai de redémarrage" spécifie le nombre de secondes pendant lequel le point d'extrémité a été déconnecté. Les connexions établies ne sont pas affectées.

* Une méthode "cancel-graceful" *(annulation de redémarrage en douceur)* indique qu'une passerelle annule une commande de redémarrage en douceur précédemment produite. Les points d'extrémité sont toujours en service.

La liste des méthodes de redémarrage peut être étendue.

Le paramètre facultatif "délai de redémarrage" est exprimée comme un nombre de secondes. Si le nombre est absent, la valeur du délai DOIT être considérée comme nulle (c'est-à-dire, zéro). Dans le cas d'une méthode "graceful", un délai nul indique que l'agent d'appel DEVRAIT simplement attendre la terminaison naturelle des connexions existantes, sans établir de nouvelles connexions. Le délai de redémarrage est toujours considéré comme nul dans le cas des méthodes "forcée" et "cancel-graceful", et donc le paramètre "délai de redémarrage" NE DOIT PAS être utilisé avec ces méthodes de redémarrage. Quand la passerelle envoie un message Redémarrage en cours de "restart" ou "graceful" avec un délai de redémarrage non zéro, la passerelle DEVRAIT envoyer un message RestartInProgress mis à jour après l'écoulement du délai de redémarrage.

Un délai de redémarrage nul pour la méthode "restart" indique que le service a déjà été restauré. Cela va normalement se produire après un démarrage/réamorçage de la passerelle. Pour atténuer les effets du changement d'adresse IP d'une passerelle par suite d'un réamorçage, l'agent d'appel PEUT souhaiter purger son antémémoire DNS pour le nom de domaine de la passerelle ou résoudre le nom de domaine de la passerelle en interrogeant le DNS sans considération du TTL d'un enregistrement de ressource DNS actuel pour la passerelle redémarrée.

Le paramètre facultatif de code de cause indique la cause du redémarrage.

Les passerelles DEVRAIENT envoyer par courtoisie un message de redémarrage en cours de "en douceur" ou "forcé" (pour les points d'extrémité pertinents) à l'agent d'appel quand elles sont mises hors service, par exemple, en étant fermées, ou mises hors service par un système de gestion de réseau, cependant l'agent d'appel ne peut pas compter recevoir toujours un tel message. Les passerelles DOIVENT envoyer un message Redémarrage en cours de "restart" (pour les points d'extrémité pertinents) avec un délai nul à leur agent d'appel quand elles sont de retour en service conformément à la procédure de redémarrage spécifiée au paragraphe 4.4.6 - les agents d'appel peuvent compter sur la réception de ce message. Les passerelles DOIVENT aussi envoyer un message Redémarrage en cours de "déconnecté" (pour les points d'extrémité pertinents) à leur "entité notifiée" actuelle conformément à la procédure "déconnecté" spécifiée au paragraphe 4.4.7.

Le message RestartInProgress va être envoyé à l'"entité notifiée" actuelle pour l'identifiant de point d'extrémité en question. Il est supposé qu'un agent d'appel par défaut, c'est-à-dire, une "entité notifiée", a été provisionnée de façon à ce que après un réamorçage/redémarrage, l'agent d'appel par défaut soit toujours la "entité notifiée" pour le point d'extrémité. Les passerelles DEVRAIENT tirer pleinement parti des conventions de caractères génériques pour minimiser le nombre de messages RestartInProgress générés quand 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 consiste en un nombre entier suivi facultativement par un commentaire.

Une NotifiedEntity peut de plus être retournée avec la réponse au RestartInProgress provenant de l'agent d'appel - ceci DEVRAIT normalement n'être fait que dans une réponse à "restart" ou "déconnecté" (voir aussi les paragraphes 4.4.6 et 4.4.7) :

* Si la réponse indiquait le succès (code de retour 200 - transaction exécutée) le redémarrage en question s'est achevé avec succès, et la NotifiedEntity retournée est la nouvelle "entité notifiée" pour le ou les points d'extrémité.
* Si la réponse provenant de l'agent d'appel indiquait une erreur, le redémarrage en question ne s'est pas bien achevé. Si un paramètre NotifiedEntity était inclus dans la réponse retournée, il spécifie une nouvelle "entité notifiée" pour le ou les points d'extrémité, qui DOIT être utilisée quand il reessaye le redémarrage en question (comme une nouvelle transaction). Ceci DEVRAIT seulement être fait avec le code d'erreur 521 (point d'extrémité redirigé).

Noter que le comportement ci-dessus pour retourner une NotifiedEntity dans la réponse n'est défini que pour les réponses RestartInProgress et NE DEVRAIT PAS être suivi pour des réponses à d'autres commandes. Tout autre comportement est indéfini.

PackageList est une liste de paquetages pris en charge qui PEUVENT être inclus avec le code d'erreur 518 (Paquetage non pris en charge).

2.4 Codes de retour et d'erreur

Toutes les commandes MGCP sont acquittées. L'accusé de réception porte un code de retour, qui indique l'état de la commande. Le code de retour est un nombre entier, pour lequel les gammes de valeurs suivantes ont été définies :
* les valeurs entre 000 et 099 indiquent un accusé de réception de réponse
* les valeurs entre 100 et 199 indiquent une réponse provisoire
* les valeurs entre 200 et 299 indiquent l'achèvement avec succès
* les valeurs entre 400 et 499 indiquent une erreur temporaire
* les valeurs entre 500 et 599 indiquent une erreur permanente
* les valeurs entre 800 et 899 sont des codes de réponse spécifiques du paquetage.

Une description des erreurs temporaires (codes d'erreur 4XX) opposée aux erreurs permanentes (codes d'erreur 5XX) est comme suit :

* Si un agent d'appel reçoit une erreur temporaire, on s'attend à la possibilité qu'une demande future soit honorée par le point d'extrémité. Dans certains cas, cela peut exiger un changement d'état dans l'environnement du point d'extrémité (par exemple, l'état du crochet dans le cas des codes d'erreur 401 ou 402 ; la disponibilité de la ressource dans le cas du code d'erreur 403, ou la disponibilité de la bande passante dans le cas du code d'erreur 404).

* Les erreurs permanantes (codes d'erreur 500 à 599) indiquent que une ou plusieurs conditions permanentes dues à une erreur de protocole ou à une incompatibilité entre le point d'extrémité et l'agent d'appel, ou à cause d'une condition d'erreur sur laquelle l'agent d'appel n'a pas de contrôle. Des exemples sont des erreurs de protocole, des demandes sur les capacités du point d'extrémité qui n'existent pas, des erreurs sur des interfaces associées au point d'extrémité, des informations manquantes ou incorrectes dans la demande ou toutes autres conditions qui ne vont simplement pas disparaître avec le temps.

Les valeurs qui ont déjà été définies sont les suivantes :
000 Accusé de réception de réponse.
100 La transaction est en cours d'exécution. Un message d'accomplissement réel va suivre.
101 La transaction a été mise en file d'attente pour exécution. Un message d'accomplissement réel va suivre.
200 La transaction demandée a été exécutée normalement. Ce code de retour peut être utilisé pour une réponse de succès à toute commande.
250 La connexion a été supprimée. Ce code de retour ne peut être utilisé que pour une réponse de succès à une commande DeleteConnection.
400 La transaction n'a pas pu être exécutée, à cause d'une erreur temporaire non spécifiée.
401 Le téléphone est déjà décroché.
402 Le téléphone est déjà raccroché.
403 La transaction n'a pas pu être exécutée, parce que le point d'extrémité n'a pas les ressources suffisantes pour l'instant.
404 Bande passante insuffisante pour l'instant.
405 La transaction n'a pas pu être exécutée, parce que le point d'extrémité est en "redémarrage".
406 Fin de temporisation de transaction. La transaction ne s'est pas achevée dans un délai raisonnable et a été interrompue.
407 Transaction interrompue. La transaction a été interrompue par une action externe, par exemple, une commande ModifyConnection interrompue par une commande DeleteConnection.
409 La transaction n'a pas pu être exécutée à cause d'une surcharge interne.
410 Pas de point d'extrémité disponible. Un caractère générique valide "any of" a été utilisé, cependant il n'y a pas de point d'extrémité disponible pour satisfaire la demande.
500 La transaction n'a pas pu être exécutée, parce que le point d'extrémité est inconnu.
501 La transaction n'a pas pu être exécutée, parce que le point d'extrémité n'est pas prêt. Cela inclut le cas où le point d'extrémité est hors service.
502 La transaction n'a pas pu être exécutée, parce que le point d'extrémité n'a pas des ressources suffisantes (condition permanente).
503 Le caractère générique "All of" est trop compliqué.
504 Commande inconnue ou non prise en charge.
505 Descripteur de connexion distante non pris en charge. Ceci DEVRAIT être utilisé quand un ou plusieurs paramètres ou valeurs obligatoires dans le descripteur de connexion distante ne sont pas pris en charge.
506 Incapable de satisfaire les options de connexion locale et le descripteur de connexion distante. Ceci DEVRAIT être utilisé quand LocalConnectionOptions et RemoteConnectionDescriptor contiennent un ou plusieurs paramètres ou valeurs obligatoires en conflit l'un avec l'autre et/ou ne peuvent pas être pris en charge au même moment (sauf pour un échec de négociation de codec - voir le code d'erreur 534).
507 Fonctionnalité non prise en charge. Des fonctions non spécifiées exigées pour exécuter la commande ne sont pas prises en charge. Noter que plusieurs autres codes d'erreur ont été définis pour des domaines spécifiques de fonction non prise

en charge (par exemple 508, 511, etc.) et ce code d'erreur DEVRAIT n'être utilisé que si il n'y a pas d'autre code d'erreur plus spécifique pour la fonction non prise en charge.

508 Traitement de quarantaine inconnu ou non pris en charge.

509 Erreur dans le descripteur de connexion distante. Ceci DEVRAIT être utilisé quand il y a une erreur de syntaxe ou de sémantique dans le RemoteConnectionDescriptor.

510 La transaction n'a pas pu être exécutée, à cause d'une erreur de protocole non spécifiée qui a été détectée. La récupération automatique d'une telle erreur va être très difficile, et donc ce code DEVRAIT n'être utilisé qu'en dernier ressort.

511 La transaction n'a pas pu être exécutée, parce que la commande contenait une extension non reconnue. Ce code DEVRAIT être utilisé pour les extensions de paramètre critique non pris en charge ("X+").

512 La transaction n'a pas pu être exécutée, parce que la passerelle n'est pas équipée pour détecter un des événements demandés.

513 La transaction n'a pas pu être exécutée, parce que la passerelle n'est pas équipée pour générer un des signaux demandés.

514 La transaction n'a pas pu être exécutée, parce que la passerelle ne peut pas envoyer l'annonce spécifiée.

515 La transaction se réfère à un identifiant de connexion incorrect (qui peut avoir été déjà supprimé).

516 La transaction se réfère à un identifiant d'appel inconnu, ou l'identifiant d'appel fourni est incorrect (par exemple, l'identifiant de connexion n'est pas associé à cet identifiant d'appel).

517 Mode non pris en charge ou invalide.

518 Paquetage non pris en charge ou inconnu. Il est RECOMMANDÉ d'inclure un paramètre PackageList avec la liste des paquetages pris en charge dans la réponse, en particulier si la réponse est générée par l'agent d'appel.

519 Le point d'extrémité n'a pas de script de numérotation.

520 La transaction n'a pas pu être exécutée, parce que le point d'extrémité "redémarre". Dans la plupart des cas, cela va être une erreur temporaire, et dans ce cas, le code d'erreur 405 DEVRAIT plutôt être utilisé. Le code d'erreur n'est inclus ici que pour la rétro compatibilité.

521 Le point d'extrémité a redirigé sur un autre agent d'appel. Le comportement de redirection associé n'est bien défini que quand cette réponse est produite pour une commande RestartInProgress.

522 Pas de tel événement ou signal. La demande se réfère à un événement ou signal qui n'est pas défini dans le paquetage pertinent (qui pourrait être le paquetage par défaut).

523 Action inconnue ou combinaison illégale d'actions.

524 Incohérence interne dans LocalConnectionOptions.

525 Extension inconnue dans LocalConnectionOptions. Ce code DEVRAIT être utilisé pour des extensions obligatoires de fabricant non prises en charge ("x+").

526 Bande passante insuffisante. Dans les cas où c'est une erreur temporaire, le code d'erreur 404 DEVRAIT plutôt être utilisé.

527 Descripteur de connexion distante manquant.

528 Version de protocole incompatible.

529 Échec interne du matériel.

530 Erreur de protocole de signalisation CAS.

531 Échec d'un groupement de cicuits (par exemple, échec de facilité).

532 Valeurs npn prises en charge dans les options de connexion locale.

533 Réponse trop grande.

534 Échec de négociation de codec.

535 Périonde de mise en paquets non prise en charge.

536 Méthode de redémarrage inconnue ou non prise en charge.

537 Extension de script de numérotation inconnue ou non prise en charge.

538 Erreur de paramètre d'événement/signal (par exemple, manquant, erroné, non pris en charge, inconnu, etc.).

539 Paramètre de commande invalide ou non pris en charge. Ce code DEVRAIT n'être utilisé que quand le paramètre n'est un paramètre d'extension ni de paquetage ni de fabricant.

540 Limite de connexions par point d'extrémité excédée.

541 Options de connexion locale invalides ou non prises en charge. Ce code DEVRAIT n'être utilisé que quand les LocalConnectionOptions ne sont une extension ni de paquetage ni de fabricant.

L'ensemble des codes de retour pourra être étendu dans une future version du protocole. Les mises en œuvre qui reçoivent un code de retour inconnu ou non pris en charge DEVRAIENT traiter le code de retour comme suit :

* Code 0xx inconnu traité comme 000.
* Code 1xx inconnu traité comme 100.
* Code 2xx inconnu traité comme 200.
* Code 3xx inconnu traité comme 521.
* Code 4xx inconnu traité comme 400.
* Code 5xx-9xx inconnu teaité comme 510.

2.5 Codes de cause

Les codes de cause sont utilisés par la passerelle quand elle supprime une connexion pour informer l'agent d'appel de la raison de la suppression de la connexion. Ils peuvent aussi être utilisés dans une commande RestartInProgress pour informer l'agent d'appel de la raison du redémarrage en cours.

Le code de cause est un nombre entier, et les valeurs suivantes ont été définies :
000 L'état du point d'extrémité est normal (ce code n'est utilisé qu'en réponse à des demandes d'examen).
900 Mauvais fonctionnement du point d'extrémité.
901 Point d'extrémité hors service.
902 Perte de la connexité à la couche inférieure (par exemple, de la synchronisation vers l'aval).
903 Perte de la réservation de ressource de qualité de service.
904 Intervention manuelle.
905 Échec de facilité (par exemple, échec de DS-0).

L'ensemble des codes de cause peut être étendu.

2.6 Utilisation d'options et descripteurs de connexion locaux

Comme indiqué précédemment, la séquence normale d'établissement d'une connexion bidirectionnelle implique au moins trois étapes :

1) L'agent d'appel demande à la première passerelle de "créer une connexion" sur un point d'extrémité. La passerelle alloue des ressources à cette connexion, et répond à la commande en fournissant une "description de session" (qu'on appelle son descripteur de connexion locale *(LocalConnectionDescriptor)*). La description de session contient les informations nécessaires pour qu'une autre partie envoie des paquets sur la nouvelle connexion créée.

2) L'agent d'appel demande alors à la seconde passerelle de "créer une connexion" sur un point d'extrémité. La commande porte la "description de session" fournie par la première passerelle (qu'on appellera désormais le RemoteConnectionDescriptor). La passerelle alloue des ressources à cette connexion, et répond à la commande en fournissant sa propre "description de session" (LocalConnectionDescriptor).

3) L'agent d'appel utilise une commande "Modifier la connexion" pour fournir cette seconde "description de session" (RemoteConnectionDescriptor) au premier point d'extrémité. Cela fait, la communication peut avoir lieu dans les deux directions.

Quand l'agent d'appel produit une commande Créer ou Modifier la connexion, il y a donc trois paramètres qui déterminent le support pris en charge par cette connexion :

* LocalConnectionOptions : fourni par l'agent d'appel pour contrôler les paramètres des supports utilisés par la passerelle pour la connexion. Quand il est fourni, la passerelle DOIT se conformer à ces paramètres de supports jusqu'à ce que la connexion soit supprimée, ou qu'une commande ModifyConnection avec de nouveaux paramètres de supports (LocalConnectionOptions ou RemoteConnectionDescriptor) soit reçue.

* RemoteConnectionDescriptor : fourni par l'agent d'appel pour porter les paramètres de supports pris en charge par l'autre côté de la connexion. Quand il est fourni, la passerelle DOIT se conformer à ces paramètres de supports jusqu'à ce que la connexion soit supprimée, ou qu'une commande ModifyConnection avec de nouveaux paramètres de supports (LocalConnectionOptions ou RemoteConnectionDescriptor) soit reçue.

* LocalConnectionDescriptor : fourni par la passerelle à l'agent d'appel pour porter les paramètres de support qu'il prend en charge pour la connexion. Quand il est fourni, la passerelle DOIT honorer les paramètres de supports jusqu'à ce que la connexion soit supprimée, ou que la passerelle produise un nouveau LocalConnectionDescriptor pour cette connexion.

Pour déterminer quels codecs fournir dans le LocalConnectionDescriptor, il y a trois listes de codecs qu'une passerelle doit prendre en compte :

* Une liste des codecs permis par les LocalConnectionOptions dans la commande actuelle (soit explicitement par la

méthode de codage, soit implicitement par la bande passante et/ou la période de mise en paquets).

* Une liste des codecs dans le RemoteConnectionDescriptor de la commande actuelle.

* Une liste interne des codecs que la passerelle peut prendre en charge pour la connexion. Une passerelle PEUT prendre en charge un ou plusieurs codecs pour une certaine connexion.

Le choix du codec (incluant tous les paramètres de supports pertinents) peut alors être décrit par les étapes suivantes :

1. Une liste approuvée des codecs est formée en prenant l'intersection de la liste interne des codecs et de la liste des codecs permis par les LocalConnectionOptions. Si les LocalConnectionOptions n'étaient pas fournies dans la commande courante, la liste approuvée des codecs contient donc la liste interne des codecs.

2. Si la liste approuvée des codecs est vide, un échec de négociation de codec s'est produit et une réponse d'erreur est générée (le code d'erreur 534 - Échec de négociation de codec, est RECOMMANDÉ).

3. Autrement, une liste négociée des codecs est formée en prenant l'intersection de la liste approuvée des codecs et des codecs permis par le RemoteConnectionDescriptor. Si un RemoteConnectionDescriptor n'a pas éte fourni dans la commande courante, la liste négociée des codecs contient donc la liste approuvée des codecs.

4. Si la liste négociée des codecs est vide, un échec de négociation de codec s'est produit et une réponse d'erreur est générée (le code d'erreur 534 - Échec de négociation de codec, est RECOMMANDÉ).

5. Autrement, la négociation de codec a réussi, et la liste négociée des codecs est retournée dans le LocalConnectionDescriptor.

Noter que le LocalConnectionOptions et le RemoteConnectionDescriptor peuvent tous deux contenir une liste de codecs ordonnée par préférence. Quand les deux sont fournies dans la commande courante, la passerelle DOIT respecter les préférences fournies dans les LocalConnectionOptions.

2.7 Réservations de ressources

Les passerelles peuvent recevoir pour instruction d'effectuer une réservation, par exemple en utilisant RSVP, sur une certaine connexion. Quand une réservation est nécessaire, l'agent d'appel va spécifier le profil de réservation à utiliser, qui est soit "charge contrôlée", soit "service garanti". L'absence de réservation peut être indiqué en demandant le service "au mieux", qui est la valeur par défaut de ce paramètre dans une commande CreateConnection. Pour une commande ModifyConnection, la valeur par défaut est simplement de conserver la valeur courante. Quand la réservation a été demandée sur une connexion, la passerelle va :
* commencer à émettre des messages RSVP "PATH" si la connexion est en mode "envoi seul", "envoi-réception", "conférence", "rebouclage réseau" ou "essai de continuité réseau" (si un descripteur de connexion distante convenable a été créé) ;
* commencer à émettre des messages RSVP "RESV" aussitôt qu'elle reçoit des messages "PATH" si la connexion est en mode "réceptiion seule", "envoi-réception", "conférence", "rebouclage réseau" ou "essai de continuité réseau".

Les filtres RSVP vont être déduits des caractéristiques de la connexion. Les profils de ressource RSVP vont être déduits des codecs, bande passante et période de mise en paquets de la connexion.

3. Protocole de contrôle de passerelle de supports

Le protocole de contrôle de passerelle de supports (MGCP, *Media Gateway Control Protocol*) met en œuvre l'interface de contrôle de passerelle de supports comme un ensemble de transactions. Les transactions sont composées d'une commande et d'une réponse obligatoire. Il y a neuf commandes :
* EndpointConfiguration *(configuration de point d'extrémité)*
* CreateConnection *(créer une connexion)*
* ModifyConnection *(modifier la connexion)*
* DeleteConnection *(supprimer la connexion)*
* NotificationRequest *(demande de notification)*
* Notify *(notifier)*

* AuditEndpoint *(examen de point d'extrémité)*
* AuditConnection *(examen de connexion)*
* RestartInProgress *(redémarrage en cours)*

Les cinq premières commandes sont envoyées par l'agent d'appel à une passerelle. La commande Notify est envoyée par la passerelle à l'agent d'appel. La passerelle peut aussi envoyer une commande DeleteConnection comme défini au paragraphe 2.3.8. L'agent d'appel peut envoyer les commandes d'audit à la passerelle, et la passerelle peut envoyer une commande RestartInProgress à l'agent d'appel.

3.1 Description générale

Toutes les commandes sont composées d'un en-tête Command, facultativement suivi par une description de session.

Toutes les réponses sont composées d'un en-tête Response, facultativement suivi par des informations de description de session.

Les en-têtes et les descriptions de session sont codés comme un ensemble de lignes de texte, séparées par un caractère retour chariot et saut à la ligne (ou, facultativement, un seul caractère saut à la ligne). Les descriptions de session sont précédées d'une ligne vide.

MGCP utilise un identifiant de transaction pour corréler les commandes et les réponses. L'identifiant de transaction est codé comme un composant de l'en-tête de commande et répété comme un composant de l'en-tête de réponse (voir les paragraphes 3.2.1.2 et 3.3).

Noter qu'une grammaire ABNF pour MGCP est fournie à l'Appendice A. Les commandes et réponses DEVRONT être codées en accord avec la grammaire, qui, selon la RFC 2234, est insensible à la casse, sauf pour la partie SDP. De même, les mises en œuvre DEVRONT être capables de décoder les commandes et réponses qui suivent la grammaire. De plus, il est RECOMMANDÉ que les mises en œuvre tolèrent des espaces blanches linéaires supplémentaires.

Certaines productions permettent l'utilisation de chaînes entre guillemets, qui peuvent être nécessaires pour éviter des problèmes de syntaxe. Lorsque la forme de chaîne entre guillemets est utilisée, le contenu va être codé en UTF-8 [RFC2279], et la valeur réelle fournie est la chaîne sans guillemets (codée en UTF-8). Lorsque les deux formes de chaîne entre guillemets et sans guillemets sont permises, l'une ou l'autre forme peut être utilisée pourvu qu'elle ne viole pas par ailleurs la grammaire.

Dans la suite de ce document, on fournit des détails suppélemntaires sur le format des commandes et réponses MGCP.

3.2 En-tête de commande

L'en-tête de commande est composée de :
* Une ligne de commande, identifiant l'action ou verbe demandé, l'identifiant de transaction, le point d'extrémité vers lequel l'action est demandée, et la version de protocole MGCP,
* un ensemble de zéro, une ou plusieurs lignes de paramètres, composées d'un nom de paramètre suivi par une valeur de paramètre.

Sauf mention contraire ou imposé par d'autres normes référencées (par exemple, SDP) chaque composant de l'en-tête de commande est insensible à la casse. Cela vaut pour les verbes comme pour les paramètres et valeurs, et donc toutes les comparaisons DOIVENT traiter les lettres majuscules et minuscules ainsi que leurs combinaisons comme égales.

3.2.1 Ligne de commande

La ligne de commande est composée de :
* le nom du verbe demandé,
* l'identification de la transaction,
* le nom du ou des points d'extrémité qui vont exécuter la commande (dans les notifications ou redémarrages, le nom du ou des points d'extrémité qui produisent la commande),
* la version du protocole.

Ces quatre élements sont codés comme des chaînes de caractères imprimables ASCII, séparés par des espaces blanches,

c'est-à-dire, les caractères espace ASCII (0x20) ou tabulation (0x09). Il est RECOMMANDÉ d'utiliser exactement un séparateur d'espace ASCII. Cependant, les entités MGCP DOIVENT être capables d'analyser les messages avec des caractères d'espaces blanches supplémentaires.

3.2.1.1 Codage du verbe demandé

Les verbes qui peuvent être demandés sont codés comme des codes ASCII de quatre lettres majuscules ou minuscules (les comparaisons DEVRONT être insensibles à la casse) comme défini dans le tableau suivant :

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

L'identifiant de transaction est codé comme une chaîne de jusqu'à 9 chiffres décimaux. Dans la ligne de commande, il suit immédiatement le codage du verbe.

De nouveaux verbes pourront être définis dans de futures versions du protocole. Il peut être nécessaire, pour des besoins expérimentaux, d'utiliser de nouveaux verbes avant qu'ils soient sanctionnés dans une version publiée de ce protocole. Les verbes expérimentaux DOIVENT être identifiés par un code de quatre lettres commençant par la lettre X, comme par exemple XPER.

3.2.1.2 Identifiants de transaction

MGCP utilise un identifiant de transaction pour corréler les commandes et les réponses. Une passerelle prend en charge deux espaces de noms d'identifiant de transaction séparés :
* un espace de noms d'identifiant de transaction pour les transactions d'envoi, et
* un espace de noms d'identifiant de transaction pour les transactions de réception.

Au minimum, les identifiants de transaction pour les commandes envoyées à une certaine passerelle DOIVENT être uniques pour la durée de vie maximum des transactions au sein de la collection des agents d'appel qui contrôlent cette passerelle. Donc, sans considération de l'agent d'appel envoyeur, les passerelles peuvent toujours détecter des transactions dupliquées en examinant simplement l'identifiant de transaction. La coordination de ces identifiants de transaction entre les agents d'appel sort cependant du domaine d'application de la présente spécification.

Les identifiants de transaction pour toutes les commandes envoyées d'une certaine passerelle DOIVENT être uniques pour la durée de vie maximum des transactions sans considération de l'agent d'appel auquel la commande est envoyée. Donc, un agent d'appel peut toujours détecter une transaction dupliquée provenant d'une passerelle par la combinaison du nom de domaine du point d'extrémité et de l'identifiant de transaction.

L'identifiant de transaction est codé comme une chaîne de jusqu'à neuf chiffres décimaux. Dans les lignes de commandes, il suit immédiatement le codage du verbe.

Les identifiants de transaction ont des valeurs entre 1 et 999 999 999 (les deux inclus). Les identifiants de transaction NE DEVRAIENT PAS utiliser de zéros en tête, bien que l'égalité soit fondée sur la valeur numérique, c'est-à-dire, que les zéros en tête sont ignorés. Une entité MGCP NE DOIT PAS réutiliser un identifiant de transaction plus tôt que trois minutes après l'achèvement de la précédente commande dans laquelle l'identifiant a été utilisé.

3.2.1.3 Codage des identifiants de point d'extrémité et des noms d'entité

Les identifiants et noms d'entité de point d'extrémité sont codés comme des adresses de messagerie électronique insensibles à la casse, comme défini dans la RFC 821, bien qu'avec des restrictions syntaxiques sur la partie locale du nom. De plus, la partie locale du nom de point d'extrémité et la partie nom de domaine peuvent chacune faire jusqu'à 255 caractères. Dans

ces adresses, le nom de domaine identifie le système où le point d'extrémité est rattaché, tandis que la partie gauche identifie un point d'extrémité ou entité spécifique sur ce système.

Des exemples de telles adresses sont :

hrd4/56@gw23.exemple.net	Circuit numéro 56 de l'interface "hrd4" de la passerelle 23 du réseau "Exemple"
Call-agent@ca.exemple.net	Agent d'appel pour le réseau "exemple"
Busy-signal@ann12.exemple.net	Le point d'extrémité virtuel "busy signal" sur le serveur d'annonces numéro 12.

Le nom d'une entité notifiée est exprimé avec la même syntaxe, avec l'ajout possible d'un numéro d'accès comme dans :

Call-agent@ca.exemple.net:5234

Si le numéro d'accès est omis dans l'entité notifiée, l'accès d'agent d'appel MGCP par défaut (2727) DOIT être utilisé.

3.2.1.4 Codage de la version de protocole

La version de protocole est codée comme le mot-clé MGCP suivi par une espace et le numéro de version, et facultativement suivi par un nom de profil. Le numéro de version est composée d'une version majeure, codée par un nombre décimal, un point, et un numéro de version mineure, codé comme un nombre décimal. La version décrite dans le présent document est la version 1.0.

Le nom de profil, si il est présent, est représenté par des chaînes, séparées par des espaces, de caractères visibles (imprimables) qui s'étendent jusqu'à la fin de la ligne. Des noms de profils peuvent être définis pour des communautés d'utilisateurs qui veulent appliquer des restrictions ou un autre profilage à MGCP.

Dans les messages initiaux, la version va être codée par : MGCP 1.0

Une entité qui reçoit une commande avec une version de protocole qu'elle n'accepte pas, DOIT répondre avec une erreur (le code d'erreur 528 - Version de protocole incompatible, est RECOMMANDÉ). Noter que ceci s'applique aussi aux profils non pris en charge.

3.2.2 Lignes de paramètres

Les lignes de paramètres sont composées d'un nom de paramètre, qui dans la plupart des cas est composé de un ou deux caractères, suivi par un caractère deux points ":", une ou des expaces blanches facultatives et la valeur du paramètre. Les paramètres qui peuvent être présents dans les commandes sont définis dans le tableau suivant :

Nom du paramètre	Code	Valeur du paramètre
BearerInformation	B	voir au paragraphe 3.2.2.1
CallId	C	voir au paragraphe 3.2.2.2
Capabilities	A	voir au paragraphe 3.2.2.3
ConnectionId	I	voir au paragraphe 3.2.2.5
ConnectionMode	M	voir au paragraphe 3.2.2.6
ConnectionParameters	P	voir au paragraphe 3.2.2.7
DetectEvents	T	voir au paragraphe 3.2.2.8
DigitMap	D	codage du texte d'un descriptif de numérotation
EventStates	ES	voir au paragraphe 3.2.2.9
LocalConnectionOptions	L	voir au paragraphe 3.2.2.10
MaxMGCPDatagram	MD	voir au paragraphe 3.2.2.11
NotifiedEntity	N	identifiant, en format de la RFC 821, composé d'une chaîne arbitraire et du nom de domaine de l'entité demandeuse, éventuellement complété par un numéro d'accès, comme dans : Call-agent@ca.exemple.net:5234. Voir aussi au paragraphe 3.2.1.3.
ObservedEvents	O	voir au paragraphe 3.2.2.12
PackageList	PL	voir au paragraphe 3.2.2.13
QuarantineHandling	Q	voir au paragraphe 3.2.2.14
ReasonCode	E	chaîne avec un entier de 3 chiffres facultativement suivi par un ensemble arbitraire de caractères (3.2.2.15).
RequestedEvents	R	voir au paragraphe 3.2.2.16
RequestedInfo	F	voir au paragraphe 3.2.2.17
RequestIdentifier	X	voir au paragraphe 3.2.2.18

ResponseAck	K	voir au paragraphe 3.2.2.19
RestartDelay	RD	nombre de secondes, codé par un nombre décimal.
RestartMethod	RM	voir au paragraphe 3.2.2.20
SecondConnectionId	I2	Identifiant de connexion
SecondEndpointId	Z2	Identifiant de point d'extrémité
SignalRequests	S	voir au paragraphe 3.2.2.21
SpecificEndPointId	Z	identifiant, en format de la RFC 821, composé d'une chaîne arbitraire, suivie par un "@" suivi par le nom de domaine de la passerelle à laquelle ce point d'extrémité est rattaché. Voir aussi le paragraphe 3.2.1.3
RemoteConnectionDescriptor	RC	description de session
LocalConnectionDescriptor	LC	description de session

Les paramètres ne sont pas nécessairement présents dans toutes les commandes. Le tableau suivant donne les associations entre paramètres et commandes. La lettre M indique "obligatoire", O "facultatif" et F "interdit". Sauf mention contraire, un paramètre NE DOIT PAS être présent plus d'une fois.

Nom du paramètre	**EP CF**	**CR CX**	**MD CX**	**DL CX**	**RQ NT**	**NT FY**	**AU EP**	**AU CX**	**RS IP**
BearerInformation	O*	O	O	O	O	F	F	F	F
CallId	F	M	M	O	F	F	F	F	F
Capabilities	F	F	F	F	F	F	F	F	F
ConnectionId	F	F	M	O	F	F	F	M	F
ConnectionMode	F	M	O	F	F	F	F	F	F
Connection Parameters	F	F	F	O*	F	F	F	F	F
DetectEvents	F	O	O	O	O	F	F	F	F
DigitMap	F	O	O	O	O	F	F	F	F
EventStates	F	F	F	F	F	F	F	F	F
LocalConnection Options	F	O	O	F	F	F	F	F	F
MaxMGCPDatagram	F	F	F	F	F	F	F	F	F
NotifiedEntity	F	O	O	O	O	O	F	F	F
ObservedEvents	F	F	F	F	F	M	F	F	F
PackageList	F	F	F	F	F	F	F	F	F
QuarantineHandling	F	O	O	O	O	F	F	F	F
ReasonCode	F	F	F	O	F	F	F	F	O
RequestedEvents	F	O	O	O	O*	F	F	F	F
RequestIdentifier	F	O*	O*	O*	M	M	F	F	F
RequestedInfo	F	F	F	F	F	F	O	M	F
ResponseAck	O	O	O	O	O	O	O	O	O
RestartDelay	F	F	F	F	F	F	F	F	O
RestartMethod	F	F	F	F	F	F	F	F	M
SecondConnectionId	F	F	F	F	F	F	F	F	F
SecondEndpointId	F	O	F	F	F	F	F	F	F
SignalRequests	F	O	O	O	O*	F	F	F	F
SpecificEndpointId	F	F	F	F	F	F	F	F	F
RemoteConnection Descriptor	F	O	O	F	F	F	F	F	F
LocalConnection Descriptor	F	F	F	F	F	F	F	F	F

Notes (*) :
* Le paramètre BearerInformation est seulement conditionnellement facultatif comme expliqué au paragraphe 2.3.2.
* Le paramètre RequestIdentifier est facultatif dans les commandes Création, Modification et suppression de connexion, cependant il devient EXIGÉ si la commande contient une demande de notification encapsulée.
* Les paramètres RequestedEvents et SignalRequests sont facultatifs dans NotificationRequest. Si ces paramètres sont omis, les listes correspondantes vont être considérées comme vides.
* Le paramètre ConnectionParameters n'est valide que dans une demande DeleteConnection envoyée par la passerelle.

