Groupe de travail Réseau
Request for Comments : 3084
Catégorie : En cours de normalisation
mars 2001

Traduction Claude Brière de L'Isle

K. Chan & J. Seligson, Nortel Networks
D. Durham & R. Yavatkar, Intel
S. Gai & K. McCloghrie, Cisco
S. Herzog, IPHighway
F. Reichmeyer, PFN
A. Smith, Allegro Networks

Utilisation de COPS pour le provisionnement de politique (COPS-PR)

Résumé
Le présent document décrit l'utilisation du protocole Service commun de politique ouverte (COPS, *Common Open Policy Service*) pour la prise en charge de l'approvisionnement de politique (COPS-PR). La présente spécification est indépendante du type de politique qui est provisionnée (QS, sécurité, etc.) mais se concentre sur les mécanismes et conventions utilisés pour communiquer les informations provisionnées entre les PDP et les PEP. Les extensions du protocole décrites dans ce document ne font aucune hypothèse sur le modèle de données de politique communiquées, mais décrivent les formats et objets de message qui portent les données de politique modélisées.

Conventions utilisées dans ce document
Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

Table des Matières

1. Introduction....2
  1.1 Pourquoi COPS pour l'approvisionnement ?....3
  1.2 Interaction entre le PEP et le PDP....3
2. Base de données d'informations de politique (PIB)....3
  2.1 Règles pour la modification et l'extension des PIB....4
  2.2 Ajout ou refus de PRC à une PIB....4
  2.3 Opérations de COPS prises en charge pour une instance d'approvisionnement....5
3. Contenu de message....5
  3.1 Demande (REQ) PEP -> PDP....5
  3.2 Décision (DEC) PDP -> PEP....6
  3.3 État de rapport (RPT) PEP -> PDP....7
4. Objets du protocole COPS-PR....7
  4.1 Identifiant complet d'instance d'approvisionnement (PRID)....8
  4.2 Préfixe de PRID (PPRID)....8
  4.3 Données d'instance d'approvisionnement codées (EPD)....9
  4.4 Objet Erreur d'approvisionnement global (GPERR)....12
  4.5 Objet Erreur d'approvisionnement de classe de PRC (CPERR)....12
  4.6 Objet Erreur de PRID (ErrorPRID)....13
5. Formats de données spécifiques de client COPS-PR....13
  5.1 Données de décision nommée....13
  5.2 Données de demande ClientSI....14
  5.3 Données de rapport d'approvisionnement de politique....14
6. Fonctionnement courant....15
7. Tolérance aux fautes....16
8. Considérations sur la sécurité....16
9. Considérations relatives à l'IANA....17
10. Remerciements....17
11. Références....17

12. Adresse des auteurs.....18
13. Déclaration complète de droits de reproduction.....18

Glossaire

PRC *(Provisioning Class)* classe d'approvisionnement : un type de données de politique.
PRI *(Provisioning Instance)* instance d'approvisionnement : une instance de PRC.
PIB *(Policy Information Base)* base de données d'information de politique.
PDP *(Policy Decision Point)* point de décision de politique : voir la [RFC2753].
PEP *(Policy Enforcement Point)* point d'application de politique : voir la [RFC2753].
PRID *(Provisioning Instance Identifier)* identifiant d'instance d'approvisionnement : identifie de façon univoque une instance de PRC.

1. Introduction

Le groupe de travail de l'IETF Protocole d'allocation de ressource (RAP, *Resource Allocation Protocol*) a défini le protocole Service commun de politique ouverte (COPS, *Common Open Policy Service*) [RFC2748] comme un protocole adaptable qui permet aux serveurs de politique (PDP) de communiquer les décisions de politique aux appareils du réseau (PEP). COPS a été conçu pour prendre en charge plusieurs types de clients de politique.

COPS est un protocole de demande/réponse qui prend en charge deux modèles courants pour le contrôle de politique : Externalisation et Configuration.

Le modèle Externalisation s'adresse au genre d'événements du PEP qui exigent une décision de politique instantanée (autorisation). Dans le scénario d'externalisation, le PEP délègue à un serveur de politique externe (PDP) la responsabilité de prendre les décisions en son nom. Par exemple, dans l'utilisation de COPS pour RSVP [RFC2749], lorsque arrive un message RSVP de réservation, le PEP doit décider si il admet ou rejette la demande. Il peut externaliser cette décision en envoyant une demande spécifique à son PDP, attendant sa décision avant d'admettre la réservation en instance.

Par ailleurs, le modèle COPS Configuration (décrit ci-dessous comme modèle d'approvisionnement) ne fait pas d'hypothèse sur une telle corrélation univoque directe entre les événements du PEP et les décisions du PDP. Le PDP peut approvisionner le PEP de façon proactive en réagissant aux événements externes (comme une entrée d'usager) aux événements de PEP, et à toutes leurs combinaisons (corrélation N:M). L'approvisionnement peut être effectué en vrac (par exemple, la configuration de QS du routeur entier) ou en parties (par exemple, mettre à jour un filtre de marquage DiffServ).

Les ressources réseau sont souvent provisionnées sur la base d'accords de niveau de service (SLA, *Service Level Agreement*) relativement statiques aux frontières de réseau. Alors que le modèle d'externalisation est traité de façon dynamique en temps réel par le PEP, le modèle d'approvisionnement est traité par le PDP d'une façon assez souple dans le temps sur une large gamme d'aspects configurables du PEP.

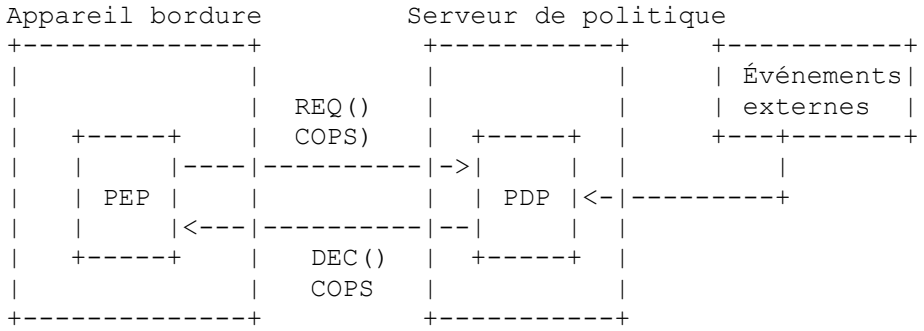


Figure 1 : Modèle d'approvisionnement COPS

Dans COPS-PR, les demandes de politique décrivent le PEP et ses paramètres configurables (plutôt qu'un événement de fonctionnement). Si un changement survient dans ces paramètres de base, une demande de mise à jour est envoyée. Donc, les demandes sont produites assez peu fréquemment. Les décisions ne sont pas nécessairement transposées directement en demandes, et sont le plus souvent produites lorsque le PDP répond à des événements externes ou des événements de PDP (mise à jour de politique/SLA).

Le présent document décrit l'utilisation du protocole COPS [RFC2748] pour la prise en charge de l'approvisionnement de politique. La présente spécification est indépendante du type de politique approvisionnée (QS, Sécurité, etc.). Il se concentre plutôt sur les mécanismes et conventions utilisées pour communiquer les informations échangées entre les PDP et les PEP. Le modèle de données supposé dans le présent document se fonde sur le concept de bases de données d'informations de politique (PIB, *Policy Information Base*) qui définissent les données de politique. Il peut y avoir une ou plusieurs PIB pour un zone de politique donnée et différentes zones de politique peuvent avoir différents ensembles de PIB.

Afin de prendre en charge un modèle qui inclut plusieurs PDP contrôlant des zones de politique sans recoupement sur un seul PEP, le type de client spécifié par le PEP au PDP est unique pour la zone de politique gérée. Un seul type de client pour une zone donnée de politique (par exemple, la QS) sera utilisé pour tous les PIB qui existent dans cette zone. Le client devrait traiter tous les types de clients COPS-PR qu'il prend en charge comme étant des espaces de noms sans recoupement et indépendants où les instances NE DOIVENT PAS être partagées .

Les exemples utilisés dans ce document sont biaisés en faveur de l'approvisionnement de politique de QS dans un environnement de services différenciés (DiffServ). Cependant, COPS-PR peut être utilisé pour d'autres types de politiques d'approvisionnement dans le même cadre.

1.1 Pourquoi COPS pour l'approvisionnement ?

COPS-PR a été conçu dans un cadre qui est optimisé pour des politiques d'approvisionnement efficace à travers des appareils, fondé sur les exigences définies dans la [RFC2753]. D'abord, COPS-PR permet un transport efficace des attributs, de grandes transactions unitaires de données, et un rapport efficace et souple des erreurs. Ensuite, comme il y a une seule connexion entre le client et le serveur de politique par zone de contrôle de politique identifiée par un type de client COPS, il garantit qu'un seul serveur met à jour une configuration de politique particulière à un instant donné. Une telle configuration de politique est effectivement verrouillée, même à partir d'une configuration de console locale, alors que PEP est connecté à un PDP via COPS. COPS utilise le transport fiable TCP, et utilise donc un mécanisme de partage/synchronisation d'état et échange seulement des mises à jour différentielles. Si le serveur ou le client est réamorcé (ou redémarré) l'autre va le savoir rapidement. Enfin, il est défini comme un mécanisme de communications en temps réel conduit par l'événement, n'exigeant jamais d'interrogations entre le PEP et le PDP.

