

Groupe de travail Réseau  
**Request for Comments : 2749**  
 Catégorie : En cours de normalisation  
 janvier 2000

Traduction Claude Brière de L'Isle

S . Herzog, Ed., IPHighway  
 J. Boyle, Level3  
 R. Cohen, Cisco  
 D. Durham, Intel  
 R. Rajan, AT&T  
 A. Sastry, Cisco

## Utilisation de COPS pour RSVP

### Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de Copyright

Copyright (C) The Internet Society (2000). Tous droits réservés.

### Résumé

Le présent document décrit les directives pour la prise en charge des services de politique COPS dans les environnements RSVP.

### Table des matières

1. Introduction.....	1
2. Valeurs RSVP pour les objets COPS.....	2
2.1 En-tête commun, type de client.....	2
2.2 Objet Contexte (Context).....	2
2.3 Informations spécifiques du client (ClientSI).....	2
2.4 Objet Décision (Decision).....	3
3. Fonctionnement de COPS pour les PEP RSVP.....	4
3.1 Flux RSVP.....	4
3.2 Associations attendues pour les demandes RSVP.....	4
3.3 Contrôle d'admission de capacité de RSVP : Commit et Delete.....	4
3.4 Contrôle de politique sur PathTear et ResvTear.....	4
3.5 Mise en antémémoire des décision COPS par le PEP.....	4
3.6 Utilisation de plusieurs fanions Contexte dans une même interrogation.....	5
3.7 Rapport d'erreur RSVP.....	5
4. Considérations pour la sécurité.....	5
5. Illustration par des exemples d'utilisation de COPS pour RSVP.....	6
5.1 Exemple de flux en envoi individuel.....	6
5.2 Flux partagés de diffusion groupée.....	7
6 Références.....	9
7. Informations sur les auteurs et remerciements.....	9
8 Déclaration complète de droits de reproduction.....	9

## 1. Introduction

Le protocole du service commun de politique ouverte (COPS) est un protocole d'interrogation/réponse qui est utilisé pour échanger des informations de politique entre un serveur de politique de réseau et un ensemble de clients [RFC2748]. COPS est développé au sein du groupe de travail Politique d'admission RSVP (RAP) de l'IETF, principalement comme un mécanisme pour fournir un contrôle d'admission fondé sur la politique sur des demandes de ressources du réseau [RFC2753].

Le présent document se fonde sur le cadre RAP [RFC2753] et en suppose une connaissance préalable, et sur le protocole COPS de base [RFC2748]. Il fournit des directives d'utilisation spécifiques pour l'usage de COPS dans des décisions de contrôle de politique d'exportation par des clients RSVP (les PEP) aux serveurs de politique (les PDP).

Étant donné la conception du protocole COPS, les directives RSVP sont principalement limitées à des lignes directrices d'applicabilité à RSVP, d'interopérabilité et d'usage, ainsi qu'à des exemples spécifiques de client.

## 2. Valeurs RSVP pour les objets COPS

L'usage de plusieurs objets COPS est affecté lorsque ils sont utilisés avec les types de client RSVP. La présente section décrit ces objets et leur usage.

### 2.1 En-tête commun, type de client

RSVP est le type de client COPS 1.

### 2.2 Objet Contexte (Context)

La sémantique de l'objet Contexte pour RSVP est la suivante :

R-Type (Fanion Type de demande)

Demande de message entrant

Ce contexte est utilisé lorsque le PEP reçoit un message RSVP entrant. Le PDP peut décider d'accepter ou rejeter le message entrant et peut aussi y appliquer d'autres objets Décision. Si le message entrant est rejeté, RSVP devrait le traiter comme s'il n'était jamais arrivé.

Demande d'allocation de ressource

Ce contexte est utilisé lorsque le PEP est sur le point d'engager des ressources locales à un flux RSVP (contrôle d'admission). Ce contexte s'applique aux seuls messages Resv. La décision d'engager des ressources locales est prise pour la fusion de toutes les réservations associées à un flux RSVP (qui sont arrivées sur une interface particulière, potentiellement de plusieurs prochains bonds RSVP).