L'ensemble des paramètres peut être étendu de deux façons differentes :
* Paramètres d'extension de paquetage (préférée)
* Paramètres d'extension de fabricant

Les paramètres d'extension de paquetage sont définis dans les paquetages qui fournissent les avantages suivants :
* un mécanisme d'enregistrement (par l'IANA) du nom de paquetage,

* un espace de noms séparé pour les paramètres,
* un groupement pratique des extensions,
* une façon simple de déterminer leur prise charge par examen.

Le mécanisme d'extension de paquetage est la méthode d'extension préférée.

Les paramètres d'extension de fabricant peuvent être utilisés si les mises en œuvre ont besoin d'expérimenter avec de nouveaux paramètres, par exemple quand elles développent une nouvelle application de MGCP. Les paramètres d'extension de fabricant DOIVENT être identifiés par des noms qui commencent par la chaîne "X-" ou "X+", comme par exemple :

3,X-Fleur: Marguerite

Les noms de paramètre qui commencent par "X+" sont des extensions de paramètre critiques. Une entité MGCP qui reçoit une extension de paramètre critique qu'elle ne peut pas comprendre DOIT refuser d'exécuter la commande. Elle DEVRAIT répondre avec le code d'erreur 511 (Extension non reconnue).

Les noms de paramètre qui commencent par "X-" sont des extensions de paramètre non critiques. Une entité MGCP qui reçoit une extension de paramètre non critique qu'elle ne peut pas comprendre DOIT simplement ignorer ce paramètre.

Noter que les paramètres d'extension de fabricant utilisent un espace de noms non géré, qui implique une éventualité de conflit de noms. Les fabricants sont par conséquent invités à inclure une chaîne spécifique du fabricant, par exemple, le nom du fabricant, dans leurs extensions.

3.2.2.1 BearerInformation

Les valeurs des informations de support sont codées comme une liste d'attributs séparés par des virgules, qui sont représentés par un nom d'attribut, éventuellement suivi par deux points et une valeur d'attribut.

Le seul attribut défini est l'attribut "encoding" (code "e") qui DOIT avoir une des valeurs "A" (loi A) ou "mu" (loi mu).

Un exemple de codage d'informations de support est :

B: e:mu

L'ensemble des attributs d'informations de support peut être étendu par des paquetages.

3.2.2.2 CallId

L'identifiant d'appel est codé comme une chaîne hexadécimale d'au plus 32 caractères. Les identifiants d'appel sont comparés comme des chaînes plutôt que comme des valeurs numériques.

3.2.2.3 Capabilities

Capabilities informe l'agent d'appel sur les capacités des points d'extrémité lors d'un audit. Le codage des capacités est fondé sur le codage des options de connexion locales pour les paramètres qui sont communs aux deux, bien qu'un code de ligne de paramètre différent soit utilisé ("A"). De plus, les capacités peuvent aussi contenir une liste de paquetages pris en charge, et une liste des modes pris en charge.

Les paramètres utilisés sont :

Une liste des codecs pris en charge. Les paramètres suivants vont s'appliquer à tous les codecs spécifiés dans cette liste. Si il est besoin de spécifier que certains paramètres, comme par exemple, suppression de silence, ne sont compatibles qu'avec certains codecs, la passerelle va alors retourner plusieurs paramètres Capability ; un pour chaque ensemble de codecs.

Période de mise en paquets : une gamme peut être spécifiée.

Bande passante : une gamme correspondant à la gamme des périodes de mise en paquets peut être spécifiée (en supposant qu'il n'y a pas de suppression de silence). Si ce paramètre est absent, les valeurs vont être déduites du type de codec.

Annulation d'écho : "on" si l'annulation d'écho est prise en charge, "off" autrement. La prise en charge est par défaut.

Suppression de silence : "on" si la suppression de silence est prise en charge pour ce codec, "off" autrement. La prise en charge est par défaut.

Contrôle de gain : "0" si le contrôle de gain n'est pas pris en charge, toutes les autres valeurs indiquent la prise en charge du contrôle de gain. La prise en charge est par défaut.

Type de service : la valeur "0" indique qu'il n'y a pas de prise en charge du type de service, toutes les autres valeurs indiquent la prise en charge du type de service. La prise en charge est par défaut.

Service de réservation de ressource : le paramètre indique les services de réservation qui sont pris en charge, en plus du service au mieux. La valeur "g" est codée quand la passerelle prend en charge les deux services garanti et à charge contrôlée, "cl" quand seul le service à charge contrôlée est pris en charge. "Au mieux" est par défaut.

Clé de chiffrement : coder une valeur quelconque indique la prise en charge du chiffrement. La non prise en charge est par défaut, qui est impliquée en omettant le paramètre.

Type de réseau : le mot-clé "nt", suivi par un deux-points et une liste séparée par des points-virgules des types de réseaux pris en charge. Ce paramètre est facultatif.

Paquetages : paquetages pris en charge par le point d'extrémité codés par le mot-clé "v", suivi par un deux-points et une chaîne de caractères. Si une liste de valeurs est spécifiée, ces valeurs vont être séparées par un point-virgule. La première valeur spécifiée va être le paquetage par défaut pour le point d'extrémité.

Modes : modes pris en charge par ce point d'extrémité codés par le mot-clé "m", suivi par un deux-points et une liste séparée par des points-virgules des modes de connexion pris en charge pour ce point d'extrémité.

La non prise en charge d'une capacité peut aussi être indiquée en excluant le paramètre de l'ensemble de capacités.

Un exemple de capacités est :

A: a:PCMU;G728, p:10-100, e:on, s:off, t:1, v:L, m:sendonly;recvonly;sendrecv;inactive

Si plusieurs capacités sont à retourner, chacune va être retournée dans une ligne de capacité séparée.

Comme les options de connexion locale peuvent être étendues, la liste des paramètres de capacités peut aussi être étendue. Des extensions individuelles peuvent définir comment elles sont rapportées comme capacités. Si une telle définition n'est pas fournie, les valeurs par défaut suivantes s'appliquent :

* attributs d'extension de paquetage : les attributs individuels ne sont pas rapportés. À la place, le nom du paquetage est simplement rapporté dans la liste des paquetages pris en charge.
* attributs d'extension de fabricant : le nom de l'attribut est rapporté sans aucune valeur.
* autres attributs d'extension : le nom de l'attribut est rapporté sans aucune valeur.

3.2.2.4 Codage des noms d'événement

Les noms d'événements sont composés d'un nom de paquetage facultatif, séparé par une barre oblique (/) du nom de l'événement réel (voir au paragraphe 2.1.7). Le caractère générique étoile ("*") peut être utilisé pour se référer à tous les paquetages. Le nom d'événement peut facultativement être suivi par un signe arobase (@) et l'identifiant d'une connexion (éventuellement en utilisant un caractère générique) sur laquelle l'événement devrait être observé. Les noms d'événement sont utilisés dans les paramètres RequestedEvents, SignalRequests, ObservedEvents, DetectEvents, et EventStates.

Les événements et signaux peuvent être qualifiés par des paramètres définis pour l'événement/signal. De tels paramètres peuvent être enclos dans des guillemets (en fait, certains paramètres DOIVENT être enclos dans des guillemets à cause de restrictions syntaxiques) et dans ce cas ils sont codés en UTF-8 ([RFC2279]).

Le nom de paramètre "!" (point d'exclamation) est réservé pour une utilisation future pour les événements et signaux.

Chaque signal est associé à un des types suivants : On/Off (OO), Time-out (TO), ou Bref (BR). (Ces types de signaux sont spécifiés dans les définitions de paquetage, et ne sont pas présents dans les messages.) Les signaux On/Off peuvent être

paramétrés avec un "+" pour activer le signal, ou un "-" pour désactiver le signal. Si un signal on/off n'est pas paramétré, le signal est activé. Les deux notations suivantes vont activer le signal vmwi (d'après la ligne de paquetage "L") :

L/vmwi(+)
L/vmwi

En plus de "!", "+" et "-", le paramètre de signal "to" est aussi réservé. Il peut être utilisé avec les signaux Time-Out pour outrepasser la valeur de fin de temporisation par défaut pour la demande en cours. Une valeur décimale en millisecondes va être fournie. La définition de signal et/ou paquetage individuel DEVRAIT indiquer si ce paramètre est pris en charge pour un ou plusieurs signaux TO dans le paquetage. Si ils ne sont pas indiqués, les signaux TO dans la version zéro du paquetage sont supposés ne pas le prendre en charge, tandis que les signaux TO dans les versions une ou plus de paquetage sont supposés le prendre en charge. Par défaut, une valeur de fin de temporisation fournie PEUT être arrondie à la plus proche valeur non zéro divisible par 1000, c'est-à-dire, une seconde entière. La définition de signal et/ou paquetage individuel peut utiliser d'autres règles d'arrondi. Toute nouvelle définition de paquetage et signal TO est vivement encouragée à prendre en charge le paramètre de signal "to".

L'exemple suivant illustre comment le paramètre "to" peut être utilisé pour s'appliquer à un signal pendant 6 secondes :

L/rg(to=6000)
L/rg(to(6000))

Voici des exemples de noms d'événement :
L/hu : transition à raccroché, dans le paquetage de ligne.
F/0 : chiffre 0 dans le paquetage MF.
hf : impulsion crochet *(Hook-flash)* en supposant que le paquetage de ligne est le paquetage par défaut pour le point d'extrémité.
G/rt@0A3F58 : signal de retour d'appel sur la connexion "0A3F58".

De plus, la notation de gamme et de caractère générique des événements peut être utilisée, à la place des noms individuels, dans les paramètres RequestedEvents et DetectEvents. Le code d'événement "all" est réservé et se réfère à tous les événements ou signaux dans un paquetage. Le signe étoile ("*") peut être utilisé pour noter "toutes les connexions", et le signe dollar ("$") peut être utilisé pour noter la connexion "en cours" (voir les détails au paragraphe 2.1.7).

Voici des exemples de ces notations :
M/[0-9] : chiffres de 0 à 9 dans le paquetage MF.
hf : (Hook-flash) impulsion crochet, en supposant que le paquetage de ligne est un paquetage par défaut pour le point d'extrémité.
[0-9*#A-D] : tous les chiffres et lettres dans les paquetages DTMF (par défaut pour le point d'extrémité).
T/all : tous les événements dans le paquetage de circuit.
R/qa@* : événement d'alerte de qualité sur toutes les connexions.
G/rt@$: retour d'appel sur la connexion en cours.

3.2.2.5 ConnectionId

L'identifiant de connexion est codé comme une chaîne hexadécimale, d'au plus 32 caractères. Les identifiants de connexion sont comparés comme des chaînes plutôt que comme des valeurs numériques.

3.2.2.6 ConnectionMode

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

Mode	**Signification**
M: sendonly	La passerelle devrait seulement envoyer des paquets
M: recvonly	La passerelle devrait seulement recevoir des paquets
M: sendrecv	La passerelle devrait envoyer et recevoir des paquets
M: confrnce	La passerelle devrait placer la connexion en mode conférence
M: inactive	La passerelle ne devrait ni envoyer ni recevoir de paquets
M: loopback	La passerelle devrait placer le circuit en mode de rebouclage arrière.
M: conttest	La passerelle devrait placer le circuit en mode essai.

M: netwloop La passerelle devrait placer la connexion en mode rebouclage de réseau.
M: netwtest La passerelle devrait placer la connexion en mode essai de continuité du réseau.

Noter que sans considération du mode de connexion, les signaux appliqués à la connexion vont résulter en l'envoi de paquets (voir le paragraphe 2.3.1).

L'ensemble des modes de connexion peut être étendu par des paquetages.

3.2.2.7 ConnectionParameters

Les paramètres de connexion sont codés comme une chaîne de paires de type et valeur, où le type est soit un identifiant de deux lettres du paramètre soit un type d'extension, et la valeur est un entier décimal. Les types sont séparés de la valeur par un signe "=". Les paramètres sont séparés les uns des autres par une virgule. Les valeurs de paramètre de connexion peuvent contenir jusqu'à neuf chiffres. Si la valeur maximum est atteinte, le compteur n'est plus mis à jour, c'est-à-dire, il ne revient pas à zéro ni ne déborde.

Les types de paramètres de connexion sont spécifié dans le tableau suivant :

Nom de paramètre de connexion	Valeur de code	Paramètre de connexion
Paquets envoyés	PS	nombre de paquets envoyés sur la connexion.
Octets envoyés	OS	nombre d'octets envoyés sur la connexion.
Paquets reçus	PR	nombre de paquets reçus sur la connexion.
Octets reçus	OR	nombre d'octets reçus sur la connexion.
Paquets perdus	PL	nombre de paquets perdus sur la connexion déduits des trous du numéro de séquence RTP.
Gigue	JI	gigue moyenne d'arrivée inter-paquets, en millisecondes, exprimée par un nombre entier.
Latence	LA	latence moyenne, en millisecondes, exprimée par un nombre entier.

L'ensemble des paramètres de connexion peut être étendu de deux façons différentes :
* Paramètres d'extension de paquetage (préférée)
* Paramètres d'extension de fabricant

Les paramètres de connexion d'extension de paquetage sont définis dans des paquetages qui fournissent les avantages suivants :
* un mécanisme d'enregistrement (par l'IANA) pour le nom du paquetage,
* un espace de noms séparé pour les paramètres,
* un groupement pratique des extensions,
* une façon simple de déterminer leur prise en charge par examen.

Le mécanisme d'extension de paquetage est la méthode d'extension préférée.

Les noms des paramètres d'extension de fabricant sont composés de la chaîne "X-" suivie par un nom de paramètre d'extension de deux lettres ou plus.

Les agents d'appel qui reçoivent des paramètre d'extensions de paquetage ou de connexion de fabricant non reconnus DEVRONT ignorer en silence ces paramètres.

Un exemple de codage de paramètre de connexion est :

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

3.2.2.8 DetectEvents

Le paramètre DetectEvents est codé comme une liste d'événements séparés par des virgules (voir le paragraphe 3.2.2.4) comme par exemple :

T: L/hu,L/hd,L/hf,D/[0-9#*]

On notera qu'aucune action ne peut être associée aux événements, cependant des paramètres d'événement peuvent être fournis.

3.2.2.9 EventStates

Le paramètre EventStates est codé comme une liste d'événements séparés par des virgules (voir le paragraphe 3.2.2.4) comme par exemple :

ES: L/hu

On notera qu'aucune action ne peut être associée aux événements ; cependant des paramètres d'événement peuvent être fournis.

3.2.2.10 LocalConnectionOptions

Les options de connexion locale décrivent les paramètres de fonctionnemenh que l'agent d'appel fournit à la passerelle dans les commandes de traitement de la connexion. Elles incluent :

* Le ou les codecs permis, codés par le mot-clé "a", suivi par deux-points et une chaîne de caractères. Si l'agent d'appel spécifie une liste de valeurs, ces valeurs vont être séparées par un point-virgule. Pour RTP, les codecs audio DEVRONT être spécifiés en utilisant les noms de codage défini dans le profil AV de RTP [RFC1890] ou son remplacement, ou par les noms de codages enregistrés par l'IANA. Les supports non audio enregistrés comme type MIME DOIVENT utiliser la forme "<type MIME>/<sous type MIME>", comme dans "image/t38".

* La période de mise en paquets en millisecondes, codée par le mot-clé "p", suivi par deux-points et un nombre décimal. Si l'agent d'appel spécifie une gamme de valeurs, la gamme va être spécifiée par deux nombres décimaux séparés par un trait d'union (comme spécifié pour le paramètre "ptime" pour SDP).

* La bande passante en kilo bits par seconde (1000 bits par seconde) codée par le mot-clé "b", suivi par deux-points et un nombre décimal. Si l'agent d'appel spécifie une gamme de valeurs, la gamme va être spécifiée par deux nombres décimaux séparés par un trait d'union.

* Le paramètre Type de service, codé par le mot-clé "t", suivi par deux-points et la valeur codée par deux chiffres hexadécimaux. Quand la connexion est transmise sur un réseau IP, les paramètres codent le paramètre de valeur de type de service de 8 bits de l'en-tête IP (autrement dit, le champ DiffServ). Le "bit" le plus à gauche dans le paramètre correspond au bit de moindre poids de l'en-tête IP.

* Le paramètre Annulation d'écho, codé par le mot-clé "e", suivi par deux-points et la valeur "on" ou "off".

* Le paramètre Contrôle de gain, codé par le mot-clé "gc", suivi par deux-points et une valeur qui peut être soit le mot-clé "auto", soit un nombre décimal (positif ou négatif) représentant le nombre de décibels de gain.

* Le paramètre suppression de silence, codé par le mot-clé "s", suivi par deux-points et la valeur "on" ou "off".

* Le paramètre réservation de ressource, codé par le mot-clé "r", suivi par deux-points et la valeur "g" (service garanti) "cl" (charge contrôlée) ou "be" (au mieux, *(best effort)*).

* La clé de chiffrement, codée par le mot-clé "k" suivi par deux-points et une spécification de clé, comme défini pour le paramètre "K" dans SDP (RFC 2327).

* Le type de réseau, codé par le mot-clé "nt" suivi par deux-points et le type de réseau codé par le mot-clé "IN" (Internet), "ATM", "LOCAL" (pour une connexion locale), ou éventuellement un autre type de réseau enregistré auprès de l'IANA selon SDP (RFC 2327).

Le codage des trois premiers attributs, quand ils sont présents, va être compatible avec les profils SDP et RTP. Noter que chacun des attributs est facultatif. Quand plusieurs attributs sont présents, ils sont séparés par une virgule.

Des exemples d'options de connexion locale sont :
L: p:10, a:PCMU

L: p:10, a:G726-32
L: p:10-20, b:64
L: b:32-64, e:off

L'ensemble des attributs d'options de connexion locale peut être étendu de trois façons différentes :
* Attributs d'extension de paquetage (préféré)
* Attributs d'extension de fabricant
* Autres attributs d'extension

Les attributs d'options de connexion locale d'extension de paquetage sont définis dans les paquetages qui procurent les avantages suivants :
* un mécanisme d'enregistrement (auprès de l'IANA) pour le nom du paquetage,
* un espace de noms séparé pour les attributs,
* un groupage pratique des extensions,
* une façon simple de déterminer leur prise en charge par l'audit.

Le mécanisme d'extension de paquetage est la méthode d'extension préférée.

Les attributs d'extension de fabricant sont composés d'un nom d'attribut, éventuellement suivi par deux-points et une valeur d'attribut. Le nom d'attribut DOIT commencer par les deux caractères "x+", pour une extension obligatoire, ou "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 (le code d'erreur 525 - Extension inconnue dans LocalConnectionOptions, est RECOMMANDÉ).

Noter que les attributs d'extension de fabricant utilisent un espace de noms non géré, ce qui implique une possibilité de conflit de noms. Les fabricants sont par conséquent encouragés à inclure une chaîne spécifique du fabricant, par exemple, son nom, dans leurs extensions.

Finalement, pour la rétro compatibilité avec certaines mises en œuvre existantes, MGCP permet aussi d'autres attributs d'extension (voir la grammaire à l'Appendice A). Noter cependant que ces extensions d'attributs ne fournisent pas les avantages de l'attribut d'extension de paquetage. L'utilisation de ce mécanisme pour de nouvelles extensions est déconseillée.

3.2.2.11 MaxMGCPDatagram

MaxMGCPDatagram peut seulement être utilisé pour l'audit, c'est-à-dire, c'est un code valide de RequestedInfo et il peut être fourni comme paramètre de réponse.

Dans les réponses, la valeur MaxMGCPDatagram est codée comme une chaîne de jusqu'à neuf chiffres décimaux -- les zéros en tête ne sont pas permis. L'exemple suivant illustre l'utilisation de ce paramètre :

MD: 8100

3.2.2.12 ObservedEvents

Le paramètre Événements observés donne la liste des événements qui ont été observés. Les codes d'événement sont les mêmes que ceux utilisés dans NotificationRequest. Les événements qui ont été accumulés en accord avec le script de numérotation peuvent être groupés dans une seule chaîne, cependant cette pratique est déconseillée ; ils DEVRAIENT être rapportés comme des listes d'événements isolés si d'autres événements ont été détectés durant l'accumulation des chiffres. Des exemples d'événements observés sont :

O: L/hu
O: D/8295555T
O: D/8,D/2,D/9,D/5,D/5,L/hf,D/5,D/5,D/T
O: L/hf, L/hf, L/hu

3.2.2.13 PackageList

La liste de paquetages peut seulement être utilisée pour l'audit, c'est-à-dire que c'est un code valide de RequestedInfo et qu'il peut être fourni comme paramètre de réponse.

Le paramètre de réponse va consister en une liste séparée par des virgules des paquetages pris en charge. Le premier paquetage retourné dans la liste est le paquetage par défaut. Chaque paquetage de la liste consiste en le nom du paquetage suivi par deux-points, et le plus fort numéro de version du paquetage prise en charge.

Un exemple de liste de paquetages est :

PL: L:1,G:1,D:0,FOO:2,T:1

Noter que pour la rétro compatibilité, la prise en charge de ce paramètre est FACULTATIVE.

3.2.2.14 QuarantineHandling

Le paramètre Traitement de quarantaine contient une liste de mots-clés séparés par des virgules :

* Le mot-clé "process" ou "discard" pour indiquer le traitement de quarantaine et les événements observés. Si ni "process" ni "discard" n'est présent, "process" est supposé.

* Le mot-clé "step" ou "loop" pour indiquer si au plus une notification par NotificationRequest est permise, ou si plusieurs notifications par NotificationRequest sont permises. Si ni "step" ni "loop" n'est présent, "step" est supposé.

Les valeurs suivantes sont des exemples valides :
Q: loop
Q: process
Q: loop,discard

3.2.2.15 ReasonCode

Les codes de cause sont des valeurs numériques de trois chiffres. Le code de cause est facultativement suivi par une espace et un commentaire, par exemple :

E: 900 Mauvais fonctionnement du point d'extrémité

Une liste des codes de cause se trouve au paragraphe 2.5.

L'ensemble des codes de cause peut être étendu par des paquetages.

3.2.2.16 RequestedEvents

Le paramètre RequestedEvents donne la liste des événements qui sont demandés. Les codes d'événement sont décrits au paragraphe 3.2.2.4.

Chaque événement peut être qualifié par une action demandée, ou par une liste d'actions. Les actions, quand elles sont spécifiées, sont codées par une liste de mots-clés, enclos entre parenthèses et séparés par des virgules. Les codes des différentes actions sont :

Action	**Code**
Notification immédiate	N
Accumuler	A
Traiter selon le script de numérotation	D
Échanger	S
Ignorer	I
Garder les signaux actifs	K
Demande de notification incorporée	E

Quand aucune action n'est spécifiée, l'action par défaut est de notifier l'événement. Cela signifie que, par exemple, ft et ft(N) sont équivalents. Les événements qui ne sont pas dans la liste sont ignorés (sauf si ils sont persistants).

L'action de transposition de chiffres *(digit-map)* DEVRAIT n'être spécifiée que pour les chiffres, lettres et temporisations

entre les chiffres dans les paquetages qui définissent le codage des chiffres, lettres, et temporisateurs (incluant les lettres d'extension de script de numérotation).

La liste des événements demandés est codée sur une seule line, avec groupes d'événements/actions séparés par des virgules. Des exemples de codage de RequestedEvents sont :

R: L/hu(N), L/hf(S,N)
R: L/hu(N), D/[0-9#T](D)

Dans le cas de l'action "Demande de notification incorporée", les paramètres de la demande de notification incorporée sont codés comme une liste de jusqu'à trois groupes de paramètres séparés par des virgules. Chaque groupe commence par un identifiant d'une lettre, suivi par une liste de paramètres enclos entre des parenthèses. Le premier groupe de paramètres facultatifs, identifié par la lettre "R", est la valeur du paramètre RequestedEvents incorporé. Le second groupe facultatif, identifié par la lettre "S", est la valeur incorporée du paramètre SignalRequests. Le troisième groupe facultatif, identifié par la lettre "D", est la valeur incorporée de DigitMap. (Noter que certaines mises en œuvre et profils existants peuvent coder ces trois composants dans un ordre différent. Les mises en œuvre sont encouragées à accepter de tels codages, mais elles NE DEVRAIENT PAS les générer.)

Si le paramètre RequestedEvents n'est pas présent, le paramètre va être réglé à une valeur nulle. Si le paramètre SignalRequests n'est pas présent, le paramètre va être réglé à une valeur nulle. Si le DigitMap est absent, la valeur actuelle DOIT être utilisée. Les exemples suivants sont des demandes incorporées valides :

R: L/hd(E(R(D/[0-9#T](D),L/hu(N)),S(L/dl),D([0-9].[#T])))
R: L/hd(E(R(D/[0-9#T](D),L/hu(N)),S(L/dl)))

Certains événements peuvent être qualifiés en ajoutant des paramètres d'événement. De tels paramètres d'événement vont être séparés par des virgules et enclos dans des parenthèses. Les paramètres d'événement peuvent être enclos dans des guillemets (en fait, certains paramètres d'événement DOIVENT être enclos dans des guillemets pour des restrictions syntaxiques) et dans ce cas la chaîne entre guillemets est elle-même codée en UTF-8. Voir au paragraphe 3.2.2.4 les détails des paramètres d'événement.

L'exemple suivant montre l'événement foobar avec un paramètre d'événement "epar" :

R: X/foobar(N)(epar=2)

On remarquera que l'action a été incluse bien qu'elle soit l'action par défaut Notifier - c'est exigé par la grammaire.

3.2.2.17 RequestedInfo

Le paramètre RequestedInfo contient une liste séparée de virgules des codes de paramètres, comme défini au paragraphe 3.2.2. Par exemple, si on veut examiner la valeur des paramètres NotifiedEntity, RequestIdentifier, RequestedEvents, SignalRequests, DigitMap, QuarantineHandling et DetectEvents, la valeur du paramètre RequestedInfo va être :

F: N,X,R,S,D,Q,T

Noter que les paramètres d'extension en général peuvent aussi être examinés. L'extension individuelle va définir l'opération d'audit.

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

F: A

3.2.2.18 RequestIdentifier

L'identifiant de demande corrèle une commande Notify avec la demande de notification qui l'a déclenchée. Un identifiant de demande est une chaîne hexadécimale d'au plus 32 caractères. Les identifiants de demande sont comparés comme des chaînes plutôt que des valeurs numériques. La chaîne "0" est réservée pour rapporter les événements persistants dans le cas où une demande de notification n'a pas encore été reçue après un redémarrage.

3.2.2.19 ResponseAck

Le paramètre Accusé de réception de réponse est utilisé pour gérer la facilité "au plus une fois" décrite au paragraphe 3.5. Il contient une liste séparée par des virgules de "gammes confirmées d'identifiants de transaction".

Chaque "gamme confirmée d'identifiants de transaction" est composée d'un nombre décimal, quand la gamme comporte exactement une transaction, ou de deux nombres décimaux séparés par un seul trait d'union, décrivant le plus bas et le plus haut identifiant de transaction inclus dans la gamme.

Un exemple d'accusé de réception de réponse est :

K: 6234-6255, 6257, 19030-19044

3.2.2.20 RestartMethod

Le paramètre RestartMethod est codé par un des mots-clés "graceful" *(en douceur)*, "forced" *(forcé)*, "restart" *(redémarre)*, "deconnected" *(déconnecté)* ou "cancel-graceful" *(annulation en douceur)* comme par exemple :

RM: restart

L'ensemble des méthodes de redémarrage peut être étendu par les paquetages.

3.2.2.21 SignalRequests

Le paramètre SignalRequests donne le nom du ou des signaux qui ont été demandés. Chaque signal est identifié par un nom, comme décrit au paragraphe 3.2.2.4.

Certains signaux, comme par exemple d'annonce ou d'affichage ADSI, peuvent être qualifiés par des paramètres supplémentaires, par exemple :
* le nom et les paramètres de l'annonce,
* la chaîne qui devrait être affichée.

De tels paramètres vont être séparés par des virgules et enclos dans des parenthèses, comme dans :

S: L/adsi("123456 Francois Gerard")
S: A/ann(http://ann.exemple.net/no-such-number.au, 1234567)

Quand une chaîne entre guillemets est fournie, la chaîne elle-même est codée en UTF-8 [RFC2279].

Quand plusieurs signaux sont demandés, leurs codes sont séparés par une virgule, comme dans :

S: L/adsi("123456 Ton ami"), L/rg

Voir au paragraphe 3.2.2.4 les détails des paramètres de signal.

3.3 Format des en-têtes de réponse

L'en-tête de réponse est composé d'une ligne de réponse, facultativement suivie par les en-têtes qui codent les paramètres de réponse.

Un exemple d'en-tête de réponse pourrait être : 200 1203 OK

La ligne de réponse commence par le code de réponse, qui est une valeur numérique de trois chiffres. Le code est suivi par une espace, et l'identifiant de transaction. Les codes de réponse définis dans les paquetages (8xx) sont suivis par une espace, une barre oblique ("/") et le nom de paquetage. Tous les codes de réponse peuvent de plus être suivis par un commentaire facultatif précédé d'une espace.

Le tableau suivant décrit les paramètres dont la présence est obligatoire ou facultative dans un en-tête de réponse, comme une fonction de la commande qui a déclenché la réponse. La lettre M signifie "obligatoire", O "facultatif" et F "Interdit". Sauf mention contraire, un paramètre NE DOIT PAS être présent plus d'une fois. Noter que le tableau reflète seulement la valeur par défaut pour les réponses qui n'ont pas défini d'autre comportement. Si une réponse est reçue avec un paramètre qui n'est pas compris ou est marqué comme interdit, le ou les paramètres en cause DOIVENT simplement être ignorés.