1.2 Interaction entre le PEP et le PDP

Lorsque un appareil s'amorce, il ouvre une connexion COPS avec son PDP primaire. Lorsque la connexion est établie, le PEP envoie au PDP des informations sur lui-même sous la forme d'une demande de configuration. Ces informations incluent des informations spécifiques du client (par exemple, le type de matériel, la version du logiciel, des informations de configuration). Durant cette phase, le client peut aussi spécifier la taille maximum de message COPS-PR acceptée.

En réponse, le PDP télécharge toutes les politiques provisionnées qui sont actuellement pertinentes pour cet appareil. À réception des politiques provisionnées, l'appareil les transpose en ses mécanismes de QS locaux, et les installe. Si les conditions changent au PDP et si le PDP détecte que des changements sont requis dans les politiques provisionnées actuellement en effet, le PDP envoie alors les changements de politique (les installe, les met à jour et/ou les supprime) au PEP, et le PEP met à jour sa configuration locale de la façon appropriée.

Si, par la suite, la configuration de l'appareil change (commande supprimée/ajoutée, nouveau logiciel installé, etc.) d'une façon non couverte par les politiques déjà connues du PEP, celui-ci envoie de façon asynchrone ces nouvelles informations non sollicitées au PDP dans une demande de mise à jour de configuration. À réception de ces nouvelles informations, le PDP envoie au PEP toutes les nouvelles politiques provisionnées supplémentaires maintenant nécessaires au PEP, ou retire les politiques qui ne sont plus requises.

2. Base de données d'informations de politique (PIB)

Les données portées par COPS-PR sont un ensemble de données de politique. Le protocole suppose une structure de données nommée, appelée base de données d'informations de politique (PIB, *Policy Information Base*) pour identifier le type et l'objet des informations de politique non sollicitées qui sont "poussées" du PDP au PEP pour provisionner une politique ou pour les envoyer du PEP au PDP comme notification. L'espace de noms de la PIB est commun au PEP et au PDP et les instances de données au sein de cet espace sont uniques dans la portée d'un certain type de client et état de demande par connexion TCP entre un PEP et un PDP. Noter qu'étant donné qu'un appareil peut mettre en œuvre plusieurs types de clients COPS, un espace unique d'instance sera fourni pour chaque type de client distinct. Il n'y a pas de partage de données d'instance entre les types de client mis en œuvre par un PEP, même si les classes qui sont instanciées sont du même type et si elles partagent le même identifiant d'instance.

La PIB peut être décrite comme une arborescence conceptuelle de noms où les branches de l'arborescence représentent des structures de données ou des classes d'approvisionnement (PRC, *Provisioning Class*) tandis que les feuilles représentent diverses instanciations d'instances d'approvisionnement (PRI, *Provisioning Instance*). Il peut y avoir plusieurs instances de données (PRI) pour chaque structure de données (PRC). Par exemple, si on veut installer plusieurs filtres de contrôle d'accès, le PRC peut représenter un type de filtre de contrôle d'accès générique et chaque PRI peut représenter un filtre de contrôle d'accès individuel à appliquer. L'arborescence pourrait être représentée comme suit :

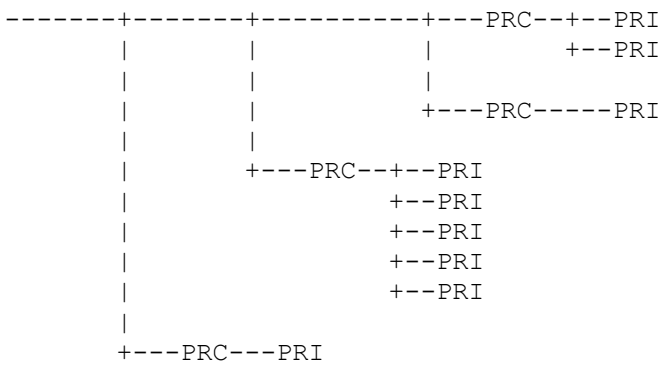


Figure 2 : Arborescence de PIB

Les instances de classes de politique (PRI) sont identifiées par un identifiant d'instance d'approvisionnement (PRID, *Provisioning Instance Identifier*). Un PRID est un nom, porté dans un objet COPS <ClientSI désigné> ou <Données de déccision désignées>, qui identifie une instance particulière d'une classe.

2.1 Règles pour la modification et l'extension des PIB

Avec plus d'expérience de la gestion fondée sur la politique, et lorsque apparaîtront de nouvelles exigences, il sera nécessaire d'apporter des changements aux PIB. Le changement à une PIB existante peut être faits de plusieurs façons.

(1) Des PRC supplémentaires peuvent être ajoutées à une PIB ou une PRC existante peut être déconseillée.
(2) Des attributs peuvent être ajoutés à une PRC existante, ou être déconseillés.
(3) Une PRC existante peut être étendue ou augmentée d'une nouvelle PRC définie dans une autre PIB (peut-être spécifique d'une entreprise).

Les règles pour chacun de des mécanismes d'extension sont décrites dans les paragraphes qui suivent. Tous ces mécanismes pour modifier une PIB permettent l'interopérabilité entre les PDP et les PEP même lorsque une partie utilise une nouvelle version de la PIB alors que l'autre utilise une ancienne version.

Noter que la SPPI [RFC3159] fournit les règles d'autorité pour mettre à jour les PIB codées en BER. L'objet de la section qui suit est d'expliquer comment de tels changements affectent les envoyeurs et les receveurs de messages COPS.

2.2 Ajout ou refus de PRC à une PIB

Une PIB publiée peut être étendue par de nouvelles PRC en révisant simplement le document et en y ajoutant des PRC supplémentaires. Ces PRC supplémentaires sont aisément identifiées avec les nouveaux PRID sous le préfixe PRID du module.

Dans le cas où un PEP qui met en œuvre la nouvelle PIB est configuré par un PDP qui met en œuvre l'ancienne PIB, le PEP va simplement ne recevoir aucune instance de la nouvelle PRC. Dans le cas où le PEP met en œuvre l'ancienne PIB et le PDP la nouvelle, le PEP peut recevoir des PRI pour la nouvelle PRC. Dans ces conditions, le PEP DOIT retourner une erreur au PDP, et revenir à son (bon) état précédent.

De même, les PRC existantes peuvent être déconseillées dans une PIB. Dans ce cas, le PEP ignore tout PRI qui lui est envoyé par un PDP qui met en œuvre l'ancienne version (non déconseillée) de la PIB. Un PDP qui met en œuvre la nouvelle version de la PIB n'envoie simplement pas d'instance de la classe déconseillée.

2.2.1 Ajout ou refus des attributs d'un PRC codé en BER

Une PIB peut être modifiée pour déconseiller un attribut existant d'une PRC ou pour en ajouter de nouveaux.

Lorsque on déconseille les attributs d'une PRC, on doit se souvenir qu'avec le protocole COPS-PR, les attributs de la PRC sont identifiés par leur ordre dans la séquence plutôt que par une étiquette explicite (ou un OID d'attribut). Par conséquent, une valeur ASN.1 DOIT être envoyée même pour les attributs déconseillées afin qu'un PDP et un PEP qui mettent en œuvre des versions différentes de la PIB soient interopérables.

Pour un attribut déconseillé, si le PDP utilise une PIB codée en BER, le PDP DOIT envoyer soit une valeur ASN.1 du type correct, soit il peut envoyer une valeur ASN.1 NULLE. Un PEP qui reçoit un ASN.1 NUL pour un attribut qui n'est pas déconseillé DEVRAIT lui substituer une valeur par défaut. Si il n'a pas de valeur par défaut à lui substituer, il DOIT retourner une erreur au PDP.

Lors de l'ajout de nouveaux attributs à une PIB, ces nouveaux attributs doivent être ajoutés en séquence après ceux existants. Un PEP qui reçoit une PRI avec plus d'attributs qu'il n'en attend DOIT ignorer les attributs supplémentaires et renvoyer un avertissement au PDP.

Un PEP qui reçoit une PRI avec moins d'attributs qu'il n'en attend DEVRAIT supposer des valeurs par défaut pour les attributs manquants. Il PEUT renvoyer un avertissement au PDP. Si les attributs manquants sont exigés, et si il n'y a pas de valeur pas défaut convenable, le PEP DOIT renvoyer une erreur au PDP. Dans tous les cas, les attributs manquants sont supposés correspondre aux derniers attributs du PRC.

2.3 Opérations de COPS prises en charge pour une instance d'approvisionnement

Une instance d'approvisionnement (PRI) contient normalement une valeur pour chaque attribut défini pour la PRC de laquelle il est une instance et elle est identifiée de façon univoque, au sein de la portée d'un certain type de client COPS et d'état de demande sur un PEP, par un identifiant d'instance d'approvisionnement (PRID). Les opérations COPS suivantes sont prises en charge sur une PRI :

- Install - Cette opération crée ou met à jour une instance nommée d'une PRC. Elle inclut deux paramètres : un objet PRID pour nommer la PRI et un objet Données codées d'instance d'approvisionnement (EPD, *Encoded Provisioning Instance Data*) avec les valeurs nouvelles/mises à jour. La valeur de PRID DOIT identifier de façon univoque une seule PRI (c'est-à-dire que le préfixe de PRID ou les valeurs de PRC sont illégales). Les mises à jour à une PRI existante sont réalisées simplement en réinstallant le même PRID avec les données d'EPD mises à jour.
- Remove - Cette opération est utilisée pour supprimer une instance d'une PRC. Elle inclut un paramètre, un objet PRID, qui désigne soit la PRI individuelle à supprimer, soit un préfixe de PRID qui désigne une ou plusieurs classes complètes de PRI. La suppression fondée sur le préfixe prend efficacement en charge le retrait d'une politique brute. La suppression d'un PRID inconnu/non existant DEVRAIT résulter en un avertissement au PDP (pas d'erreur).