Demande de message sortant (transmission d'un message RSVP sortant)Ce contexte est utilisé lorsque le PEP est sur le point de transmettre un message RSVP sortant. Le PDP peut décider de permettre ou de refuser le message sortant, ainsi que de fournir un objet de données de politique sortant.

M-Type (Type de message)

Le champ M-Type dans l'objet Contexte identifie le type de message RSVP applicable. Les valeurs de M-Type sont identiques aux valeurs utilisées dans le champ "msg type" dans l'en-tête RSVP [RFC2205].

Les types de message RSVP suivants sont pris en charge dans COPS :

- Path (chemin)
- Resv (réservation)
- PathErr (erreur de chemin)
- ResvErr (erreur de réservation)

Les autres types de message tels que PathTear, ResvTear, et Resv Confirm ne sont pas pris en charge. La liste des types de message pris en charge ne peut être étendue que dans des versions ultérieures de RSVP et/ou des versions ultérieures du présent document.

### 2.3 Informations spécifiques du client (ClientSI)

Tous les objets qui ont été reçus dans un message RSVP sont encapsulés au sein de l'objet Informations spécifiques du client (ClientSI signalé) envoyé du PEP au PDP distant (voir au paragraphe 3.1 sur les flux multiples empaquetés dans un seul message RSVP).

Le PEP et le PDP partagent l'état RSVP, et le PDP est supposé mettre en œuvre la même spécification fonctionnelle RSVP que le PEP. Dans le cas où un PDP détecte l'absence des objets exigés par la [RFC2205] il devrait retourner une <Erreur> dans le message Décision indiquant "Manque des informations obligatoires spécifiques du client". Si, d'un autre côté, le PDP détecte l'absence d'objets RSVP facultatifs qui sont nécessaires pour approuver la demande au regard des politiques en cours, le PDP devrait retourner une <Décision> négative.

À la différence des contextes entrants et sortants, "l'allocation de ressource" n'est pas toujours directement associée au message RSVP spécifique. Dans une session de diffusion groupée, elle peut représenter la fusion de plusieurs réservations

entrantes. Donc, l'objet ClientSI devrait précisément contenir les objets SESSION et STYLE ainsi que la FLOWSPEC fusionnée, la liste FILTERSPEC, et l'objet SCOPE (chaque fois qu'il est pertinent).

## 2.4 Objet Décision (Decision)

COPS fournit au PDP des contrôles souples sur l'utilisation par le PEP de réponses aux messages de RSVP. Lorsque ils acceptent un message RSVP, les PDP peuvent fournir une priorité de préemption, déclencher des alertes, remplacer les objets RSVP, et plus encore, utiliser les commandes, fanions et objets de décision.

### Commandes de décision

Seules deux commandes s'appliquent à RSVP

#### Installe

Réponse positive :

Accepte/Permet/Admet un message RSVP ou une allocation de ressource locale.

#### Retire

Réponse négative :

Refuse/Rejette/Supprime un message RSVP ou une allocation de ressource locale.

### Fanions de décision

Un seul fanion de décision s'applique à RSVP :

#### Déclencher erreur

Si ce fanion est établi, RSVP devrait programmer un PathErr, en réponse à un message Path, ou une ResvErr (en réponse à un message Resv).

### Données de politique sans état

Cet objet peut comporter un ou plusieurs éléments de politique (comme spécifié pour l'objet RSVP Données de politique [RFC2750]) qui sont supposés être bien compris par le LPDP du client. Le PEP devrait considérer cela comme un ajout à la décision déjà reçue du PDP (il peut seulement ajouter, mais ne peut pas l'outrepasser).

Par exemple, soient des éléments de politique qui spécifient une priorité de préemption d'un flux, ces éléments peuvent être inclus dans un message Resv entrant ou peuvent être fournis par le PDP qui répond à une interrogation.