Nom du paramètre	**EP CF**	**CR CX**	**MD CX**	**DL CX**	**RQ NT**	**NT FY**	**AU EP**	**AU CX**	**RS IP**
BearerInformation	F	F	F	F	F	F	O	F	F
CallId	F	F	F	F	F	F	F	O	F
Capabilities	F	F	F	F	F	F	O*	F	F
ConnectionId	F	O*	F	F	F	F	O*	F	F
ConnectionMode	F	F	F	F	F	F	F	O	F
Connection Parameters	F	F	F	O*	F	F	F	O	F
DetectEvents	F	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	F	O	F	F
LocalConnectionOptions	F	F	F	F	F	F	F	O	F
MaxMGCPDatagram	F	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	F	O	O	O
ObservedEvents	F	F	F	F	F	F	O	F	F
QuarantineHandling	F	F	F	F	F	F	O	F	F
PackageList	O*	O*	O*	O*	O*	O*	O	O*	O*
ReasonCode	F	F	F	F	F	F	O	F	F
RequestIdentifier	F	F	F	F	F	F	O	F	F
ResponseAck	O*	O*	O*	O*	O*	O*	O*	O*	O*
RestartDelay	F	F	F	F	F	F	O	F	F
RestartMethod	F	F	F	F	F	F	O	F	F
RequestedEvents	F	F	F	F	F	F	O	F	F
RequestedInfo	F	F	F	F	F	F	F	F	F
SecondConnectionId	F	O	F	F	F	F	F	F	F
SecondEndpointId	F	O	F	F	F	F	F	F	F
SignalRequests	F	F	F	F	F	F	O	F	F
SpecificEndpointId	F	O	F	F	F	F	O*	F	F
LocalConnectionDescriptor	F	O*	O	F	F	F	F	O*	F
RemoteConnectionDescriptor	F	F	F	F	F	F	F	O*	F

Notes (*) :

* Le paramètre PackageList n'est permis qu'avec le code de retour 518 (Paquetage non pris en charge) sauf pour AuditEndpoint, où il peut aussi être retourné en cas d'audit.
* Le paramètre ResponseAck NE DOIT PAS être utilisé avec d'autre réponse qu'une réponse finale produite après une réponse provisoire pour la transaction en question. Dans ce cas, la présence du paramètre ResponseAck DEVRAIT déclencher un accusé de réception de réponse - toute valeur ResponseAck fournie sera ignorée.
* Dans le cas d'un message CreateConnection, la ligne de réponse est suivie par un paramètre Connection-Id et un LocalConnectionDescriptor. Elle peut aussi être suivie d'un paramètre Specific-Endpoint-Id, si la demande de création a été envoyée à un Endpoint-Id avec un caractère générique. Les paramètres connexion-Id et LocalConnectionDescriptor sont marqués comme facultatifs dans le tableau. En fait, ils sont obligatoires avec toutes les réponses positives, quand une connexion a été créée, et interdits quand la réponse est négative, et qu'aucune connexion n'a été créée.
* Un LocalConnectionDescriptor DOIT être transmis avec une réponse positive (code 200) à un CreateConnection. Il DOIT aussi être transmis en réponse à une commande ModifyConnection, si la modification a résulté en une modification des paramètres de session. Le LocalConnectionDescriptor est codé comme une "description de session", comme défini au paragraphe 3.4. Il est séparé de l'en-tête de réponse par une ligne vide.
* Les paramètres de connexion ne sont valides que dans une réponse à une commande DeleteConnection sans caractère générique envoyée par l'agent d'appel.
* Les paramètres multiples ConnectionId, SpecificEndpointId, et Capabilities peuvent être présents dans la réponse à une commande AuditEndpoint.
*Quand plusieurs descripteurs de session sont codés dans la même réponse, ils sont codés l'un après l'autre, séparés par une ligne vide. C'est le cas par exemple quand la réponse à une demande d'audit de connexion porte à la fois une description de session locale et une description de session distante, comme dans :

200 1203 OK

```
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:PCMU;G726-32
M: sendrecv
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 1296 RTP/AVP 0

v=0
o=- 33343 346463 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

Dans cet exemple, en accord avec la syntaxe de SDP, chaque description commence avec une ligne "version", (v=...). La description locale est toujours transmise avant la description distante. Si un descripteur de connexion est demandé, mais qu'il n'en existe pas pour la connexion examinée, ce descripteur de connexion va apparaître avec seulement le champ de version de protocole SDP.

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

3.3.1 Réponse CreateConnection

Dans le cas d'u message CreateConnection, la ligne de réponse est suivie par un paramètre Connection-Id avec une réponse de succès (code 200). Un LocalConnectionDescriptor est de plus transmis avec une réponse positive. Le LocalConnectionDescriptor est codé comme une "description de session", comme défini par SDP (RFC 2327). Il est séparé de l'en-tête de 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
```

Quand une réponse provisoire a été précédemment fournie, la réponse finale DEVRAIT de plus contenir le paramètre Accusé de réception de réponse (les réponses finales produites pas des entités qui suivent la présente spécification vont inclure le paramètre, mais les anciennes mises en œuvre de la RFC 2705 PEUVENT ne pas le faire) :

```
200 1204 OK
K:
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
```

La réponse finale DEVRAIT alors être acquittée par un accusé de réception de réponse :

```
000 1204
```

3.3.2 Réponse ModifyConnection

Dans le cas d'un message ModifyConnection réussi, la ligne de réponse est suivie par un LocalConnectionDescriptor, si la modification a résulté en une modification des paramètres de session (par exemple, changer seulement le mode d'une connexion n'altère pas les paramètres de session). Le LocalConnectionDescriptor est codé comme une "description de session", comme défini par SDP. Il est séparé de l'en-tête de 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
```

Quand une réponse provisoire a été produite précédemment, la réponse finale DEVRAIT de plus contenir le paramètre Accusé de réception de réponse comme dans :

```
200 1207 OK
K:
```

La réponse finale DEVRAIT alors être acquittée par un accusé de réception de réponse:

```
000 1207 OK
```

3.3.3 Réponse DeleteConnection

Selon la variante du message DeleteConnection, la ligne de réponse peut être suivie par une ligne de paramètre Paramètres de connexion, comme défini au paragraphe 3.2.2.7.

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

3.3.4 Réponse NotificationRequest

Une réponse réussie de NotificationRequest n'inclut pas de paramètres de réponse supplémentaires.

3.3.5 Réponse Notify

Une réponse Notify réussie n'inclut pas de paramètres de réponse supplémentaires.

3.3.6 Réponse AuditEndpoint

Dans le cas d'un AuditEndPoint réussi, la ligne de réponse peut être suivie par des informations pour chacun des paramètres demandés - chaque paramètre va apparaître sur une ligne séparée. Les paramètres pour lesquels aucune valeur n'existe actuellement, par exemple, le script de numérotation, vont quand même être fournis mais avec une valeur vide. Chaque nom de point d'extrémité local "étendu" avec un caractère générique va apparaître sur une ligne séparée, en utilisant le code de paramètre "SpecificEndPointId", par exemple :

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

Quand les identifiants de connexion sont examinés et que plusieurs connexions existent sur le point d'extrémité, une liste

séparée par des virgules d'identifiants de connexion DEVRAIT être retournée comme dans :

```
200 1200 OK
I: FDE234C8, DFE233D1
```

Autrement, plusieurs lignes de paramètre Identifiant de connexion peuvent être retournées - les deux formes ne devraient pas être mélangées bien que le faire ne constitue pas une erreur.

Quand les capacités sont examinées, la réponse peut inclure plusieurs lignes de paramètres capabilities comme dans :

```
200 1200 OK
A: a:PCMU;G728, p:10-100, e:on, s:off, t:1, v:L, m:sendonly;recvonly;sendrecv;inactive
A: a:G729, p:30-90, e:on, s:on, t:1, v:L, m:sendonly;recvonly;sendrecv;inactive;confrnce
```

3.3.7 Réponse AuditConnection

Dans le cas d'un AuditConnection réussi, la réponse peut être suivie par des informations sur chacun des paramètres demandés. Les paramètres pour lesquels aucune valeur n'existe actuellement peuvent quand même être fournis. Les descripteurs de connexion vont toujours apparaître en dernier et chacun va être 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 les deux descripteurs de connexion locale et distante sont fournis, le descrpteur de connexion locale va être le premier des deux. Si un descripteur de connexion est demandé, mais si il n'existe pas pour la connexion examinée, ce descripteur de connexion va apparaître avec seulement le champ Version de protocole SDP ("v=0") comme par exemple :

```
200 1203 OK

v=0
```

3.3.8 Réponse RestartInProgress

Une réponse RestartInProgress réussie peut inclure un paramètre NotifiedEntity, mais ne pas inclure par ailleurs d'autre paramètre de réponse supplémentaire.

Aussi, une réponse 521 à un RestartInProgress DOIT inclure un paramètre NotifiedEntity avec le nom d'un autre agent d'appel à contacter quand le premier agent d'appel redirige le point d'extrémité sur un autre agent d'appel comme dans :

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

3.4 Codage de la description de session (SDP)

La description de session (SDP) est codée conformément au protocole de description de session, SDP. Il est EXIGÉ des mises en œuvre de MGCP qu'elles soient pleinement capables d'analyser tout message SDP conforme, et elles DOIVENT envoyer des descriptions de session qui se conforment strictement au SDP standard.

La description générale et l'explication des paramètres SDP se trouvent dans la RFC 2327 (ou ses successeurs). En particulier, on notera que
* Origine ("o=")
* Nom de session ("s=") et
* Temps d'activité ("t=")
sont tous obligatoires dans la RFC 2327. Bien qu'ils soient de peu d'utilité pour MGCP, ils DOIVENT néanmoins être fournis conformément à la RFC 2327. On suggère ci-après les valeurs à utiliser pour chacun des champs, cependant le lecteur est invité à consulter les détails dans la RFC 2327 (ou ses successeurs) :

Origine : o = <nom d'utilisateur> <identifiant de session> <version> <type de réseau> <type d'adresse> <adresse>
* Le nom d'utilisateur DEVRAIT être réglé à trait d'union ("-").
* Il est RECOMMANDÉ que l'identifiant de session soit un horodatage NTP comme suggéré dans la RFC 2327.
* La version est un numéro de version qui DOIT être incrémenté à chaque changement du SDP. Un compteur initialisé à zéro ou un horodatage NTP comme suggéré dans la RFC 2327 est RECOMMANDÉ.
* Pour les sessions RTP, le type de réseau DEVRAIT être "IN".
* Pour les sessions RTP, le type d'adresse DEVRAIT être "IP4" (ou "IP6").
* L'adresse DEVRAIT être la même adresse que fournie dans le champ Informations de connexion ("c=").

Nom de session : s = <nom de session>
Le nom de session devrait être un trait d'union ("-").

Temps d'activité : t = <heure de début> <heure de fin>
* L'heure de début peut être réglée à zéro.
* L'heure de fin devrait être réglée à zéro.

Chacun de ces trois champs peut être ignoré à réception.

Pour s'accommoder encore plus des principes d'extensibilité de MGCP, les mises en œuvre sont encouragées à prendre en charge l'attribut PINT "a=require" - voir les détails dans la [RFC2848].

L'usage de SDP dépend en fait du type de session établie. On décrit ci-dessous l'usage de SDP pour un service audio qui utilise le profil RTP/AVP [RFC1890], ou l'interconnexion LOCAL définie dans le présent document. En cas de conflit entre ce qui est décrit ci-dessous et SDP (RFC 2327 ou ses successeur) c'est la spécification SDP qui a le pas.

3.4.1 Usage de SDP pour un service audio

Dans une passerelle de téléphonie, on a seulement à décrire des sessions qui utilisent exactement un support, l'audio. L'usage de SDP pour cela est direct et est décrit en détails dans la RFC 2327.

Voici un exemple d'une description de session conforme à la RFC 2327 pour une connexion audio :

```
v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

3.4.2 Usage de SDP pour connexions locales

Quand MGCP est utilisé pour établir des connexions internes au sein d'une seule passerelle, le format SDP est utilisé pour coder les paramètres de cette connexion. Les paramètres de connexion et de supports vont être utilisés comme suit :

* Le paramètre de connexion (c=) va spécifier que la connexion est locale, en utilisant le mot-clé "LOCAL" comme type de réseau, le mot-clé "EPN" (nom de point d'extrémité) comme type d'adresse, et le nom local du point d'extrémité comme adresse de connexion.

* Le paramètre "m=audio" va spécifier un numéro d'accès, qui va toujours être réglé à 0, le type de protocole, toujours

réglé au mot-clé LOCAL, et le type de codage, en utilisant les mêmes conventions que pour le profil RTP AVP (numéros de charge utile RTP). Le type de codage devrait normalement être réglé à 0 (PCMU).

Un attribut de niveau session qui identifie la connexion PEUT de plus être présent. Cela permet aux points d'extrémité de prendre en charge plusieurs connexions LOCAL. L'utilisation de cet attribut est FACULTATIVE et bien sûr inutile pour les points d'extrémité qui ne prennent en charge qu'une seule connexion LOCAL. L'attribut est défini comme suit :

a=MGCPlocalcx:<ConnectionID>

L'attribut de connexion locale MGCP est un attribut insensible à la casse, seulement de niveau session, qui identifie la connexion MGCP LOCAL sur le point d'extrémité identifié dans les informations de connexion auxquelles le SDP s'applique. Le ConnectionId est une chaîne hexadécimale contenant au plus 32 caractères. Le ConnectionId lui-même est insensible à la casse. L'attribut de connexion locale MGCP n'est pas soumis à l'attribut charset *(jeu de caractères)*.

Un exemple de description de session LOCALE pourrait être :

```
v=0
o=- A7453949499 0 LOCAL EPN X35V3+A4/13
s=-
c=LOCAL EPN X35V3+A4/13
t=0 0
a=MGCPlocalcx:FDE234C8
m=audio 0 LOCAL 0
```

Noter que l'attribut MGCP Connexion locale est spécifié au niveau de la session et qu'il pourrait avoir été omis dans le cas où une seule connexion LOCALE par point d'extrémité serait prise en charge.

3.5 Transmission sur UDP

Les messages MGCP sont transmis sur UDP. Les commandes sont envoyées à une des adresses IP définies dans le DNS pour le point d'extrémité spécifié. Les réponses sont renvoyées à l'adresse de source (c'est-à-dire, à l'adresse IP et au numéro d'accès UDP) des commandes - la réponse peut ou non arriver de la même adresse que celle à laquelle la commande a été envoyée.

Quand aucun accès n'est spécifié pour le point d'extrémité, les commandes DOIVENT être envoyées par défaut :
* par les agents d'appel, à l'accès MGCP par défaut pour les passerelles, 2427 ;
* par les passerelles, à l'accès MGCP par défaut pour les agents d'appel, 2727.

3.5.1 Fourniture de la fonctionnalité Au plus une fois

Les messages MGCP, étant portés sur UDP, peuvent être sujets à des pertes. En l'absence d'une réponse en temps utile, les commandes sont retransmises. La plupart des commandes MGCP ne sont pas idempotentes. L'état de la passerelle deviendrait imprévisible si, par exemple, les commandes CreateConnection étaient exécutées plusieurs fois. Les procédures de transmission DOIVENT donc fournir une fonction "au plus une fois".

Les entités MGCP sont supposées garder en mémoire une liste des réponses qu'elles ont envoyé aux transactions récentes, et une liste des transactions qui sont actuellement en cours d'exécution. La valeur numérique des identifiants de transaction des commandes entrantes est comparée aux identifiants de transaction des réponses récentes. Si une correspondance est trouvée, l'entité MGCP n'exécute pas à nouveau la transaction, mais renvoie simplement la réponse. Les commandes restantes vont être comparées à la liste des transactions en cours, c'est-à-dire, les transactions reçues précédemment qui n'ont pas encore fini d'être exécutées. Si une correspondance est trouvée, l'entité MGCP n'exécute pas à nouveau la transaction, mais une réponse provisoire (paragraphe 3.5.5) DEVRAIT être produite pour accuser réception de la commande.

La procédure utilise une longue valeur de temporisateur, notée T-HIST dans ce qui suit. Le temporisateur DOIT être réglé à plus que la durée maximum d'une transaction, et DOIT tenir compte du nombre maximum de répétitions, de la valeur maximum du temporisateur de répétition et du délai maximum de propagation d'un paquet dans le réseau. Une valeur suggérée est 30 secondes.

La copie des réponses PEUT être détruite soit T-HIST secondes après la production de la réponse, soit quand la passerelle

(ou l'agent d'appel) reçoit une confirmation que la réponse a été créée, par le "accusé de réception de réponse". Pour les transactions qui sont aquittées par cet attribut, la passerelle DEVRA garder une copie de l'identifiant de transaction (par opposition à la réponse de transaction entière) pendant T-HIST secondes après la production de la réponse, afin de détecter et ignorer les copies dupliquées de la demande de transaction qui pourraient être produites par le réseau.

3.5.2 Identifiants de transaction et prise de contact à trois phases

Les identifiants de transaction sont des nombres entiers dans la gamme de 1 à 999 999 999 (les deux inclus). Les agents d'appel peuvent décider d'utiliser un espace de nombres spécifique pour chacune des passerelles qu'ils gèrent, ou d'utiliser le même espace de nombres pour toutes les passerelles qui appartiennent à un groupe arbitraire. Les agents d'appel peuvent décider de partager la charge de la gestion d'une grande passerelle entre plusieurs processus indépendants. Ces processus DOIVENT alors partager l'espace de nombres de transaction. Il y a plusieurs mises en œuvre possibles de ce partage, comme d'avoir une allocation centralisée des identifiants de transaction, ou de pré-allouer des gammes sans chevauchement des identifiants aux différents processus. Les mises en œuvre DOIVENT garantir que des identifiants de transaction univoques sont alloués à toutes les transactions qui sont originaires d'un agent d'appel logique, comme défini à la Section 4. Les passerelles peuvent simplement détecter les transactions dupliquées en regardant seulement l'identifiant de transaction.

L'attribut Accusé de réception de réponse peut se trouver dans toute commande. Il porte un ensemble de "gammes d'identifiants de transaction confirmés" pour les réponses finales reçues - les réponses provisoires NE DOIVENT PAS être confirmées. Une réponse donnée NE DEVRAIT PAS être confirmée dans deux messages séparés.

Les entités MGCP PEUVENT choisir de supprimer les copies des réponses (mais pas l'identifiant de transaction) pour les transactions dont l'identifiant est inclus dans les "gammes d'identifiants de transaction confirmés" reçues dans les messages de confirmation de réponse (commande ou réponse). Elles DEVRAIENT ensuite éliminer en silence les autres commandes provenant de cette entité quand l'identifiant de transaction tombe dans ces gammes, et que la réponse a été produite moins de T-HIST secondes auparavant.

Les entités DOIVENT faire attention quand elles accusent réception des réponses. En particulier, une réponse DEVRAIT seulement être acquittée si l'accusé de réception de réponse est envoyé à la même entité que celle à laquelle la commande correspondante (c'est-à-dire, la commande dont la réponse est acquittée) a été envoyée.

De même, les entités NE DEVRAIENT PAS accepter aveuglément un accusé de réception de réponse pour une certaine réponse. Cependant il est considéré qu'il est sûr d'accepter un accusé de réception de réponse pour une certaine réponse, quand cet accusé de réception de réponse est envoyé par la même entité que la commande qui a généré cette réponse.

On devrait noter que l'utilisation d'accusés de réception de réponse dans les commandes (par opposition aux réponses d'accusé de réception de réponse suivant une réponse provisoire) est FACULTATIVE. L'avantage de son utilisation est que cela réduit la consommation globale de mémoire. Cependant, afin d'éviter de grands messages, les mises en œuvre NE DEVRAIENT PAS générer de grandes listes d'accusés de réception de réponse. Une stratégie est de gérer les réponses aux commandes sur la base du point d'extrémité. Une commande pour un point d'extrémité peut confirmer une réponse à une commande plus ancienne pour ce même point d'extrémité. Les réponses aux commandes avec des noms de point d'extrémité avec caractère générique peuvent être confirmées de façon sélective en considérant avec attention la taille du message, ou autrement ne simplement être pas acquittées (sauf si la réponse exige explicitement un accusé de réception de réponse). Il faut faire attention à ne pas confirmer deux fois les mêmes réponse ou une réponse qui est vieille de plus de T-HIST secondes.

Les valeurs de "gammes d'identifiants de transaction confirmés" NE DEVRONT PAS être utilisées si plus de T-HIST secondes se sont écoulées depuis que l'entité a produit sa dernière réponse à l'autre entité, ou quand une entité revient en fonctionnement. Dans cette situation, les commandes DOIVENT être acceptées et traitées, sans aucun essai sur l'identifiant de transaction.

Les commandes qui portent l'attribut "Accusé de réception de réponse" peuvent être transmises dans le désordre. L'union des "gammes d'identifiants de transaction confirmés" reçues dans les récents messages DEVRA être conservée.

3.5.3 Calcul des temporisateurs de retransmission

Il est de la responsabilité de l'entité demandeuse de fournir des temporisations convenables pour toutes les commandes en instance, et de réessayer les commandes quand les temporisations ont été dépassées. De plus, quand des commandes répétées ne sont toujours pas acquittées, il est de la responsabilité de l'entité demandeuse de chercher des services redondants et/ou de supprimer les associations existantes ou en instance.

La présente spécification évite à dessein de spécifier des valeurs pour les temporisateurs de retransmission. Ces valeurs sont normalement dépendantes du réseau. Les temporisateurs de retransmission DEVRAIENT normalement estimer le temporisateur en mesurant le temps passé entre l'envoi d'une commande et le retour de la première réponse à la commande. Au minimum, une stratégie de retransmission impliquant un retard exponentiel DOIT être mise en œuvre. Une possibilité est avec l'algorithme mis en œuvre dans TCP/IP, qui utilise deux variables :

* le delai moyen d'accusé de réception, AAD *(average acknowledgment delay)* estimé par une moyenne lissée exponentiellement des délais observés,
* l'écart type, ADEV *(average deviation)* estimé par une moyenne lissée exponentiellement de la valeur absolue de la différence entre le délai observé et la moyenne actuelle.

Le temporisateur de retransmission, RTO, dans TCP, est réglé à la somme du délai moyen plus N fois l'écart type, où N est une constante. Dans MGCP, la valeur maximum du temporisateur DEVRAIT cependant être limitée, afin de garantir qu'aucun paquet répété ne va être reçu par les passerelles après T-HIST secondes. Une valeur maximum suggérée pour RTO (RTO-MAX) est 4 secondes. Les mises en œuvre DEVRAIENT aussi envisager de limiter la valeur minimum de ce temporisateur [Allman].

Après toute retransmission, l'entité MGCP DEVRAIT faire ce qui suit :

*elle devrait doubler la valeur estimée du délai d'accusé de réception pour cette transaction, T-DELAY ;
* elle devrait calculer une valeur aléatoire, uniformément distribuée entre 0,5 T-DELAY et T-DELAY ;
* elle devrait régler le temporisateur de retransmission (RTO) au minimum de : - la somme de cette valeur aléatoire et N fois l'écart type, - RTO-MAX.

Cette procédure a deux effets. Parce qu'elle inclut un composant d'accroissement exponentiel, elle va automatiquement ralentir le flux de messages en cas d'encombrement. Parce qu'elle inclut un composant aléatoire, elle va casser la potentielle synchronisation entre les notifications déclenchées par le même événement externe.

Noter que les estimations de AAD et ADEV NE DEVRAIT PAS être mises à jour pour les transactions qui impliquent des retransmissions. Aussi, la première nouvelle transmission suivant une retransmission réussie DEVRAIT utiliser le RTO pour cette dernière retransmission. Si cette transmission réussit sans aucune retransmission, les estimations de AAD et ADEV sont mises à jour et RTO est déterminé à nouveau comme d'habitude. Voir par exemple les détails dans [TCP/IP].

3.5.4 Taille maximum de datagramme, fragmentation et ré-assemblage

Les messages MGCP transmis sur UDP s'appuient sur IP pour la fragmentation et le réassemblage des grands datagrammes. La taille maximum théorique d'un datagramme IP est de 65535 octets. Avec un en-tête IP de 20 octets et un en-tête UDP de 8 octets, cela laisse un maximum théorique de message MGCP de 65507 octets quand on utilise UDP.

Cependant, IP n'exige pas qu'un hôte reçoive des datagrammes IP de plus de 576 octets [RFC1122], ce qui donnerait une taille de message MGCP d'une petitesse inacceptable. Par conséquent, MGCP rend obligatoire que les mises en œuvre DOIVENT prendre en charge les datagrammes MGCP jusqu'à au moins 4000 octets, ce qui exige que la fragmentation et le réassemblage IP correspondants soient pris en charge. Noter que la limite de 4000 octets s'applique au niveau MGCP. Les frais généraux de couche inférieure vont exiger la prise en charge de datagrammes plus grands que cela : les frais généraux de UDP et de IP vont être d'au moins 28 octets, et, par exemple, l'utilisation de IPsec va ajouter des frais généraux supplémentaires.

On devrait noter que cela s'applique aux agents d'appel et aux points d'extrémité. Les agents d'appel peuvent examiner les points d'extrémité pour déterminer si ils prennent en charge de plus grands datagrammes MGCP que ce qui est spécifié ci-dessus. Les points d'extrémité n'ont actuellement pas de capacité similaire pour déterminer si un agent d'appel prend en charge de plus grandes tailles de datagramme MGCP.

3.5.5 Portage

Il y a des cas où un agent d'appel va vouloir envoyer plusieurs messages en même temps aux mêmes passerelles, et vice versa. Quand plusieurs messages MGCP doivent être envoyés dans le même datagramme, ils DOIVENT être séparés par une ligne de texte qui contient un seul point, comme par exemple dans :

```
200 2005 OK
.
DLCX 1244 card23/21@tgw-7.exemple.net MGCP 1.0
```

C: A3C47F21456789F0
I: FDE234C8

Les messages portés DOIVENT être traités exactement comme si ils avaient été reçus un à la fois dans plusieurs datagrammes distincts. Chaque message dans le datagramme DOIT être traité jusqu'au bout et dans l'ordre en commençant par le premier message, et chaque commande DOIT être recevoir une réponse. Les erreurs rencontrées dans un message qui était porté NE DOIVENT PAS affecter un autre message reçu dans ce datagramme - chaque message est traité de façon autonome.

Le portage peut être utilisé pour réaliser deux choses :
* la garantie de la livraison et du traitement dans l'ordre des messages,
* le partage du sort de la livraison des messages.

Quand le portage est utilisé pour garantir la livraison dans l'ordre des messages, les entités DOIVENT s'assurer que cette propriété de livraison dans l'ordre est conservée à la retransmissions des messages individuels. Un exemple en est quand plusieurs Notify sont envoyés en utilisant le portage (comme décrit au paragraphe 4.4.1).

Le partage de sort de la livraison de message assure que soit tous les messages sont livrés, soit aucun ne l'est. Quand le portage est utilisé pour garantir ce partage de sort, les entités DOIVENT aussi s'assurer que cette propriété est conservée lors des retransmissions. Par exemple, à réception d'un Notify provenant d'un point d'extrémité opérant en mode verrouillé, l'agent d'appel peut souhaiter envoyer la réponse et une nouvelle commande NotificationRequest dans un seul datagramme pour assurer le partage de sort de la livraison de message des deux.

3.5.6 Réponses provisoires

L'exécution de certaines transactions peut exiger du temps. De longs temps d'exécution peuvent interagir avec la procédure de retransmission fondée sur le temporisateur.

Il peut en résulter un nombre excessif de retransmissions, ou des valeurs de temporisateur qui deviennent trop longues pour être efficaces.

Les passerelles (et agents d'appel) qui peuvent prédire qu'une transaction va exiger un long temps d'exécution DEVRAIENT envoyer une réponse provisoire avec le code de réponse 100. À titre d'indication, une transaction qui exige une communication externe pour s'achever, par exemple, une réservation de ressource du réseau, DEVRAIT produire une réponse provisoire. De plus, les entités DEVRAIENT envoyer une réponse provisoire si elles reçoivent une répétition d'une transaction dont l'exécution n'est pas encore finie.

Les passerelles (ou agents d'appel) qui commencent à construire des files d'attentes de transactions à exécuter peuvent envoyer une réponse provisoire avec le code de réponse 101 pour l'indiquer (voir les détails au paragraphe 4.4.8).

La pure sémantique transactionnelle impliquerait que les réponses provisoires NE DEVRAIENT PAS retourner d'autre information que le fait que la transaction est en cours d'exécution, cependant une approche optimiste permettant que des informations soient retournées permet une réduction du délai qui serait autrement subi par le système.

Afin de réduire les délais dans le système, il est RECOMMANDÉ d'inclure un identifiant de connexion et une description de session dans une réponse provisoire 100 à la commande CreateConnection. Si une description de session devrait être retournée par la commande ModifyConnection, la description de session DEVRAIT aussi être incluse dans la réponse provisoire. Si la transaction s'achève avec succès, les informations retournées dans la réponse provisoire DOIVENT être répétées dans la réponse finale. Il est considéré que c'est une erreur de protocole de ne pas répéter ces informations ou de changer une des informations précédemment fournies dans une réponse de succès. Si la transaction échoue, un code d'erreur est retourné - l'information retournée précédemment n'est plus valide.

Une transaction CreateConnection ou ModifyConnection en cours d'exécution DOIT être annulée si une commande DeleteConnection est reçue pour le point d'extrémité. Dans ce cas, une réponse finale pour la transaction annulée DEVRAIT quand même être retournée automatiquement (le code d'erreur 407 - transaction interrompue, est RECOMMANDÉ) et une réponse finale pour la transaction annulée DOIT être retournée si une retransmission de la transaction annulée est détectée (voir aussi au paragraphe 4.4.4).

Les entités MGCP qui reçoivent une réponse provisoire DEVRONT passer à un plus long temporisateur de répétition (LONGTRAN-TIMER) pour cette transaction. L'objet de ce temporisateur est principalement de détecter des défaillances de traitement. La valeur par défaut de LONGTRAN-TIMER est 5 secondes, cependant le processus de provisionnement

peut changer cela. Noter que les retransmissions DOIVENT quand même satisfaire les exigences de temps spécifiées aux paragraphes 3.5.1 et 3.5.3. Par conséquent LONGTRAN-TIMER DOIT être plus petit que T-HIST (il devrait en fait être considérablement plus petit). Aussi, les entités NE DOIVENT PAS laisser une transaction tourner sans délai. Une transaction qui est arrivée en fin de temporisation pour l'entité DEVRAIT retourner le code d'erreur 406 (fin de temporisation de transactionout). Selon la définition de T-HIST (paragraphe 3.5.1) le temps maximum d'exécution d'une transaction est plus petit que T-HIST (dans un réseau à faible délai, it peut raisonnablement être approximé comme T-HIST moins T-MAX en toute sécurité) et une réponse finale devrait être reçue pas plus de T-HIST secondes après l'envoi initial de la commande. Néanmoins, les entités DEVRAIENT attendre 2*T-HIST secondes avant d'abandonner l'idée de recevoir une réponse finale. La retransmission de la commande DOIT quand même cesser après T-MAX secondes. Si une réponse n'est pas reçue, le résultat de la transaction n'est pas connu. Si l'entité qui envoie la commande est une passerelle, elle devient maintenant "déconnectée" et DEVRA initier la procédure "déconnectée" (voir au paragraphe 4.4.7).

Quand la transaction finit l'exécution, la réponse finale est envoyée et la réponse provisoire maintenant obsolète est supprimée. Afin d'assurer la détection rapide d'une réponse finale perdue, les réponses finales produites après des réponses provisoires pour une transaction DEVRAIENT être acquittées (malheureusement les mises en œuvre de la RFC 2705 ne peuvent pas le faire, ce qui est la seule raison pour que ce ne soit pas une exigence absolue).

Le point d'extrémité DEVRAIT donc inclure un paramètre "ResponseAck" vide dans les seules réponses finales. La présence du paramètre "ResponseAck" dans la réponse finale DEVRAIT déclencher une réponse "Accusé de réception de réponse" à renvoyer au point d'extrémité. La réponse "Accusé de réception de réponse" va alors inclure l'identifiant de transaction de la réponse qu'elle acquitte dans l'en-tête de réponse. Noter que, pour la rétro compatibilité, les entités ne peuvent pas dépendre de la réception d'un tel "Accusé de réception de réponse", cependant il est fortement RECOMMANDÉ de prendre en charge ce comportement, car autrement des délais excessifs dans le cas de perte de paquet ainsi que des retransmissions excessives peuvent se produire.

La réception d'une réponse "Accusé de réception de réponse" est soumise aux mêmes stratégies et procédures de temporisation et de retransmission que les réponses aux commandes, c'est-à-dire que l'envoyeur de la réponse finale va la retransmettre si un "Accusé de réception de réponse" n'est pas reçu dans les temps. Pour la rétro compatibilité, la non réception d'un "Accusé de réception de réponse" NE DEVRAIT PAS affecter les estimations de délai d'aller-retour pour les commandes suivantes, et de plus NE DOIT PAS conduire le point d'extrémité a devenir "déconnecté". La réponse "Accusé de réception de réponse" n'est jamais acquittée.

4. États, reprise sur défaillance et conditions de concurrence

Afin de mettre en œuvre une signalisation d'appel appropriée, l'agent d'appel doit garder trace de l'état du point d'extrémité, et la passerelle doit s'assurer que les événements sont proprement notifiés à l'agent d'appel. Il existe des conditions particulières quand la passerelle ou l'agent d'appel sont redémarrés : la passerelle doit être redirigée sur un nouvel agent d'appel durant les procédures de "reprise sur défaillance", l'agent d'appel doit entreprendre une action particulière quand la passerelle est mise hors ligne, ou redémarrée.

4.1 Hypothèses de reprise sur défaillance et points à souligner

Les faits marquants suivants du protocole sont importants pour comprendre les mécanismes de reprise sur défaillance de l'agent d'appel :

* Les agents d'appel sont identifiés par leur nom de domaine (et l'accès facultatif) et non par leurs adresses réseau, et plusieurs adresses peut être associées à un nom de domaine.

* Un point d'extrémité a un et un seul agent d'appel associé à tout moment. L'agent d'appel associé à un point d'extrémité est la valeur courante de "l'entité notifiée". "L'entité notifiée" détermine où la passerelle va envoyer ses commandes. Si "l'entité notifiée" n'inclut pas de numéro d'accès, le numéro d'accès par défaut d'agent d'appel (2727) est supposé.

* NotifiedEntity est un paramètre envoyé par l'agent d'appel à la passerelle pour établir "l'entité notifiée" pour le point d'extrémité.

* "L'entité notifiée" pour un point d'extrémité est la dernière valeur de paramètre NotifiedEntity reçue pour ce point d'extrémité. Si aucun paramètre explicite NotifiedEntity n'a jamais été reçu, "l'entité notifiée" va par défaut être à une valeur provisionnée. Si aucune valeur n'a été provisionnée ou si un paramètre NotifiedEntity vide a été fourni (les deux

sont fortement déconseillés) rendant ainsi "l'entité notifiée" vide, "l'entité notifiée" est réglée à l'adresse de source de la dernière commande non d'audit pour le point d'extrémité. Donc l'audit ne va pas changer "l'entité notifiée".

* Les réponses aux commandes sont envoyées à l'adresse de source de la commande, sans considération de "l'entité notifiée" en cours. Quand un message Notify a besoin d'être porté avec la réponse, le datagramme est envoyé à l'adresse de source de la nouvelle commande reçue, sans considération de "l'entité notifiée" en cours.

La capacité de "l'entité notifiée" de se résoudre en plusieurs adresses réseau permet à une "entité notifiée" de représenter un agent d'appel avec plusieurs interfaces physiques et/ou un agent d'appel logique constitué de plusieurs systèmes physiques. L'ordre des adresses réseau quand un nom DNS se résout en plusieurs adresses est non déterministe de sorte que les schémas de reprise sur défaillance de l'agent d'appel NE DOIVENT PAS dépendre d'un ordre quelconque (par exemple, une passerelle DOIT être capable d'envoyer un "Notify" à toute adresse réseau résolue). Par ailleurs, le système va probablement être plus efficace si la passerelle envoie les commandes à l'interface sur laquelle elle a déjà une association en cours. Il est RECOMMANDÉ que les passerelles utilisent l'algorithme suivant pour atteindre ce but :

* Si "l'entité notifiée" se résout en plusieurs adresses réseau, et si l'adresse de source de la demande est une de ces adresses, l'adresse réseau est l'adresse de destination préférée pour les commandes.
* Si par ailleurs, l'adresse de source de la demande n'est pas une des adresses résolues, la passerelle doit choisir une des adresses résolues pour les commandes.
* Si la passerelle échoue à contacter l'adresse réseau choisie, elle DOIT essayer les solutions de remplacement dans la liste des adresses résolues comme décrit au paragraphe 4.3.

Si un agent d'appel entier devient indisponible, les points d'extrémité gérés par cet agent d'appel vont éventuellement devenir "déconnectés". La seule façon pour que ces points d'extrémité deviennent à nouveau connectés est soit que l'agent d'appel défaillant redevienne disponible, soit qu'un agent d'appel de secours contacte les points d'extrémité affectés avec une nouvelle "entité notifiée".

Quand un agent d'appel de secours a pris le contrôle d'un groupe de points d'extrémité, il est supposé que l'agent d'appel défaillant communique et se synchronise avec l'agent d'appel de secours afin de retransférer le contrôle des points d'extrémité affectés à l'agent d'appel d'origine. Autrement, l'agent d'appel défaillant pourrait simplement devenir l'agent d'appel de secours.

On devrait noter que la résolution de conflit de transfert inter-cellulaire entre des CA séparées n'est pas en place - on s'appuie strictement sur l'idée que les CA savent ce qu'elles font et communiquent ensemble (bien que AuditEndpoint puisse être utilisé pour apprendre quelle est "l'entité notifiée" en cours). Si ce n'est pas le cas, un comportement inattendu peut se produire.