3. Contenu de message

Le protocole COPS permet à différents clients COPS de définir leur propre "désignation", c'est-à-dire, des informations spécifiques du client pour divers messages. Cette section décrit les messages échangés entre un serveur COPS (PDP) et les clients COPS d'approvisionnement de politique (PEP) qui portent des objets de données spécifiques du client. Tous les messages COPS utilisés par COPS-PR se conforment aux spécifications de message définies dans le protocole COPS de base [RFC2748].

Note : L'utilisation du caractère '*' représenté dans le présent document est cohérente avec l'ABNF [RFC2234] et signifie zéro, une ou plusieurs des entités suivantes.

3.1 Demande (REQ) PEP -> PDP

Le message REQ est envoyé par les clients d'approvisionnement de politique pour produire une 'demande de configuration' au PDP comme spécifié dans l'objet Contexte COPS. Le bride de client *(Client Handle)* associée au message REQ générée par un client d'approvisionnement DOIT être unique pour ce client. La bride de client est utilisée pour identifier un état de demande spécifique. Donc, un client peut éventuellement ouvrir plusieurs états de demande de configuration, chacun identifié de façon univoque par sa bride. Des états de demande différents sont utilisés pour isoler des informations de configuration désignées de façon similaire dans des contextes qui ne se chevauchent pas (ou des espaces de noms logiquement isolés). Donc, une instance d'informations désignées est unique par rapport à un type de client particulier et est unique par rapport à un état de demande particulier pour ce type de client, même si les informations étaient identifiées de façon similaire dans d'autres états de demande (c'est-à-dire, utilisent le même PRID). Donc, la bride de client fait aussi partie de l'identification d'instance des informations de configuration communiquées.

Le message de demande de configuration sert de demande du PEP au PDP de données de politique d'approvisionnement que le PDP peut avoir pour le PEP, comme des listes de contrôle d'accès, etc. Cela inclut la politique que le PDP peut avoir au moment où la demande est reçue aussi bien que toutes données de politique futures ou mises à jour de ces données.

Le message de demande de configuration devrait inclure les informations de client d'approvisionnement pour fournir au PDP des informations de configuration ou de capacités spécifiques du client sur le PEP. Les informations fournies par le PEP devraient inclure les ressources du client (par exemple, les capacités de mise en file d'attente) et les informations de configuration de politique par défaut (par exemple, les combinaisons de rôle par défaut) ainsi que les données d'incarnation sur les politiques existantes. Ces informations n'incluent normalement pas les informations précédemment installées par un PDP mais plutôt devraient inclure des sommes de contrôle ou des références abrégées à des informations précédemment installées pour des besoins de synchronisation. Ces informations du client assistent le serveur pour décider quels types de politiques le PEP peut installer et mettre en application. Le format des informations encapsulées dans un ou plusieurs des objets COPS ClientSI désignés est décrit à la Section 5. Noter que le ou les messages de demande de configuration sont générés et envoyés au PDP en réponse à la réception d'un message Demande d'état de synchronisation (SSQ, *Synchronize State Request*) provenant du PDP. De même, un message de demande de configuration mis à jour (utilisant la même valeur de bride de client que la demande d'origine qui est maintenant mise à jour) peut aussi être généré par le PEP et envoyé au PDP à tout moment à cause de modifications locales de l'état interne du PEP. De cette façon, le PDP va être synchronisé avec l'état interne pertinent du PEP à tout moment.

Les informations de politique fournies par le PDP DOIVENT être cohérentes avec les données de décision désignées définies pour le client d'approvisionnement de politique. Le PDP répond à la demande de configuration par un message DEC contenant les données de politique d'approvisionnement disponibles.
Le message REQ a le format suivant :

<Demande> ::= <En-tête commun> <Bride de client> <Contexte= demande de configuration> *(<ClientSI désigné>)
[<Intégrité>]

Noter que les objets COPS IN-Int, OUT-Int et LPDPDecisions ne sont pas inclus dans une demande COPS-PR.

3.2 Décision (DEC) PDP -> PEP

Le message DEC est envoyé du PDP à un client d'approvisionnement de politique en réponse au message REQ reçu du PEP. La bride de client DOIT être la même bride reçue dans le message REQ correspondant.

Le message DEC est envoyé comme réponse immédiate à une demande de configuration avec le fanion de message sollicité établi *(à 1)* dans l'en-tête de message COPS. Les messages DEC suivants peuvent aussi être envoyés à tout moment après le message DEC d'origine pour fournir au PEP des informations supplémentaires de politique mise à jour sans que le fanion de message sollicité soit établi dans l'en-tête de message COPS (car ce sont des décisions non sollicitées).

Chaque message DEC peut contenir plusieurs décisions. Cela signifie qu'un seul message peut installer des politiques et en supprimer d'autres. En général, un seul message DEC COPS-PR DOIT contenir d'abord toutes les décisions de suppression exigées, suivies par toutes les décisions d'installation exigées. Ceci est utilisé pour résoudre une question de préséance, et non une question de programmation : la décision de suppression retire ce qu'elle spécifie, sauf les éléments qui sont installés dans le même message.

Le message DEC peut aussi être utilisé par le PDP pour commander au PEP d'ouvrir un nouvel état de demande ou de supprimer un état de demande existant selon ce qui est identifié dans la bride de client. Pour accomplir cela, COPS-PR définit un nouveau fanion pour l'objet Fanion de décision COPS. Le fanion 0x02 est à utiliser par les types de client COPS-PR et on le désigne ci-après sous le nom de fanion "État de demande". Une décision Install (Fanion de décision : Code de commande = Install) avec le fanion État de demande établi dans l'objet Fanions de décision COPS cause la production par le PEP d'une nouvelle demande avec une nouvelle bride de client, ou autrement, spécifie l'erreur appropriée dans un message Rapport COPS. Une décision Remove (Fanion de décision : Code de commande = Remove) avec le fanion État de demande établi dans l'objet Fanions de décision COPS va être cause que le PEP envoie un message COPS État suppression de demande (DRQ, *Delete Request State*) pour l'état de demande identifié par la bride de client dans le message DEC. Chaque fois que le fanion État de demande est établi dans l'objet Fanions de décision COPS dans le message DEC, aucun objet COPS Données de décision désignées ne peut être inclus dans la décision correspondante (car elle ne sert à rien pour ce fanion de décision). Noter qu'une seule décision peut être présente avec le fanion État de demande par message DEC, et, si elle est présente, elle DOIT être la seule décision dans ce message. Comme on le décrit plus loin, le PEP DOIT répondre à chaque DEC par un rapport sollicité correspondant.

Un message DEC COPS-PR DOIT être traité comme une seule "transaction", c'est-à-dire que soit toutes les décisions dans un message DEC réussissent, soit elles échouent toutes. Si elles échouent, le PEP va revenir à son bon état précédent, qui est la dernière transaction DEC réussie, si il en est. Cela permet au PDP de ne supprimer des politiques que si d'autres politiques peuvent être installées à leur place. Le message DEC a le format suivant :

<Message Décision> ::= <En-tête commun> <bride de client> *(<Décision>) | <Erreur> [<Intégrité>]

<Décision> ::= <Contexte> <Fanions Décision:> [<Données de décision désignées: Approvisionnement >]

Noter que l'objet Données de décision désignées (Approvisionnement) est inclus dans une décision COPS-PR lorsque c'est une décision Install ou Remove avec aucun fanion Décision établi. Les autres types d'objets de données de décision COPS (par exemple, Sans état, Remplacement) ne sont pas prises en charge par les types de client COPS-PR. L'objet Données de décision désignées NE DOIT PAS être inclus dans la décision si le code de commande de l'objet Fanions de décision est NUL (signifiant qu'il n'y a pas d'informations de configuration à installer pour le moment) ou si le fanion État de demande est établi dans l'objet Fanions de décision.

Pour chaque décision dans le message DEC, le PEP effectue l'opération spécifiée dans le code de commande et le champ Fanions dans l'objet Fanions de décision sur les Données de décision désignées. Pour le client d'approvisionnement de politiques, le format de ces données est défini dans le contexte de la base de données d'informations de politique (voir à la Section 5). En réponse à un message DEC, le client d'approvisionnement de politique DOIT renvoyer un message RPT *(rapport)* avec le fanion Message sollicité établi, au PDP pour l'informer de l'action entreprise.

3.3 État de rapport (RPT) PEP -> PDP

Le message RPT est envoyé du client d'approvisionnement de politiques au PDP pour rapporter des informations de comptabilité associées à la politique provisionnée, ou pour notifier au PDP des changements dans le PEP (Type de rapport = 'Accounting') en relation avec le client approvisionneur.