Les objets sans état doivent être bien compris, mais pas nécessairement pris en charge par tous les PEP. Par exemple, en supposant un élément de politique standard pour la priorité de préemption, il est parfaitement légitimé que certains PEP ne prennent pas en charge une telle préemption et qu'ils l'ignorent. Le PDP doit faire attention lorsque il utilise de tels objets. En particulier, il doit être prêt à ce que ces objets soient ignorés par le PEP.

Les données de politique sans état peuvent être retournées dans les décisions et elles s'appliquent individuellement à chacun des contextes marqués par des fanions dans les messages REQ. Lorsque elles sont appliquées aux entrants, il est supposé qu'elles ont été reçues comme objet POLICY\_DATA dans le message entrant. Lorsque elles s'appliquent à Allocation de ressource, elles sont supposées avoir été reçues sur tous les messages entrants fusionnés. Enfin, lorsque elles sont appliquées aux messages sortants, elles sont supposées avoir été reçues dans tous les messages contribuant au message sortant.

### Données de remplacement

L'objet Remplacement peut contenir plusieurs objets RSVP à remplacer (par rapport à la demande RSVP d'origine). Le remplacement normal est effectué sur la demande "Transmission sortante" (par exemple, remplacer les données de politique sortante) mais ne s'y limite pas, et peut aussi être effectué sur d'autres contextes (tels que "Demande d'allocation de ressource"). Dans d'autres cas, le remplacement de l'objet FlowSpec RSVP peut être utile pour contrôler des ressources à travers une zone de confiance (avec des nœuds ignorants de la politique (PIN, *policy ignorant node*). Actuellement, les clients RSVP sont seulement obligés de permettre le remplacement de trois objets : POLICY\_DATA, ERROR\_SPEC, et FLOWSPEC, mais peuvent facultativement prendre en charge le remplacement d'autres objets.

Le remplacement d'objet RSVP est effectué de la manière suivante :

Si aucune décision Données de remplacement n'apparaît dans un message de décision, tous les objets signalés sont traités comme si le PDP n'était pas là. Lorsque un objet d'un certain C-Num apparaît, il remplace TOUTES les instances d'objet C-Num dans le message RSVP. Si il apparaît vide (avec une longueur de 4) il retire simplement toutes les instances des objets C-Num sans rien ajouter.

### 3. Fonctionnement de COPS pour les PEP RSVP

#### 3.1 Flux RSVP

Le contrôle de politique est effectué par flux RSVP, qui est défini par l'unité atomique d'une réservation RSVP (réservation TC). Les styles de réservation peuvent aussi impacter la définition des flux ; un ensemble d'expéditeurs qui sont considérés comme un seul flux pour la réservation WF sont considérés comme un ensemble de flux individuels lorsque le style FF est utilisé.

Plusieurs flux FF peuvent être empaquetés dans un seul message Resv. Un message empaqueté doit être dépaqueté lorsque une demande séparée est produite pour chacun des flux empaquetés comme si ils étaient des messages RSVP individuels. Chaque demande COPS devrait comporter les objets associés de POLICY\_DATA, qui sont, par défaut, tous les objets POLICY\_DATA dans le message empaqueté. Des PEP sophistiqués, capables de regarder dans les objets de politique, peuvent examiner l'objet POLICY\_DATA ou SCOPE pour restreindre la liste des flux associés (comme optimisation).

Prière de noter que les règles qui gouvernent le message RSVP Empaqueté s'appliquent également au contexte REQ Entrant aussi bien que Sortant.

#### 3.2 Associations attendues pour les demandes RSVP

Lorsque il prend une décision de politique, le PDP peut considérer la Resv aussi bien que son état Path correspondant (état associé). L'association d'état est directe dans le cas courant d'envoi individuel car le flux RSVP comporte un état Path et un état Resv. Dans le cas de diffusion groupée, cette correspondance peut être plus compliquée, car la correspondance peut être de beaucoup à beaucoup. Le protocole COPS suppose que le PDP est à capacité RSVP et est capable de déterminer ces associations sur la base du contenu du message REQ Client et en particulier de l'objet ClientSI.