Noter que comme mentionné précédemment, "l'entité notifiée" par défaut est provisionnée et peut inclure le nom de domaine et l'accès. Pour les petites passerelles, le provisionnement peut être fait sur la base du point d'extrémité. Pour les plus grosses passerelles, un seul élément de provisionnement peut être fourni pour plusieurs points d'extrémité ou même pour la passerelle entière. Dans les deux cas, une fois la passerelle sous tension, chaque point d'extrémité DOIT avoir sa propre "entité notifiée", de sorte que les valeurs provisionnées pour une agrégation de points d'extrémité DOIVENT être copiées à "l'entité notifiée" pour chaque point d'extrémité dans l'agrégation avant que le fonctionnement se poursuive. Lorsque possible, la commande RestartInProgress au redémarrage DEVRAIT être envoyée à "l'entité notifiée" provisionnée sur la base d'une agrégation qui permet d'utiliser le caractère générique "all of". Cela va réduire le nombre de messages RestartInProgress.

Une autre façon de voir l'utilisation de "l'entité notifiée" est en termes d'associations entre passerelles et agents d'appel. Une "entité notifiée" est un moyen pour établir cette association, et gouverne où la passerelle va envoyer les commandes. Les commandes reçues par la passerelle peuvent cependant venir de toute source. L'association est initialement provisionnée avec une "entité notifiée" provisionnée, de sorte qu'à la mise sous tension, RestartInProgress et les d'événements persistants qui se produisent avant la première NotificationRequest provenant d'agents d'appel vont être envoyés à l'agent d'appel provisionné. Une fois qu'un agent d'appel fait une demande, il peut cependant inclure le paramètre NotifiedEntity et établir une nouvelle association. Comme "l'entité notifiée" persiste à travers les appels, l'association reste intacte jusqu'à ce qu'une nouvelle "entité notifiée" soit fournie.

4.2 Communication avec les passerelles

Le nom de point d'extrémité dans les passerelles inclut un nom local indiquant le point d'extrémité spécifique et un nom de domaine indiquant l'hôte/passerelle où le point d'extrémité réside. Les passerelles peuvent avoir plusieurs interfaces pour la redondance.

Dans les passerelles qui ont une capacité d'acheminement, le nom de domaine peut se résoudre en une seule adresse réseau avec un acheminement interne à cette adresse à partir de toute interface de la passerelle. Dans d'autres, le nom de domaine peut se résoudre en plusieurs adresses réseau, une pour chaque interface. Dans ce cas, si un agent d'appel échoue à contacter la passerelle sur une des adresses, il DOIT essayer les autres.

4.3 Retransmission, et détection des associations perdues

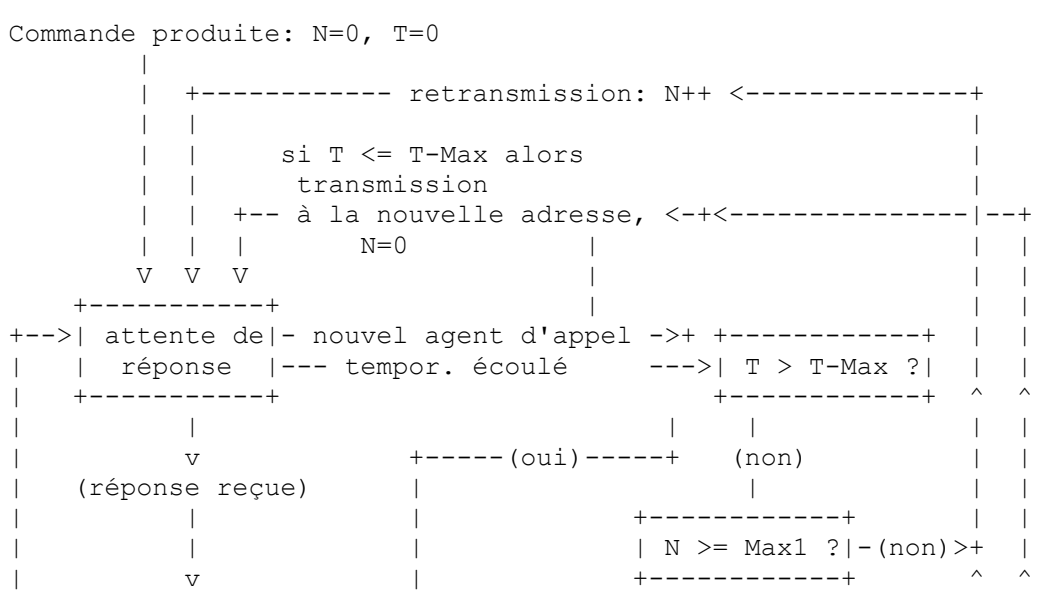
Le protocole de contrôle de passerelle de supports est organisé comme un ensemble de transactions, dont chacune est composée d'une commande et d'une réponse, couramment appelée un accusé de réception. Les messages MGCP, étant portés sur UDP, peuvent être l'objet de pertes. En l'absence d'une réponse en temps utile, les commandes sont retransmises. Les entités MGCP DOIVENT garder en mémoire une liste des réponses envoyées aux transactions récentes, c'est-à-dire, une liste de toutes les réponses qu'elles ont envoyé sur les dernières T-HIST secondes, et une liste des transactions qui n'ont pas encore fini leur exécution.

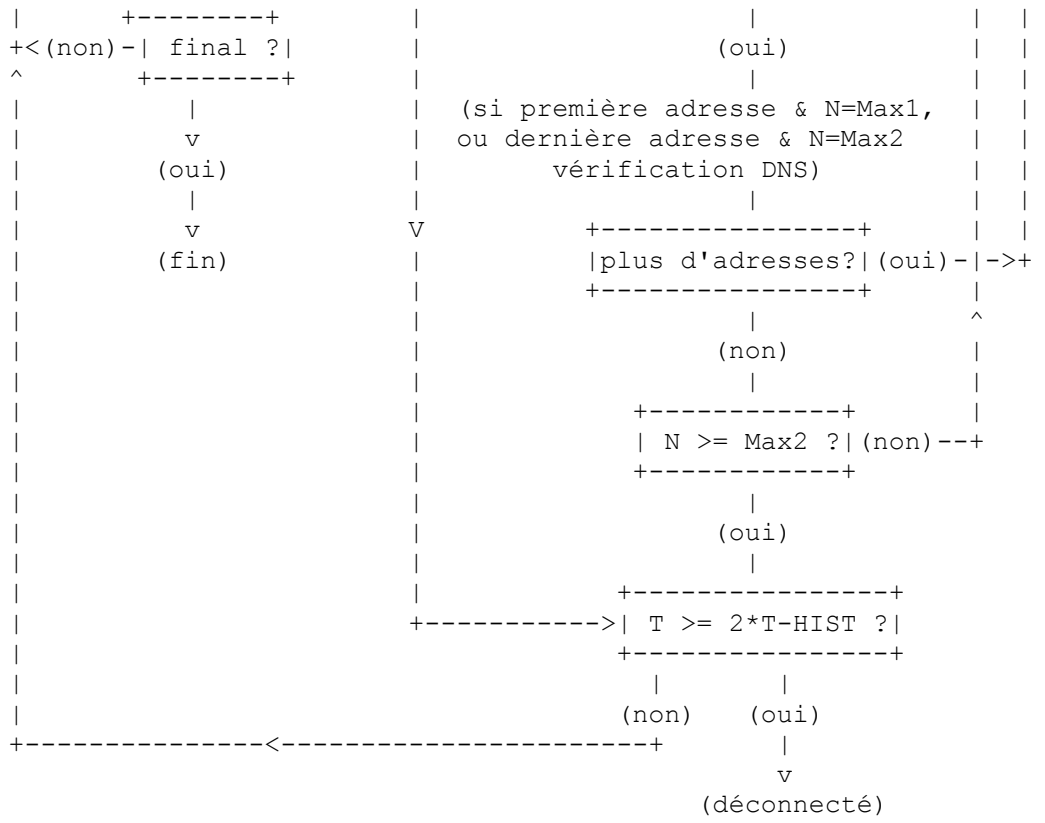
Les identifiants de transaction des commandes entrantes sont comparés aux identifiants de transaction des réponses récentes. Si une correspondance est trouvée, l'entité MGCP n'exécute pas la transaction, mais répète simplement la réponse. Si une correspondance à une transaction à une réponse précédente n'est pas trouvée, l'identifiant de transaction de la commande entrante est comparé à la liste des transactions dont l'exécution n'est pas encore finie. Si une correspondance est trouvée, l'entité MGCP n'exécute pas à nouveau la transaction, mais DEVRAIT simplement envoyer une réponse provisoire - une réponse finale sera fournie quand l'exécution de la commande sera achevée (voir les détails au paragraphe 3.5.6).

Le mécanisme de répétition est utilisé pour se garder contre quatre types d'erreurs possibles :
* erreurs de transmission, quand par exemple un paquet est perdu à cause du bruit sur une ligne ou de l'encombrement dans une file d'attente,
* défaillance d'un composant, quand par exemple une interface à un agent d'appel devient indisponible,
* défaillance de l'agent d'appel, quand par exemple un agent d'appel entier devient indisponible,
* défaillance, quand un nouvel agent d'appel "prend le contrôle" de façon transparente.

Les éléments devraient être capables de déduire de l'histotrique une estimation du taux de pertes de paquets dû à des erreurs de transmission. Dans un système correctement configuré, ce taux de pertes devrait être très bas, normalement, moins de 1 %. Si un agent d'appel ou une passerelle doit répéter un message plus de quelques fois, il est légitime de supposer que quelque chose d'autre qu'une erreur de transmission se produit. Par exemple, avec un taux de pertes de 1 %, la probabilité que 5 tentatives consécutives de transmission échouent est de 1 sur 100 milliards, un événement qui devrait se produire moins de une fois tous les 10 jours pour un agent d'appel qui traite 1000 transactions par seconde. (Bien sûr, le nombre de retransmissions qui est considéré comme excessif devrait être une fonction du taux de perte de paquets dominant.) On devrait noter que le "seuil de suspicion", qu'on appellera "Max1", est normalement inférieur au "seuil de déconnexion", qu'on appellera "Max2". Max2 DOIT être réglé à une valeur supérieure à Max1.

L'algorithme de retransmission MGCP est illustré dans la Figure ci-dessous et expliqué ensuite :





Un algorithme de retransmission classique compterait simplement le nombre de répétitions successives, et en conclurait que l'association est rompue après la retransmission du 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 :

* On exige que la passerelle vérifie toujours la présence d'un nouvel agent d'appel. Il peut être remarqué soit :
  - en recevant une commande où la NotifiedEntity pointe sur le nouvel agent d'appel, soit
  - en recevant une réponse de redirection pointant sur un nouvel agent d'appel.

  Si un nouvel agent d'appel est détecté, la passerelle DOIT commencer à retransmettre les commandes en instance pour le ou les points d'extrémité redirigés sur ce nouvel agent d'appel. Les réponses aux nouvelles ou aux anciennes commandes sont encore transmises à l'adresse de source de la commande.

* Avant toute retransmission, on vérifie que le temps écoulé depuis l'envoi du datagramme initial n'est pas supérieur à T-MAX. Si plus de T-MAX s'est écoulé, les retransmissions DOIVENT alors cesser. Si plus de 2*T-HIST s'est écoulé, alors le point d'extrémité devient déconnecté.

* Si le nombre de répétitions pour cet agent d'appel est égal à "Max1", et si son nom de domaine n'a pas été résolu récemment (par exemple, dans les 5 dernières secondes ou autre disposition) et si il n'est pas en cours de résolution, alors la passerelle PEUT activement interroger le serveur de noms de domaine afin de détecter le possible changement des intefaces de l'agent d'appel. Noter que la première répétition est la seconde transmission.

* La passerelle peut avoir appris plusieurs adresses IP pour l'agent d'appel. Si le nombre de répétitions pour cette adresse IP est supérieur ou égal à "Max1" et inférieur à "Max2", et si il y a plus d'adresses qui n'ont pas été essayées, la passerelle DOIT alors diriger les retransmissions sur des adresses de remplacement. Aussi, la réception de notifications explicites du réseau comme, par exemple, réseau ICMP, hôte, protocole, ou accès injoignable DEVRAIT conduire la passerelle à essayer des adresses de remplacement (en examinant avec soin les possibles problèmes de sécurité).

* Si il n'y a plus d'autre interface à essayer, et si le nombre de répétitions pour cette adresse est Max2, la passerelle DEVRAIT alors contacter une fois de plus le DNS pour voir si d'autres interfaces sont devenues disponibles, sauf si le nom de domaine a été résolu récemment (par exemple, dans les 5 dernières secondes ou autre disposition) ou si il est déjà dans le processus de résolution. Si il n'y a encore plus d'interface à essayer, la passerelle est alors déconnectée et DOIT initier la procédure "déconnectée" (voir au paragraphe 4.4.7).

Afiin d'adapter automatiquement la charge du réseau, MGCP spécifie des temporisateurs à accroissement exponentiel. Si le temporisateur initial est réglé à 200 millisecondes, la perte d'une cinquième retransmission va être détectée après environ 6 secondes. C'est probablement un délai d'attente acceptable pour détecter une reprise sur défaillance. Les répétitions devraient continuer après ce délai non seulement afin de peut-être surmonter un problème de connexité temporaire, mais aussi afin de donner un peu plus de temps pour l'exécution d'une reprise sur défaillance - attendre un délai total de 30 s est probablement acceptable.

Il est cependant important que le délai maximum de retransmissions soit limité. Avant toute retransmission, on vérifie que le temps écoulé (T) depuis l'envoi du datagramme initial n'est pas supérieur à T-MAX. Si plus de T-MAX s'est écoulé, les retransmissions DOIVENT cesser. Si plus de 2*T-HIST s'est écoulé, le point d'extrémité devient déconnecté. La valeur T-MAX est en relation avec la valeur T-HIST : la valeur T-HIST DOIT être supérieure ou égale à T-MAX plus le délai maximum de propagation dans le réseau.

La valeur par défaut pour T-MAX est 20 secondes. Donc, si le délai maximum de propagation supposé est de 10 secondes, les réponses aux vieilles transactions vont devoir être conservées pendant une période d'au moins 30 secondes. L'importance d'avoir un accord de l'envoyeur et du receveur sur ces valeurs ne peut pas être surestimée.

La valeur par défaut pour Max1 est 5 retransmissions et la valeur par défaut pour Max2 est 7 retransmissions. Ces deux valeurs peuvent être changées par le processus de provisionnement.

Le processus de provisionnement DOIT être capable de désactiver une ou les deux interrogations DNS de Max1 et Max2.

4.4 Conditions de concurrence

MGCP traite les conditions de concurrence par la notion d'une "liste de quarantaine" et par la détection explicite de la désynchronisation, par exemple, pour un état de crochet discordant dû à une double prise pour un point d'extrémité.

MGCP ne suppose pas que le mécanisme de transport va conserver l'ordre des commandes et réponses. Ceci peut causer des conditions de concurrence, qui peuvent être atténuées par un comportement approprié de l'agent d'appel. (Noter que certaines conditions de concurrence sont inhérentes aux systèmes répartis ; elles vont quand même se produire, même si les commandes ont été transmises dans un ordre strict.)

Dans certains cas, de nombreuses passerelles peuvent décider de redémarrer le fonctionnement en même temps. Ceci peut arriver, par exemple, si une zone perd l'alimentation ou sa capacité de transmission durant un tremblemnent de terre ou une tempête de neige. Quand l'alimentation et la transmission sont rétablies, de nombreuses passerelles peuvent décider d'envoyer simultanément des commandes "RestartInProgress", conduisant à un fonctionnement très instable.

4.4.1 Liste de quarantaine

Les passerelles contrôlées par MGCP vont recevoir des "demandes de notification" qui leur demandent de surveiller une liste des "événements". Les éléments de protocole qui déterminent le traitement de ces événements sont la liste "Événements demandés", le "script de numérotation", le "traitement de quarantaine", et la liste "Événements détectés".

Quand le point d'extrémité est initialisé, la liste des événements demandés consiste seulement en événements persistants pour le point d'extrémité, et le script de numérotation est supposé vide. À ce point, le point d'extrémité PEUT utiliser une NotificationRequest implicite avec l'identifiant de demande réservé de zéro ("0") pour détecter et rapporter un événement persistant, par exemple, décroché. Une condition de décroché pré-existante DOIT ici résulter en la génération aussi de l'événement "décroché".

Le point d'extrémité atttend la réception d'une commande NotificationRequest, après quoi la passerelle commence à observer le point d'extrémité pour des occurrences des événements mentionnés dans la liste, incluant des événements persistants.

Les événements sont examinés comme ils arrivent. L'action qui suit est déterminée par le paramètre "action" associé à l'événement dans la liste des événements demandés, et aussi par le script de numérotation. Les événements qui sont définis comme "accumuler" ou "accumuler selon le script de numérotation" sont accumulés dans une liste d'événements, les événements qui sont marqués comme "accumuler selon le script de numérotation" vont de plus être accumulés dans la "chaîne de numérotation en cours". Cela va se poursuivre jusqu'à ce que un événement soit rencontré qui déclenche une notification qui va être envoyée à "l'entité notifiée" en cours.

La passerelle, à ce point, va transmettre la commande Notify et va placer le point d'extrémité dans un état "Notification". Tant que le point d'extrémité est dans cet état Notification, les événements qui sont à détecter sur le point d'extrémité sont mémorisés dans une mémoire tampon de "quarantaine" (FIFO, premier entré premier sorti) pour traitement ultérieur. Les événements sont, en un sens, "en quarantaine". Tous les événements qui sont spécifiés par l'union du paramètre RequestedEvents et le plus récemment reçu paramètre DetectEvents ou, en l'absence de ce dernier, tous les événements qui sont mentionnés dans les RequestedEvents, DEVRONT être détectés et mis en quarantaine, sans considération de l'action associée à l'événement. Les événements persistants sont vus ici comme implicitement inclus dans RequestedEvents. Si la mémoire tampon de quarantaine atteint la limite de capacité du point d'extrémité, un événement "Débordement de mémoire tampon de quarantaine" (voir l'Appendice B) DEVRAIT être généré (quand cet événement est pris en charge, le point d'extrémité DOIT s'assurer qu'il a la capacité d'inclure l'événement dans la mémoire tampon de quarantaine). Les événements en excédent vont alors être éliminés.

Le point d'extrémité sort de l'état "Notification" quand est reçue la réponse (de succès ou d'échec) à la commande Notify. La commande Notify peut être retransmise dans l'état "Notification", comme spécifié au paragraphe 3.5 et à la Section 4. Si le point d'extrémité est ou devient déconnecté (voir au paragraphe 4.3) durant cela, une réponse à la commande Notify ne va jamais être reçue. La commande Notify est alors perdue et donc n'est plus considérée comme en instance, alors que le point d'extrémité est encore dans l'état "Notification". Si cela devait se produire, l'achèvement de la procédure de déconnexion spécifiée au paragraphe 4.4.7 DEVRA alors conduire le point d'extrémité à sortir de l'état "Notification".

Quand le point d'extrémité sort de l'état "Notification", il réinitialise la liste des événements observés et la "chaîne de numérotation en cours" du point d'extrémité à une valeur nulle.

Après cela, le comportement de la passerelle dépend de la valeur du paramètre QuarantineHandling dans la commande de déclenchement NotificationRequest :

Si l'agent d'appel avait spécifié qu'il attendait au plus une notification en réponse à la commande Demande de notification, alors la passerelle DEVRA simplement garder les événements accumulés dans la mémoire tampon de quarantaine jusqu'à ce qu'elle reçoive la prochaine commande Demande de notification.

Si, cependant, la passerelle est autorisée à envoyer plusieurs commandes Notify successives, il va procéder comme suit. Quand la passerelle sort de l'état "Notification", elle réinitialise la liste des événements observés et la "chaîne de numérotation en cours" du point d'extrémité à une valeur nulle et commence le traitement de la liste des événements en quarantaine, en utilisant la liste déjà reçue des événements demandés et le script de numérotation. Quand elle traite ces événements, la passerelle peut rencontrer un événement qui déclenche l'envoi d'une commande Notify. Si c'est le cas, la passerelle peut adopter un des deux comportements suivants :

* elle peut immédiatement transmettre une commande Notify qui va rapporter tous les événements qui ont été accumulés dans la liste des événements observés jusqu'à l'événement déclencheur inclus, laissant les événements non traités dans la mémoire tampon de quarantaine,
* ou elle peut tenter de vider la mémoire tampon de quarantaine et transmettre une seule commande Notify rapportant plusieurs ensembles d'événements (dans une seule liste d'événements observés) et éventuellement plusieurs chaînes de numérotation. La "chaîne de numérotation en cours" est remise à une valeur nulle après chaque événement déclencheur. Les événements qui suivent le dernier événement déclencheur sont laissés dans la mémoire tampon de quarantaine.

Si la passerelle transmet une commande Notify, le point d'extrémité va rentrer et rester dans l'état "Notification" jusqu'à la réception de l'accusé de réception (comme décrit ci-dessus). Si la passerelle ne trouve pas un événement en quarantaine qui déclenche une commande Notify, il place le point d'extrémité dans un état normal. Les événements sont ensuite traités comme ils viennent, exactement de la même façon que si une commande NotificationRequest venait d'être reçue.

Une passerelle peut recevoir à tout moment une nouvelle commande NotificationRequest pour le point d'extrémité, incluant le cas où le point d'extrémité est déconnecté. Activer une demande de notification incorporée est vu ici comme recevoir aussi une nouvelle demande de notification, sauf que la liste courante des ObservedEvents reste inchangée plutôt que d'être traitée à nouveau. Quand une nouvelle demande de notification est reçue dans l'état Notification, la passerelle DEVRA s'assurer que le Notify en instance est reçu par l'agent d'appel avant un nouveau Notify (noter qu'un Notify perdu à cause d'une déconnexion n'est plus considéré comme en instance). Elle fait cela en utilisant la fonction de "portage" du protocole. Les messages vont alors être envoyés dans un seul paquet à "l'entité notifiée" en cours. Les étapes impliquées sont les suivantes :

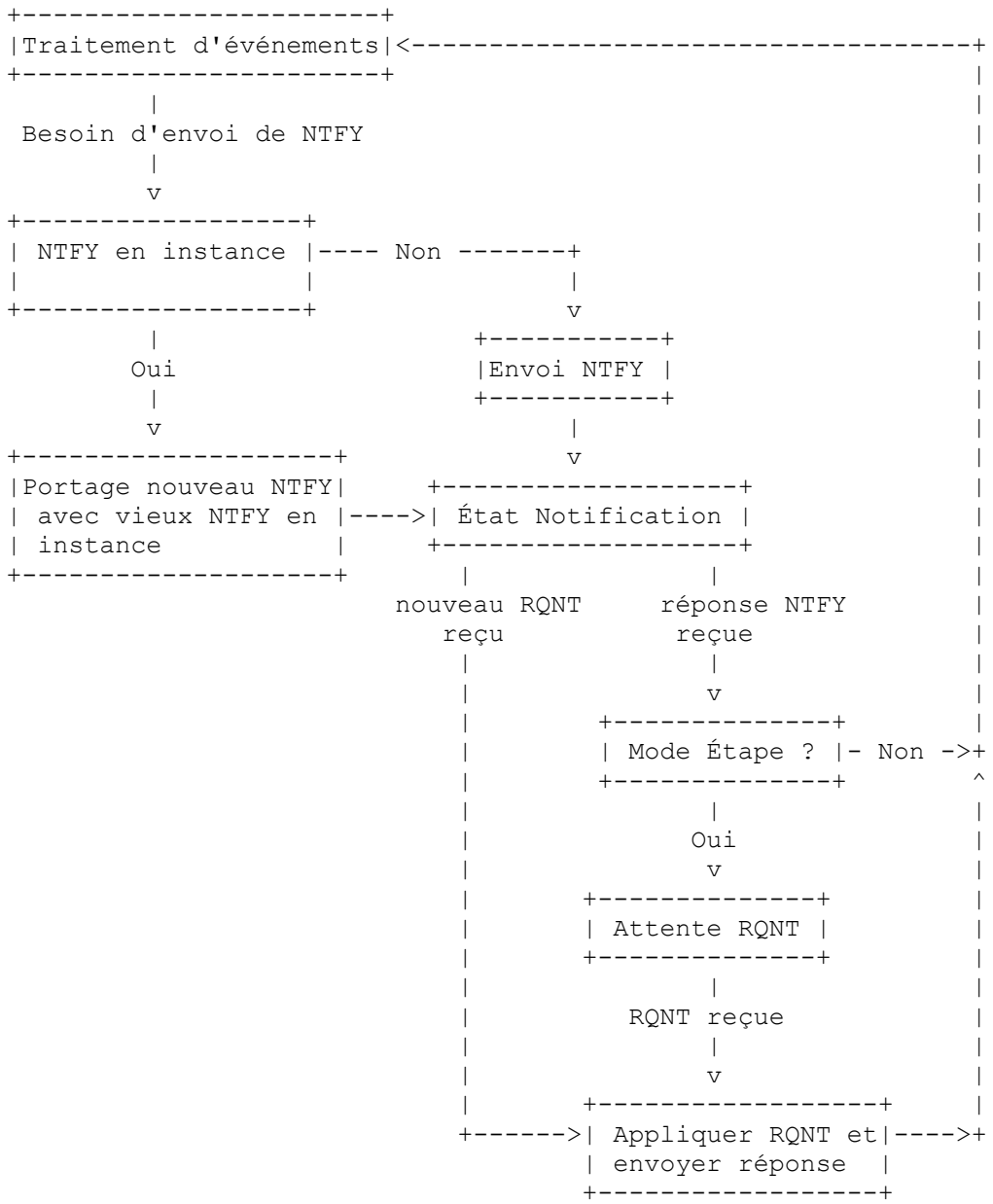
a) La passerelle envoie une réponse à la nouvelle demande de notification.
b) Le point d'extrémité est alors sorti de l'état "Notification" sans attendre l'accusé de réception de la commande Notify en instance.
c) Une copie de la commande Notify non acquittée est conservée jusqu'à ce qu'un accusé de réception soit reçu. Si un

temporisateur arrive à expiration, le Notify va être retransmis.

d) Si la passerelle a un nouveau Notify à transmettre avant que le ou les précédents Notify soient acquittés, elle construit un paquet qui porte une répétition du ou des vieux Notify et le nouveau Notify (ordonnés par age avec le plus ancien en premier). Ce datagramme va être envoyé à "l'entité notifiée" en cours.

f) Les passerelles qui ne peuvent pas porter plusieurs messages dans le même datagramme et donc garantir une livraison dans l'ordre des deux (ou plus) Notify DEVRONT laisser le point d'extrémité dans l'état "Notification" tant que le dernier Notify n'est pas acquitté.

La procédure est illustrée par le diagramme suivant :



Les passerelles peuvent aussi tenter de livrer le Notify en instance avant une réponse de succès à la nouvelle NotificationRequest en utilisant la fonction de "portage" du protocole. C'était en fait le comportement exigé dans la RFC 2705, cependant il y a plusieurs complications à faire cela, et les avantages sont discutables. En particulier, le mécanisme de la RFC 2705 ne garantissait pas la livraison dans l'ordre des Notify et des réponses aux NotificationRequest en général, et donc les agents d'appel avaient de toutes façons à traiter la livraison dans le désordre de ces messages. Le changement au statut facultatif est donc rétro compatible tout en réduisant grandement la complexité.

Après la réception de la commande Demande de notification, la liste des événements demandés et le script de numérotation (si un nouveau a été fourni) sont remplacés par les nouveaux paramètres reçus, et la chaîne de numérotation en cours est

réinitialisée à une valeur nulle. De plus, quand la demande de notification a été reçue dans l'état "Notification", la liste des événements observés est remise à une valeur nulle. Le comportement suivant est conditionné par la valeur du paramètre QuarantineHandling. Le paramètre peut spécifier que les événements en quarantaine (et les événements observés qui dans ce cas sont maintenant une liste vide) devraient être éliminés, ce que dans ce cas ils vont être. Si le paramètre spécifie que les événements en quarantaine (et observés) sont à traiter, la passerelle va commencer à traiter la liste des événements en quarantaine (et observés) en utilisant la nouvelle liste reçue des événements demandés et le script de numérotation (si il est fourni). Quand elle traite ces événements, la passerelle peut rencontrer un événement qui exige l'envoi d'une commande Notify. Si tel est le cas, la passerelle va immédiatement transmettre une commande Notify qui va rapporter tous les événements qui ont été accumulés dans la liste des événements observés jusqu'à l'événement déclencheur, en laissant les événements non traités dans la mémoire tampon de quarantaine, et va entrer dans l'état "Notification".

Une nouvelle demande de notification peut être reçue alors que la passerelle a accumulé des événements conformément à la précédente demande de notification, mais n'a pas encore détecté un événement déclencheur de notification, c'est-à-dire, le point d'extrémité n'est pas dans l'état "Notification". Le traitement des événements pas encore notifiés est déterminé, comme avec les événements en quarantaine, par le paramètre Traitement de quarantaine :

* si il spécifie que les événements en quarantaine devront être ignorés, la liste des événements observés est simplement réinitialisée ;
* si il spécifie que les événements en quarantaine devront être traités, la liste des événements observés est transférée à la liste des événements en quarantaine. La liste des événements observés est alors réinitialisée, et la liste des événements en quarantaine est traitée.

Les agents d'appel qui contrôlent des points d'extrémité en mode verrouillage DEVRAIENT fournir la réponse à un message Notify réussi et la nouvelle NotificationRequest dans le même datagramme en utilisant le mécanisme de portage.

4.4.2 Détection explicite

Un élément clé de l'état de plusieurs points d'extrémité est la position du crochet. Une condition de concurrence peut se produire quand l'utilisateur décide de décrocher avant que l'agent d'appel ait le temps de demander à la passerelle de notifier un événement de décroché (la condition de "double prise" bien connue en téléphonie) ou si l'utilisateur raccroche avant que l'agent d'appel ait le temps de demander la notification de l'événement.

Pour éviter cette condition de concurrence, la passerelle DOIT vérifier la condition du point d'extrémité avant d'accuser réception d'une NotificationRequest. Elle DOIT retourner une erreur :

1. si il est demandé à la passerelle de notifier une transition de "décroché" alors que le téléphone est déjà décroché, (code d'erreur 401 - téléphone décroché)
2. si il est demandé à la passerelle de notifier une condition de "raccroché" ou "impulsion crochet" alors que le téléphone est déjà raccroché (code d'erreur 402 - téléphone raccroché).

De plus, des définitions de signal individuelles peuvent spécifier qu'un signal ne va fonctionner que sous certaines conditions, par exemple, la sonnerie peut n'être possible que si le téléphone est déjà décroché. Si de tels pré-requis existent pour un signal donné, la passerelle DOIT retourner l'erreur spécifiée dans la définition du signal si le pré-requis n'est pas satisfait.

On devrait noter que la vérification de condition est effectuée au moment de la réception de la demande de notification, tandis que l'événement réel qui a causé la condition actuelle peut avoir été rapporté, ou ignoré précédemment, ou il peut être actuellement en quarantaine.

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

Quand une NotificationRequest échoue, qu'elle soit incluse dans une commande de traitement de connexion ou non, la passerelle DOIT simplement continuer comme si la commande n'avait jamais été reçue. Comme toutes les autres transactions, la NotificationRequest DOIT opérer comme une transaction atomique, donc tout changement initié par suite de la commande DOIT être inversé.

Une autre condition de concurrence peut survenir quand un Notify est produit peu avant la réception par la passerelle d'une NotificationRequest. L'identifiant de demande est utilisé pour corréler les 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 que la passerelle a reçu la nouvelle NotificationRequest. Ceci est particulièrement important pour éviter des impasses en mode "étape".

4.4.3 Sémantique des transactions

Lorsque le temps d'achèvement de transaction potentiel augmente, par exemple, dû à des réservations de ressources externes, une définition précise de la sémantique transactionnelle devient de plus en plus importante. En particulier, le problème des conditions de concurrence, par exemple, comme il se rapporte à l'état du crochet, doit être défini précisément.

Un point important à considérer est que l'état d'une pré-condition (par exemple, l'état du crochet) peut en fait changer entre le moment du début d'une transaction et celui où elle s'achève avec succès (transaction engagée) ou échoue. En général, on peut dire que le succès de l'exécution d'une transaction dépend de une ou plusieurs pré-conditions où l'état d'une ou plusieurs des pré-conditions peut changer dynamiquement entre le début de la transaction et l'engagement de la transaction.

La plus simple sémantique pour cela est simplement d'exiger que toutes les pré-conditions soient satisfaites depuis le moment où la transaction est initiée jusqu'à l'engagement de la transaction. Si une des pré-conditions n'est pas satisfaite avant l'achèvement de la transaction, la transaction va aussi échouer.

Par exemple, considérons une transaction qui inclut une demande de l'événement "décroché". Quand la transaction est initiée, le téléphone est "raccroché" et cette pré-condition est donc satisfaite. Si l'état du crochet change à "décroché" avant que la transaction s'achève, la pré-condition n'est plus satisfaite, et donc la transaction échoue immédiatement.

Finalement, on doit considérer le moment où une nouvelle transaction prend effet et où le traitement par le point d'extrémité selon une ancienne transaction s'arrête. Par exemple, supposons que la transaction T1 a été exécutée avec succès et que le traitement d'événement est actuellement fait selon la transaction T1. On reçoit maintenant une nouvelle transaction T2 qui spécifie un nouveau traitement d'événement (par exemple, un CreateConnection avec une demande de notification encapsulée). Comme on ne sait pas si T2 va s'achever avec succès ou non, on ne peut pas commencer à traiter les événements selon T2 tant que le résultat de T2 n'est pas connu. Alors qu'on pourrait suspendre tout le traitement d'événement jusqu'à ce que le résultat de T2 soit connu, cela pourrait aller pour un système moins performant et donc NE DEVRAIT PAS être fait. Quand une nouvelle transaction Ty est reçue et que Ty modifie le traitement en accord avec une ancienne transaction Tx, le traitement selon Tx DEVRAIT plutôt rester actif aussi longtemps que possible, jusqu'à ce qu'il soit connu qu'un bon résultat de Ty s'est produit. Si Ty échoue, le traitement selon Tx va bien sûr se poursuivre normalement. Tous les changements induits par Ty prennent logiquement effet quand Ty s'engage. Donc, si le point d'extrémité était dans l'état Notification quand Ty s'engage, et si Ty contenait une NotificationRequest, le point d'extrémité va être sorti de l'état Notification quand Ty s'engage. Noter que ceci est indépendant de si le point d'extrémité était dans l'état Notification quand Ty a été initié. Par exemple, un Notify pourrait être généré à cause du traitement selon Tx entre le début et l'engagement de Ty. Si l'engagement de Ty conduit le point d'extrémité à entrer dans l'état Notification, une nouvelle NotificationRequest (Tz) est nécessaire pour sortir de l'état Notification. Cela découle du fait que l'exécution de la transaction respecte l'ordre causal.