RPT est aussi utilisé comme mécanisme pour informer le PDP de l'action entreprise au PEP en réponse à un message DEC. Par exemple, en réponse à une décision 'Install', le PEP indique au PDP si les données de politique sont installées (Type de rapport = 'Succès') ou non (Type de rapport = 'Échec'). Les rapports qui sont en réponse à un message DEC DOIVENT établir le fanion Message sollicité dans leur en-tête de message COPS. Chaque rapport sollicité DOIT être envoyé pour son DEC correspondant dans l'ordre où les messages DEC ont été reçus. En cas d'échec sollicité, ont attend du PEP qu'il revienne à son (bon) état précédent comme si la transaction DEC erronée ne s'était pas produite. Le PEP DOIT toujours répondre à un DEC par un RPT sollicité même en réponse à un DEC NUL, auquel cas, le type de rapport sera 'Réussite'.

Les rapports peuvent aussi être non sollicités et tous les rapports non sollicités NE DOIVENT PAS établir le fanion Message sollicité dans leur en-tête de message COPS. Les exemples de rapports non sollicités incluent les types de rapport 'Comptabilité', qui n'étaient pas déclanchés par des messages DEC spécifiques, ou les types de rapport 'Échec', qui indiquent un échec dans une configuration précédemment installée avec succès (noter que, dans le cas de tels échecs non sollicités, le PEP ne peut pas revenir à un "bon" état antérieur car dans ces conditions asynchrones, ce que l'état correct pourrait être n'est pas évident).

Le message RPT peut contenir des informations de client d'approvisionnement comme des paramètre comptables ou des erreurs/avertissements relatifs à une décision. Le format des données pour cette information est défini dans le contexte de la base de données d'informations de politique (voir à la Section 5). Le message RPT a le format suivant :

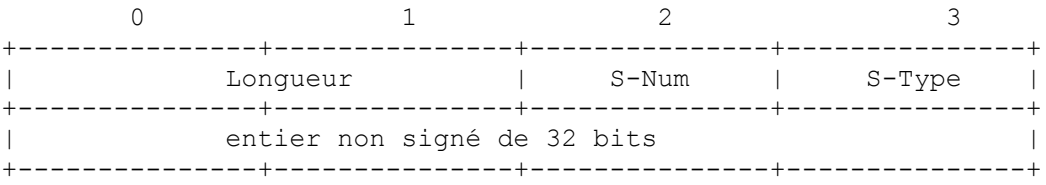
<État de rapport > ::= <En-tête commun> <bride de client> <Type de rapport> *(<ClientSI désigné>) [<Intégrité>]

4. Objets du protocole COPS-PR

Les clients d'approvisionnement de politique COPS encapsulent plusieurs nouveaux objets au sein des objets COPS Informations spécifiques du client désigné et Données de décision désignées existants. La présente section définit le format de ces nouveaux objets.

COPS-PR classe les données de politique en fonction de "liens", où un lien consiste en un identifiant d'instance d'approvisionnement et en données d'instance d'approvisionnement, codées dans le contexte de la base de données d'informations de politique d'approvisionnement (voir à la Section 5).

Le format de ces nouveaux objets est le suivant :



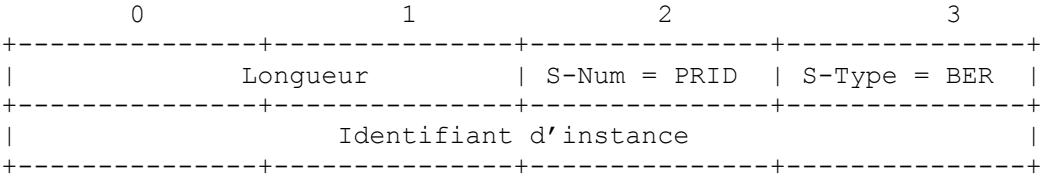
Le S-Num et le S-Type sont similaires aux C-Num et C-Type utilisés dans l'objet COPS de base. La différence est que S-Num et S-Type ne sont utilisés que pour les clients COPS-PR et sont encapsulés au sein des objets COPS ClientSI désigné ou Données de décision désignées existants. Le S-Num identifie l'objectif général de l'objet, et le S-Type décrit le codage spécifique utilisé pour l'objet. Toutes les descriptions et tous les exemples d'objet dans ce document utilisent les règles de codage de base (BER) comme type de codage (S-Type = 1). Des codages supplémentaires peuvent être définis pour les S-Type restants à l'avenir (par exemple, un S-Type supplémentaire pourrait être utilisé pour porter des codages fondés sur la chaîne XML [XML] comme un EPD de données d'instance de PRI, où les URN identifient les PRC [RFC2141] et où des XPointers seraient utilisés pour les PRID).

Longueur est une valeur de deux octets qui décrit le nombre des octets (y compris l'en-tête) qui composent l'objet. Si la longueur en octets ne tombe pas sur une limite de mot de 32 bits, un bourrage DOIT être ajouté à la fin de l'objet afin qu'il soit aligné sur la prochaine limite de 32 bits avant que l'objet puisse être envoyé sur le réseau. Du côté receveur, la limite de l'objet suivant sera trouvée en arrondissant simplement la longueur déclarée d'objet de l'objet en cours sur la prochaine limite de 32 bits. Les valeurs du bourrage DOIVENT être toutes de zéros.

4.1 Identifiant complet d'instance d'approvisionnement (PRID)

S-Num = 1 (PRID complet), S-Type = 1 (BER), Longueur = variable.

Cet objet est utilisé pour porter l'identifiant, ou PRID, d'une instance d'approvisionnement. L'identifiant est codé suivant les règles qui ont été définies pour le codage des valeurs d'identifiant d'objet (OID, *Object Identifier*) SNMP . Précisément, les valeurs de PRID sont codées en utilisant le format de type/longueur/valeur (TLV) et le paquetage de sous identifiant initial qui est spécifié par les règles de codage binaire [BER] utilisées pour les identifiants d'objet dans une PDU SNMP.



Par exemple, un PRID (fictif) égal à 1.3.6.1.2.2.8.1 serait codé comme suit (valeurs en hexadécimal) :

06 07 2B 06 01 02 02 08 01

L'objet PRID entier serait codé comme suit :

00 0D	- Longueur01 - S-Num
01	- S-Type (PRID complet)
06 07 2B 06 01 02 02 08 01	-PRID codé
00 00 00	- Bourrage

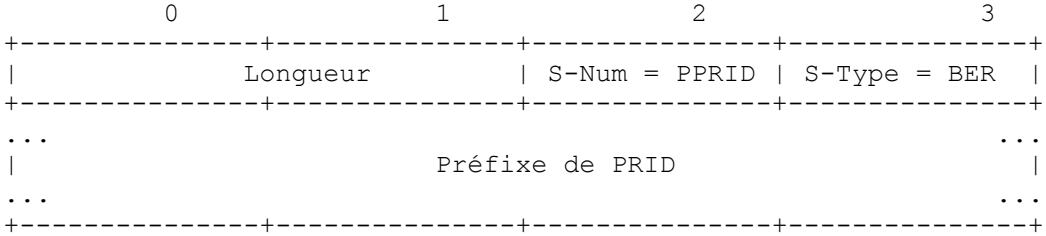
Note : Lors du codage d'un type d'objet xxxEntry d'une xxxTable comme défini par le SMI [RFC2578] et SPPI [RFC3159], l'OID va contenir tous les sous identifiants jusque et y compris l'OID de xxxEntry mais pas les identifiants colonnaires pour les attributs au sein de la séquence de xxxEntry. Le dernier identifiant (suffixe) est l'INDEX d'une instance de la xxxEntry entière incluant sa SEQUENCE d'attributs codés dans le EPD (défini plus loin). Cela constitue une instance (PRI) d'une classe (PRC) dans les termes du SMI.

Un PRID pour l'OID d'une valeur scalaire (non colonnaire) est codé directement comme la PRC où le suffixe de l'identifiant d'instance est toujours zéro car il y aura seulement une instance d'une valeur scalaire. Le EPD sera alors utilisé pour convoyer la valeur scalaire.

4.2 Préfixe de PRID (PPRID)

Certaines opérations, comme les retraits de décision, peuvent être optimisées en spécifiant un préfixe de PRID avec l'intention que l'opération demandée soit appliquée à toutes les PRI qui correspondent au préfixe (par exemple, toutes les instances de la même PRC). Les objets Préfixe de PRID DOIVENT être utilisés seulement dans l'opération de protocole COPS <Retrait de décision> lorsque il peut être optimal d'effectuer un retrait de décision en bloc en utilisant les préfixes de classe au lieu d'une séquence d'opérations individuelle de <Retrait de décision>. D'autres opérations COPS, par exemple, les opérations <Installer une décision> exigent toujours une spécification de PRID individuelle.

S-Num = 2 (Préfixe de PRID), S-Type = 1 (BER), Longueur = variable.