Par exemple, le PDP devrait être capable de reconnaître l'activation et la désactivation de l'état de blocage RSVP qui suit des événements discrets comme l'arrivée d'un message ResvErr (qui active l'état de blocage) aussi bien qu'un changement dans le message Resv sortant.

#### 3.3 Contrôle d'admission de capacité de RSVP : Commit et Delete

Dans RSVP, l'admission d'une nouvelle réservation exige à la fois une approbation administrative (contrôle de politique) et un contrôle de capacité d'admission. Après avoir été approuvé par les deux, et après que la réservation a été bien installée, le PEP la notifie au PDP distant en envoyant un message de rapport qui spécifie le type Commit (*engagé*). Le message de rapport du type Commit signale quand la facturation devrait effectivement commencer et si effectuer les opérations plus lourdes différées (par exemple, le débit d'une carte de crédit) est permis par le PDP.

Si, au lieu de cela, une réservation approuvée par le PDP échoue à l'admission à cause d'un manque de ressources, le PEP doit produire un rapport de non engagement et un repli et envoyer une demande de mise à jour à son état précédent (la réservation installée précédemment). Si aucun état n'était précédemment installé, le PEP devrait produire une commande Supprimer (DRQ).

#### 3.4 Contrôle de politique sur PathTear et ResvTear

Les messages PathTear et ResvTear ne sont pas contrôlés par cette architecture de politique. Cela s'appuie sur deux hypothèses : d'abord, que l'authentification MD-5 vérifie que la commande Tear est reçue du même nœud que celui qui a envoyé la réservation initiale, et ensuite, qu'il est fonctionnellement équivalent pour ce nœud de détenir des rafraîchissements pour cette réservation. Lorsque une ResvTear ou PathTear est reçue au PEP, tous les états affectés installés sur le PDP devraient être supprimés ou mis à jour par le PEP.

#### 3.5 Mise en antémémoire des décisions COPS par le PEP

Comme COPS est un protocole à états pleins, les rafraîchissements pour les messages RSVP Path et Resv n'ont pas besoin d'être constamment envoyés au PDP distant. Une fois qu'une décision a été retournée pour une demande, le PEP peut mettre en antémémoire cette décision et l'appliquer aux rafraîchissements futurs. Lorsque le PEP détecte un changement du message Resv ou Path correspondant, il devrait mettre à jour le PDP avec le nouvel état de demande. Les PEP peuvent continuer d'utiliser l'état mis en antémémoire jusqu'à ce qu'ils reçoivent la réponse du PDP. Ce cas est très différent de celui de l'admission initiale d'un flux ; étant donné que des accreditifs valides et l'authentification ont déjà été établis, la période relativement longue de rafraîchissement RSVP, et le court temps de réponse de PEP à PDP, le compromis entre mise à jour opportune et prévention d'attaque penche en faveur de l'opportunisme. Cependant, ceci est réellement le choix du PEP, et n'est pas pertinent pour les PDP.

Si la connexion est perdue entre le PEP et le PDP, l'état RSVP en antémémoire peut être conservé pendant la période de temporisation RSVP pour être utilisé pour les flux précédemment admis (mais il ne peut pas être appliqué à l'état nouveau ou mis à jour). Si la connexion ne peut pas être rétablie avec le PDP ou un PDP de sauvegarde après la fin de la période de temporisation, on attend du PEP qu'il purge toutes les décisions placées en antémémoire. Sans décision applicable en antémémoire, le PEP doit soit rejeter le flux, soit revenir à son LPDP (si il est disponible) pour les décisions.

Une fois qu'une connexion est rétablie avec un nouveau PEP (ou celui d'origine) le PDP peut produire une demande SSQ. Dans ce cas, le PEP doit produire à nouveau les demandes qui correspondent à l'état RSVP en cours (comme si tout l'état avait été uis à jour récemment). Il devrait aussi inclure dans son LPDPDecision la décision en cours (en antémémoire) concernant chacun de ces états correspondant.