Un autre problème en rapport est l'utilisation de caractères génénriques, en particulier "all of", qui peut correspondre à plus d'un point d'extrémité. Quand une commande est demandée, et que l'identifiant de point d'extrémité correspond à plus d'un point d'extrémité, la sémantique de transaction s'applique encore. Donc, la commande DOIT soit réussir pour tous les points d'extrémité, soit DOIT échouer pour tous. Une seule réponse est par conséquent toujours produite.

4.4.4 Ordre des commandes, et traitement des désordres

MGCP ne rend pas obligatoire que le protocole de transport sous-jacent garantisse une livraison dans l'ordre des commandes à une passerelle ou un point d'extrémité. Cette propriété tend à maximiser l'à-propos des actions, mais elle a des inconvénients. Par exemple :

* les commandes Notify peuvent être retardées et arriver à l'agent d'appel après la transmission d'une nouvelle commande Demande de notification,
* si une nouvelle NotificationRequest est transmise avant qu'une précédente soit acquittée, il n'est pas garanti que la précédente va être reçue et exécutée après la nouvelle.

Les agents d'appel qui veulent garantir un fonctionnement cohérent des points d'extrémité peuvent utiliser les règles suivantes :

1) Quand une passerelle traite plusieurs points d'extrémité, les commandes relevant des différents points d'extrémité peuvent être envoyées en parallèle, par exemple, en suivant un modèle où chaque point d'extrémité est contrôlé par son propre processus ou son propre fil.
2) Quand plusieurs connexions sont créées sur le même point d'extrémité, les commandes relevant de différentes connexions peuvent être envoyées en parallèle.

3) Sur une connexion donnée, il devrait normalement y avoir seulement une commande en instance (créer ou modifier). Cependant, une commande DeleteConnection peut être produite à tout moment. En conséquence, une passerelle peut parfois recevoir une commande ModifyConnection qui s'applique à une connexion précédemment supprimée. Ces commandes vont échouer, et un code d'erreur DOIT être retourné (le code d'erreur 515 - Identifiant de connexion incorrect, est RECOMMANDÉ).
4) Sur un point d'extrémité donné, il devrait normalement y avoir seulement une commande NotificationRequest en instance à tout moment. Le paramètre RequestId DOIT être utilisé pour corréler les commandes Notify avec la demande de notification déclencheuse.
5) Dans certains cas, une commande DeleteConnection implicitement ou explicitement munie de caractères génériques, qui s'applique à un groupe de points d'extrémité, peut caler en face d'une commande CreateConnection en instance. L'agent d'appel devrait supprimer individuellement toutes les connexions dont l'achèvement était en instance au moment de la commande DeleteConnection globale. Aussi, de nouvelles commandes CreateConnection pour des points d'extrémité désignés par le caractère générique NE DEVRAIENT PAS être envoyées tant que la commande DeleteConnection avec caractères génériques n'est pas acquittée.
6) Quand des commandes sont incorporées au sein d'une autre, les exigences de séquençage pour toutes les commandes doivent être respectées. Par exemple, une commande CreateConnection avec une NotificationRequest en elle doit respecter en même temps les exigences de séquençage associées aux deux CreateConnection et NotificationRequest.
7) AuditEndpoint et AuditConnection ne sont soumises à aucune exigence de séquençage.
8) RestartInProgress DOIT toujours être la première commande envoyée par un point d'extrémité, comme défini par la procédure de redémarrage. Toute autre commande ou réponse non de redémarrage (voir le paragraphe 4.4.6) excepté les réponses à audit, DOIVENT être livrées après cette commande RestartInProgress (le portage est permis).
9) Quand plusieurs messages sont portés dans un seul paquet, les messages sont toujours traités dans l'ordre.
10) Sur un point d'extrémité donné, il devrait normalement y avoir seulement une commande EndpointConfiguration en instance à tout moment.

Les passerelles NE DOIVENT faire aucune hypothèse sur si les agents d'appel suivent ou non ces règles. Par conséquent, les passerelles DOIVENT toujours répondre aux commandes, sans considération de si elles respectent ou non les règles ci-dessus. Pour assurer un fonctionnement cohérent, les passerelles DEVRAIENT se comporter comme spécifié ci-dessous quand une ou plusieurs des règles ci-dessus ne sont pas suivies :

* Lorsque une seule commande en instance est attendue (ModifyConnection, NotificationRequest, et EndpointConfiguration) mais que la même commande est reçue dans une nouvelle transaction avant que l'ancienne ait fini de s'exécuter, la passerelle DEVRAIT faire échouer la commande précédente. Cela inclut le cas où une ou plusieurs des commandes étaient encapsulées. L'utilisation du code d'erreur 407 (Transaction interrompue) est RECOMMANDÉE.

* Si une commande ModifyConnection est reçue pour une commande CreateConnection en instance, la commande ModifyConnection DEVRAIT simplement être rejetée. L'utilisation du code d'erreur 400 (Erreur temporaire) est RECOMMANDÉE. Noter que cette situation constitue une erreur de programmation de l'agent d'appel.

* Si une commande DeleteConnection est reçue pour une commande CreateConnection ou ModifyConnection en instance, la commande en instance DOIT être interrompue. L'utilisation du code d'erreur 407 (Transaction interrompue) est RECOMMANDÉE.

Noter que quand la réception d'une nouvelle commande conduit à l'interruption d'une ancienne commande, l'ancienne commande DEVRAIT être interrompue sans considération de si la nouvelle commande réussit ou non. Par exemple, si une commande ModifyConnection est interrompue par une commande DeleteConnection qui échoue elle-même à cause d'une demande de notification encapsulée, la commande ModifyConnection est quand même interrompue.

4.4.5 États de service de point d'extrémité

Comme décrit précédemment, les points d'extrémité configurés pour fonctionner peuvent être en service ou hors service. L'état de service réel du point d'extrémité est reflété par la combinaison des paramètres RestartMethod et RestartDelay, qui sont envoyés avec des commandes RestartInProgress (paragraphe 2.3.12) et de plus peuvent être examinés dans des commandes AuditEndpoint (paragraphe 2.3.10).

L'état de service d'un point d'extrémité affecte la façon dont il traite une commande. Un point d'extrémité en service DOIT traiter toute commande reçue, tandis qu'un point d'extrémité hors service DOIT rejeter les commandes qui ne sont pas d'audit, mais DEVRAIT traiter les commandes d'audit si possible. Pour la rétro compatibilité, les commandes d'audit pour un point d'extrémité hors service peuvent aussi être rejetées. Toute commande rejetée parce qu'un point d'extrémité est hors service DEVRAIT générer un code d'erreur 501 (Point d'extrémité pas prêt/hors service).

Noter que (selon le paragraphe 2.1.2) sauf mention contraire pour une commande, les noms de point d'extrémité contenant le caractère générique "any of" se réfèrent seulement aux points d'extrémité en service, tandis que les noms de point d'extrémité contenant le caractère générique "all of" se réfèrent à tous les points d'extrémité, sans considération de l'état de service.

Les relations ci-dessus sont illustrées dans le tableau suivant qui montre les états de service en cours et le traitement par la passerelle des commandes en fonction de la commande RestartInProgress envoyée et de la réponse (si il en est) qui en est reçue. La dernière colonne fait aussi la liste (entre parenthèses) de la RestartMethod à retourner en cas d'audit :

Méthode de redémarrage	**Délai de redémarrage**	**2xx reçu ?**	**État de service**	**Réponse à nouvelle commande**
graceful	zéro	oui/non	en service	non audit : 2xx audit : 2xx (graceful)
graceful	non zéro	oui/non	en service*	non audit : 2xx audit : 2xx (graceful)
forced	N/A	oui/non	hors service	non audit : 501 audit : 2xx (forced)
restart	zéro	non	en service	non audit : 2xx, 405* audit : 2xx (restart)
restart	zéro	oui	en service	non audit : 2xx audit : 2xx (restart)
restart	non zéro	non	hors service*	non audit : 501* audit : 2xx(restart)
restart	non zéro	oui	hors service*	non audit : 501* audit : 2xx (restart)
deconnected	zéro/non zéro	non	en service	non audit : 2xx, audit : 2xx (déconnecté)|
deconnected	zéro/non zéro	oui	en service	non audit : 2xx audit : 2xx (restart)
cancel-graceful	N/A	oui/non	en service	non audit : 2xx audit : 2xx (restart) |

Notes (*):

* Les trois états de service marqués avec "*" vont changer après l'expiration de RestartDelay moment auquel une commande RestartInProgress mise à jour DEVRAIT être envoyée.
* Si le point d'extrémité retourne 2xx quand la procédure de redémarrage n'est pas encore achevée, la livraison dans l'ordre DOIT quand même être satisfaite, c'est-à-dire, le portage est à utiliser. Si cependant la commande n'est pas traitée, 405 DEVRAIT être retourné.
* À la suite d'un RestartInProgress "restart" avec un RestartDelay non zéro, le code d'erreur 501 n'est retourné que si le point d'extrémité entre en service, c'est-à-dire, jusqu'à l'expiration de RestartDelay.

4.4.6 Combattre l'avalanche de redémarrages

Supposons qu'un grand nombre de passerelles soient mises sous tension simultanément. Si elles initient toutes une transaction RestartInProgress, l'agent d'appel va très probablement être débordé, conduisant à des pertes de messages et à l'encombrement du réseau durant la période critique de restauration du service. Afin d'empêcher de telles avalanches, le comportement suivant est EXIGÉ :

1) Quand une passerelle est mise sous tension, elle DOIT initier un temporisateur de redémarrage à une valeur aléatoire, uniformément distribuée entre 0 et un délai maximum d'attente (MWD, *Maximum Waiting Delay*). On devrait faire attention à éviter la synchronisation de la génération de nombres aléatoires entre plusieurs passerelles qui utiliseraient le même algorithme.

2) La passerelle DOIT alors attendre la fin de ce temporisateur, la réception d'une commande provenant de l'agent d'appel, ou la détection d'une activité de l'utilisateur local, comme par exemple une transition à décroché sur une passerelle résidentielle.

3) Quand le temporisateur arrive à expiration, quand une commande est reçue, ou quand une activité est détectée, la passerelle DOIT initier la procédure de redémarrage.

La procédure de redémarrage exige simplement que le point d'extrémité garantisse que la première commande non d'audit,

ou une réponse non de redémarrage (c'est-à-dire, des codes d'erreur autres que 405, 501, et 520) à une commande non d'audit que l'agent d'appel voit de ce point d'extrémité soit une commande RestartInProgress "restart". Le point d'extrémité est libre de tirer pleinement parti du portage pour réaliser cela. Les points d'extrémité qui sont considérés comme en service auront une RestartMethod de "restart", tandis que les points d'extrémité considérés comme hors service auront une RestartMethod de "forced" (voir aussi au paragraphe 4.4.5). Les commandes rejetées à cause d'un point d'extrémité qui n'a pas encore achevé la procédure de redémarrage DEVRAIENT utiliser le code d'erreur 405 (Point d'extrémité "en redémarrage").

La procédure de redémarrage est achevée une fois qu'une réponse de succès a été créée. Si une réponse d'erreur est reçue, le comportement suivant dépend du code d'erreur en question :

* Si le code d'erreur indique une erreur temporaire (4xx), la procédure de redémarrage DOIT alors être réinitiée (comme une nouvelle transaction).

* Si le code d'erreur est 521, le point d'extrémité est alors redirigé, et la procédure de redémarrage DOIT être réinitiée (comme une nouvelle transaction). La réponse 521 DOIT avoir inclus une NotifiedEntity qui est alors "l'entité notifiée" sur laquelle le redémarrage est initié. Si elle n'incluait pas de NotifiedEntity, la réponse est traitée comme toute autre erreur permanente (voir ci-dessous).

* Si l'erreur est une autre erreur permanente (5xx), et si le point d'extrémité n'est pas capable de rectifier l'erreur, alors le point d'extrémité n'initie plus la procédure de redémarrage de lui-même (jusqu'à ce qu'il soit réamorcé/redémarré) sauf spécification contraire. Si une commande est reçue pour le point d'extrémité, le point d'extrémité DOIT réinitier la procédure de redémarrage.

Noter que si le RestartInProgress est porté avec la réponse (R) à une commande reçue lors du redémarrage, alors la retransmission du RestartInProgress n'exige pas le portage de la réponse R. Cependant, lorsque le point d'extrémité redémarre, le renvoi de la réponse R exige que le RestartInProgress soit porté pour assurer la livraison dans l'ordre des deux.

Si la passerelle entrait dans l'état "déconnecté" alors qu'elle effectue la procédure de redémarrage, la procédure de déconnecté spécifiée au paragraphe 4.4.7 DOIT être effectuée, sauf qu'un message "redémarrage" plutôt que "déconnecté" est envoyé durant la procédure.

Chaque point d'extrémité dans une passerelle va avoir un agent d'appel provisionnable, c'est-à-dire, une "entité notifiée", pour y diriger le message initial de redémarrage. Quand la collection des points d'extrémité dans une passerelle est gérée par plus d'un agent d'appel, la procédure ci-dessus DOIT être effectuée pour chaque collection de points d'extrémité gérée par un agent d'appel donné. La passerelle DOIT tirer pleinement parti de l'utilisation de caractères génériques pour minimiser le nombre de messages RestartInProgress générés quand plusieurs points d'extrémité dans une passerelle redémarrent et que les points d'extrémité sont gérés par le même agent d'appel. Noter que durant le démarrage, il est possible que des points d'extrémité démarrent comme étant hors service, et ensuite deviennent en service au titre de la procédure d'initialisation de la passerelle. Une passerelle peut donc choisir d'envoyer un premier RestartInProgress "forced" pour tous ses points d'extrémité, et ensuite un RestartInProgress "restart" pour les points d'extrémité qui viennent en service. Autrement, la passerelle peut simplement envoyer un RestartInProgress "restart" pour les seuls points d'extrémité qui sont en service, et un RestartInProgress "forced" pour les points d'extrémité spécifiques qui sont hors service. Des caractères génériques DOIVENT quand même être utilisés pour minimiser le nombre de messages envoyés.

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

Les agents d'appel sont normalement dimensionnés pour traiter la charge de trafic de l'heure de pointe, durant laquelle, en moyenne, 10 % des lignes vont être occupées, passant des appels d'une durée moyenne de normalement 3 minutes. Le traitement d'un appel implique normalement 5 à 6 transactions MGCP entre chaque point d'extrémité et l'agent d'appel. Ce simple calcul montre que l'agent d'appel est supposé traiter 5 à 6 transactions pour chaque point d'extrémité, en moyenne toutes les 30 minutes, ou, pour le dire autrement, environ une transaction par point d'extrémité toutes les 5 à 6 minutes en moyenne. Cela suggère qu'une valeur raisonnable de MWD pour une passerelle résidentielle serait de 10 à 12 minutes. En l'absence de configuration explicite, les passerelles résidentielles devraient adopter une valeur de 600 secondes pour MWD.

Le même raisonnement suggère que la valeur de MWD devrait être plus courte pour les passerelles de circuits ou pour les passerelles d'affaires, parce qu'elles traitent un grand nombre de points d'extrémité, et aussi parce que le taux d'utilisation de ces points d'extrémité est plus élevé que 10 % durant l'heure de pointe , une valeur typique étant de 60 %. Ces points d'extrémité, durant l'heure de pointe, sont donc supposés contribuer à environ une transaction par minute à la charge de

l'agent d'appel. Un algorithme raisonnable est de prendre une valeur de MWD par point d'extrémité de "circuits" six fois plus courte que la MWD de passerelle résidentielle, et aussi inversement proportionnelle au nombre de points d'extrémité qui redémarrrent. Par exemple, MWD devrait être réglé à 2,5 secondes pour une passerelle qui traite une ligne T1, ou à 60 millisecondes pour une passerelle qui traite une ligne T3.

4.4.7 Points d'extrémité déconnectés

En plus de la procédure de redémarrage, les passerelles ont aussi une procédure "déconnecté", qui DOIT être initiée quand un point d'extrémité devient "déconnecté" comme décrit au paragraphe 4.3. On devrait noter ici que les points d'extrémité peuvent seulement devenir déconnectés quand ils tentent de communiquer avec l'agent d'appel. Les étapes suivantes DOIVENT être suivies par un point d'extrémité qui devient "déconnecté" :

1. Un temporisateur "déconnecté" est initialisé à une valeur aléatoire, uniformément distribuée entre 1 et un délai d'attente inital "déconnecté" (Tdinit) provisionnable, par exemple, de 15 secondes. On DOIT faire attention à éviter de synchroniser la génération de nombre aléatoire entre plusieurs passerelles et points d'extrémité qui utiliseraient le même algorithme.

2. La passerelle attend ensuite la fin de ce temporisateur, la réception d'une commande pour le point d'extrémité de la part de l'agent d'appel, ou la détection d'une activité d'utilisateur local pour le point d'extrémité, comme par exemple une transition à décroché.

3. Quand le temporisateur "déconnecté" s'écoule pour le point d'extrémité, et qu'une commande est reçue pour le point d'extrémité, ou quand une activité d'utilisateur local est détectée pour le point d'extrémité, la passerelle initie la procédure "déconnecté" pour le point d'extrémité - si une procédure "déconnecté" est déjà en cours pour le point d'extrémité, elle est simplement remplacée par la nouvelle. De plus, dans le cas d'une activité d'utilisateur local, un délai d'attente minimum "déconnecté" (Tdmin) provisionnable DOIT s'être écoulé depuis que le point d'extrémité est devenu déconnecté ou depuis la dernière fois qu'il a terminé la procédure "déconnecté" afin de limiter le taux d'utilisation de la procédure. Si Tdmin n'est pas passé, le point d'extrémité repasse simplement à l'étape 2, sans affecter de procédure déconnecté déjà en cours.

4. Si la procédure "déconnecté" laisse encore le point d'extrémité déconnecté, le temporisateur "déconnecté" est alors doublé, sous réserve d'un délai provisionnable d'attente "déconnecté" maximum (Tdmax) par exemple, 600 secondes, et la passerelle refait l'étape 2 (en utilisant un nouvel identifiant de transaction).

La procédure "déconnecté" est similaire à la procédure de redémarrage en ce qu'elle déclare simplement que le point d'extrémité DOIT envoyer une commande RestartInProgress à l'agent d'appel pour l'informer que le point d'extrémité a été déconnecté. De plus, le point d'extrémité DOIT garantir que le premier message non d'audit (commande non d'audit ou réponse à une commande non d'audit) que l'agent d'appel voit provenant de ce point d'extrémité DOIT informer l'agent d'appel que le point d'extrémité est déconnecté (sauf si le point d'extrémité passe hors service). Quand une commande (C) est reçue, ceci est réalisé par l'envoi d'un datagramme porté avec une commande RestartInProgress "déconnecté" et la réponse à la commande C à l'adresse de source de la commande C par opposition à "l'entité notifiée" en cours. Ce RestartInProgress porté n'est pas automatiquement retransmis par le point d'extrémité mais s'appuie simplement sur le partage de sort avec la réponse portée pour garantir l'exigence de livraison dans l'ordre. L'agent d'appel envoie quand même une réponse au RestartInProgress porté, cependant, comme d'habitude, la réponse peut être perdue. En plus de la commande RestartInProgress portée, une nouvelle procédure "déconnecté" est déclenchée par la commande reçue. Cela va conduire à une copie non portée (c'est-à-dire, dans la même transaction) de la commande RestartInProgress "déconnecté" qui est envoyée de façon fiable à "l'entité notifiée" en cours.

Quand l'agent d'appel apprend que le point d'extrémité est déconnecté, l'agent d'appel peut alors par exemple décider d'examiner le point d'extrémité, ou simplement de supprimer toutes les connexions pour le point d'extrémité. Noter que chacune de ces procédures "déconnecté" va résulter en une nouvelle commande RestartInProgress, qui va être soumise aux procédures normales de retransmission spécifiées au paragraphe 4.3. À la fin de la procédure, le point d'extrémité peut donc être encore "déconnecté". Si le point d'extrémité passe hors service alors qu'il est déconnecté, il DEVRAIT envoyer un message RestartInProgress "forcé" comme décrit au paragraphe 2.3.12.

La procédure "déconnecté" est achevée quand une réponse de succès a été créée. Les réponses d'erreur sont traitées de façon similaire à la procédure de redémarrage (paragraphe 4.4.6). Si la procédure "déconnecté" doit être initiée à nouveau suite à une réponse d'erreur, les considérations de temporisateur de limitation de taux spécifiées ci-dessus s'appliquent.

Noter que si le RestartInProgress est porté avec la réponse (R) à une commande reçue alors qu'il est déconnecté, la retransmission de ce RestartInProgress particulier n'exige pas le portage de la réponse R. Cependant, quand le point

d'extrémité est déconnecté, le renvoi de la réponse R exige bien que le RestartInProgress soit porté avec la réponse pour assurer la livraison dans l'ordre des deux.

Si un ensemble de points d'extrémité déconnectés a la même "entité notifiée", et si l'ensemble de points d'extrémité peut être désigné avec un caractère générique, la passerelle PEUT remplacer les procédures "déconnecté" individuelles par une procédure "déconnecté" munie des caractères génériques appropriés. Dans ce cas, le délai de redémarrage pour la commande RestartInProgress "déconnecté" avec les caractères génériques DEVRA être le délai de redémarrage correspondant à la plus ancienne procédure "déconnecté" remplacée. Noter que si seul un sous ensemble de ces points d'extrémité change ensuite son "entité notifiée" et/ou ne sont plus déconnectés, alors cette procédure "déconnecté" avec caractères génériques ne peut plus être utilisée. Les procédures "déconnecté" individuelles restantes DOIVENT alors être reproduites à nouveau.

Un point d'extrémité déconnecté peut souhaiter envoyer une commande (en dehors de RestartInProgress) alors qu'il est déconnecté. Le faire ne va réussir que lorsque l'agent d'appel est à nouveau accessible, ce qui soulève la question de quoi faire entre temps d'une telle commande. D'un côté, le point d'extrémité pourrait éliminer directement la commande, cependant cela ne fonctionnerait pas très bien quand l'agent d'appel est en fait disponible, mais que le point d'extrémité n'a pas encore achevé la procédure "déconnecté" (considérons par exemple le cas où une NotificationRequest a juste été reçue qui a immédiatement résulté en la génération d'un Notify). Pour empêcher de tels scénarios, les points d'extrémité déconnectés NE DEVRONT PAS éliminer aveuglément de nouvelles commandes à envoyer pendant une durée de T-MAX secondes après la réception d'une commande non d'audit.

Une façon de satisfaite cette exigence est d'employer une mise en mémoire tampon temporaire des commandes à envoyer, cependant en faisant ainsi, le point d'extrémité DOIT s'assurer que :
* il ne construit pas une longue file d'attente de commandes à envoyer,
* il ne submerge pas l'agent d'appel en lui envoyant rapidement trop de commandes une fois qu'il est reconnecté.

Mettre en mémoire tampon les commandes pendant T-MAX secondes et, une fois que le point d'extrémité est réconnecté, limiter le taux auquel les commandes mises en mémoire tampon sont envoyées à une commande en instance par point d'extrémité est considéré comme acceptable (voir aussi au paragraphe 4.4.8, en particulier si on utilise des caractères génériques). Si le point d'extrémité n'est pas connecté dans les T-MAX secondes, mais qu'une procédure "déconnecté" est initiée dans les T-MAX secondes, le point d'extrémité PEUT porter la ou les commandes en mémoire tampon avec ce RestartInProgress. Noter qu'une fois qu'une commande a été envoyée, sans considération de si elle a été initialement mise en mémoire tampon, ou portée précédemment, les retransmissions de cette commande DOIVENT cesser T-MAX secondes après l'envoi initial, comme décrit au paragraphe 4.3.

La présente spécification ne spécifie délibérément aucun comportement supplémentaire pour un point d'extrémité déconnecté. Les fabricants PEUVENT par exemple choisir de fournir du silence, d'exécuter une tonalité de remise en ordre, ou même de permettre l'exécution d'un fichier wav téléchargé.

La valeur par défaut est 15 secondes pour Tdinit, 15 secondes pour Tdmin, et 600 secondes pour Tdmax.

4.4.8 Contrôle de charge en général

Les paragraphes précédents ont décrit plusieurs mécanismes MGCP pour traiter l'encombrement et la surcharge, à savoir :
* la stratégie de retransmission UDP qui s'adapte à l'encombrement du réseau et de l'agent d'appel sur la base du point d'extrémité,
* les lignes directrices sur l'ordre des commandes qui limitent le nombre de commandes produites en parallèle,
* la procédure de redémarrage qui empêche l'inondation en cas d'avalanche de redémarrages, et
* la procédure "déconnecté" qui empêche l'inondation en cas d'un grand nombre de points d'extrémité déconnectés.

Il est cependant toujours possible à un certain ensemble de points d'extrémité, soit sur la même, soit sur des passerelles différentes, de produire une ou plusieurs commandes à un certain moment. Bien qu'on pourrait objecter que les agents d'appel devraient être dimensionnés pour traiter un message par point d'extrémité desservi à tout moment, cela peut n'être pas toujours le cas en pratique. De même, les passerelles peuvent n'être pas capables de traiter un message pour tous leurs points d'extrémité à tout moment. En général, ces problèmes peuvent être traités par l'utilisation d'un mécanisme fondé sur le crédit, ou en surveillant et adaptant automatiquement le comportement observé. On opte pour cette dernière approche comme suit .

Conceptuellement, on suppose que les agents d'appel et les passerelles vont tenir une file d'attente des transactions entrantes à exécuter. Associée à cette file d'attente de transactions est une marque de hautes et de basses eaux. Une fois que la file d'attente atteint la marque des hautes eaux, l'entité DEVRAIT commencer à produire des réponses provisoires 101

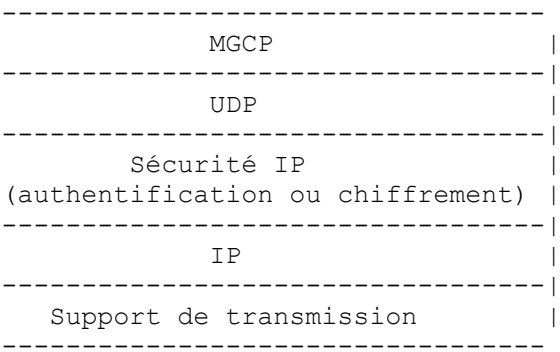
(transaction en file d'attente) jusqu'à ce que la longueur de la fils d'atttente tombe à la marque des basses eaux. Ceci s'applique aux nouvelles transactions aussi bien qu'aux retransmissions. Si l'entité n'est pas capable de traiter de nouvelles transactions à ce moment, elle DEVRAIT retourner le code d'erreur 409 (Surcharge de traitement).

De plus, les passerelles DEVRAIENT ajuster le taux d'envoi des nouvelles commandes à un certain agent d'appel en surveillant les temps de réponse observés en provenance de cet agent d'appel à un *ensemble* de points d'extrémité. Si le temps de réponse moyen lissé observé augmente soudainement de façon significative au dessus d'un certain seuil, ou si la passerelle reçoit une réponse 101 (Transaction mise en file d'attente) ou 409 (Surcharge) la passerelle DEVRAIT ajuster en conséquence le taux d'envoi des nouvelles commandes à cet agent d'appel. Les détails de l'algorithme de lissage de la moyenne, le taux d'ajustement, et les seuils impliqués feront l'objet d'études complémentaires, cependant ils DOIVENT être configurables.

De même, les agents d'appel DEVRAIENT ajuster le taux d'envoi des nouvelles transactions à une certaine passerelle en surveillant les temps de réponse observés en provenance de cette passerelle pour un *ensemble* de points d'extrémité. Si le temps de réponse moyen lissé observé augmente soudainement de façon significative au dessus d'un certain seuil, ou si l'agent d'appel reçoit une réponse 101 (Transaction mise en file d'attente) ou 409 (Surcharge) l'agent d'appel DEVRAIT ajuster en conséquence le taux d'envoi des nouvelles commandes à cette passerelle. Les détails de l'algorithme de lissage de la moyenne, le taux d'ajustement, et les seuils impliqués feront l'objet d'études complémentaires, cependant ils DOIVENT être configurables.

5. Exigences pour la sécurité

Toute entité peut envoyer une commande à un point d'extrémité MGCP. Si une entité non autorisée pouvait utiliser le MGCP, elle serait capable d'établir des appels non autorisés, ou d'interférer avec des appels autorisés. On s'attend à ce que les messages MGCP soient toujours portés sur des connexions Intenet sécurisées, comme défini dans l'architecture de sécurité IP définie dans la RFC 2401, en utilisant l'en-tête d'authentification IP, défini dans la RFC 2402, ou la charge utile de sécurité encapsulante IP, définie dans la RFC 2406. La pile de protocole MGCP complète va donc inclure les couches suivantes :



Une protection adéquate des connexions va être réalisée si les passerelles et les agents d'appel acceptent seulement des messages pour lesquels la sécurité IP a fourni un service d'authentification. Un service de chiffrement va fournir une protection supplémentaire contre l'espionnnage, empêchant donc des tiers de surveiller les connexions établies par un certain point d'extrémité.

Le service de chiffrement va aussi être demandé si les descriptions de session sont utilisées pour porter des clés de session, comme défini dans SDP.

Ces procédures ne protègent pas nécessairement contre les attaques de déni de service par des passerelles ou agents d'appel qui se comportent mal. Cependant, elles vont fournir une identification de ces entités au mauvais comportement, qui devraient alors être privées de leur autorisation par les procédures de maintenance.

5.1 Protection des connexions de supports

MGCP permet aux agents d'appel de fournir aux passerelles des "clés de session" qui peuvent être utilisées pour chiffrer les messages audio, protégeant contre l'espionnage.

Un problème spécifique des réseaux de paquets est "l'irruption non contrôlée". Cette attaque peut être effectuée en dirigeant des paquets sur l'adresse IP et l'accès UDP utilisés par une connexion. Si aucune protection n'est mise en œuvre, les paquets vont être décodés et les signaux vont être exécutés sur le "côté ligne".

Une protection basique contre cette attaque est de n'accepter de paquets que de sources connues, cependant ceci entre en conflit avec les principes de RTP. Cela a aussi deux inconvénients : cela ralentit l'établissement de la connexion et cela peut être trompé par une usurpation de source :

* Pour permetttre la protection fondée sur l'adresse, l'agent d'appel doit obtenir l'adresse de source de la passerelle de sortie et la passer à la passerelle d'entrée. Cela exige au moins un aller-retour de réseau, et laisse un dilemme : soit permettre à l'appel de se poursuivre sans attendre que l'aller-retour soit achevé, et risquer par exemple de "couper" une annonce distante, soit attendre la fin de l'aller-retour complet et se résoudre à de plus lentes procédures d'établissement.
* L'usurpation de source n'est effective que si l'attaquant peut obtenir des paires valides d'adresse et d'accès de source et de destination, par exemple en écoutant une fraction du trafic. Pour combattre l'usurpation de source, on pourrait essayer de contrôler tous les points d'accès au réseau. Mais ceci est en pratique très difficile à réaliser.

Une solution de remplacement à la vérification de l'adresse de source est de chiffrer et authentifier les paquets, en utilisant une clé secrète qui est portée durant la procédure d'établissement de l'appel. Cela ne va pas ralentir l'établissement d'appel, et fournit une forte protection contre l'usurpation d'adresse.

6. Paquetages

Comme décrit au paragraphe 2.1.6, les paquetages sont la façon préférée pour étendre MGCP. Dans cette section, on décrit les exigences associées à la définition d'un paquetage.

Un paquetage DOIT avoir un nom de paquetage défini de façon univoque. Le nom de paquetage DOIT être enregistré par l'IANA, sauf si il commence par les caractères "x-" ou "x+" qui sont réservés aux paquetages expérimentaux. Prière de se référer à l'Appendice C pour les considérations relatives à l'IANA.

Un paquetage DOIT aussi avoir une version définie qui est simplement un entier non négatif. La version initiale et par défaut d'un paquetage est zéro, la version suivante est un, etc. Les nouvelles versions de paquetage DOIVENT être complètement rétro compatibles, c'est-à-dire qu'une nouvelle version d'un paquetage NE DOIT PAS redéfinir ou supprimer d'extensions fournies dans une version antérieure du paquetage. Si un tel besoin apparaît, un nouveau nom de paquetage DOIT être utilisé à la place.

Les paquetages qui contiennent des signaux de type fin de temporisation PEUVENT indiquer si le paramètre "to" est pris en charge pour tous les signaux de fin de temporisation dans le paquetage ainsi que les règles d'arrondi par défaut qui leur sont associées (voir au paragraphe 3.2.2.4). Si une telle définition n'est pas fournie, chaque signal de fin de temporisation DEVRAIT fournir ces définitions.

Un paquetage définit une ou plusieurs des extensions suivantes :
* Actions
* BearerInformation *(informations de support)*
* ConnectionModes *(modes de connexion)*
* ConnectionParameters *(paramètres de connexion)*
* DigitMapLetters *(lettres du script de numérotation)*
* Événements et signaux
* ExtensionParameters *(paramètres d'extension)*
* LocalConnectionOptions *(options de connexion locale)*
* ReasonCodes *(codes de cause)*
* RestartMethods *(méthodes de redémarrage)*
* Codes de retour

Pour chacun des types d'extensions ci-dessus pris en charge par le paquetage, la définition du paquetage DOIT contenir une description de l'extension comme définie dans les paragraphes suivants. Noter que les extensions de paquetage, comme toute autre extension, DOIVENT respecter la grammaire de MGCP.

6.1 Actions

L'extension Actions DEVRA inclure :
* Le nom et le codage de l'extension Action.
* Si l'extension Action prend des paramètres d'action, le nom, le codage, et les valeurs possibles de ces paramètres.
* Une description du fonctionnement de l'extension Action.
* Une liste des actions dans cettte spécification auxquelles l'extension peut être combinée. Si une telle liste n'est pas fournie, il est supposé que l'extension Action ne peut pas être combinée avec une autre action dans cette spécification.
* Si plus d'une extension Action est définie dans le paquetage, une liste des actions dans le paquetage avec lesquelles l'extension peut être combinée. Si une telle liste n'est pas fournie, il est supposé que l'extension Action ne peut pas être combinée avec une autre action dans le paquetage.