En continuant avec l'exemple précédent, un préfixe de PRID qui est égal à 1.3.6.1.2.2 serait codé comme suit (les valeurs sont en hexadécimal) : 06 05 2B 06 01 02 02

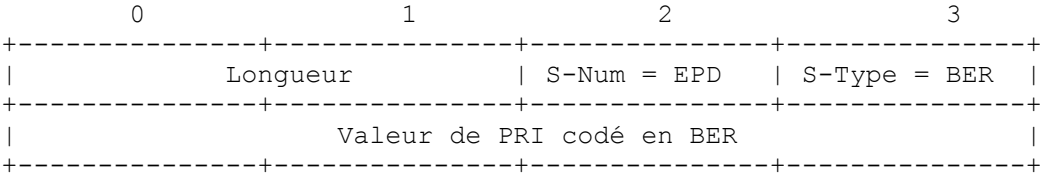
L'objet PPRID entier serait codé comme suit :

00 0B	- Longueur
02	- S-Num = PRID de préfixe
01	- S-Type = BER
06 05 2B 06 01 02 02	- préfixe de PRID codé
00	- bourrage

4.3 Données d'instance d'approvisionnement codées (EPD)

S-Num = 3 (EPD), S-Type = 1 (BER), Longueur = variable.

Cet objet est utilisé pour porter la valeur codée d'une instance d'approvisionnement. La valeur PRI, qui contient toutes les valeurs individuelles des attributs qui composent la classe (qui correspond au type d'objet xxxEntry de SMI définissant la SEQUENCE des attributs composant un tableau [RFC2578], [RFC3159]) est codée comme une série de sous composants TLV. Chaque sous composant représente la valeur d'un seul attribut et suit les BER. Noter que l'ordre des attributs non scalaires (multiples) au sein des EPD est dicté par leurs suffixes d'OID colonnaires respectifs quand ils sont définis dans la [RFC2578]. Donc, l'attribut qui a le plus petit suffixe d'OID colonnaire va apparaître en premier et l'attribut avec le plus fort numéro de suffixe d'OID colonnaire sera le dernier.



À titre d'exemple,une définition fictive d'une classe de filtre de paquet IPv4 pourrait être décrite en utilisant la SMI comme suit :

IDENTIFIANT D'OBJET ipv4FilterIpFilter ::= { someExampleOID 1 }

-- Tableau de filtre IP

TYPE D'OBJET	ipv4FilterTable
SYNTAXE	SEQUENCE DE Ipv4FilterEntry
MAX-ACCESS	non accessible
STATUT	actuel

DESCRIPTION : "Définitions de filtre. Un paquet doit correspondre à tous les champs dans un filtre. Des caractères génériques peuvent être spécifiés pour les champs qui ne sont pas pertinents."
::= { ipv4FilterIpFilter 1 }

TYPE D'OBJET ipv4FilterEntry
SYNTAXE Ipv4FilterEntry
MAX-ACCESS non accessible
STATUT actuel
DESCRIPTION : "Une instance de la classe de filtre."
INDEX { ipv4FilterIndex }
::= { ipv4FilterTable 1 }

```
Ipv4FilterEntry ::= SEQUENCE {
    ipv4FilterIndex          Unsigned32,
    ipv4FilterDstAddr        IpAddress,
    ipv4FilterDstAddrMask    IpAddress,
    ipv4FilterSrcAddr        IpAddress,
    ipv4FilterSrcAddrMask    IpAddress,
    ipv4FilterDscp           Integer32,
    ipv4FilterProtocol       Integer32,
    ipv4FilterDstL4PortMin   Integer32,
    ipv4FilterDstL4PortMax   Integer32,
    ipv4FilterSrcL4PortMin   Integer32,
    ipv4FilterSrcL4PortMax   Integer32,
    ipv4FilterPermit         TruthValue
}
```

TYPE D'OBJET ipv4FilterIndex
SYNTAXE Unsigned32
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION "Indice entier pour identifier de façon univoque le filtre parmi tous les filtres."
::= { ipv4FilterEntry 1 }

TYPE D'OBJET ipv4FilterDstAddr
SYNTAXE IpAddress
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Adresse IP à comparer à l'adresse IP de destination du paquet."
::= { ipv4FilterEntry 2 }

TYPE D'OBJET ipv4FilterDstAddrMask
SYNTAXE IpAddress
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Gabarit pour la correspondance de l'adresse IP de destination. Un bit zéro dans le gabarit signifie que le bit correspondant dans l'adresse correspond toujours."
::= { ipv4FilterEntry 3 }

TYPE D'OBJET ipv4FilterSrcAddr
SYNTAXE IpAddress
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Adresse IP à comparer à l'adresse IP de source du paquet."
::= { ipv4FilterEntry 4 }

TYPE D'OBJET ipv4FilterSrcAddrMask
SYNTAXE IpAddress
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Gabarit pour la correspondance de l'adresse IP de source."
::= { ipv4FilterEntry 5 }

TYPE D'OBJET ipv4FilterDscp
SYNTAXE Integer32 (-1 | 0..63)
MAX-ACCESS lecture-écriture

STATUT actuel
DESCRIPTION : "Valeur que peut avoir et à laquelle peut correspondre le DSCP dans le paquet. Une valeur de -1 indique que n'a pas été définie une valeur spécifique de DSCP et que donc toutes les valeurs de DSCP sont considérées comme correspondantes."
::= { ipv4FilterEntry 6 }

TYPE D'OBJET ipv4FilterProtocol
SYNTAXE Integer32 (0..255)
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Le protocole IP à comparer au protocole du paquet. Une valeur de zéro signifie que tout correspond."
::= { ipv4FilterEntry 7 }

TYPE D'OBJET ipv4FilterDstL4PortMin
SYNTAXE Integer32 (0..65535)
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Valeur minimum que peut avoir le numéro d'accès de destination de couche 4 de ce paquet et à laquelle correspond ce filtre."
::= { ipv4FilterEntry 8 }

TYPE D'OBJET ipv4FilterDstL4PortMax
SYNTAXE Integer32 (0..65535)
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Valeur maximum que peut avoir le numéro d'accès de destination de couche 4 de ce paquet et à laquelle correspond ce filtre."
::= { ipv4FilterEntry 9 }

TYPE D'OBJET ipv4FilterSrcL4PortMin
SYNTAXE Integer32 (0..65535)
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Valeur minimum que peut avoir le numéro d'accès de source de couche 4 de ce paquet et à laquelle correspond ce filtre."
::= { ipv4FilterEntry 10 }

TYPE D'OBJET ipv4FilterSrcL4PortMax
SYNTAXE Integer32 (0..65535)
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "Valeur maximum que peut avoir le numéro d'accès de source de couche 4 de ce paquet et à laquelle correspond ce filtre."
::= { ipv4FilterEntry 11 }

TYPE D'OBJET ipv4FilterPermit
SYNTAXE TruthValue
MAX-ACCESS lecture-écriture
STATUT actuel
DESCRIPTION : "À faux, l'évaluation est niée. C'est-à-dire qu'une correspondance valide sera évaluée comme une non correspondance et vice versa."
::= { ipv4FilterEntry 12 }

Une instance fictive de la classe de filtre définie ci-dessus pourrait alors être codée comme suit :

```
02 01 08              :ipv4FilterIndex/Unsigned32/Valeur = 8
40 04 C0 39 01 05     :ipv4FilterDstAddr/IpAddress/Valeur = 192.57.1.5
40 04 FF FF FF FF     :ipv4FilterDstMask/IpAddress/Valeur=255.255.255.255
40 04 00 00 00 00     :ipv4FilterSrcAddr/IpAddress/Valeur = 0.0.0.0
40 04 00 00 00 00     :ipv4FilterSrcMask/IpAddress/Valeur = 0.0.0.0
02 01 FF              :ipv4FilterDscp/Integer32/Valeur = -1 (non utilisé)
02 01 06              :ipv4FilterProtocol/Integer32/Valeur = 6 (TCP)
05 00                 :ipv4FilterDstL4PortMin/NULL/non pris en charge
05 00                 :ipv4FilterDstL4PortMax/NULL/non pris en charge
```

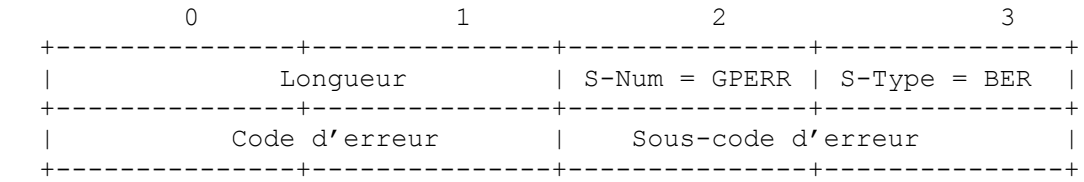
05 00 :ipv4FilterSrcL4PortMin/NULL/non pris en charge
05 00 :ipv4FilterSrcL4PortMax/NULL/non pris en charge
02 01 01 :ipv4FilterPermit/TruthValue/Valeur = 1 (vrai)

L'objet EPD entier pour cette instance serait alors codé comme suit :
00 30 - Longueur
03 - S-Num = EPD
01 - S-Type = BER
02 01 08 - ipv4FilterIndex
40 04 C0 39 01 05 - ipv4FilterDstAddr
40 04 FF FF FF FF - ipv4FilterDstMask
40 04 00 00 00 00 - ipv4FilterSrcAddr
40 04 00 00 00 00 - ipv4FilterSrcMask
02 01 FF - ipv4FilterDscp
02 01 06 - ipv4FilterProtocol
05 00 - ipv4FilterDstL4PortMin
05 00 - ipv4FilterDstL4PortMax
05 00 - ipv4FilterSrcL4PortMin
05 00 - ipv4FilterSrcL4PortMax
02 01 01 - ipv4FilterPermit

Noter que les attributs non acceptés dans une classe sont quand même retournés dans le EPD pour une PRI. Par convention, une valeur NUL est retournée pour les attributs qui ne sont pas pris en charge. Dans l'exemple précédent, les attributs de numéro d'accès de source et de destination ne sont pas pris en charge.