### 3.6 Utilisation de plusieurs fanions Contexte dans une même interrogation

RSVP est un protocole de contrôle à remise différée (*store-and-forward*) où les messages sont traités en trois étapes distinctes (entrée, allocation de ressource, et sortie). Chaque étape exige une décision de politique distincte comme indiqué par les fanions de contexte (voir au paragraphe 2.2). Dans de nombreux cas, établir plusieurs fanions de contexte pour lier ensemble deux ou trois opérations en une seule demande peut significativement optimiser le fonctionnement du protocole.

Les règles suivantes s'appliquent pour établir plusieurs fanions Contexte :

- a. Plusieurs fanions de contexte ne peuvent être établis que dans deux cas génériques, qui représentent une portion substantielle des transactions attendues de COPS, et dont on peut garantir qu'elles ne causent pas d'ambiguïté.

FF en envoi individuel :

[Entrant + Allocation + Sortant]

Diffusion groupée avec seulement un message Resv reçu sur l'interface

[Entrant + Allocation]

- b. Les événement de contexte sont rangés par heure car chaque message doit d'abord être traité comme Entrant, puis comme Allocation de ressource et ensuite seulement comme Sortant. Lorsque plusieurs fanions de contexte sont établis, tous les objets ClientSI inclus dans la demande sont supposés être traités conformément au dernier fanion. Cette règle s'applique à la fois au contexte de la demande (REQ) et au contexte de la décision (DEC).

Par exemple, lorsque on combine Entrant + Allocation pour un message Resv entrant, la flowspec incluse dans le ClientSI serait celle qui correspond au contexte Ressource-Allocation (TC).

- c. Chaque décision est liée à un objet Contexte, qui détermine à quelle portion du contexte de la décision il s'applique. Lorsque des décisions individuelles s'appliquent à différents sous-groupes du contexte, le PDP devrait envoyer chaque groupe d'objets de décision encapsulés par l'objet Fanion de contexte avec les fanions de contexte applicables à ces objets établis (*à 1*) (voir les exemples à la Section 5).

### 3.7 Rapport d'erreur RSVP

RSVP utilise l'objet ERROR\_SPEC dans les messages PathErr et ResvErr pour faire rapport des erreurs de politique. Alors que le contenu de l'objet ERROR\_SPEC est défini dans les [RFC2205] et [RFC2750], le PDP est dans la meilleure position pour fournir son contenu (sous-codes). Ceci s'effectue de la manière suivante : d'abord, le PEP (RSVP) interroge le PDP avant d'envoyer un PathErr ou ResvErr, puis le PDP retourne la ERROR\_SPEC construite dans l'objet de décision Données de remplacement.

## 4. Considérations pour la sécurité

Le présent document s'appuie sur COPS pour sa signalisation et sa sécurité. Prière de se référer à la section "Considérations pour la sécurité" da la [RFC2748].

La sécurité pour les messages RSVP est fournie par l'authentification MD5 [RFC1321] inter-routeur, en supposant un modèle de chaîne de confiance. Un scénario de développement vraisemblable plaide en faveur du déploiement des PEP

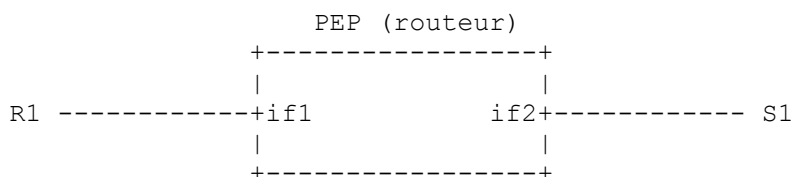
seulement sur la bordure du réseau (aux nœuds frontières) alors que le cœur du réseau consiste en nœuds PIN. Dans ce scénario la confiance MD5 (par l'authentification) est établie entre les PEP frontières (non voisins). Une telle confiance peut être réalisée par une signature interne (l'intégrité) de l'objet Données de politique lui-même, qui est laissé non modifié lorsque il passe à travers les nœuds PIN (voir la [RFC2750]).