Les extensions Action définies dans deux paquetages différents ou plus NE DEVRAIENT PAS être utilisées simultanément, sauf à considérer très attentivement leur potentielle interaction et les effets collatéraux.

6.2 BearerInformation

Les extensions BearerInformation DEVRONT inclure :
* Le nom et le codage de l'extension BearerInformation.
* Les valeurs possibles et le codage de ces valeurs qui peuvent être allouées à l'extension BearerInformation.
* Une description du fonctionnement de l'extension BearerInformation. Au titre de cette description, la valeur par défaut (si il en est) si l'extension est omise dans une commande EndpointConfiguration DOIT être définie. Il peut être nécessaire de faire une distinction entre la valeur par défaut avant et après l'application initiale du paramètre, par exemple, si le paramètre conserve sa valeur précédente une fois spécifié, jusqu'à ce qu'elle soit explicitement altérée. Si les valeurs par défaut ne sont pas décrites, alors le paramètre d'extension est simplement vide par défaut dans toutes les commandes EndpointConfiguration.

Noter que l'extension DEVRA être incluse dans le résultat pour une commande AuditEndpoint examinant les BearerInformation.

6.3 ConnectionModes

L'extension Modes de connexion DEVRA inclure :
* Le nom et le codage de l'extension Modes de connexion.
* Une description du fonctionnement de l'extension Modes de connexion.
* Une description de l'interaction qu'une connexion dans l'extension Modes de connexion aura avec d'autres connexions dans chacun des modes définis dans cette spécification. Si une telle description n'est pas fournie, l'extension Modes de connexion NE DOIT PAS avoir d'interaction avec d'autres connexions sur le point d'extrémité.

L'extension Modes de connexion NE DEVRA PAS être incluse dans la liste des modes dans une réponse à un AuditEndpoint sur les capacités, car le paquetage va être rapporté dans la liste des paquetages.

6.4 ConnectionParameters

L'extension Paramètres de connexion DEVRA inclure :
* Le nom et le codage de l'extension Paramètres de connexion.
* Les valeurs possibles et le codage de ces valeurs qui peuvent être alloués à l'extension Paramètres de connexion.
* Une description de la façon dont ces valeurs sont déduites.

Noter que l'extension Paramètres de connexion DOIT être incluse dans le résultat pour une commande AuditConnection qui examine les paramètres de connexion.

6.5 DigitMapLetters

L'extension Lettres du script de numérotation DEVRA inclure :
* Le nom et le codage de l'extension Lettres du script de numérotation.
* Une description de la signification de l'extension Lettres du script de numérotation.

Noter que l'extension DigitMapLetters dans un script de numérotation ne suit pas les conventions normales de désignation

pour les extensions définies dans les paquetages. Plus précisément, le nom de paquetage et la barre oblique ("/") ne vont pas faire partie du nom de l'extension, formant ainsi un espace de noms plat et limité avec de potentiels conflits de noms.

Donc, un paquetage NE DEVRA PAS définir d'extension Lettre de script de numérotation dont le codage serait déjà utilisé dans un autre paquetage. Si deux paquetages ont utilisé le même codage pour une extension Lettre de script de numérotation, et si ces deux paquetages sont pris en charge par le même point d'extrémité, le résultat de l'utilisation de cette extension Lettre de script de numérotation est indéfini.

Noter que bien qu'une extension DigitMapLetter n'inclue pas de préfixe de nom de paquetage ni de barre oblique ("/") au titre du nom d'extension au sein d'un script de numérotation, le préfixe de nom de paquetage et la barre oblique sont inclus quand l'événement qui code pour l'événement correspondant au DigitMapLetter est rapporté comme événement observé. En d'autres termes, le script de numérotation définit juste les règles de correspondance, mais l'événement est quand même rapporté comme tout autre événement.

6.6 Événements et signaux

La définition d'événement/signal DEVRA inclure le nom précis de l'événement/signal (c'est-à-dire, le code utilisé dans MGCP) une définition textuelle de l'événement/signal, et, quand c'est approprié, la définition précise des événements/signaux correspondants, par exemple les fréquences exactes des signaux audio comme les tonalités de numérotation ou les tonalités DTMF.

La description de paquetage DOIT fournir, pour chaque événement/signal, les informations suivantes :

* La description de l'événement/signal et son objet, qui DEVRAIT inclure le signal réel généré par le client (par exemple, tonalité FSK de xx ms) ainsi que le résultat observé par l'utilisateur (par exemple, indicateur lumineux de message en attente).
* Le code d'événement utilisé pour l'événement/signal.
* Les caractéristiques détaillées de l'événement/signal, comme par exemple les fréquences et amplitude des signaux audio, les modulations et les répétitions. Ces détails peuvent être spécifiques du pays.
* La durée typique et maximale de l'événement/signal si applicable.
* Si le signal ou événement peut être appliqué à une connexion (à travers un flux de supports) il DOIT être indiqué explicitement. Si une telle indication n'est pas fournie, il est supposé que le signal ou événement ne peut pas être appliqué à une connexion.

Pour les événements, ce qui suit DOIT aussi être fourni :

* L'indication de si l'événement est persistant. Par défaut, les événements ne sont pas persistants - définir des événements comme persistants est déconseillé (voir à l'Appendice B une solution de remplacement préférée). Noter que les événements persistants vont automatiquement déclencher un Notify quand ils se produisent, sauf si l'agent d'appel a explicitement donné pour instruction au point d'extrémité de faire autrement. Ceci non seulement viole le modèle MGCP normal, mais aussi suppose que l'agent d'appel prend en charge le paquetage en question. Une telle hypothèse a peu de chances d'être vérifiée en général.
* L'indication de si il y a un état d'événement auditable associé à l'événement. Par défaut, les événements n'ont pas d'état d'événement auditable.
* Si l'événement Paramètres est pris en charge, il DOIT être déclaré explicitement. Sa syntaxe et sa sémantique précise DOIVENT alors être fournies (sous réserve de la grammaire de l'Appendice A). Il DEVRAIT aussi être spécifié si ces paramètres s'appliquent aux RequestedEvents, ObservedEvents, DetectEvents et EventStates. Si ce n'est pas spécifié autrement, il est supposé que :
  * ils ne s'appliquent pas aux RequestedEvents,
  * ils s'appliquent aux ObservedEvents,
  * ils s'appliquent de la même façon aux DetectEvents qu'aux RequestedEvents pour un certain paramètre d'événement,
  * ils s'appliquent de la même façon aux EventStates qu'aux ObservedEvents pour un certain paramètre d'événement.
* Si l'événement est supposé être utilisé dans la confrontation au script de numérotation, ce DEVRAIT être déclaré explicitement. Noter que seuls les événements avec des codes de paramètre d'une seule lettre ou chiffre peuvent faire cela. Voir les détails au paragraphe 2.1.5.

Pour les signaux, ce qui suit DOIT aussi être fourni :

* Le type de signal (OO, TO, BR).
* Les signaux de temporisation DEVRAIENT avoir une indication de la valeur de fin de temporisation par défaut. Dans certains cas, les valeurs de fin de temporisation peuvent être variables (si elles dépendent de l'achèvement d'une action comme des chiffres à impulsion).
* Si les paramètres de signal sont pris en charge, ils DOIVENT être déclarés explicitement. Leur syntaxe et sémantique

précise DOIVENT alors être fournies (sous réserve de la grammaire de l'Appendice A).

* Les signaux de fin de temporisation (TO) peuvent aussi indiquer si le paramètre "to" est pris en charge ou non ainsi que ce que sont les règles d'arrondi associées. Si elle est omise de la définition de signal, la définition pour l'ensemble du paquetage est supposée (voir la Section 6). Si la définition de paquetage ne spécifiait pas cela, les règles d'arrondi par défaut sont à la plus proche seconde non zéro, tandis que la prise en charge du paramètre "to" est par défaut "non" pour la version de paquetage zéro, et "oui" pour les versions de paquetage une et supérieures.

Le format suivant est RECOMMANDÉ pour définir les événements et signaux conformément à ce qui précède :

```
-------------------------------------------------------------
| Symbole |   Définition                |  R  |   S     Durée    |
|---------|-----------------------------|-----|------------------|
|         |                             |     |                  |
|         |                             |     |                  |
-------------------------------------------------------------
```

où ;

* Symbole indique le code d'événement utilisé pour l'événement/signal, par exemple, "hd".
* Définition donne une brève définition de l'événement/signal
* R contient un "x" si l'événement peut être détecté ou un ou plusieurs des symboles suivants :
  - "P" si l'événement est persistant.
  - "S" si l'événement est un état d'événement qui peut être audité.
  - "C" si l'événement peut être détecté sur une connexion.
* S contient une des valeurs suivantes si il est un signal :
  - "OO" si le signal est un signal On/Off *(ouvert/fermé)*.
  - "TO" si le signal est un signal Time-Out *(temporisation)*.
  - "BR" si le signal est un signal Bref.
* S contient aussi :
  - "C" si le signal peut être appliqué sur une connexion.

Le tableau DEVRAIT alors être suivi par une description plus complète de chaque événement/signal défini.

6.6.1 Événements par défaut et réservés

Tous les paquetages qui contienent des signaux de type Time-Out contiennent les événements Échec d'opération ("of", *operation failure*) et Opération achevée ("oc", *operation complete*) sans considération de si ils sont fournis au titre de la description du paquetage ou non. Ces événements sont nécessaires pour prendre en charge les signaux Time-Out et ne peuvent pas être outrepassés dans les paquetages avec des signaux Time-Out. Ils PEUVENT être étendus si nécessaire, cependant une telle pratique est déconseillée.

Si un paquetage sans signal Time-Out contient des définitions pour les événements "oc" et "of", les définitions d'événement fournies dans le paquetage PEUVENT outrepasser celles indiquées ici. Une telle pratique est cependant déconseillée et est simplement admise pour éviter de potentiels problèmes de rétro compatibilité.

Il est considéré comme de bonne pratique de mentionner explicitement que les deux événements sont pris en charge en accordance avec leurs définitions par défaut, qui sont comme suit :

Symbole	**Définition**	**R**	**S**	**Durée**
oc	Opération achevée	x		
of	Échec d'opération	x		

Opération achevée (oc) : l'événement Opération achevée est généré quand il a été demandé à la passerelle d'appliquer un ou plusieurs signaux de type TO sur le point d'extrémité ou la connexion, et que un ou plusieurs de ces signaux se sont achevés sans être arrêtés par la détection d'un événement demandé comme une transition à décroché ou des chiffres numérotés. Le rapport d'achèvement devrait porter comme paramètre le nom du du signal qui arrive en fin de vie, comme dans :

O: G/oc(G/rt)

Dans ce cas, l'événement observé s'est produit parce que le signal "rt" dans le paquetage "G" est arrivé en fin de temporisation.

Si le signal rapporté était appliqué à une connexion, le paramètre fourni va inclure aussi le nom de la connexion, comme dans :

O: G/oc(G/rt@0A3F58)

Quand l'événement Opération achevée est demandé, il ne peut pas être paramétré avec des paramètres d'événement. Quand le nom de paquetage est omis (ce qui est déconseillé) au titre du nom du signal, le paquetage par défaut est supposé.

Échec d'opération (of) : l'événement Échec d'opération est généré quand il a été demandé au point d'extrémité d'appliquer un ou plusieurs signaux de type TO sur le point d'extrémité ou la connexion, et que un ou plusieurs de ces signaux a échoué avant 'arriver en fin de temporisation. Le rapport d'achèvement devrait porter comme paramètre le nom du signal qui a échoué, comme dans :

O: G/of(G/rt)

Dans ce cas, un échec s'est produit en produisant le signal "rt" dans le paquetage "G".

Quand le signal rapporté était appliqué sur une connexion, le paramètre fourni va inclure aussi le nom de la connexion, comme dans :

O: G/of(G/rt@0A3F58)

Quand l'événement Échec d'opération est demandé, des paramètres d'événement ne peuvent pas être spécifiés. Quand le nom de paquetage est omis (ce qui est déconseillé) le nom de paquetage par défaut est supposé.

6.7 ExtensionParameters

Les extensions Paramètre d'extension DEVRONT inclure :
* Le nom et le codage du paramètre d'extension.
* Les valeurs possibles et le codage de ces valeurs qui peuvent être allouées au paramètre d'extension.
* Pour chacune des commandes définies dans cette spécification, si le paramètre d'extension est obligatoire, facultatif, ou interdit dans les demandes et les réponses. Noter que les paramètres d'extension NE DEVRAIENT normalement PAS être obligatoires.
* Une description du fonctionnement du paramètre d'extension. Au titre de cette description, la valeur par défaut (si il y en a une) si l'extension est omise dans une commande, DOIT être définie. Il peut être nécessaire de faire une distinction entre la valeur par défaut avant et après l'application initiale du paramètre, par exemple, si le paramètre conserve sa valeur précédente une fois spécifié, jusqu'à ce qu'il soit explicitement modifié. Si les valeurs par défaut ne sont pas décrites, alors le paramètre d'extension prend simplement par défaut la valeur vide dans toutes les commandes.
* Si l'extension peut être examinée dans AuditEndpoint et/ou AuditConnection ainsi que les valeurs retournées. Si rien n'est spécifié, alors l'examen du paramètre d'extension peut seulement être fait pour AuditEndpoint, et la valeur retournée DEVRA être la valeur courante pour l'extension. Noter qu'elle peut être vide.

6.8 LocalConnectionOptions

Les extensions LocalConnectionOptions DEVRONT inclure :
* Le nom et le codage de l'extension LocalConnectionOptions.
* Les valeurs possibles et le codage de ces valeurs qui peuvent être allouées à l'extension LocalConnectionOptions.
* Une description du fonctionnement de l'extension LocalConnectionOptions. Au titre de cette description, on DOIT spécifier :
  - La valeur par défaut (si il en est) si l'extension est omise dans une commande CreateConnection.
  - La valeur par défaut si l'extension est omise dans une commande ModifyConnection. Ce peut être de simplement conserver la valeur précédente (si il en est une) ou d'appliquer la valeur par défaut. Si rien n'est spécifié, la valeur courante est conservée si possible.
  - Si l'audit des capacités va résulter en le retour de l'extension, une description à cet effet ainsi que les valeurs possibles et leur codage (noter que le paquetage lui-même va toujours être retourné). Si rien n'est spécifié, l'extension NE DEVRA PAS être retournée quand on examine les capacités.

Noter aussi que l'extension DOIT être incluse dans le résultat d'une commande AuditConnection examinant les LocalConnectionOptions.

6.9 Codes de cause

Les codes de cause d'extension DEVRONT inclure :
* Le numéro du code de cause. Le numéro DOIT être dans la gamme de 800 à 899.
* Une description du code de cause d'extension incluant les circonstances qui conduisent à la génération du code de cause. Ces circonstances DEVRAIENT être limitées aux événements causés par une autre extension définie dans le paquetage pour assurer que le receveur va être capable d'interpréter correctement le code de cause d'extension.

Noter que le code de cause d'extension peut devoir être fourni dans le résultat pour une commande AuditEndpoint qui examine le code de cause.

6.10 RestartMethods

L'extension Méthodes de redémarrage DEVRA inclure :
* Le nom et le codage pour la méthode de redémarrage.
* Une description de la méthode de redémarrage incluant les circonstances qui conduisent à la génération de la méthode de redémarrage. Ces circonstances DEVRAIENT être limitées aux événements causés par une autre extension définie dans le paquetage pour assurer que le receveur va être capable d'interpréter correctement l'extension de méthode de redémarrage.
* Une indication de si le paramètre RestartDelay est à utiliser avec l'extension. Si rien n'est spécifié, il est supposé qu'il n'est pas à utiliser. Dans ce cas, RestartDelay DOIT être ignoré si il est présent.
* Si la méthode de redémarrage définit un état de service, la description DOIT explicitement le déclarer et le décrire. Dans ce cas, l'extension de méthode de redémarrage peut alors être fournie dans le résultat pour une commande AuditEndpoint qui examine la méthode de redémarrage.

6.11 Codes de retour

L'extension Codes de retour DEVRA inclure :
* Le numéro du code de retour d'extension. Le numéro DOIT être dans la gamme de 800 à 899.
* Une description du code de retour d'extension incluant les circonstances qui conduisent à la génération du code de retour d'extension. Ces circonstances DEVRAIENT être limitées aux événements causés par une autre extension définie dans le paquetage pour assurer que le receveur va être capable d'interpréter correctement le code de retour d'extension.

7. Versions et compatibilité

7.1 Changements par rapport à la RFC 2705

La RFC 2705 a été produite en octobre 1999, comme dernière mise à jour du projet de version 0.5. Le présent document mis à jour bénéficie de l'expérience de mise en œuvre ultérieure. Les principaux changements par rapport à la RFC 2705 sont :
* Plusieurs précisions, changements rédactionnels et la résolution des incohérences connues.
* Utilisation du langage de spécification en accord avec la RFC 2119 et ajout du paragraphe des conventions de la RFC 2119.
* Précision du comportement de l'utilisation de caractères génériques mixtes dans les noms de point d'extrémité.
* Suppression de l'exigence de désignation d'avoir le premier terme qui identifie la passerelle physique quand la passerelle consiste en plusieurs passerelles physiques. Ajout aussi des recommandations sur l'utilisation de caractères génériques dans les désignatisons seulement à partir de la droite, ainsi que l'usage de caractères génériques mixtes.
* Précision que les formes synonymes et valeurs pour les noms de point d'extrémité ne sont pas librement interchangeables.
* Permettre les adresses IPv6 dans les noms de point d'extrémité.
* Précision des règles de confrontation de script de numérotation.
* Ajout de la sémantique manquante pour les symboles utilisés dans les scripts de numérotation.
* Ajout de la description du temporisateur T dans les scripts de numérotation.
* Ajout de la recommandation de prendre en charge les tailles de script de numérotation d'au moins 2048 octets par point d'extrémité.
* Précision sur l'utilisation de caractères génériques dans plusieurs commandes.
* Définition formelle des paramètres Événement et signal pour les événements et signaux.

* Les événements persistants sont maintenant permis dans le protocole de base MGCP.
* Ajout de détails supplémentaires sur les caractères génériques de connexion.
* Précision sur le comportement de rebouclage, et les modes de connexion d'essai de continuité pour le mixage et pour plusieurs connexions dans ces modes.
* Modification de BearerInformation pour le rendre facultatif et conditionnel dans la commande EndpointConfiguration.
* Précision sur l'opération d'action "swap audio" pour un scénario spécifique et noté que le fonctionnement pour les autres scenarios est indéfini.
* Ajout de la recommandation que toutes les mises en œuvre prennent en charge le codage MIC U pour l'interopérabilité.
* Changement de la valeur de bande passante de LocalConnectionOptions de l'exclusion à l'inclusion des frais généraux de la couche IP et au delà pour la cohérence avec SDP.
* Précision que le mode de seconde connexion dans une commande CreateConnection va être réglé à "envoi/réception".
* Le type de service par défaut est changé en zéro.
* Ajout de détails supplémentaires sur l'annulation d'écho, la suppression de silence, et le contrôle de gain. Ajout aussi d'une recommandation que les agents d'appel ne spécifient pas le traitement de l'annulation d'écho et du contrôle de gain.
* Ajout de l'exigence qu'une connexion ait un descripteur de connexion distante afin d'utiliser les modes "rebouclage réseau" et "essai de continuité réseau".
* Suppression des procédures et spécification pour les NAS (qui vont à la place être fournies comme paquetage).
* Suppression des procédures et spécification pour ATM (qui vont à la place être fournies comme paquetage).
* Ajout du paramètre facultatif manquant NotifiedEntity à la commande MGCP DeleteConnection (provenant de l'agent d'appel).
* Ajout d'un nouveau code facultatif d'informations demandées MaxMGCPDatagram pour AuditEndpoint pour permettre d'examiner la taille maximum des datagrammes MGCP pris en charge.
* Ajout d'un nouveau code facultatif d'informations demandées PackageList pour AuditEndpoint pour permettre l'examen des paquetages avec un numéro de version de paquetage. Le paramètre PackageList est aussi permis avec le code de retour 518 (Paquetage non pris en charge).
* Ajout des attributs manquants dans Capacités.
* Précision qu'à l'expiration d'un délai de redémarrage non zéro, un RestartInProgress mis à jour devrait être envoyé. Précisé aussi qu'une nouvelle NotifiedEntity peut seulement être retournée en réponse à une commande RestartInProgress.
* Ajout de la réponse Accusé de réception de réponse (code de retour 000) et inclusion dans la prise de contact en trois phases.
* Le paramètre ResponseAck a été changé pour être permis dans toutes les commandes.
* Ajout des codes de retour 101, 405, 406, 407, 409, 410, 503, 504, 505, 506, 507, 508, 509, 533, 534, 535, 536, 537, 538, 539, 540, 541, et définition des codes de retour dans la gamme 800-899 comme étant des codes de retour spécifiques du paquetage. Du texte supplémentaire est fourni pour quelques codes de retour et des détails supplémentaires sur la façon de traiter les codes de retour inconnus.
* Ajout des codes de cause 903, 904, 905 et définition des codes de cause 800-899 comme étant spécifiques du paquetage.
* Ajout d'un paragraphe précisant la procédure de négociation de codec.
* Précision que les paramètres de réservation de ressource dans une commande ModifyConnection reviennent pas défaut à la valeur courante utilisée.
* Précision que le mode de connexion est facultatif dans les commandes ModifyConnection.
* Il est corrigé que le LocalConnectionDescriptor soit facultatif dans une réponse à la commande CreateConnection (en cas de défaillance).
* Précision que les chaînes entre guillemets sont codées en UTF-8 et l'interchangeabilité des chaînes entre guillemets et des chaînes sans guillemets.
* Précision que les identifiants de transaction sont comparés comme des valeurs numériques.
* Précision de l'ordre des bits pour les options de connexion locale TypeOfService.
* Précision de l'utilisation de l'identifiant de demande zéro.
* Ajout de paragraphes d'exemples pour les commandes, réponses, et certains flux d'appels.
* Correction de l'usage et des exigences pour que SDP soit être strictement conforme à la RFC 2327.
* Ajout de l'exigence que toutes les mises en œuvre de MGCP prennent en charge les datagrammes MGCP jusqu'à au moins 4000 octets. Ajout aussi d'un nouveau paragraphe sur la taille maximum de datagramme, la fragmentation et le réassemblage.
* Généralisation du schéma de retransmission par portage pour déclarer seulement les exigences sous-jacentes à satisfaire.
* Précisions au paragraphe sur le calcul des temporisateurs de retransmission.
* Précision sur le fonctionnement des transactions de longue durée, incluant des réponses provisoires, des retransmissions et des échecs.
* Amélioration de la description des réponses provisoires et de l'interaction avec la prise de contact à trois phases.
* Amélioration de la description de la reprise sur défaillance et du rôle de "l'entité notifiée". Une "entité notifiée" vide a

été permise, bien que fortement déconseillée.
* Précision sur la procédure de retransmission et suppression des considérations de "mauvaise clé". Correction des incohérences entre les limites de retransmission de Max1 et Max2 et du diagramme de flux associé.
* Mise à jour de la résolution de nom de domaine pour que la procédure de retransmission entraîne moins de frais généraux quand plusieurs points d'extrémité retransmettent.
* Suppression de l'exigence de livraison dans l'ordre des réponses aux demandes de notification et commandes Notify. Les commandes Notify sont quand même livrées dans l'ordre.
* Précision que l'activation d'une demande de notification incorporée ne supprime pas la liste des ObservedEvents.
* Définition des interactions entre l'état Déconnecté et l'état Notification.
* Ajout d'un paragraphe sur la sémantique des transactions.
* Définition du comportement d'une passerelle quand plusieurs transactions interagissantes sont reçues.
* Fourniture de détails supplémentaires sur les états de service. Précision des relations entre les états de service de point d'extrémité, les méthodes de redémarrage, et le traitement associé des commandes.
* Précision sur le fonctionnement de la transition de la "procédure de redémarrage" à "l'état déconnecté".
* Il est permis aux commandes et réponses d'examen d'outrepasser les procédures "redémarrage" et "déconnecté".
* Précision sur le fonctionnement de la "procédure déconnecté" et en particulier le fonctionnement des messages portés RestartInProgress "déconnecté".
* Ajout de l'option d'agréger les messages RestartInProgress "déconnecté" dans certaines conditions pour réduire le volume de messages.
* Définition d'un comportement supplémentaire pour les points d'extrémité qui souhaitent envoyer des commandes quand elles sont dans l'état "déconnecté".
* Ajout d'un nouveau paragraphe sur le contrôle de charge en général qui inclut deux nouveaux codes d'erreur (101 et 409) pour traiter la surcharge.
* Suppression de la commande "ProposedMoveConnection".
* Suppression des paquetages de la spécification du protocole (ils seront fournis dans des documents distincts).
* Le concept de paquetage est formellement étendu pour être le principal mécanisme d'extension permettant maintenant de définir des extensions aussi dans les paquetages :
  - Informations de support
  - Options locales de connexion
  - Paramètres d'extension
  - Modes de connexion
  - Actions
  - Lettres de script de numérotation
  - Paramètres de connexion
  - Méthodes de redémarrage
  - Codes de cause
  - Codes de retour
* Ajout des exigences et du format suggéré pour les définitions de paquetage.
* Défini que les événements "opération achevée" et "échec d'opération" sont automatiquement présents dans les paquetages avec des signaux Time-Out.
* Suppression de la liste des différences qu'il y avait avant la RFC 2705.
* Ajout du paquetage de base pour traiter le débordement de mémoire tampon de quarantaine, d'événements observés, d'échec de demande de notification incorporée, et pour permettre que des événements soient demandés de façon persistante. Une nouvelle commande "Message" est aussi incluse.
* Ajout des procédures d'enregistrement par l'IANA pour les paquetages et autres extensions.
* Mise à jour de la grammaire pour corriger des erreurs connues et prendre en charge de nouvelles extensions de façon rétro compatible. Ajout des nouvelles extensions (facultatives) PackageList et MaxMGCPDatagram pour l'audit Changement des règles d'espaces blanches explicites dans certaines productions pour rendre la grammaire plus cohérente.
* Ajout du tableau d'interaction des modes de connexion.
* Ajout de détails supplémentaires sur les conventions de désignation de point d'extrémité virtuel. Et aussi d'une convention de point d'extrémité passerelle suggérée et une option "Range Wildcard" pour les conventions de désignation de point d'extrémité.

8. Considérations sur la sécurité

Les questions de sécurité sont discutées à la Section 5.

9. Remerciements

Des remerciement particuliers sont dus aux auteurs de la spécification MGCP 1.0 originale : Mauricio Arango, Andrew Dugan, Isaac Elliott, Christian Huitema, et Scott Picket.

Nous voulons aussi remercier les nombreux relecteurs qui ont fourni des commentaires sur la conception de SGCP et ensuite MGCP, notamment Sankar Ardhanari, Francois Berard, David Auerbach, Bob Biskner, David Bukovinsky, Charles Eckel, Mario Edini, Ed Guy, Barry Hoffner, Jerry Kamitses, Oren Kudevitzki, Rajesh Kumar, Troy Morley, Dave Oran, Jeff Orwick, John Pickens, Lou Rubin, Chip Sharp, Paul Sijben, Kurt Steinbrenner, Joe Stone, et Stuart Wray.

La version 0.1 de MGCP était largement inspirée par le "Contrôle d'appareil de protocole Internet" (IPDC) conçu par le comité consultatif technique établi par le groupe "Level 3 Communications". Des parties de texte entières ont été empruntées au protocole de contrôle de connexion IP, au protocole de contrôle de support IP, et à la gestion d'appareil IP. Les auteurs souhaitent reconnaître les contributions à ces protocoles faites par Ilya Akramovich, Bob Bell, Dan Brendes, Peter Chung, John Clark, Russ Dehlinger, Andrew Dugan, Isaac Elliott, Cary FitzGerald, Jan Gronski, Tom Hess, Geoff Jordan, Tony Lam, Shawn Lewis, Dave Mazik, Alan Mikhak, Pete O'Connell, Scott Pickett, Shyamal Prasad, Eric Presworsky, Paul Richards, Dale Skran, Louise Spergel, David Sprague, Raj Srinivasan, Tom Taylor et Michael Thomas.

10. Références

[Allman] Allman, M., Paxson, V. "On Estimating End-to-End Network Path Properties", Proc. SIGCOMM'99, 1999.

[H.225] Recommandation UIT-T H.225, "Protocoles de signalisation d'appel et mise en paquets de flux de supports pour les systèmes de communications multimédia fondés sur le paquet".

[H.245] Recommandation UIT-T H.245, "Protocole de contrôle pour communications multimédia", mai 2006.

[H.323] Recommandation UIT-T H.323v3, "Systèmes de communication multimédia fondés sur le paquet", UIT, Genève, septembre 1999.

[LSSGR] Bellcore, "LSSGR: Switching System Generic Requirements for Call Control Using the Integrated Services Digital Network User Part (ISDNUP)", GR-317-CORE, Issue 2, décembre 1997.

[Q.761] Recommandation UIT-T Q.761, "Description fonctionnelle du sous système utilisateur RNIS du système de signalisation n° 7", (Malaga-Torremolinos, 1984 ; modifiée à Helsinki, 1993).

[Q.762] Recommandation UIT-T Q.762, "Fonction générale des messages et signaux du sous système utilisateur RNIS du système de signalisation n° 7", (Malaga-Torremolinos, 1984 ; modifiée à Helsinki, 1993).

[RFC1122] R. Braden, "Exigences pour les hôtes Internet – couches de communication", STD 3, octobre 1989. *(MàJ par* RFC6633, 8029)

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson, "RTP : protocole de transport pour applications en temps réel", janvier 1996. *(Obsolète, voir* RFC3550 STD64)

[RFC1890] H. Schulzrinne, "Profil RTP pour conférences audio et vidéo avec contrôle minimal", janvier 1996. (*Obsolète, voir* RFC3551) *(P.S.)*

[RFC2026] S. Bradner, "Le processus de normalisation de l'Internet -- Révision 3", (BCP0009) octobre 1996. *(Remplace* RFC1602, RFC1871) (*MàJ par* RFC3667, 3668, 3932, 3979, 3978, 5378, 6410, 8179, 8789)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir* RFC5234)

[RFC2279] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", janvier 1998. (*Obsolète, voir*

RFC3629) *(D.S.)*

[RFC2326] H. Schulzrinne, A. Rao et R. Lanphier, "Protocole de flux directs en temps réel (RTSP)", avril 1998. *(Remplacée par* RFC7826)

[RFC2327] M. Handley et V. Jacobson, "SDP : Protocole de description de session", avril 1998. *(Obsolète; voir* RFC4566)

[RFC2401] S. Kent et R. Atkinson, "Architecture de sécurité pour le protocole Internet", novembre 1998. (*Obsolète, voir* RFC4301)

[RFC2402] S. Kent et R. Atkinson, "En-tête d'authentification IP", novembre 1998. (*Obsolète, voir* RFC4302, 4305)

[RFC2406] S. Kent et R. Atkinson, "Encapsulation de charge utile de sécurité IP (ESP)", novembre 1998. *(Obsolète, voir* RFC4303)

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Obsolète, voir la* RFC5226)

[RFC2848] S. Petrack, L. Conroy, "Protocole de service PINT : extensions à SIP et SDP pour l'accès IP aux services de téléphone", juin 2000. *(P.S.)*

[RFC2974] M. Handley, C. Perkins, E. Whelan, "Protocole d'annonce de session (SAP)", octobre 2000. *(Expérimentale)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par les RFC* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[TCP/IP] Stevens, W. Richard, "TCP/IP Illustrated, Volume 1, The Protocols", Addison-Wesley, 1994.