4.4 Objet Erreur d'approvisionnement global (GPERR)

S-Num = 4 (GPERR), S-Type = 1 (pour BER), Longueur = 8.

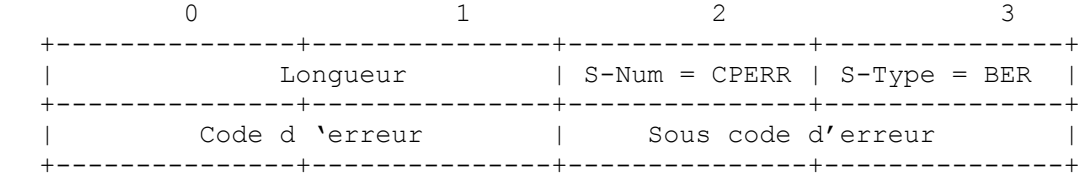


L'objet Erreur globale d'approvisionnement a le même format que l'objet Erreur dans COPS [RFC2748], sauf que C-Num et C-Type sont remplacés par les valeurs de S-Num et S-Type montrées. L'objet Erreur globale d'approvisionnement est utilisé pour communiquer des erreurs générales qui ne se transposent pas en une PRC spécifique.

Les codes d'erreur globale suivant sont définis :
availMemLow(1) *(- La mémoire disponible est faible)*
availMemExhausted(2) *(- La mémoire disponible est épuisée)*
unknownASN.1Tag(3) - Le type d'étiquette erronée DEVRAIT être spécifié dans le champ Sous-code d'erreur.
maxMsgSizeExceeded(4) - Le message COPS (la transaction) est trop gros.
unknownError(5) *(- Erreur inconnue)*
maxRequestStatesOpen(6) - Aucun autre état de demande ne peut être créé par le PEP (en réponse à un message DEC tentant d'ouvrir un nouvel État de demande).
invalidASN.1Length(7) - La longueur d'un objet ASN.1 était incorrecte.
invalidObjectPad(8) - Le bourrage de l'objet est incorrect.
unknownPIBData(9) - Certaines des données fournies par le PDP sont inconnues/non acceptées par le PEP (bien que formatées correctement). Les codes d'erreur spécifiques de PRC sont à utiliser pour donner plus d'informations.
unknownCOPSPRObject(10) - Le sous-code (octet 2) contient un S-num d'objet inconnu et (octet 3) contient un S-Type d'objet inconnu.
malformedDecision(11) - La décision n'a pas pu être analysée.

4.5 Objet Erreur d'approvisionnement de classe de PRC (CPERR)

S-Num = 5 (CPERR), S-Type = 1 (for BER), Longueur = 8.



L'objet Erreur d'approvisionnement spécifique d'une classe a le même format que l'objet Erreur dans COPS [RFC2748], excepté pour C-Num et C-Type qui sont remplacés par les valeurs de S-Num et S-Type montrées. L'objet Erreur spécifique de classe est utilisé pour communiquer des erreurs relatives à des PRC spécifiques et DOIT avoir un objet PRID d'erreur associé.

Les erreurs génériques spécifiques de classe sont définies :

priSpaceExhausted(1) - Aucune autre instance ne peut actuellement être installée dans cette classe.
priInstanceInvalid(2) - L'instance de classe spécifiée est actuellement invalide ce qui interdit l'installation ou la suppression.
attrValueInvalid(3) - La valeur spécifiée pour des attributs identifiés est illégale.
attrValueSupLimited(4) - La valeur spécifiée pour l'attribut identifié est légale mais pas prise en charge actuellement par l'appareil.
attrEnumSupLimited(5) - L'énumération spécifiée pour l'attribut identifié est légale mais pas prise en charge actuellement par l'appareil.
attrMaxLengthExceeded(6) - La longueur globale de la valeur spécifiée pour l'attribut identifié excède les limitations de l'appareil.
attrReferenceUnknown(7) - L'instance de classe spécifiée par l'identifiant d'instance de politique n'existe pas.
priNotifyOnly(8) - La classe n'est actuellement prise en charge que pour l'utilisation par des messages de demande ou de rapport, ce qui interdit l'installation de décision.
unknownPrc(9) - Tentative d'installer une PRI d'une classe non prise en charge par le PEP.
tooFewAttrs(10) - La PRI reçue a moins d'attributs qu'exigé.
invalidAttrType(11) - La PRI reçue a un attribut d'un mauvais type.
deletedInRef(12) - La PRI supprimée est encore référencée par d'autres PRI non supprimées.
priSpecificError(13) - Le champ Sous code d'erreur contient le code d'erreur spécifique de la PRC

Lorsque approprié (erreurs 3, 4, 5, 6, 7 ci-dessus) les sous codes d'erreur DEVRAIENT identifier le sous identifiant d'OID de l'attribut associé à l'erreur.

4.6 Objet Erreur de PRID (ErrorPRID)

S-Num = 6 (ErrorPRID), S-Type = 1 (BER), Longueur = variable.

Cet objet est utilisé pour porter l'identifiant, ou PRID, d'une instance d'approvisionnement qui a causé une erreur d'installation ou n'a pas pu être installée ou supprimée. L'identifiant est codé et formaté exactement comme dans l'objet PRID comme décrit au paragraphe 4.1.

5. Formats de données spécifiques de client COPS-PR

La présente section décrit le format des informations spécifiques du client désigné pour le client d'approvisionnement de politique COPS. Les formats de ClientSI sont définis pour l'objet Données de décision désignées du message Décision, pour l'objet ClientSI désigné du message Demande et pour l'objet ClientSI désigné du message Rapport. Le contenu réel des données est défini par la base de données d'informations de politique pour un type de client d'approvisionnement spécifique (voir ci-dessous).

5.1 Données de décision nommée

Les formats encapsulés par l'objet Données de décision désignées pour les types de client d'approvisionnement de politique dépendent du type de décision. Install et Remove sont les deux types de décisions qui dictent le format interne de l'objet COPS Données de décision désignées et ils exigent sa présence. Install et Remove se réfèrent respectivement aux codes de commande 'Install' et 'Remove', spécifiés dans l'objet COPS Fanions de décision lorsque aucun fanion de décision n'est établi. Les données sont en général composées d'un ou plusieurs liens. Chaque lien associe un objet PRID et un objet EPD.

L'objet PRID est toujours présent dans les deux décisions Install et Remove. L'objet EPD DOIT être présent dans le cas d'une décision Install et NE DOIT PAS être présent dans le cas d'une décision Remove.
Le format de ces données est encapsulé au sein de l'objet COPS Données de décision désignées comme suit :

<Données de décision désignées > ::= <<Décision Install> | <Décision Remove>><Décision Install> ::= *(<PRID> <EPD>)
<Décision Remove> ::= *(<PRID>|<PPRID>)

Noter que les objets PRID dans une décision Remove peuvent spécifier des valeurs de préfixe de PRID. La suppression explicite et implicite des politiques installées est prise en charge par un client. Les données de la décision Install DOIVENT être explicites (c'est-à-dire que les valeurs de préfixe de PRID sont illégales et DOIVENT être rejetées par un client).

5.2 Données de demande ClientSI

Les données de demande du client d'approvisionnement vont utiliser les mêmes liens que décrits ci-dessus. Le format de ces données est encapsulé dans l'objet COPS ClientSI désigné comme suit :

<Demande de ClientSI: désigné> ::= <*(<PRID> <EPD>)>

5.3 Données de rapport d'approvisionnement de politique

L'objet COPS ClientSI désigné est utilisé dans le message RPT en conjonction avec l'objet COPS d'accompagnement Type de rapport pour encapsuler les informations de rapport COPS-PR provenant du PEP au PDP. Les types de rapport peuvent être 'Succès' ou 'Échec', indiquant au PDP le succès ou l'échec de l'installation/suppression d'un ensemble particulier de politiques d'approvisionnement sur le PEP, ou 'Comptabilité'.

5.3.1 Format de données de type de rapport de réussite ou d'échec

Les types de rapport peuvent être 'Succès' ou 'Échec' pour indiquer au PDP le succès ou l'échec de l'installation/suppression d'un ensemble particulier de politiques d'approvisionnement sur le PEP. Les données de rapport d'approvisionnement consistent en les liens décrits ci-dessus et des informations d'erreur/avertissement globales et spécifiques. Les erreurs spécifiques sont associées à une instance particulière. Pour un type de rapport de 'Succès', une erreur spécifique est une indication d'un avertissement relatif à une politique spécifique qui a été installée, mais qui n'est pas pleinement mise en œuvre (par exemple, ses paramètres ont été approximés) comme identifié par l'objet ErrorPRID. Pour un type de rapport 'Échec', c'est un code d'erreur spécifique d'un lien, là encore, identifié par l'objet ErrorPRID. Des erreurs spécifiques peuvent aussi inclure des liens réguliers <PRID><EPD> pour porter des informations supplémentaires d'une façon générique afin que les erreurs/avertissements spécifiques puissent être décrits de façon plus littérale et associés à l'objet ErrorPRID erroné.