## 5. Illustration par des exemples d'utilisation de COPS pour RSVP

Cette section détaille les deux scénarios typiques d'envoi individuel et de diffusion groupée.

### 5.1 Exemple de flux en envoi individuel

Ce paragraphe détaille les étapes en utilisant COPS pour contrôler un flux RSVP en envoi individuel. Il détaille le contenu des messages COPS par rapport à la Figure 1.



**Figure 1 : Exemple d'envoi individuel : vue d'un seul PEP**

Le routeur PEP a deux interfaces (if1, if2). L'envoyeur S1 envoie au receveur R1.

Un message Path arrive de S1 :

```

PEP --> PDP  REQ := <Manette A> <Contexte : entrant & sortant, Path>
                <Interface-entrante if2> <Interface-sortante if1>
                <ClientSI : tous les objets dans le message Path>
  
```

```

PDP --> PEP  DEC := <Manette A> <Contexte : entrant & sortant, Path>
                <Décision : Commande, Installe>
  
```

Un message Resv arrive de R1 :

```

PEP --> PDP  REQ := <Manette B>
                <Contexte : entrant & allocation & sortant, Resv>
                <Interface-entrante if1> <Interface-sortante if2>
                <ClientSI : tous les objets dans le message Resv>
  
```

```

PDP --> PEP  DEC := <Manette B>
                <Contexte : entrant, Resv>
                <Décision : commande, Installe>
                <Contexte : allocation, Resv>
                <Décision : commande, Installe>
                <Décision : Sans état, Priorité = 7>
                <Contexte : sortant, Resv>
                <Décision : commande, Installe>
                <Décision : remplacement, POLICY-DATA1>
  
```

```

PEP --> PDP  RPT := <Manettee B>
                <Type de rapport : commit>
  
```

Remarque que la Décision a été partagée à cause du besoin de spécifier des objets de décision différents pour les différents contextes.

Du temps passe, le PDP change sa décision :

```

PDP --> PEP  DEC := <Manette B>
                <Contexte : allocation, Resv>
  
```

```
<Décision : commande, Installe>
<Décision : Sans état, Priorité = 3>
```

Comme la priorité est trop faible, le PEP préempte le flux :

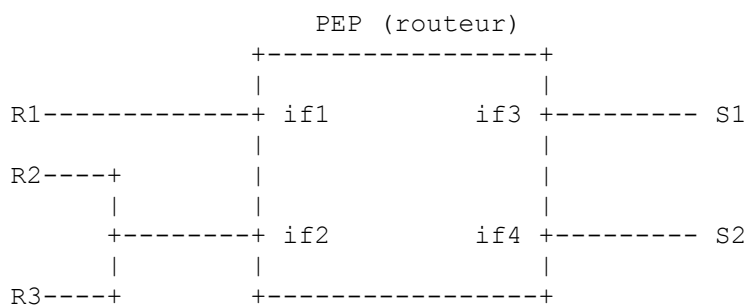
```
PEP --> PDP DRQ := <Manette B>
<Code de cause : Préempté>
```

Du temps passe, l'envoyeur S1 cesse d'envoyer des messages Path :

```
PEP --> PDP DRQ := <Manette A>
<Raison : Fin de temporisation>
```

## 5.2 Flux partagés de diffusion groupée

Ce paragraphe détaille les étapes en utilisant COPS pour contrôler un flux RSVP en envoi individuel. Il détaille le contenu des messages COPS par rapport à la Figure 2.



**Figure 2 : Exemple de diffusion groupée : vue d'un seul PEP**

La Figure 2 montre un PEP RSVP (routeur) qui a deux envoyeurs (S1, S2) et trois receveurs (R1, R2, R3) pour la même session de diffusion groupée. L'interface if2 est connectée à un support partagé. Dans cet exemple, on suppose que les membres de la diffusion groupée sont déjà en place. Aucun message RSVP précédent n'a été reçu, et le premier qui arrive est un message Path sur l'interface if3 provenant de l'envoyeur S1 :

```
PEP --> PDP REQ := <Manette A> <Contexte: entrant, Path>
<Interface-entrante if3>
<ClientSI : tous les objets dans le Path entrant>
```

```
PDP --> PEP DEC := <Manette A> <Contexte : entrant, Path>
<Décision : commande, Installe>
```