Appendice A. Description de la syntaxe formelle du protocole

Dans cette Section, on donne une description formelle de la syntaxe du protocole, suivant le "BNF augmenté pour les spécifications de syntaxe" défini dans la RFC 2234. La syntaxe utilise le cœur des règles définies au paragraphe 6.1 de la RFC 2234, qui ne sont pas incluses ici. De plus, la syntaxe suit les règles de sensibilité à la casse de la RFC 2234, c'est-à-dire, MGCP est insensible à la casse (mais SDP ne l'est pas). On devrait noter que l'ABNF ne fournit pas de spécification implicite des espaces blanches linéaires et les messages MGCP DOIVENT donc suivre les règles d'espace blanche linéaire explicites fournies dans la grammaire ci-dessous. Cependant, en accord avec les principes généraux de robustesse, les mises en œuvre sont fortement encouragées à tolérer des espaces blanches linéaires additionnelles dans les messages reçus.

```
MGCPMessage = MGCPCommand / MGCPResponse

MGCPCommand = MGCPCommandLine 0*(MGCPParameter) [EOL *SDPinformation]

MGCPCommandLine = MGCPVerb 1*(WSP)transaction-id 1*(WSP)endpointName 1*(WSP) MGCPversion EOL

MGCPVerb = "EPCF" / "CRCX" / "MDCX" / "DLCX" / "RQNT" / "NTFY" / "AUEP" / "AUCX" / "RSIP" /
           extensionVerb

extensionVerb  = ALPHA 3(ALPHA / CHIFFRE)          ; expérimental commence par X

transaction-id = 1*9(CHIFFRE)

endpointName = LocalEndpointName "@" DomainName
LocalEndpointName = LocalNamePart 0*("/" LocalNamePart)
LocalNamePart = AnyName / AllName / NameString
AnyName = "$"
AllName = "*"
NameString = 1*(gamme-de-caractères-permis)                ; VCHAR sauf "$", "*", "/", "@"
gamme-de-caractères-permis = %x21-23 / %x25-29 / %x2B-2E / %x30-3F / %x41-7E
```

```
DomainName = 1*255(ALPHA / CHIFFRE / "." / "-")            ; comme défini dans la RFC 821
                    / "#" number / "[" IPv4address / IPv6address "]"       ; voir la RFC 2373

; Réécrit dans l'ABNF de la RFC 821
number =1*CHIFFRE

; de la RFC 2373
IPv6address = hexpart [ ":" IPv4address ]
IPv4address = 1*3CHIFFRES "." 1*3CHIFFRES "." 1*3CHIFFRES "." 1*3CHIFFRES

; cette production, bien que apparaissant dans la RFC2373, n'est pas référencée
IPv6prefix = hexpart "/" 1*2CHIFFRES
hexpart = hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
hexseq  = hex4 *( ":" hex4)
hex4 = 1*4HEXDIG

MGCPversion = "MGCP" 1*(WSP) 1*(CHIFFRE) "." 1*(CHIFFRE) [1*(WSP) ProfileName]
ProfileName = VCHAR *( WSP / VCHAR)

MGCPParameter = ParameterValue EOL

; Vérifier infoCode si plus de valeurs de paramètre sont définies
; La plupart des valeurs facultatives peuvent seulement être omises dans un audit.
ParameterValue = ("K"  ":" 0*(WSP)  [ResponseAck])
            / ("B"  ":" 0*(WSP)  [BearerInformation])
            / ("C"  ":" 0*(WSP)  CallId)
            / ("I"  ":" 0*(WSP)  [ConnectionId])
            / ("N"  ":" 0*(WSP)  [NotifiedEntity])
            / ("X"  ":" 0*(WSP)  [RequestIdentifier])
            / ("L"  ":" 0*(WSP)  [LocalConnectionOptions])
            / ("M"  ":" 0*(WSP)  ConnectionMode)
            / ("R"  ":" 0*(WSP)  [RequestedEvents])
            / ("S"  ":" 0*(WSP)  [SignalRequests])
            / ("D"  ":" 0*(WSP)  [DigitMap])
            / ("O"  ":" 0*(WSP)  [ObservedEvents])
            / ("P"  ":" 0*(WSP)  [ConnectionParameters])
            / ("E"  ":" 0*(WSP)  ReasonCode)
            / ("Z"  ":" 0*(WSP)  [SpecificEndpointID])
            / ("Z2" ":" 0*(WSP)  SecondEndpointID)
            / ("I2" ":" 0*(WSP)  SecondConnectionID)
            / ("F"  ":" 0*(WSP)  [RequestedInfo])
            / ("Q"  ":" 0*(WSP)  QuarantineHandling)
            / ("T"  ":" 0*(WSP)  [DetectEvents])
            / ("RM" ":" 0*(WSP)  RestartMethod)
            / ("RD" ":" 0*(WSP)  RestartDelay)
            / ("A"  ":" 0*(WSP)  [Capabilities])
            / ("ES" ":" 0*(WSP)  [EventStates])
            / ("PL" ":" 0*(WSP)  [PackageList])                    ; seulement en audit.
            / ("MD" ":" 0*(WSP)  MaxMGCPDatagram)                  ; seulement en audit.
            / (extensionParameter ":" 0*(WSP) [parameterString])

; Une réponse finale peut inclure un ResponseAck vide.
ResponseAck = confirmedTransactionIdRange *( ","  0*(WSP) confirmedTransactionIdRange )

confirmedTransactionIdRange = transaction-id ["-" transaction-id]

BearerInformation = BearerAttribute 0*("," 0*(WSP) BearerAttribute)
BearerAttribute = ("e" ":" BearerEncoding) / (BearerExtensionName [":" BearerExtensionValue])
BearerExtensionName = PackageLCOExtensionName
BearerExtensionValue = LocalOptionExtensionValue
```

```
BearerEncoding = "A" / "mu"

CallId = 1*32(HEXDIG)

; La réponse à une demande d'audit peut inclure une liste d'identifiants.
ConnectionId = 1*32(HEXDIG) 0*("," 0*(WSP) 1*32(HEXDIG))
SecondConnectionID = ConnectionId

NotifiedEntity = [LocalName "@"] DomainName [":" portNumber]
LocalName  = LocalEndpointName                    ; Pas de structure interne

portNumber = 1*5(CHIFFRE)

RequestIdentifier = 1*32(HEXDIG)

LocalConnectionOptions = LocalOptionValue 0*(WSP) 0*("," 0*(WSP) LocalOptionValue 0*(WSP))
LocalOptionValue = ("p"  ":" packetizationPeriod)
        / ("a"  ":" compressionAlgorithm)
        / ("b"  ":" bande passante)
        / ("e"  ":" echoCancellation)
        / ("gc" ":" gainControl)
        / ("s"  ":" silenceSuppression)
        / ("t"  ":" typeOfService)
        / ("r"  ":" resourceReservation)
        / ("k"  ":" encryptiondata)
        / ("nt" ":" ( typeOfNetwork / supportedTypeOfNetwork))
        / (LocalOptionExtensionName [":" LocalOptionExtensionValue])

Capabilities =  CapabilityValue 0*(WSP) 0*("," 0*(WSP) CapabilityValue 0*(WSP))
CapabilityValue = LocalOptionValue / ("v" ":" supportedPackages) / ("m" ":" supportedModes)

PackageList = pkgNameAndVers 0*("," pkgNameAndVers)
pkgNameAndVers  = paquetageName ":" paquetageVersion
paquetageVersion = 1*(CHIFFRE)

packetizationPeriod  = 1*4(CHIFFRE) ["-" 1*4(CHIFFRE)]
compressionAlgorithm = algorithmName 0*(";" algorithmName)

algorithmName = 1*(SuitableLCOCharacter)
bande passante = 1*4(CHIFFRE) ["-" 1*4(CHIFFRE)]
echoCancellation = "on" / "off"
gainControl = "auto" / ["-"] 1*4(CHIFFRE)
silenceSuppression = "on" / "off"
typeOfService = 1*2(HEXDIG)                       ; 1 hex seulement pour les capacités.
resourceReservation  = "g" / "cl" / "be"

; les paramètres de chiffrement sont codés comme dans SDP (RFC 2327)
; Note : la clé de chiffrement peut contenir un algorithme comme spécifié dans la RFC 1890
encryptiondata = ( "clear" ":" encryptionKey ) / ( "base64" ":" encodedEncryptionKey )
        / ( "uri" ":" URItoObtainKey ) / ( "prompt" )      ; défini dans SDP, non utilisable dans MGCP !

encryptionKey = 1*(SuitableLCOCharacter) / quotedString    ; Voir la RFC 2045
encodedEncryptionKey = 1*(ALPHA / CHIFFRE / "+" / "/" / "=")
URItoObtainKey = 1*(SuitableLCOCharacter) / quotedString

typeOfNetwork = "IN" / "ATM" / "LOCAL" / OtherTypeOfNetwork   ; Enregistré par l'IANA - voir la RFC 2327
OtherTypeOfNetwork = 1*(SuitableLCOCharacter)
supportedTypeOfNetwork = typeOfNetwork *(";" typeOfNetwork)

supportedModes = ConnectionMode 0*(";" ConnectionMode)
```

```
supportedPackages = packageName 0*(";" packageName)

packageName = 1*(ALPHA / CHIFFRE / HYPHEN)            ; Hyphen ni premier ni dernier.

LocalOptionExtensionName = VendorLCOExtensionName / PackageLCOExtensionName / OtherLCOExtensionName
VendorLCOExtensionName = "x" ("+"/"-") 1*32(SuitableExtLCOCharacter)
PackageLCOExtensionName = packageName "/" 1*32(SuitablePkgExtLCOCharacter)
; Ne doit pas commencer par "x-" ou "x+".
OtherLCOExtensionName = 1*32(SuitableExtLCOCharacter)

LocalOptionExtensionValue = (1*(SuitableExtLCOValChar) / quotedString)
                        *(";" (1*(SuitableExtLCOValChar) / quotedString))

; Note : Pas de mode "data".
ConnectionMode = "sendonly" / "recvonly" / "sendrecv" / "confrnce" / "inactive" / "loopback"
                         / "conttest" / "netwloop" / "netwtest" / ExtensionConnectionMode
ExtensionConnectionMode = PkgExtConnectionMode
PkgExtConnectionMode = packageName "/" 1*(ALPHA / CHIFFRE)

RequestedEvents = requestedEvent 0*("," 0*(WSP) requestedEvent)
requestedEvent = (eventName ["(" requestedActions ")"]) / (eventName "(" requestedActions ")" "(" eventParameters ")" )
eventName = [(packageName / "*") "/"] (eventId / "all" / eventRange / "*" / "#")            ; pour le DTMF
                ["@" (ConnectionId / "$" / "*")]
eventId = 1*(ALPHA / CHIFFRE / HYPHEN)                    ; Hyphen ni premier ni dernier.
eventRange = "[" 1*(DigitMapLetter / (CHIFFRE "-" CHIFFRE) / (DTMFLetter "-" DTMFLetter)) "]"
DTMFLetter = "A" / "B" / "C" / "D"

requestedActions = requestedAction 0*("," 0*(WSP) requestedAction)
requestedAction = "N" / "A" / "D" / "S" / "I" / "K" / "E" "(" EmbeddedRequest ")" / ExtensionAction
ExtensionAction = PackageExtAction
PackageExtAction = packageName "/" Action ["(" ActionParameters ")"]
Action = 1*ALPHA
ActionParameters = eventParameters                                ; Peut contenir des actions.

; Note : Devrait tolérer un ordre différent pour la réception, par exemple, de NCS.
EmbeddedRequest = (     "R" "(" EmbeddedRequestList ")"
                ["," 0*(WSP) "S" "(" EmbeddedSignalRequest ")"]
                ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"]      )
            / (     "S" "(" EmbeddedSignalRequest ")" ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )
            / (     "D" "(" EmbeddedDigitMap ")" )

EmbeddedRequestList = RequestedEvents
EmbeddedSignalRequest = SignalRequests
EmbeddedDigitMap = DigitMap

SignalRequests = SignalRequest 0*("," 0*(WSP) SignalRequest )
SignalRequest = eventName [ "(" eventParameters ")" ]

eventParameters = eventParameter 0*("," 0*(WSP) eventParameter)
eventParameter = eventParameterValue / eventParameterName "=" eventParameter
                        / eventParameterName "(" eventParameters ")"
eventParameterString = 1*(SuitableEventParamCharacter)
eventParameterName = eventParameterString
eventParameterValue = eventParameterString / quotedString

DigitMap = DigitString  / "(" DigitStringList ")"
DigitStringList = DigitString 0*( "|" DigitString )
DigitString = 1*(DigitStringElement)
DigitStringElement = DigitPosition ["."]
DigitPosition = DigitMapLetter / DigitMapRange
; Note:  "X" est maintenant inclus.
```

```
DigitMapLetter  = CHIFFRE / "#" / "*" / "A" / "B" / "C" / "D" / "T" / "X" / ExtensionDigitMapLetter
ExtensionDigitMapLetter = "E" / "F" / "G" / "H" / "I" / "J" / "K" / "L" / "M" / "N" / "O" / "P" / "Q" / "R"
                          / "S" / "U" / "V" / "W" / "Y" / "Z"
; Note :"[x]"  est maintenant inclus
DigitMapRange = "[" 1*DigitLetter "]"
DigitLetter = *((CHIFFRE "-" CHIFFRE) / DigitMapLetter)

ObservedEvents = SignalRequests

EventStates = SignalRequests

ConnectionParameters = ConnectionParameter 0*( "," 0*(WSP) ConnectionParameter )

ConnectionParameter  = ( "PS" "=" packetsSent )
                / ( "OS" "=" octetsSent )
                / ( "PR" "=" packetsReceived )
                / ( "OR" "=" octetsReceived )
                / ( "PL" "=" packetsLost )
                / ( "JI" "=" jitter )
                / ( "LA" "=" averageLatency )
                / ( ConnectionParameterExtensionName "=" ConnectionParameterExtensionValue )
packetsSent  = 1*9(CHIFFRE)
octetsSent = 1*9(CHIFFRE)
packetsReceived = 1*9(CHIFFRE)
octetsReceived = 1*9(CHIFFRE)
packetsLost = 1*9(CHIFFRE)
jitter = 1*9(CHIFFRE)
averageLatency = 1*9(CHIFFRE)

ConnectionParameterExtensionName = VendorCPExtensionName / PackageCPExtensionName
VendorCPExtensionName = "X" "-" 2*ALPHA
PackageCPExtensionName = packageName "/" CPName
CPName = 1*(ALPHA / CHIFFRE / HYPHEN)
ConnectionParameterExtensionValue = 1*9(CHIFFRE)

MaxMGCPDatagram = 1*9(CHIFFRE)

ReasonCode = 3DIGIT
        [1*(WSP) "/" packageName]          ; Seulement pour 8xx.
        [WSP 1*(%x20-7E)]

SpecificEndpointID = endpointName
SecondEndpointID = endpointName

RequestedInfo = infoCode 0*("," 0*(WSP) infoCode)

infoCode = "B" / "C" / "I" / "N" / "X" / "L" / "M" / "R" / "S" / "D" / "O" / "P" / "E" / "Z" / "Q" / "T" / "RC" / "LC"
      / "A" / "ES" / "RM" / "RD" / "PL" / "MD" / extensionParameter

QuarantineHandling = loopControl / processControl / (loopControl "," 0*(WSP) processControl )
loopControl  = "step" / "loop"
processControl = "process" / "discard"

DetectEvents = SignalRequests

RestartMethod = "graceful" / "forced" / "restart" / "deconnected" / "cancel-graceful" / extensionRestartMethod
extensionRestartMethod = PackageExtensionRM
PackageExtensionRM = packageName "/" 1*32(ALPHA / CHIFFRE / HYPHEN)
RestartDelay = 1*6(CHIFFRE)

extensionParameter = VendorExtensionParameter / PackageExtensionParameter / OtherExtensionParameter
```

```
VendorExtensionParameter  = "X" ("-"/"+") 1*6(ALPHA / CHIFFRE)
PackageExtensionParameter = packageName "/" 1*32(ALPHA / CHIFFRE / HYPHEN)
                                                    ; ne doit pas commencer par "x-" ou x+"
OtherExtensionParameter = 1*32(ALPHA / CHIFFRE / HYPHEN)

; Si le premier caractère est un guillemet, c'est une chaîne entre guillemets.
parameterString = (%x21 / %x23-7F) *(%x20-7F)    ; le premier et le dernier ne doivent pas être une espace blanche.
                  / quotedString

MGCPResponse = MGCPResponseLine 0*(MGCPParameter) *2(EOL *SDPinformation)

MGCPResponseLine = responseCode 1*(WSP) transaction-id
                           [1*(WSP) "/" packageName]          ; Seulement pour 8xx.
                           [WSP responseString] EOL

responseCode = 3DIGIT
responseString = *(%x20-7E)

SuitablePkgExtLCOCharacter = SuitableLCOCharacter

SuitableExtLCOCharacter = CHIFFRE / ALPHA / "+" / "-" / "_" / "&" / "!" / "'" / "|" / "=" / "#" / "?"
                                / "." / "$" / "*" / "@" / "[" / "]" / "^" / "`" / "{" / "}" / "~"

SuitableLCOCharacter = SuitableExtLCOCharacter / "/"

SuitableExtLCOValChar = SuitableLCOCharacter / ":"

; VCHAR sauf """, "(", ")", ",", et "=".
SuitableEventParamCharacter = %x21 / %x23-27 / %x2A-2B / %x2D-3C / %x3E-7E

; Note : codé en UTF8.
quotedString = DQUOTE 0*(quoteEscape / quoteChar) DQUOTE
quoteEscape = DQUOTE DQUOTE
quoteChar = (%x00-21 / %x23-FF)

EOL = CRLF / LF

HYPHEN = "-"

; Voir dans la RFC 2327 la grammaire SDP appropriée à la place.
SDPinformation = SDPLine CRLF *(SDPLine CRLF)          ; voir la RFC 2327 pour la définition appropriée.
SDPLine = 1*(%x01-09 / %x0B / %x0C / %x0E-FF)
```

Appendice B. Paquetage de base

Nom de paquetage : B
Version : 0

La spécification MGCP définit un paquetage de base qui contient un ensemble de paramètres d'événements et d'extensions qui sont d'utilisation générale pour le protocole. Bien que ce ne soit pas exigé, il est fortement RECOMMANDÉ de prendre en charge ce paquetage car il fournit des fonctions importantes pour le protocole de base.

B.1 Événements

Le tableau ci-dessous donne la liste des événements :

Symbole	Définition	R	S	Durée
enf(##)	défaillance RQNT incorporée	x		
oef	événements observés pleins	x		
qbo	débordement de mémoire tampon de quarantaine	x		

Les événements sont définis comme suit :

Défaillance de demande de notification incorporée (enf) : l'événement Défaillance de demande de notification incorporée (enf) est généré quand une défaillance de demande de notification incorporée survient. Quand l'événement est demandé, il devrait faire partie de la demande de notification incorporée elle-même. Quand l'événement est rapporté, il peut être paramétré avec un code d'erreur (voir le paragraphe 2.4) qui détaille l'erreur qui s'est produite. Quand il est demandé, il ne peut pas être paramétré.

Événements observés pleins (oef) : l'événement est généré quand le point d'extrémité est incapable d'accumuler plus d'événements dans la liste des ObservedEvents. Si cet événement se produit, et si il n'est pas utilisé pour déclencher un Notify, les événements suivants qui auraient dû avoir été ajoutés à la liste vont être perdus.

Débordement de mémoire tampon de quarantaine (qbo) : l'événement est généré quand la mémoire tampon de quarantaine déborde et qu'un ou plusieurs événements ont été perdus.

B.2 Paramètres d'extension

B.2.1 PersistentEvents

PersistentEvents : liste des événements qu'il est demandé à la passerelle de détecter et rapporter de façon persistante. Le paramètre est facultatif mais peut être fourni dans toute commande où le paramètre DetectEvents peut être fourni. La valeur initiale par défaut du paramètre est vide. Quand le paramètre est omis d'une commande, il garde sa valeur courante. Quand le paramètre est fourni, il remplace complètement la valeur courante. Fournir un événement dans cette liste est similaire (mais préférable) à définir cet événement particulier comme persistant. La liste courante des PersistentEvents va implicitement s'appliquer aux demandes de notification courantes aussi bien que suivantes, cependant aucune détection de double prise, etc. ne va être effectuée (similaire à DetectEvents). Si un événement fourni dans cette liste est inclus dans une liste de RequestedEvents, les paramètres d'action et d'événement utilisés dans les RequestedEvents vont remplacer les paramètres d'action et d'événement associés aux événement dans la liste des PersistentEvents pour la durée de vie de la liste des RequestedEvents, après quoi les paramètres d'action et d'événement de PersistentEvents sont restaurés. Les événements avec des états d'événement demandés par ce paramètre vont être inclus dans la liste des EventStates en cas d'audit.

PersistentEvents peut aussi être utilisé pour détecter des événements sur les connexions. L'utilisation du caractère générique "all connexions" est directe, tandis que l'utilisation de PersistentEvents avec une ou plusieurs connexions spécifiques doit être considéré avec précaution. Une fois que la connexion en question est supprimée, une demande de notification suivante sans une nouvelle valeur de PersistentEvents va échouer (le code d'erreur 515 - Identifiant de connexion incorrect, est RECOMMANDÉ) car elle se réfère implicitement à la connexion supprimée.

Le paramètre génère les codes d'erreur pertinents à partir du protocole de base, par exemple, le code d'erreur 512 si un événement inconnu est spécifié.

Le paramètre PersistentEvents peut être examiné, et dans ce cas, il va retourner sa valeur en cours. L'examen d'un RequestedEvents n'est pas affecté par cette extension, c'est-à-dire, les événements spécifiés dans cette liste ne sont pas automatiquement rapportés quand les RequestedEvents sont examinés.

Le nom de paramètre pour PersistentEvents est "PR" et il est défini par la production :

```
PersistentEvents = "PR" ":" 0*WSP  RequestedEvents]
```

L'exemple suivant illustre l'utilisation du paramètre :

```
B/PR: L/hd(N), L/hf(N), L/hu(N), B/enf, B/oef, B/qbo
```

qui donne pour instruction au point d'extrémité de détecter de façon persistante et rapporter les événements "décroché", "impulsion crochet", et "raccroché". Il dit aussi au point d'extrémité de détecter de façon persistante et de rapporter les échecs de demande de notification incorporée, les événements observés pleins, et le débordement de mémoire tampon de

quarantaine.

B.2.2 NotificationState

NotificationState est un paramètre RequestedInfo qui peut être examiné avec la commande AuditEndpoint. Il peut être utilisé pour déterminer si le point d'extrémité est ou non dans l'état Notification.

Le paramètre est interdit dans toute commande. Dans les réponses, il n'est un paramètre valide de réponse que pour AuditEndpoint.

Il est défini par la grammaire suivante :

NotificationState = "NS" ":" 0*WSP NotificationStateValue
NotificationStateValue = "ns" / "ls" / "o"

Il est demandé au titre de l'audit en incluant le code de paramètre dans RequestedInfo, comme dans :

F: B/NS

Le paramètre de réponse va contenir la valeur "ns" si le point d'extrémité est dans l'état "Notification", la valeur "ls" si le point d'extrémité est dans l'état "verrrouillé" (c'est-à-dire, en attente d'une RQNT après qu'une réponse à un NTFY a été créée quand on fonctionne en mode "step") ou la valeur "o" autrement, comme par exemple :

B/NS: ns

B.3 Verbes

Les paquetages MGCP ne sont pas destinés à définir de nouvelles commandes, cependant une exception est faite dans ce cas afin d'ajouter une capacité générale importante qui manque actuellement, à savoir la capacité pour la passerelle d'envoyer un message générique à l'agent d'appel.

La définition de la nouvelle commande est :

ReturnCode
<-- Message(EndpointId
[, ...])

EndpointId est le nom pour le ou les points d'extrémité dans la passerelle qui produisent la commande Message. L'identifiant DOIT être un identifiant de point d'extrémité pleinement qualifié, incluant le nom de domaine de la passerelle. La partie locale du nom de point d'extrémité NE DOIT PAS utiliser le caractère générique "any of".

Le seul paramètre spécifié dans la définition de la commande Message est le EndpointId, cependant, il est envisagé que des extensions définiront des paramètres supplémentaires à utiliser avec la commande Message. Ces extensions NE DOIVENT PAS altérer ou autrement interférer avec le fonctionnement normal du protocole MGCP de base. Elles peuvent cependant définir des capacités supplémentaires au dessus et au delà de celles fournies par le protocole MGCP de base. Par exemple, une extension pour permettre à la passerelle d'examiner les paquetages pris en charge par l'agent d'appel pourrait être définie, tandis que l'utilisation de la commande Message comme un moyen de remplacement pour rapporter les événements observés serait illégal, car cela altèrerait le comportement normal du protocole MGCP.

Afin de ne pas interférer avec le fonctionnement normal de MGCP, l'absence de réponse à la commande Message NE DOIT PAS conduire le point d'extrémité à devenir déconnecté. Le ou les points d'extrémité DOIVENT être prêts à traiter cela de façon transparente et à continuer le traitement normal sans qu'il en soit affecté.

Si le ou les points d'extrémité reçoivent une réponse indiquant que l'agent d'appel ne prend pas en charge la commande Message, le ou les points d'extrémité NE DOIVENT PAS renvoyer une commande Message jusqu'à ce que "l'entité notifiée" en cours ait changé. De même, si le ou les points d'extrémité reçoivent une réponse indiquant que l'agent d'appel ne prend pas en charge un ou plusieurs paramètres dans la commande Message, le ou les points d'extrémité NE DOIVENT PAS envoyer à nouveau une commande Message avec ces paramètres jusqu'à ce que "l'entité notifiée" en cours ait changé.

La commande Message est codée par MESG, comme montré dans l'exemple suivant :

MESG 1200 aaln/1@rgw.whatever.net MGCP 1.0

Appendice C. Considérations relatives à l'IANA

C.1 Nouveau sous registre de paquetage MGCP

L'IANA a établi un nouveau sous-registre pour les paquetages MGCP à http://www.iana.org/assignments/mgcp-packages.

Les paquetages peuvent être enregistrés auprès de l'IANA selon les procédures suivantes :

Le paquetage DOIT avoir un nom de chaîne unique qui NE DOIT PAS commencer par les deux caractères "x-" ou "x+".

Les titre, nom, et version de paquetage (zéro supposé par défaut) DOIVENT être enregistrés par l'IANA ainsi qu'une référence au document qui décrit le paquetage. Le document DOIT avoir un URL stable et DOIT être contenu sur un serveur public de la Toile.

Les paquetages peuvent définir une ou plusieurs Extensions Lettres de script de numérotation, cependant elles sont prises dans un espace de noms limité et plat. Pour empêcher les conflits de noms, l'IANA NE DEVRA PAS enregistrer un paquetage qui définit une extension Lettre de script de numérotation déjà définie dans un autre paquetage enregistré par l'IANA. Pour faciliter cette tâche, de tels paquetages DEVRONT contenir la ligne "Extension Lettres de script de numérotation: " suivie par une liste des extensions Lettres de script de numérotation définies dans le paquetage au début de la définition du paquetage.

Un nom de contact, une adresse de messagerie et postale DOIVENT être fournis pour le paquetage. Les informations de contact DEVRONT être mises à jour par l'organisation qui les définit en tant que de besoin.

Finalement, avant d'enregistrer un paquetage, l'IANA DOIT avoir une relecture du paquetage par un expert désigné [RFC2434]. L'expert réviseur enverra un message à l'IANA sur la détermination globale de la révision.

C.2 Nouveau paquetage MGCP

Le présent document définit un nouveau paquetage de base MGCP dans l'Appendice B, qui a été enregistré par l'IANA.

C.3 Nouveau sous registre de LocalConnectionOptions MGCP

L'IANA a établi un nouveau sous registre pour les LocalConnectionOptions MGCP à http://www.iana.org/assignments/mgcp-localconnectionoptions

Les paquetages sont le mécanisme d'extension préféré, cependant pour la rétro compatibilité, les options de connexion locale au-delà de celles fournies dans la présente spécification peuvent être enregistrées auprès de l'IANA. Chacune de ces options de connexion locale DOIT avoir un nom de chaîne unique qui NE DOIT PAS commencer par "x-" ou "x+". Le nom du champ d'option de connexion locale et le nom du codage DOIVENT aussi être enregistrés auprès de l'IANA ainsi qu'une référence au document qui décrit une option de connexion locale. Le document DOIT avoir un URL stable et DOIT être contenu sur un serveur public de la Toile.

Un nom de contact, une adresse de messagerie et postale DOIVENT être fournis pour l'option de connexion locale. Les informations de contact DEVRONT être mises à jour par l'organisation qui les définit en tant que de besoin.

Finalement, avant d'enregistrer une LocalConnectionOption, l'IANA DOIT avoir une relecture de l'option de connexion locale par un expert désigné [RFC2434]. L'expert réviseur enverra un message à l'IANA sur la détermination globale de la révision.

Appendice D. Interactions de mode

Un point d'extrémité MGCP peut établir un ou plusieurs flux de supports. Ces flux sont entrants (d'un point d'extrémité distant) ou sortants (générés au microphone du combiné). Le paramètre "mode de connexion" établit la direction et la génération de ces flux. Quand il y a seulement une connexion à un point d'extrémité, la transposition de ces flux est directe ; le combiné exécute le flux entant sur le combiné du locuteur et génère le flux sortant à partir du signal du microphone du combiné, selon le paramètre de mode.

Cependant, quand plusieurs connexions sont établies avec un point d'extrémité, il peut y avoir de nombreux flux entrants et sortants. Selon le mode de connexion utilisé, ces flux peuvent interagir différemment les uns avec les autres et avec les flux entrants et sortants du combiné.

Le tableau ci-dessous décrit comment les différentes connexions DEVRONT être mixées quand une ou plusieurs connexions sont concurremment "actives". Une connexion active est définie ici comme une connexion qui est dans un des modes suivants :

* "send/receive" *(envoi/réception)*
* "send only" *(envoi seul)*
* "receive only" *(réception seule)*
* "conference" *(conférence)*

Les connexions dans les modes "bouclage arrière réseau", "essai de continuité réseau", ou "inactif" ne sont pas affectées par les connexions en mode "actif". Le tableau utilise les conventions suivantes :

* Ai est le flux de supports entrant provenant de Connexion A
* Bi est le flux de supports entrant provenant de Connexion B
* Hi est le flux de supports entrant provenant du microphone du combiné
* Ao est le flux de supports sortant vers la Connexion A
* Bo est le flux de supports sortant vers la Connexon B
* Ho est le flux de supports sortant vers l'écouteur du combiné
* NA n'indique aucun flux de quelque sorte que ce soit (supposant qu'il n'y a pas de signal appliqué sur la connexion)

"netw" dans le tableau suivant indique le mode "netwloop" ou "netwtest".

	Mode de Connexion A					
M	sendonly	recvonly	sendrecv	confrnce	inactive	netw
o sendonly	Ao=Hi	Ao=NA	Ao=Hi	Ao=Hi	Ao=NA	Ao=Ai
d	Bo=Hi	Bo=Hi	Bo=Hi	Bo=Hi	Bo=Hi	Bo=Hi
e	Ho=NA	Ho=Ai	Ho=Ai	Ho=Ai	Ho=NA	Ho=NA
recvonly		Ao=NA	Ao=Hi	Ao=Hi	Ao=NA	Ao=Ai
d		Bo=NA	Bo=NA	Bo=NA	Bo=NA	Bo=NA
e		Ho=Ai+Bi	Ho=Ai+Bi	Ho=Ai+Bi	Ho=Bi	Ho=Bi
sendrecv			Ao=Hi	Ao=Hi	Ao=NA	Ao=Ai
c			Bo=Hi	Bo=Hi	Bo=Hi	Bo=Hi
o			Ho=Ai+Bi	Ho=Ai+Bi	Ho=Bi	Ho=Bi
n confrnce				Ao=Hi+Bi	Ao=NA	Ao=Ai
n				Bo=Hi+Ai	Bo=Hi	Bo=Hi
e				Ho=Ai+Bi	Ho=Bi	Ho=Bi
x inactive					Ao=NA	Ao=Ai
i					Bo=NA	Bo=NA
o					Ho=NA	Ho=NA
n netw						Ao=Ai
						Bo=Bi
B						Ho=NA

Si il y a trois connexions "actives" ou plus, elles vont quand même interagir comme défini dans le tableau ci-dessus avec le flux de supports sortant mixé pour chaque interaction (union de tous les flux). Si des ressources internes sont utilisées et que les flux ne peuvent pas être mixés, la passerelle DOIT retourner une erreur (les codes d'erreur 403 ou 502, Pas assez de ressources, sont RECOMMANDÉS).

Appendice E. Conventions de désignation des points d'extrémité

Les paragraphes qui suivent donnent des conventions de dénomination de point d'extrémité RECOMMANDÉES.

E.1 Points d'extrémité de ligne d'accès analogique

La chaîne "aaln" devrait être utilisée comme premier terme dans un nom de point d'extrémité locale pour des points d'extrémité de ligne d'accès analogique. Les termes qui suivent "aaln" devraient respecter la hiérarchie physique de la passerelle afin que si la passerelle a un certain nombre d'accès RJ11, le nom de point d'extrémité local pourrait ressembler à ce qui suit :

aaln/#

où "#" est le nombre de lignes analogiques (accès RJ11) sur la passerelle.

Par ailleurs, la passerelle peut avoir un certain nombre d'unités physiques de connexion, dont chacune contient un certain nombre d'accès RJ11, et dans ce cas, le nom de point d'extrémité local pourrait ressembler à :

aaln/<unit #>/#

où <unit #> est le numéro de la prise en unités dans la passerelle et "#" est le numéro de la ligne analogique (accès RJ11) sur cette unité. Des zéros en tête NE DOIVENT PAS être utilisés dans les numéros ("#") ci-dessus.

E.2 Circuits numériques

La chaîne "ds" devrait être utilisée pour le premier terme des points d'extrémité numériques avec une convention de dénomination qui suit la hiérarchie physique et numérique comme dans :

ds/<unit-type1>-<unit #>/<unit-type2>-<unit #>/.../<channel #>

où : <unit-type> identifie le niveau de hiérarchie particulier. Des exemples de valeurs de <unit-type> sont : "s", "su", "oc3", "ds3", "e3", "ds2", "e2", "ds1", "e1" où "s" indique numéro d'intervalle et "su" indique une sous unité au sein d'un intervalle. Des zéros en tête NE DOIVENT PAS être utilisés dans les numéros ("#") ci-dessus.

<unit #> est un nombre décimal qui est utilisé pour faire référence à une instance particulière d'un <unit-type> à ce niveau de la hiérarchie. Le nombre de niveaux et la dénomination de ces niveaux se fonde sur la hiérachie physique au sein de la passerelle de supports.

E.3 Points d'extrémité virtuels

Un autre type de point d'extrémité est celui qui n'est pas associé à une interface physique (comme un point d'extrémité analogique ou numérique). Ce type de point d'extrémité est appelé un point d'extrémité virtuel et est souvent utilisé pour représenter des ressources de processeur à signal numérique (DSP, *Digital Signal Processor*) qui donne certaines capacités au point d'extrémité. Des exemples sont des appareils d'annonces, de réponse vocale interactive (IVR, *interactive voice response)* ou de pont de conférence. Ces appareils peuvent avoir plusieurs instances de fonctions DSP de sorte qu'une convention de dénomination possible est :

<type-de-point-d'extrémité-virtuel>/<point-d'extrémité-#>

où <type-de-point-d'extrémité-virtuel> peut être une chaîne représentant le type du point d'extrémité (comme "ann" pour un serveur d'annonces ou "cnf" pour un serveur de conférence) et <point-d'extrémité-#> va identifier un point d'extrémité virtuel particulier au sein de l'appareil. Des zéros en tête NE DOIVENT PAS être utilisés dans le numéro ("#") ci-dessus. Si la hiérarchie physique du serveur inclut des cartes DSP enfichées, un autre niveau de hiérarchie dans le nom du point d'extrémité local peut être utilisé pour décrire l'unité enfichée.

Un point d'extrémité virtuel peut être créé par suite de l'utilisation du caractère générique "any of". De même, un point d'extrémité virtuel peut cesser d'exister une fois que la dernière connexion sur le point d'extrémité virtuel est supprimée. La définition du point d'extrémité virtuel DOIT détailler ces deux aspects.

Quand un <type-de-point-d'extrémité-virtuel> crée et supprime automatiquement des points d'extrémité virtuels, il va y avoir des cas où aucun point d'extrémité virtuel n'existe au moment où une commande RestartInProgress va être produite. Dans ce cas, la passerelle DEVRAIT simplement utiliser le caractère générique "all of" au lieu d'un <point-d'extrémité-#> spécifique, comme par exemple dans :

ann/*@mygateway.whatever.net

Si la commande RestartInProgress se réfère à tous les points d'extrémité dans la passerelle (virtuels ou non) le <identifiant-de-point-d'extrémité-virtuel> peut être omis comme par exemple dans :

*@mygateway.whatever.net

Les commandes reçues par la passerelle vont quand même devoir se référer à un point d'extrémité réel (éventuellement créé par cette commande en utilisant le caractère générique "any of") afin que la commande soit traitée.