Les erreurs globales ne sont pas liées à un ErrorPRID spécifique. Dans un message de rapport 'Succès', une erreur globale est une indication d'un avertissement général au niveau du PEP (par exemple, mémoire faible). Dans un message RPT 'Échec', c'est une indication d'une erreur générale au niveau du PEP (par exemple, mémoire saturée).

Dans le cas d'un type de rapport 'Échec', le PEP DOIT rapporter au moins la première erreur et DEVRAIT rapporter autant d'erreurs que possible. Dans ce cas, le PEP DOIT revenir à sa configuration de la dernière bonne transaction avant la réception du message de décision erroné.

Le format de ces données est encapsulé dans l'objet COPS ClientSI désigné comme suit :

<ClientSI désigné : Rapport> ::= <[<GPERR>] *(<rapport>)>
<rapport> ::= <ErrorPRID> <CPERR> *(<PRID><EPD>)

5.3.2 Format de données de type de rapport comptable

De plus, les rapports peuvent être utilisés pour porter des informations de comptabilité lorsque ils spécifient le type de rapport 'Comptabilité'. Ce message de rapport de comptabilité va normalement porter des informations statistiques ou d'événements relatifs à la configuration installée pour l'usage du PDP. Ces informations sont codées comme un ou plusieurs liens <PRID><EPD> qui décrivent généralement les informations comptables rapportées du PEP au PDP.

Le format de ces données est encapsulé dans l'objet COPS ClientSI désigné comme suit :

<ClientSI désigné : Rapport> ::= <*(<PRID><EPD>)>

Note : La [RFC2748] définit un objet facultatif Temporisateur de comptabilité (AcctTimer) à utiliser dans le message COPS Client-Accept. Des rapports comptables périodiques pour les clients COPS-PR sont aussi obligatoirement réglés par ce temporisateur. Les rapports comptables périodiques NE DEVRAIENT PAS être générés par le PEP plus fréquemment que la période spécifiée par le AcctTimer COPS. Donc, la période entre les nouveaux rapports comptables DEVRAIENT être supérieurs ou égaux à la période spécifiée (si elle l'est) dans le AcctTimer. Si aucun objet AcctTimer n'est spécifié par le PDP, il n'y a alors pas de contrainte imposée à l'intervalle comptable du PEP.

6. Fonctionnement courant

La présente section décrit, de façon générale, les échanges normaux entre un PDP et un client d'approvisionnement de politique COPS.

D'abord, une connexion TCP est établie entre le client et le serveur et le PEP envoie un message Client-Ouvert spécifiant un type de client COPS-PR (l'utilisation de l'objet ClientSI dans le message Client-Ouvert est actuellement non défini pour les clients COPS-PR). Si le PDP prend en charge le type de client d'approvisionnement spécifié, le PDP répond par un message Client-Accept (CAT). Si le type de client n'est pas accepté, un message Client-Clos (CC) est retourné par le PDP au PEP, éventuellement identifiant un autre serveur connu pour prendre en charge la politique pour le type de client d'approvisionnement spécifié.

Après avoir reçu le message CAT, le PEP peut envoyer des demandes au serveur. La REQ d'un client d'approvisionnement de politique contient un objet de contexte COPS 'Demande de configuration' et, facultativement, toutes informations spécifiques du client désigné pertinents du PEP. Les informations fournies par le PEP devraient inclure les ressources disponibles du client (par exemple, les classes/attributs pris en charge) et les informations de configuration de politique par défaut ainsi que les données incarnées sur la politique existante. Le message de demande de configuration d'un client d'approvisionnement sert à deux objets. D'abord, c'est une demande au PDP de toutes les données de configuration d'approvisionnement que le PDP peut avoir actuellement qui conviendraient au PEP, comme les filtres de contrôle d'accès, etc., selon les informations que le PEP a spécifiées dans sa REQ. Aussi, la demande de configuration ouvre effectivement un canal qui va permettre au PDP d'envoyer de façon asynchrone les données de politique au PEP, quand le PDP décide que c'est nécessaire, tant que le PEP garde ouvert l'état de sa demande (c'est-à-dire, tant que le PEP n'envoie pas un DRQ avec la bride de client de l'état de demande). Ces données asynchrones peuvent être de nouvelles données de politique ou une mise à jour des données de politique envoyées précédemment. Tout changement pertinent de l'état interne du PEP peut être communiqué au PDP par l'envoi par le PEP d'un message REQ de mise à jour. Le PEP est libre d'envoyer de tels messages REQ mis à jour à tout moment après un message CAT pour communiquer les changements de son état local.

Après l'envoi par le PEP d'une REQ, si le PDP a des informations de configuration de politique d'approvisionnement pour le client, ces informations sont retournées au client dans un message DEC contenant les données de politique de client d'approvisionnement de politique dans l'objet COPS Données de décision désignées et en spécifiant un code de commande "Install" dans l'objet Fanions de décision. Si aucun filtre n'est défini, le message DEC va simplement spécifier qu'il n'y a pas de filtre en utilisant le code de commande "Décision NULLE" dans l'objet Fanions de décision. Comme le PEP DOIT spécifier une bride de client dans le message de demande, le PDP DOIT traiter la bride de client et la copier dans le message de décision correspondant. Un message DEC DOIT être produit par le PDP avec le fanion Message sollicité établi dans l'en-tête de message COPS, sans considérer si le PDP a ou non des informations de configuration pour le PEP au moment de la demande. Ceci est pour empêcher le PEP d'arriver en fin de temporisation pour la demande et supprimer la bride de client.

Le PDP peut alors ajouter de nouvelles données de politique ou mettre à jour/supprimer des configurations existantes en envoyant un ou des messages DEC non sollicités ultérieurs au PEP, avec la même bride de client. Les configurations installées précédemment sur le PEP sont mises à jour par le PDP en réinstallant simplement à nouveau les mêmes informations d'instance de configuration (écrasant effectivement les vieilles données). Le PEP est responsable de la suppression de la bride de client lorsque elle n'est plus nécessaire, par exemple lorsque une interface est fermée, et d'informer le PDP que la bride de client est à supprimer via le message COPS DRQ.

Pour les besoins de l'approvisionnement de politique, l'état d'accès, et les demandes d'accès au serveur de politique, peuvent être initiés par d'autres sources en dehors du PEP. Les exemples d'autres sources incluent les usagers rattachés qui demandent des services réseau via une interface de la Toile dans une application de gestion centrale, ou des serveurs H.323 qui demandent des ressources au nom d'un utilisateur d'une application de visioconférence. Lorsque une telle demande est acceptée, l'appareil bordure affecté par la décision (le point où le flux va entrer dans le réseau) a besoin d'être informé de la

décision. Comme le PEP dans l'appareil bordure n'a pas initié la demande, les spécificités de la demande, par exemple, la spécification du flux, le filtre de paquet, et le PHB à appliquer, ont besoin d'être communiquées au PEP par le PDP. Ces informations sont envoyées au PEP en utilisant le message Décision contenant les objets Données de décision d'approvisionnement de politique désignées dans l'objet Décision COPS comme spécifié. Toute mise à jour des informations d'état, par exemple dans le cas d'un changement de politique on de suppression d'un appel, est communiquée au PEP par des messages DEC non sollicités ultérieurs contenant la même bride de client et l'état de demande d'approvisionnement de politique mis à jour. Les mises à jour peuvent spécifier que les données de politique sont à installer, supprimer, ou mettre à jour (réinstaller).

Les PDP peuvent aussi commander au PEP d'ouvrir un nouvel état de demande ou d'en supprimer un existant en produisant une décision avec le fanion État de demande de l'objet Fanions de décision établi. Si le code de commande est "install", le PDP commande alors au PEP de créer un nouvel état de demande, et donc de produire un nouveau message REQ spécifiant une nouvelle bride de client ou autrement de produire un rapport "Échec" spécifiant la condition d'erreur appropriée. Chaque état de demande représente un espace de noms indépendant et logiquement sans chevauchement, identifié par la bride de client, sur lequel chaque transaction (autrement dit, installations, suppressions, mises à jour de configuration) peut être effectuée. Les autres états de demande existants ne seront pas affectés par le nouvel état de demande car ils sont indépendants (et donc, aucune instance de données de configuration ne peut être affectée au sein d'un état de demande par les décisions pour un autre état de demande telle qu'identifiée par la bride de client). Si le code de commande est "Supprimer", le PDP commande alors au PEP de supprimer l'état de demande existant spécifié par la bride de client du message DEC, causant par là la production par le PEP d'un message DRQ pour cette bride.

Le PEP DOIT accuser réception d'un message DEC et spécifier quelle action a été prise en envoyant un message RPT avec un objet Type de rapport de "Succès" ou "Échec" avec le fanion Message sollicité établi dans l'en-tête de message COPS. Cela sert d'indication au PDP que le demandeur (par exemple, un serveur H.323) peut être notifié de l'acceptation ou non de la demande par le réseau. Si le PEP a besoin de rejeter l'opération DEC pour une raison quelconque, un message RPT est envoyé avec un type de rapport qui a la valeur "Échec" et facultativement un objet Informations spécifiques du client qui spécifie que les données de politique ont été rejetées. Dans de telles conditions d'échec de rapport sollicité, le PEP DOIT toujours revenir à son (bon) état précédemment installé comme si la décision n'était pas intervenue. Le PDP est alors libre de modifier sa décision et d'essayer de nouveau.