Le PEP consulte son tableau de transmission et trouve deux interface sortantes pour le chemin (if1, if2). L'échange ci-dessous est pour l'interface if1, un autre échange se ferait de même pour if2 en utilisant la nouvelle manette B2.

```
PEP --> PDP REQ := <Manette B1> <Contexte : sortant, Path>
<Interface-sortante if1>
<clientSI : tous les objets dans le Path sortant>
```

```
PDP --> PEP DEC := <Manette B1>
<Contexte : sortant, Path>
<Décision : commande, Installe>
<Décision : Remplacement, POLICY-DATA1>
```

Ici, le PDP a décidé de permettre la transmission du message Path et de fournir l'objet de données de politique approprié pour l'interface if1.

Ensuite, un message WF Resv provenant du receveur R2 arrive sur l'interface if2.

```
PEP --> PDP REQ := <Manette C> <Contexte : entrant & allocation, Resv>
<Interface-entrante if2>
```

<ClientSI : tous les objets dans le message Resv, RSpec1 incluse>

PDP --> PEP DEC := <Manette C>  
 <Contexte: entrant, Resv>  
 <Décision : commande, Installe>  
 <Contexte : allocation, Resv>  
 <Décision : commande, Installe>  
 <Décision : Sans état, priorité = 5>

PEP --> PDP RPT := <Manette C> <Commit>

Ici, le PDP approuve la réservation et alloue une priorité de préemption de 5. Le PEP a répondu par un rapport Commit.

Le PEP a besoin de transmettre en amont le message Resv vers S1 :

PEP --> PDP REQ := <Manette E> <Contexte: sortant, Resv>  
 <Interface-sortante if3>  
 <Client info : tous les objets dans la Resv sortante>

PDP --> PEP DEC := <Manette E>  
 <Contexte : sortant, Resv>  
 <Décision : commande, Installe>  
 <Décision : rmeplacement, POLICY-DATA2>

Note : L'objet Contexte fait partie de ce message DEC même si il peut paraître redondant parce que la REQ ne spécifiait q'un seul fanion Contexte.

Ensuite, un nouveau message Resv WF provenant du receveur R3 arrive sur l'interface if2 avec une plus forte RSpec (RSpec2). Comme deux réservations sont arrivées sur if2, il ne peut pas effectuer une demande avec plusieurs fanions Contexte, et doit les produire séparément.

Le PEP produit à nouveau ue manette mise à jour, C REQ, avec un nouvel objet Contexte <Contexte : entrant , Resv>, et reçoit une DEC pour la manette C.

PEP --> PDP REQ := <Manette F> <Contexte: entrant , Resv>  
 <Interface-entrante if2>  
 <ClientSI : tous les objets dans le message Resv, y compris RSpec2>

PDP --> PEP DEC := <Manette F> <Contexte : entrant , Resv>  
 <Décision : commande, Installe>

PEP --> PDP REQ := <Manette G> <Contexte : allocation, Resv>  
 <Interface-entrante if2>  
 <ClientSI : tous les objets dans la Resv fusionnée, y compris RSpec2>

PDP --> PEP DEC := <Manette G>  
 <Contexte : allocation, Resv>  
 <Décision : commande, Installe>  
 <Décision : Sans état, Priorité = 5>

PEP --> PDP RPT := <Manette G> <Commit>

Étant donné le changement des réservations entrantes, le PEP a besoin de transmettre un nouveau message Resv sortant en l'amont vers S1. Il répète exactement l'interaction précédente de la manette E, sauf que les objets ClientSI reflètent maintenant la fusion des deux réservations.