E.4 Passerelle de supports

MGCP définit seulement le fonctionnement sur les points d'extrémité dans une passerelle de supports. Il peut être avantageux de définir un point d'extrémité qui représente la passerelle elle même par opposition aux points d'extrémité gérés par la passerelle. Les mises en œuvre qui souhaitent le faire devraient utiliser le nom de point d'extrémité local "mg" (pour "passerelle de supports") comme dans :

mg@mygateway.whatever.net

Noter que définir un tel point d'extrémité ne change pas la sémantique du protocole, c'est-à-dire, le point d'extrémité "mg" et les autres points d'extrémité (par exemple, des circuits numériques) dans la passerelle sont toujours des points d'extrémité indépendants et DOIVENT être traités comme tels. Par exemple, les commandes RestartInProgress DOIVENT toujours être produites pour tous les points d'extrémité dans la passerelle comme d'habitude.

E.5 Caractères génériques de gamme

Comme décrit au paragraphe 2.1.2, le schéma de dénomination de point d'extrémité MGCP définit les caractères génériques "all of" et "any of" pour les termes individuels dans un nom de point d'extrémité local. Bien que le caractère générique "all of" soit très utile pour réduire le nombre de messages, il ne peut par définition être utilisé que quand on souhaite se référer à toutes les instances d'un terme donné dans le nom du point d'extrémité local. De plus, dans le cas où une commande est à envoyer par la passerelle à l'agent d'appel, le caractère générique "all of" ne peut être utilisé que si tous les points d'extrémité qu'il désigne ont la même "entité notifiée". Les mises en œuvre qui préfèrent un schéma de caractères génériques plus fin peuvent utiliser le schéma de caractères génériques de gamme décrit ici.

Un caractére générique de gamme est défini comme suit :

```
RangeWildcard    = "[" NumericalRange *( "," NumericalRange ) "]"
NumericalRange   = 1*(CHIFFRE) [ "-" 1*(CHIFFRE) ]
```

Noter que les espaces blanches ne sont pas permises. Aussi, comme les caractères génériques de gamme utilisent le caractère "[" pour indiquer le début d'une gamme, le caractère "[" NE DOIT PAS être utilisé dans des noms de point d'extrémité qui utilisent des caractères génériques de gamme. La longueur d'un caractére générique de gamme DEVRAIT être limitée à une valeur raisonnablement petite, par exemple, 128 caractères.

Les caractères génériques de gamme peuvent être utilisés partout où un caractère générique "all of" peut être utilisé. La sémantique est identique pour les points d'extrémité désignés. Cependant, on DOIT noter que l'utilisation du schéma de caractères génériques de gamme exige la prise en charge par la passerelle et par l'agent d'appel. Donc, une passerelle NE DOIT PAS supposer que son agent d'appel prend en charge les caractères génériques de gamme et vice versa. En pratique, cela signifie normalement que la passerelle et l'agent d'appel vont devoir tous les deux être provisionnés de façon cohérente afin d'utiliser les caractères génériques de gamme. Aussi, si une passerelle ou un agent d'appel qui utilise les caractères génériques de gamme reçoit une réponse d'erreur qui pourrait indiquer un possible problème de désignation de point d'extrémité, ils DOIVENT être capables de revenir automatiquement à la non utilisation des caractères génériques de gamme.

Les exemples suivants illustrent l'utilisation des caractères génériques de gamme :

```
ds/ds1-1/[1-12]
ds/ds1-1/[1,3,20-24]
ds/ds1-[1-2]/*
ds/ds3-1/[1-96]
```

L'exemple suivant illustre comment l'utiliser dans une commande :

```
RSIP 1204 ds/ds3-1/[1-96]@tgw-18.whatever.net MGCP 1.0
RM: restart
RD: 0
```

Appendice F. Exemples de codage de commandes

Cet appendice donne des exemples de commandes et réponses montrées avec le codage réel utilisé. Des exemples sont fournis pour chaque commande. Tout commentaire montré dans les commandes et réponses est facultatif.

F.1 NotificationRequest

Le premier exemple illustre une demande de notification qui va faire sonner un téléphone et chercher un événement de décroché :

```
RQNT 1201 aaln/1@rgw-2567.whatever.net MGCP 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: l/hd(N)
S: l/rg
```

La réponse indique que la transaction a réussi :

```
200 1201 OK
```

Le second exemple illustre une demande de notification qui va chercher et accumuler un événement de décroché, et ensuite fournir une tonalité de numérotation et accumuler les chiffres en accord avec le script de numérotation fourni. La "entité notifiée" est réglée à "ca@ca1.whatever.net:5678", et comme le paramètre SignalRequests est vide (il pourrait aussi avoir été omis) tous les signaux TO actuellement actifs vont être arrêtés. Tous les événements dans la mémoire tampon de quarantaine vont être traités, et la liste des événements à détecter dans l'état "notification" va inclure des tonalités de télécopie en plus des "événements demandés" et des événements persistants :

```
RQNT 1202 aaln/1@rgw-2567.whatever.net MGCP 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
R: L/hd(A, E(S(L/dl),R(L/oc, L/hu, D/[0-9#*T](D))))
D: (0T|00T|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
S:
Q: process
T: G/ft
```

La réponse indique que la transaction a réussi :

```
200 1202 OK
```

F.2 Notify

L'exemple ci-dessous illustre un message Notify qui notifie un événement de décroché suivi par un numéro de 12 chiffres commençant par "91". Un identifiant de transaction corrélant le Notify avec la demande de notification dont il résulte est

inclus. La commande est envoyé à "l'entité notifiée" en cours, qui va normalement être la valeur réelle fournie dans le paramètre NotifiedEntity, c'est-à-dire, "ca@ca1.whatever.net:5678" - une situation de reprise sur défaillance pourrait l'avoir changée :

```
NTFY 2002 aaln/1@rgw-2567.whatever.net MGCP 1.0
N: ca@ca1.whatever.net:5678
X: 0123456789AC
O: L/hd,D/9,D/1,D/2,D/0,D/1,D/8,D/2,D/9,D/4,D/2,D/6,D/6
```

La réponse Notify indique que la transaction a réussi :

```
200 2002 OK
```

F.3 CreateConnection

Le premier exemple illustre une commande CreateConnection pour créer une connexion sur le point d'extrémité spécifié. La connexion va faire partie de l'identifiant d'appel spécifié. Les options de connexion locale spécifient que la Loi mu de G.711 va être le codec utilisé et que la période de mise en paquets va être de 10 ms. Le mode de connexion va être "receive only" :

```
CRCX 1204 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: recvonly
```

La réponse indique que la transaction a réussi, et un identifiant de connexion pour la nouvelle connexion créée est donc inclus. Une description de session pour la nouvelle connexion est aussi incluse - noter 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 second exemple illustre une commande CreateConnection contenant une demande de notification et un descripteur de connexion distante :

```
CRCX 1205 aaln/1@rgw-2569.whatever.net MGCP 1.0
C: A3C47F21456789F0
L: p:10, a:PCMU
M: sendrecv
X: 0123456789AD
R: L/hd
S: L/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, parce que le téléphone était déjà décroché. Par conséquent, ni un identifiant de connexion ni une description de session ne sont retournés :

```
401 1205 Téléphone décroché
```

Notre troisième exemple illustre l'utilisation de la réponse provisoire et de la prise de contact en trois phases. On crée une autre connexion et on accuse réception de la précédente réponse reçue en utilisant le paramètre Accusé de réception de réponse :

```
CRCX 1206 aaln/1@rgw-2569.whatever.net MGCP 1.0
K: 1205
C: A3C47F21456789F0
L: p:10, a:PCMU
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 retourné 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:
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 accuse réception de la réponse finale comme demandé :

```
000 1206
```

et la transaction est achevée.

F.4 ModifyConnection

Le premier exemple montre une commande ModifyConnection qui établit simplement le mode de connexion d'une connexion à "send/receive" - "l'entité notifiée" est aussi établie :

```
MDCX 1209 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
N: ca@ca1.whatever.net
M: sendrecv
```

La réponse indique que la transaction a réussi :

```
200 1209 OK
```

Dans le second exemple, on passe une description de session et on inclut une demande de notification avec la commande ModifyConnection. Le point d'extrémité commencer à exécuter les tonalités de retour d'appel chez l'utilisateur :

```
MDCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
M: recvonly
X: 0123456789AE
R: L/hu
S: G/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 réussi :

```
200 1206 OK
```

F.5 DeleteConnection (de l'agent d'appel)

Dans cet exemple, l'agent d'appel donne simplement pour instruction à la passerelle de supprimer la connexion "FDE234C8" sur le point d'extrémité spécifié :

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
```

La réponse indique le succès, et que la connexion a été supprimée. Les paramètres de connexion pour la connexion sont donc aussi inclus :

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

F.6 DeleteConnection (de la passerelle)

Dans cet exemple, la passerelle envoie une commande DeleteConnection à l'agent d'appel pour lui dire qu'une connexion sur le point d'extrémité spécifié a été supprimée. Le code de cause spécifie la raison de la suppression, et les paramètres de connexion pour la connexion sont aussi fournis :

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
I: FDE234C8
E: 900 - Erreur du matériel
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
```

F.7 DeleteConnection (plusieurs connexions de l'agent d'appel)

Dans le premier exemple, l'agent d'appel donne pour instruction à la passerelle de supprimer toutes les connexions relatives à l'appel "A3C47F21456789F0" sur le point d'extrémité spécifié :

```
DLCX 1210 aaln/1@rgw-2567.whatever.net MGCP 1.0
C: A3C47F21456789F0
```

La réponse indique le succès et que la ou les connexions ont été supprimées :

```
250 1210 OK
```

Dans le second exemple, l'agent d'appel donne pour instruction à la passerelle de supprimer toutes les connexions relatives à tous les points d'extrémité spécifiés :

```
DLCX 1210 aaln/*@rgw-2567.whatever.net MGCP 1.0
```

La réponse indique le succès :

```
250 1210 OK
```

F.8 AuditEndpoint

Dans le premier exemple, l'agent d'appel veut savoir quels points d'extrémité sont présents sur la passerelle spécifiée, d'où l'utilisation du caractère générique "all of" pour la portion locale du nom du point d'extrémité :

```
AUEP 1200 *@rgw-2567.whatever.net MGCP 1.0
```

La passerelle indique le succès et inclut une liste des noms de point d'extrémité :

```
200 1200 OK
Z: aaln/1@rgw-2567.whatever.net
Z: aaln/2@rgw-2567.whatever.net
```

Dans le second exemple, les capacités d'un des points d'extrémité sont demandées :

```
AUEP 1201 aaln/1@rgw-2567.whatever.net MGCP 1.0
F: A
```

La réponse indique le succès et aussi les capacités. Deux codecs sont pris en charge, cependant avec des capacités différentes. Par conséquent, deux ensembles séparés de capacités sont retournés :

```
200 1201 OK
A: a:PCMU, p:10-100, e:on, s:off, v:L;S, m:sendonly;recvonly;sendrecv;inactive;netwloop;netwtest
A: a:G729, p:30-90, e:on, s:on, v:L;S, m:sendonly;recvonly;sendrecv;inactive;confrnce;netwloop
```

Dans le troisième exemple, l'agent d'appel examine plusieurs types d'informations pour le point d'extrémité :

```
AUEP 2002 aaln/1@rgw-2567.whatever.net MGCP 1.0
F: R,D,S,X,N,I,T,O,ES
```

La réponse indique le succès :

```
200 2002 OK
R: L/hu,L/oc(N),D/[0-9](N)
D:
S: L/vmwi(+)
X: 0123456789B1
N: [128.96.41.12]
I: 32F345E2
T: G/ft
```

O: L/hd,D/9,D/1,D/2
ES: L/hd

La liste des événements demandés contient trois événements. Lorsque aucun nom de paquetage n'est spécifié, le paquetage par défaut est supposé. Il en va de même pour les actions, donc l'action par défaut - Notify - doit être supposée pour l'événement "L/hu". L'omission d'une valeur pour le "script de numérotation" signifie que le point d'extrémité n'a actuellement pas de script de numérotation. Il n'y a pas de signaux de temporisation actuellement actifs, cependant le signal OO "vmwi" est actuellement actif et est par conséquent inclus - dans ce cas il a été paramétré, cependant le paramètre pourrait avoir été exclu. La "entité notifiée" en cours se réfère à une adresse IP et une seule connexion existe pour le point d'extrémité. La valeur actuelle de DetectEvents est "G/ft", et la liste des ObservedEvents contient les quatre événements spécifiés. Finalement, les états d'événement examinés révèlent que le téléphone était décroché au moment où la transaction a été traitée.

F.9 AuditConnection

Le premier exemple montre une commande AuditConnection où on examine les paramètres CallId, NotifiedEntity, LocalConnectionOptions, Connection Mode, LocalConnectionDescriptor, et Connection :

AUCX 2003 aaln/1@rgw-2567.whatever.net MGCP 1.0
I: 32F345E2
F: C,N,L,M,LC,P

La réponse indique le succès et inclut les informations pour les RequestedInfo :

200 2003 OK
C: A3C47F21456789F0
N: ca@ca1.whatever.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 second exemple, on demande à examiner RemoteConnectionDescriptor et LocalConnectionDescriptor :

AUCX 1203 aaln/2@rgw-2567.whatever.net MGCP 1.0
I: FDE234C8
F: RC,LC

La réponse indique le succès, et inclut les informations pour RequestedInfo. Dans ce cas, aucun RemoteConnectionDescriptor n'existe, donc seul le 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

F.10 RestartInProgress

Le premier exemple illustre un message RestartInProgress envoyé par une passerelle pour informer l'agent d'appel que le point d'extrémité spécifié va être mis hors service dans 300 secondes :

```
RSIP 1200 aaln/1@rgw-2567.whatever.net MGCP 1.0
RM: graceful
RD: 300
```

de la passerelle's que la transaction a réussi :

```
200 1200 OK
```

Dans le second exemple, le message RestartInProgress envoyé par la passerelle informe l'agent d'appel que tous les points d'extrémité de la passerelle vont être placés en service dans 0 seconde, c'est-à-dire, ils sont actuellement en service. Le délai de redémarrage pourrait aussi avoir été omis :

```
RSIP 1204 *@rgw-2567.whatever.net MGCP 1.0
RM: restart
RD: 0
```

La réponse de l'agent d'appel indique le succès, et de plus fournit aux points d'extrémité en question une nouvelle "entité notifiée" :

```
200 1204 OK
N: CA-1@whatever.net
```

Autrement, la commande pourrait avoir échoué avec une nouvelle "entité notifiée" comme dans :

```
521 1204 OK
N: CA-1@whatever.net
```

Dans ce cas, la commande devrait être reessayée afin de satisfaire la "procédure de redémarrage", cette fois allant à l'agent d'appel "CA-1@whatever.net".

Appendice G. Exemples de flux d'appels

Les tableaux de flux de messages de cette Section utilisent les abréviations suivantes :
* rgw = passerelle résidentielle
* ca = agent d'appel
* n+ = l'étape 'n' est répétée une ou plusieurs fois

Noter que l'utilisation de majuscules et minuscules dans le texte des messages est pour faciliter la lisibilité et n'est en aucune façon une exigence. La seule exigence impliquant la casse est d'être tout le temps insensible à la casse.

G.1 Redémarrage

G.1.1 Redémarrage de passerelle résidentielle

Le tableau suivant montre une séquence de messages qui pourrait survenir quand un agent d'appel (ca) est contacté par deux passerelles résidentielles (rgw1 et rgw2) indépendantes qui ont redémarré.

Tableau G.1 : Redémarrage de passerelle résidentielle

N° d'étape	usager1	rgw1	ca	rgw2	usager2
1		rsip ->			
			<- ack		
2			<- auep		
		ack ->			

```
3+                              <- rqnt
                    ack ->
4                                       <- rsip
                                ack ->
5                               auep ->
                                        <- ack
6+                              rqnt ->
                                        <- ack
```

Étape 1 - RestartInProgress (rsip) de rgw1 à ca

rgw1 utilise le DNS pour déterminer le nom de domaine de ca et envoie à l'accès par défaut de 2727. La commande consiste en ce qui suit :

```
rsip 1 *@rgw1.whatever.net mgcp 1.0
rm: restart
```

Le "*" est utilisé pour informer ca que tous les points d'extrémité de rgw1 sont en redémarrage, et "restart" est spécifié comme la méthode de redémarrage. L'agent d'appel "ca" accuse réception de la commande avec un message d'accusé de réception contenant l'identifiant de transaction (dans ce cas 1) pour la commande. Il envoie l'accusé de réception à rgw1 en utilisant le même accès que spécifié comme accès de source pour le rsip. Si aucun n'était indiqué, il utilise l'accès par défaut de 2727.

```
200 1 ok
```

Un code de réponse est obligatoire. Dans ce cas, "200", indique "la transaction demandée a été exécutée normalement". la chaîne de réponse est facultative. Dans ce cas, "ok" est inclus comme description supplémentaire.

Étape 2 - AuditEndpoint (auep) de ca à rgw1

La commande consiste en ce qui suit :

```
auep 153 *@rgw1.whatever.net mgcp 1.0
```

Le "*" est utilisé pour demander les informations d'audit à rgw1 sur tous ses points d'extrémité. rgw1 accuse réception de la commande avec un message d'accusé de réception contenant l'identifiant de transaction (dans ce cas 153) de la commande, et elle inclut une liste de ses points d'extrémité. Dans cet exemple, rgw1 a deux points d'extrémité, aaln/1 et aaln/2.

```
200 153 ok
Z: aaln/1@rgw1.whatever.net
Z: aaln/2@rgw1.whatever.net
```

Une fois qu'il a la liste des identifiants de point d'extrémité, ca peut envoyer des commandes AuditEndpoint individuelles dans lesquelles le "*" est remplacé par l'identifiant du point d'extrémité donné. Comme réponse, rgw1 va remplacer la liste des identifiants de point d'extrémité retournée dans l'exemple par les informations demandées pour le point d'extrémité. Cet échange de messages facultatif n'est pas montré dans cet exemple.

Étape 3 - NotificationRequest (rqnt) de ca à chaque point d'extrémité de rgw1

Dans ce cas, ca envoie deux rqnts, une pour aaln/1 :

```
rqnt 154 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hd(n)
x: 3456789a0
```

et une seconde pour aaln/2:

```
rqnt 155 aaln/2@rgw1.whatever.net mgcp 1.0
r: l/hd(n)
x: 3456789a1
```

Noter que dans la ligne de paramètre d'événements demandés, l'événement est pleinement spécifié comme "l/hd", c'est-à-dire, avec le nom de paquetage, afin d'éviter toute ambiguîté potentielle. C'est le comportement recommandé. Pour être clair, l'action, qui dans ce cas est de notifier, est explicitement spécifiée en incluant le "(n)". Si aucune action n'est spécifiée, Notify est supposé par défaut, sans considération de l'événement. Si aucune autre action n'est désirée, cela doit être déclaré explicitement.

La réponse attendue de rgw1 à ces demandes est un accusé de réception de aaln/1 comme suit :

```
200 154 ok
```

et de aaln/2 :

```
200 155 ok
```

Étape 4 RestartInProgress (rsip) de rgw2 à ca

```
rsip 0 *@rgw2.whatever.net mgcp 1.0
rm: restart
```

suivi par l'accusé de réception de ca :

```
200 0 ok
```

Étape 5 - AuditEndpoint (auep) de ca à rgw2

```
auep 156 *@rgw2.whatever.net mgcp 1.0
```

suivi par un accusé de réception de rgw2 :

```
200 156 ok
z: aaln/1@rgw2.whatever.net
z: aaln/2@rgw2.whatever.net
```

Étape 6 - NotificationRequest (rqnt) de ca à chaque point d'extrémité de rgw2

```
rqnt 157 aaln/1@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
x: 3456789a2
```

suivi par :

```
rqnt 158 aaln/2@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
x: 3456789a3
```

avec l'accusé de réception de rgw2 pour aaln/1 :

```
200 157 ok
```

et pour aaln/2 :

```
200 158 ok
```

G.1.2 Redémarrage de l'agent d'appel

Le tableau suivant montre la séquence de messages qui se produit quand un agent d'appel (ca) redémarre. Comment il détermine les informations d'adresse des passerelles, dans ce cas rgw1 et rgw2, n'est pas couvert dans le présent document. Pour l'interopérabilité, il est RECOMMANDÉ de fournir la capacité de configurer l'agent d'appel à envoyer AUEP (*) aux adresses et acccès spécifiques.

Tableau G.2 : Redémarrage de l'agent d'appel

N°	usager1	rgw1	ca	rgw2	usager2
1			<- auep		
		ack ->			
2+			<- rqnt		
		ack ->			
3			auep ->		
				<- ack	
4+			rqnt ->		
				<- ack	

Étape 1 - AuditEndpoint (auep) de ca à rgw1. La commande consiste en ce qui suit :

```
auep 0 *@rgw1.whatever.net mgcp 1.0
```

Le "*" est utilisé pour demander à rgw1 les informations d'audit de tous ses points d'extrémité. rgw1 accuse réception de la command avec un message d'accusé de réception contenant l'identifiant de transaction (dans ce cas 0) de la commande, et elle inclut une liste de ses points d'extrémité. Dans cet exemple, rgw1 a deux points d'extrémité, aaln/1 et aaln/2.

```
200 0 ok
z: aaln/1@rgw1.whatever.net
z: aaln/2@rgw1.whatever.net
```

Une fois qu'il a la liste des identifiants de point d'extrémité, ca peut envoyer des commandes individuelles AuditEndpoint dans lesquelles le "*" est remplacé par l'identifiant du point d'extrémité concerné. Comme réponse, rgw1 va remplacer la liste des identifiants de point d'extrémité retournée dans l'exemple par les informations demandées pour le point d'extrémité. Cet échange de messages facultatif n'est pas montré dans cet exemple.

Étape 2 - Demande de notification (rqnt) de décroché de ca à rgw1
Dans ce cas, ca envoie deux rqnts, une pour aaln/1 :

```
rqnt 1 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hd(n)
x: 234567890
```

et une seconde pour aaln/2 :

```
rqnt 2 aaln/2@rgw1.whatever.net mgcp 1.0
r: l/hd(n)
x: 234567891
```

La réponse attendue de rgw1 à ces demandes est un accusé de réception de aaln/1 comme suit :

```
200 1 ok
```

et de aaln/2 :

```
200 2 ok
```

Étape 3 - AuditEndpoint (auep) de ca à rgw2

```
auep 3 *@rgw2.whatever.net mgcp 1.0
```

suivi par un accusé de réception de rgw2 :

```
200 3 ok
z: aaln/1@rgw2.whatever.net
z: aaln/2@rgw2.whatever.net
```

Étape 4 - NotificationRequest (rqnt) de ca à chaque point d'extrémité de rgw2 :

rqnt 4 aaln/1@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
x: 234567892

suivi par :

rqnt 5 aaln/2@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
x: 234567893

avec un accusé de réception de rgw2 pour aaln/1 :

200 4 ok

et pour aaln/2 :

200 5 ok

G.2 Création de connexion

G.2.1 De passerelle résidentielle à passerelle résidentielle

Le tableau suivant montre la séquence de messages qui se produit quand un utilisateur (usager1) fait un appel à travers une passerelle résidentielle (rgw1) à un utilisateur desservi par une autre passerelle résidentielle (rgw2). Cet exemple illustre seulement la communication entre les passerelles résidentielles et l'agent d'appel (ca). Le nom local des points d'extrémité dans cet exemple est aaln/1 pour les deux passerelles, et les références au sein de la description des étapes à rgw1 et rgw2 peuvent être supposées se référer à aaln/1 de rgw1 et aaln/1 de rgw2. Noter que ceci est seulement un exemple et n'est pas le seul scénario d'appel légal.

Tableau G.3 : Création d'une connexion de passerelle résidentielle

N°	usager1	rgw1	ca	rgw2	usager2
1	décroché ->	ntfy ->			
			<- ack		
2	<- dialtone		<- rqnt		
		ack ->			
3	chiffres ->	ntfy ->			
			<- ack		
4			<- rqnt		
		ack ->			
5	<- recvonly		<- crcx		
		ack ->			
6			crcx ->		sendrcv ->
				<- ack	
7	<- recvonly		<- mdcx		
		ack ->			
8	<- ringback		<- rqnt		
		ack ->			
9			rqnt ->		ringing ->
				<- ack	
10				<- ntfy	<- décroché
			ack ->		
11			rqnt ->		
				<- ack	
12			<- rqnt		
		ack ->			
13	<- sendrcv		<- mdcx		
		ack ->			

Étape 1 - Notifie (ntfy)le décroché de rgw1 à ca
Ce ntfy est le résultat du décroché de usager1 et suppose que ca avait précédemment envoyé une rqnt avec le RequestId "445678944" à rgw1 demandant la notification d'un événement de décroché *(offhook)* :

```
ntfy 12 aaln/1@rgw1.whatever.net mgcp 1.0
o: l/hd
x: 445678944
```

Accusé de réception provenant de ca :

```
200 12 ok
```

Étape 2 - Demande de notification (rqnt) des chiffres de ca à rgw1
Demande à rgw1 de notifier un raccroché et de collecter les chiffres en accord avec le script de numérotation, et de fournir la tonalité de numérotation *(dialtone)* :

```
rqnt 1057 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hu(n), d/[0-9#*T](d)
s: l/dl
x: 445678945
d: 5xxx
```

Accusé de réception provenant de rgw1 :

```
200 1057 ok
```

Étape 3 - Notifie (ntfy) les chifrfes de rgw1 à ca

```
ntfy 13 aaln/1@rgw1.whatever.net mgcp 1.0
o: d/5, d/0, d/0, d/1
x: 445678945
```

Accusé de réception provenant de ca :

```
200 13 ok
```

Étape 4 - Demande de notification (rqnt) de ca à rgw1
Demande à rgw1 de notifier l'événement d'une transition à raccroché *(on-hook)* :

```
rqnt 1058 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hu(n)
x: 445678946
```

Accusé de réception de rgw1 :

```
200 1058 ok
```

Étape 5 - Création de connexion (crcx) de ca à rgw1
Demande une nouvelle connexion sur rgw1 avec les options de connexion locale spécifiées, incluant 20 ms comme période de mise en paquets, le codec en loi mu G.711, et le mode réception seule :

```
crcx 1059 aaln/1@rgw1.whatever.net mgcp 1.0
c: 9876543210abcdef
l: p:20, a:PCMU
m: recvonly
```

Accusé de réception de rgw1 qu'une nouvelle connexion, "456789fedcba5", a été créée, suivi par une ligne blanche et ensuite les paramètres SDP :

```
200 1059 ok
i: 456789fedcba5
```

```
v=0
o=- 23456789 98765432 IN IP4 192.168.5.7
s=-
c=IN IP4 192.168.5.7
t=0 0
m=audio 6058 RTP/AVP 0
```

Étape 6 - Créer une connexion (crcx) de ca à rgw2
Demande une nouvelle connexion sur rgw2. La demande inclut la description de session retournée par rgw1 comme quoi une connexion bidirectionnelle peut être initiée :

```
crcx 2052 aaln/1@rgw2.whatever.net mgcp 1.0
c: 9876543210abcdef
l: p:20, a:PCMU
m: sendrecv

v=0
o=- 23456789 98765432 IN IP4 192.168.5.7
s=-
c=IN IP4 192.168.5.7
t=0 0
m=audio 6058 RTP/AVP 0
```

Accusé de réception de rgw2 qu'une nouvelle connexion, "67890af54c9", a été créée ; suivi par une ligne blanche et ensuite les paramètres SDP :

```
200 2052 ok
i: 67890af54c9

v=0
o=- 23456889 98865432 IN IP4 192.168.5.8
s=-
c=IN IP4 192.168.5.8
t=0 0
m=audio 6166 RTP/AVP 0
```

Étape 7 - Modifier la connexon (mdcx) de ca à rgw1
Demande à rgw1 de modifier la connexion existante, "456789fedcba5", pour utiliser la description de session retournée par rgw2 qui établit une connexion semi duplex qui, bien que non utilisée dans cet exemple, pourrait être utilisée pour fournir à usager1 une tonalité de retour d'appel dans la bande, des annonces, etc. :

```
mdcx 1060 aaln/1@rgw1.whatever.net mgcp 1.0
c: 9876543210abcdef
i: 456789fedcba5
l: p:20, a:PCMU
M: recvonly

v=0
o=- 23456889 98865432 IN IP4 192.168.5.8
s=-
c=IN IP4 192.168.5.8
t=0 0
m=audio 6166 RTP/AVP 0
```

Accusé de réception de rgw1 :

```
200 1060 ok
```

Étape 8 - Demande de notification (rqnt) provenant de ca pour que rgw1 fournisse le retour d'appel *(ringback)*
Demande à rgw1 de notifier l'événement d'une transition à raccroché, et aussi de fournir la tonalité de retour d'appel :

rqnt 1061 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hu(n)
s: g/rt
x: 445678947

Accusé de réception de rgw1 :

200 1061 ok

Étape 9 - Demande de notification (rqnt) de ca à rgw2 pour fournir la sonnerie
Demande à rgw2 de continuer à chercher un décroché et de fournir la sonnerie :

rqnt 2053 aaln/1@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
s: l/rg
x: 445678948

Accusé de réception provenant de rgw2 :

200 2053 ok

Étape 10 - Notifie (ntfy) le décroché de rgw2 à ca

ntfy 27 aaln/1@rgw2.whatever.net mgcp 1.0
o: l/hd
x: 445678948

Accusé de réception provenant de ca :

200 27 ok

Étape 11 - Demande de notification (rqnt) de raccroché de ca à rgw2

rqnt 2054 aaln/1@rgw2.whatever.net mgcp 1.0
r: l/hu(n)
x: 445678949

Accusé de réception provenant de rgw2 :

200 2054 ok

Étape 12 - Demande de notification (rqnt) de raccroché de ca à rgw1

rqnt 1062 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hu(n)
x: 445678950

Accusé de réception provenant de rgw1 :

200 1062 ok

Étape 13 - Modifier la connexion (mdcx) de ca à rgw1
Demande à rgw1 de modifier la connexion existante, "456789fedcba5", en envoi/réception *(sendrecv)* afin qu'une connexion bidirectionnelle soit initiée :

mdcx 1063 aaln/1@rgw1.whatever.net mgcp 1.0
c: 9876543210abcdef
i: 456789fedcba5
m: sendrecv

Accusé de réception provenant de rgw1 :

```
200 1063 ok
```

G.3 Suppression de connexion

G.3.1 De passerelle résidentielle à passerelle résidentielle

Le tableau suivant montre la séquence de messages qui se produit quand un utilisateur (usager2) initie la suppression d'une connexion existante sur une passerelle résidentielle (rgw2) avec un utilisateur desservi par une autre passerelle résidentielle (rgw1). Cet exemple illustre seulement la communication entre les passerelles résidentielles et l'agent d'appel (ca). Le nom local des points d'extrémité dans cet exemple est aaln/1 pour les deux passerelles, et les références au sein de la description des étapes pour rgw1 et rgw2 peuvent être supposées se référer à aaln/1 de rgw1 et aaln/1 de rgw2.

Tableau G.4 : Suppression de connexion de passerelle résidentielle

N°	usager1	rgw1	ca	rgw2	usager2
1				<- ntfy	<- raccroché
			ack ->		
2			dlcx ->		
				<- ack	
3			<- dlcx		
		ack ->			
4			rqnt ->		
				<- ack	
5	raccroché->	ntfy ->			
			<- ack		
6			<- rqnt		
		ack ->			

Étape 1 - Notifier (ntfy) décroché de rgw1 à ca
Ce ntfy est le résultat de usager2 qui raccroche et suppose que ca avait précédemment envoyé une rqnt à rgw2 demandant la notification d'un événement de raccroché (voir la fin de la séquence Création de connexion):

```
ntfy 28 aaln/1@rgw2.whatever.net mgcp 1.0
o: l/hu
x: 445678949
```

Accusé de réception provenant de ca :

```
200 28 ok
```

Étape 2 - Supprimer la connexion (dlcx) de ca à rgw2
Demande à rgw2 de supprimer la connexion "67890af54c9" :

```
dlcx 2055 aaln/1@rgw2.whatever.net mgcp 1.0
c: 9876543210abcdef
i: 67890af54c9
```

Accusé de réception provenant de rgw2. Noter le code de réponse "250" qui signifie "la connexion est supprimée" :

```
250 2055 ok
```

Étape 3 - Supprimer la connexion (dlcx) de ca à rgw1
Demande à rgw1 de supprimer la connexion "456789fedcba5":

```
dlcx 1064 aaln/1@rgw1.whatever.net mgcp 1.0
c: 9876543210abcdef
i: 456789fedcba5
```

Accusé de réception provenant de rgw1 :

```
250 1064 ok
```

Étape 4 - Demande de notification (rqnt) de ca àrgw2
Demande à rgw2 de notifier à ca l'événement d'une transition de décroché :

```
rqnt 2056 aaln/1@rgw2.whatever.net mgcp 1.0
r: l/hd(n)
x: 445678951
```

Accusé de réception provenant de rgw2 :

```
200 2056 ok
```

Étape 5 - Notifie (ntfy) un raccroché de rgw1 à ca
Notifie à ca que usager1 à rgw1 est revenu à raccroché :

```
ntfy 15 aaln/1@rgw1.whatever.net mgcp 1.0
o: l/hu
x: 445678950
```

Accusé de réception provenant de ca :

```
200 15 ok
```

Étape 6 - Demande de notification (rqnt) de décroché de ca à rgw1
Demande à rgw1 de notifier à ca de l'événement d'une transition à décroché :

```
rqnt 1065 aaln/1@rgw1.whatever.net mgcp 1.0
r: l/hd(n)
x: 445678952
```

Accusé de réception provenant de rgw1 :

```
200 1065 ok
```

Adresse des auteurs

Flemming Andreasen
Cisco Systems
499 Thornall Street, 8th Floor
Edison, NJ 08837
USA
mél : fandreas@cisco.com

Bill Foster
Cisco Systems
771 Alder Drive
Milpitas, CA 95035
USA
mél : bfoster@cisco.com

Déclaration complète de droits de reproduction

les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses successeurs ou ayant droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.