Le PEP peut faire rapport au PDP de l'état actuel de tout état de demande installé lorsque approprié. Ces informations sont envoyées dans un message État de rapport (RPT) avec le fanion "Comptabilité" établi. L'état de demande qui est rapporté est identifié via la bride de client associée dans le message de rapport.

Finalement, les messages Client-Clos (CC) sont utilisés pour annuler le message Client-Ouvert correspondant. Le message CC informe l'autre côté que le type de client spécifié n'est plus pris en charge.

7. Tolérance aux fautes

Lorsque la communication est perdue entre le PEP et le PDP, le PEP tente de rétablir la connexion TCP avec le PDP avec lequel il était connecté en dernier. Si ce serveur ne peut pas être joint, le PEP tente alors de se connecter à un PDP secondaire, supposé être configuré manuellement (ou connu par ailleurs) au PEP.

Lorsque une connexion est finalement rétablie avec un PDP, le PEP envoie un message OPN *(ouvrir)* avec un objet <LastPDPAddr> fournissant l'adresse du plus récent PDP pour lequel il a toujours des décisions en antémémoire. Si aucune décision ne figure en antémémoire sur le PEP (à cause d'un réamorçage ou d'une fin de temps de vie de l'état) le PEP NE DOIT PAS inclure les informations de dernière adresse de PDP. Sur la base de cet objet, le PDP peut demander au PEP de resynchroniser ses informations d'état en cours (en produisant un message COPS SSQ). Si, après reconnexion, le PDP ne demande pas la synchronisation, le client peut supposer que le serveur le reconnaît et que l'état actuel au PEP est correct, de sorte qu'il n'est pas nécessaire d'envoyer un message REQ. Cependant, tout changement d'état survenu au PEP que celui-ci ne pourrait pas communiquer au PDP du fait de la perte de communications DOIT être rapporté au PDP via l'envoi par le PEP d'un message REQ mis à jour. Chaque fois que la resynchronisation est demandée, le PEP DOIT produire à nouveau tous messages REQ pour tous les états de demande connus et le PDP DOIT émettre des messages DEC pour supprimer les PRID individuels ou préfixes appropriés pour s'assurer de la cohérence de l'état connu au PEP.

Tandis que le PEP est déconnecté du PDP, l'état de demande actif au PEP est utilisé pour les décisions de politique. Si le PEP ne peut pas se reconnecter dans un délai pré-spécifié, tous les états de demande installés sont à supprimer et leurs brides associées sont retirées. La même chose est vraie pour le PDP ; détectant la défaillance d'une connexion TCP, le temporisateur est déclanché pour tous les états de demande associés au PEP et ces états sont supprimés après l'expiration de la période administrativement spécifiée sans qu'une connexion soit rétablie.

8. Considérations sur la sécurité

Le protocole COPS [RFC2748], duquel découle le présent document, décrit les mécanismes de sécurité obligatoires qui DOIVENT être pris en charge par toutes les mises en œuvre de COPS. Ces mécanismes de sécurité obligatoires sont utilisés par le protocole COPS pour transférer des informations opaques du PEP au PDP et vice versa d'une façon authentifiée et sûre. COPS pour l'approvisionnement de politique définit simplement une structure pour ces informations opaques déjà portées par le protocole COPS. À ce titre, les mécanismes de sécurité décrits pour le protocole COPS seront aussi déployés dans un environnement COPS-PR, assurant par là l'intégrité des informations COPS-PR communiquées. De plus, afin de décrire pleinement un ensemble pratique de données structurées à utiliser avec COPS-PR, une base de données d'informations de politique (PIB, *Policy Information Base*) sera probablement décrite dans un document distinct. Les auteurs d'une tel document de PIB doivent être conscients des problèmes de sécurité associés aux données spécifiques qu'ils définissent. Ces problèmes DOIVENT être pleinement spécifiés dans la section des considérations sur la sécurité du document de PIB ainsi que les mécanismes de sécurité requis pour transporter ces nouvelles données.

9. Considérations relatives à l'IANA

COPS pour l'approvisionnement de politique suit les mêmes considérations relatives à l'IANA que les objets COPS du protocole COPS de base [RFC2748]. COPS-PR a défini une valeur de fanion de décision supplémentaire de 0x02, étendant le protocole de base COPS de cette seule valeur. Aucun nouveau type de client COPS n'est défini par le présent document.

COPS-PR introduit aussi un nouvel espace de numéros d'objet où chaque objet est identifié par sa paire de valeurs S-Num et S-Type. Ces objets sont encapsulés au sein des objets COPS existants ClientSI désigné ou Données de décision désignées [RFC2748] et donc, ne sont pas en conflit avec les numéros alloués dans le protocole COPS de base. Des paires supplémentaires de S-Num et S-Type ne peuvent être ajoutées à COPS-PR qu'en utilisant la règle du consensus de l'IETF définie dans la [RFC2434]. Ces deux numéros sont toujours à traiter comme une paire, avec un ou plusieurs S-Types définis pour chaque S-Num. Le présent document définit les valeurs de S-Num 1 à 6 et le S-Type 1 pour chacune de ces six valeurs (noter que la valeur de S-Type de 2 est réservée pour le transport des données codées en XML). Une liste de toutes les paires de S-Num et S-Type définie dans le présent document se trouve aux paragraphes 4.1 à 4.6.

De même, des codes d'erreur d'approvisionnement global et d'erreur d'approvisionnement spécifique de classe définis pour COPS-PR ne peuvent être ajoutés que par consensus de l'IETF. Le présent document définit les valeurs de code d'erreur d'approvisionnement global de 1 à 11 au paragraphe 4.4 pour l'objet Erreur d'approvisionnement global (GPERR, *Global Provisioning Error*). Le présent document définit aussi les valeurs de code d'erreur spécifique de classe de 1 à 13 au paragraphe 4.5 pour l'objet Erreur de provisionnement de classe (CPERR, *Class Provisioning Error*).

10. Remerciements

Le présent document a été développé avec l'implication active d'un certain nombre de sources. Les auteurs tiennent à remercier spécifiquement les apports précieux de Michael Fine, Scott Hahn, et Carol Bell.

11. Références

[ASN1] International Organization for Standardization, International Standard 8824, "Information processing systems - Open Systems Interconnection, "Specification of Abstract Syntax Notation One (ASN.1)", décembre 1987.

[BER] International Organization for Standardization. International Standard 8825, "Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)", (décembre 1987).

[RFC2141] R. Moats, "Syntaxe des URN", mai 1997.

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir* RFC5234)

[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. *(Rendue obsolète par la* RFC5226)

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang et W. Weiss, "Architecture pour services différenciés", décembre 1998. *(MàJ par* RFC3260)

[RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure des informations de gestion, version 2 (SMIv2)", avril 1999. (STD0058)

[RFC2748] D. Durham et autres, "Protocole COPS (Service commun de politique ouverte)", janvier 2000. (*MàJ par* RFC4261) *(P.S.)*

[RFC2749] S. Herzog, et autres, "Utilisation de COPS avec RSVP", janvier 2000. *(P.S.)*

[RFC2753] R. Yavatkar, D. Pendarakis, R. Guerin, "Cadre pour le contrôle d'admission fondé sur la politique", janvier 2000. *(Info.)*

[RFC3159] K. McCloghrie et autres, "Structure des informations d'approvisionnement de politique (SPPI)", août 2001. *(P.S.)*

[XML] World Wide Web Consortium (W3C), "Extensible Markup Language (XML)," W3C Recommendation, février 1998, http://www.w3.org/TR/1998/REC-xml-19980210

12. Adresse des auteurs

Kwok Ho Chan
Nortel Networks, Inc.
600 Technology Park Drive
Billerica, MA 01821
téléphone : (978) 288-8175
mél : khchan@nortelnetworks.com

David Durham
Intel
2111 NE 25th Avenue
Hillsboro, OR 97124
téléphone : (503) 264-6232
mél : david.durham@intel.com

Silvano Gai
Cisco Systems, Inc.
170 Tasman Dr.
San Jose, CA 95134-1706
téléphone : (408) 527-2690
mél : sgai@cisco.com

Shai Herzog
IPHighway Inc.
69 Milk Street, Suite 304
Westborough, MA 01581
téléphone : (914) 654-4810
mél : Herzog@iphighway.com

Keith McCloghrie
téléphone : (408) 526-5260
mél l: kzm@cisco.com

John Seligson
Nortel Networks, Inc.
4401 Great America Parkway
Santa Clara, CA 95054
téléphone : (408) 495-2992
mél : jseligso@nortelnetworks.com

Francis Reichmeyer
PFN, Inc.
University Park at MIT
26 Landsdowne Street
Cambridge, MA 02139
téléphone : (617) 494 9980
mél l: franr@pfn.com

Raj Yavatkar
téléphone : (503) 264-9077
mél l: raj.yavatkar@intel.com

Andrew Smith
Allegro Networks
6399 San Ignacio Ave.
San Jose, CA 95119
USA
mél : andrew@allegronetworks.com

13. Déclaration complète de droits de reproduction

ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent 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.