Si une ResvErr arrive de S1, le PEP la transpose en R3 seulement (parce que il a une flowspec supérieure : RSpec2) et il arrive ce qui suit :

PEP --> PDP REQ := <Manette H> <Contexte : entrant, ResvErr>  
 <Interface-entrante if3>  
 <ClientSI : tous les objets de la ResvErr entrante>



PDP --> PEP DEC := <Manette H> <Contexte : entrant, ResvErr>  
<Décision : commande, Installe>

PEP --> PDP REQ := <Manette I> <Contexte : sortant, ResvErr>  
<Interface-sortante if2>  
<ClientSI : tous les objets dans la ResvErr sortante>

PDP --> PEP DEC := <Manette I>  
<Contexte : sortant, ResvErr>  
<Décision : commande, Installe>  
<Décision : Remplacement, POLICY-DATA3>

Lorsque S2 se joint à la session en envoyant un message Path, les demandes Path entrantes et sortantes sont produites pour le nouveau Path. Une nouvelle demande Resv sortante sera envoyée à S2.

## 6 Références

- [RFC1321] R. Rivest, "Algorithme de [résumé de message MD5](#)", avril 1992. (*Information*)
- [RFC2205] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "[Protocole de réservation de ressource \(RSVP\)](#) -- version 1, spécification fonctionnelle", septembre 1997. (*MàJ par RFC2750, RFC3936, RFC4495*) (*P.S.*)
- [RFC2748] D. Durham et autres, "[Protocole COPS](#) (Service commun de politique ouverte)", janvier 2000. (*MàJ par RFC4261*) (*P.S.*)
- [RFC2750] S. Herzog, "[Extensions à RSVP](#) pour le contrôle de politique", janvier 2000. (*P.S.*)
- [RFC2753] R. Yavatkar, D. Pendarakis, R. Guerin, "[Cadre pour le contrôle d'admission](#) fondé sur la politique", janvier 2000. (*Info.*)

## 7. Informations sur les auteurs et remerciements

Des remerciement particuliers à Andrew Smith et Timothy O'Malley, les présidents de nos groupes de travail, et à Fred Baker, Laura Cunningham, Russell Fenger, Roch Guerin, Ping Pan, et Raj Yavatkar, pour leurs précieuses contributions.

Jim Boyle  
Level 3 Communications  
1025 Eldorado Boulevard  
Broomfield, CO 80021  
téléphone : 720.888.1192  
mél : jboyle@Level3.net

Ron Cohen  
CISCO Systems  
4 Maskit St.  
Herzeliya Pituach 46766 Israel  
téléphone : 972.9.9700064  
mél : [ronc@cisco.com](mailto:ronc@cisco.com)

David Durham  
Intel  
2111 NE 25th Avenue  
Hillsboro, OR 97124  
téléphone : 503.264.6232  
mél : David.Durham@intel.com

Raju Rajan  
AT&T Labs Research  
180 Park Ave., P.O. Box 971  
Florham Park, NJ 07932  
téléphone : 973.360.7229  
mél : rajur@research.att.com

Shai Herzog  
IPHighway, Inc.  
55 New York Avenue  
Framingham, MA 01701  
téléphone : 508.620.1141  
mél : herzog@iphighway.com

Arun Sastry  
Cisco Systems  
4 The Square  
Stockley Park  
Uxbridge, Middlesex UB11 1BN  
UK  
téléphone : +44-208-756-8693  
mél : asastry@cisco.com

## 8 Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (1998). Tous droits réservés.

Ce document et ses traductions peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de droits de reproduction ci-dessus et ce paragraphe soient inclus sur

toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant le droit d'auteur ou les références à l'Internet Society ou d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définies dans les processus des normes de l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ou ses successeurs ou ayants droit.

Ce document et les renseignements qu'il contient sont fournis "TELS QUELS" et l'INTERNET SOCIETY et l'INTERNET ENGINEERING TASK FORCE déclinent toute garantie, expresse ou implicite, y compris mais sans s'y limiter, toute garantie que l'utilisation de l'information ici présente n'enfreindra aucun droit ou aucune garantie implicite de commercialisation ou d'adaptation à un objet particulier.

**Remerciement**

Le financement de la fonction d'éditeur des RFC est actuellement fourni par la Internet Society.