Groupe de travail Réseau
**Request for Comments : 2746**
Catégorie : En cours de normalisation

Traduction Claude Brière de L'Isle

A. Terzis, UCLA
J. Krawczyk, ArrowPoint Communications
J. Wroclawski, MIT LCS
L. Zhang, UCLA
janvier 2000

# Fonctionnement de RSVP sur tunnels IP



**RésuméLe** présent document décrit une approche pour la fourniture de services de protocole RSVP sur les tunnels IP. On décrit brièvement le problème, les caractéristiques de solutions possibles, et les objectifs de conception de notre approche. On présente ensuite les détails d'une mise en œuvre satisfaisant aux objectifs de conception.

## 1. Introduction

Les "tunnels IP dans IP sont devenus un mécanisme largement répandu pour transporter les datagrammes dans l'Internet. Normalement, un tunnel est utilisé pour acheminer tes paquets à travers des portions de réseau qui ne mettent pas directement en œuvre le service désiré (par exemple, IPv6) ou pour augmenter et modifier le comportement de l'architecture d'acheminement déployée (par exemple, l'acheminement en diffusion groupée, IP mobile, réseau privé virtuel).

De nombreux protocoles de tunnelage IP dans IP existent aujourd'hui. La [RFC2003] détaille une méthode de tunnelage qui utilise un en-tête IPv4 supplémentaire. La [RFC2004] décrit un moyen de réduire la taille de l'en-tête IP "interne" utilisé dans la [RFC2003] lorsque le datagramme original n'est pas fragmenté. La méthode générique de tunnelage dans la [RFC2473] peut être utilisée pour tunneler des paquets IPv4 ou IPv6 au sein de IPv6. La [RFC1933] décrit comment tunneler les datagrammes IPv6 à travers les réseaux IPv4. La [RFC1701] décrit une encapsulation d'acheminement générique, alors que la [RFC1702] applique cette encapsulation à IPv4. Finalement, la [RFC1827] décrit un mécanisme qui peut être utilisé pour tunneler un datagramme IP chiffré.

Dans la perspective traditionnelle de la livraison au mieux des paquets IP, un tunnel se comporte comme toute autre liaison. Les paquets entrent à une extrémité du tunnel, et sont livrés à l'autre extrémité sauf si la surcharge de la ressource ou une erreur a causé leur perte.

Le protocole d'établissement de RSVP [RFC2205] est un des composants d'un cadre conçu pour étendre IP à la prise en charge de plusieurs classes de service contrôlées sur une grande variété de technologies de niveau liaison. Pour déployer cette technologie avec le maximum de souplesse, il est souhaitable que les tunnels agissent comme des liaisons contrôlables au sein du réseau.

Un tunnel, et en fait toute sorte de liaison, peut participer à un réseau à capacité RSVP d'une de trois façons, selon les capacités de l'équipement à partir duquel est construit le tunnel et les désirs de l'opérateur.

1. La liaison (logique) peut ne pas prendre en charge du tout la réservation de ressource ou le contrôle de qualité de service. C'est une liaison au mieux. Dans ce document, on appelle cela un tunnel au mieux ou de type 1.
2. La liaison (logique) peut être capable de promettre qu'un certain niveau global de ressources soit disponible pour porter du trafic, mais pas pour allouer des ressources à des flux de données individuels spécifiques. Une allocation de ressource configurée sur un tunnel en est un exemple. Dans ce document, on appelle ce cas un tunnel de type 2.
3. La liaison (logique) peut être capable de faire les réservations pour des flux individuels de données de bout en bout. On appelle ce cas un tunnel de type 3. Noter que la caractéristique clé qui distingue les tunnels de type 3 des tunnels de type 2 est que dans le tunnel de type 3 les nouvelles réservations sont créées et supprimées de façon dynamique au fil

des réservations de bout en bout.

Les tunnels de type 1 existent lorsque au moins un des routeurs comportant les points d'extrémité du tunnel ne prend pas en charge le schéma que nous décrivons ici. Dans ce cas, le tunnel agit comme une liaison au mieux. Notre but est simplement de s'assurer que les messages RSVP traversent correctement la liaison, et la présence de la liaison non contrôlée est détectée, comme exigé par le cadre des services intégrés.

Lorsque les deux points d'extrémité du tunnel sont capables de prendre en charge RSVP sur les tunnels, on aimerait que les ressources appropriées soient réservées le long du tunnel. Selon les exigences de la situation, cela peut signifier que les flux de données d'un client sont placés dans une réservation agrégée plus grande (tunnels de type 2) ou qu'éventuellement une nouvelle réservation distincte est faite pour le flux de données (tunnels de type 3). Noter qu'une réservation RSVP entre les deux points d'extrémité de tunnel ne signifie pas nécessairement que tous les routeurs intermédiaires le long du chemin du tunnel prennent en charge RSVP ; cela est équivalent au cas d'une session RSVP de bout en bout existante qui passe de façon transparente à travers un nuage non RSVP.

Actuellement cependant, la signalisation RSVP n'est pas possible sur les tunnels. Les paquets RSVP qui entrent dans le tunnel sont encapsulés avec un en-tête IP externe qui a un numéro de protocole autre que 46 (par exemple, c'est 4 pour l'encapsulation IP dans IP) et ils ne portent pas l'option Alerte de routeur, ce qui les rend virtuellement "invisibles" aux routeurs RSVP entre les deux points d'extrémité du tunnel. De plus, le schéma actuel d'encapsulation IP dans IP n'ajoute qu'un en-tête IP comme enveloppe externe. Il est impossible de distinguer entre les paquets qui utilisent des réservations et ceux qui n'en utilisent pas, ou de différencier les paquets qui appartiennent à des sessions RSVP différentes lorsque ils sont dans le tunnel, parce qu'aucune information distinctive comme un accès UDP n'est disponible dans l'encapsulation.

Le présent document décrit un mécanisme d'amélioration du tunnelage IP qui permet à RSVP de faire des réservations à travers les tunnels IP dans IP. Ce mécanisme est capable de prendre en charge les tunnels de type 2 aussi bien que de type 3, comme décrits ci-dessus, et n'exige qu'un minimum de changements à RSVP et aux autres parties du cadre des services intégrés.

## 2. Le concept

### 2.1 Objectifs du concept

Les choix de ce concept sont motivés par plusieurs objectifs.

* Coexister avec la plupart, sinon tous, des schémas actuels de tunnelage IP dans IP.
* Limiter les changements à la spécification RSVP au minimum possible.
* Limiter les changements nécessaires aux seuls deux points d'extrémité d'un tunnel. Cette exigence conduit à un développement plus simple, à une redondance plus faible dans les routeurs intermédiaires, et à une plus faible probabilité de défaillance lorsque l'ensemble des routeurs intermédiaires est modifié à cause de changements d'acheminement.
* Prendre en charge une interopération correcte avec les routeurs RSVP qui n'ont pas été mis à niveau pour traiter RSVP sur les tunnels et avec les routeurs de point d'extrémité de tunnel non RSVP. Dans ces cas, le tunnel se comporte comme une liaison non RSVP.

### 2.2 Approche de base

L'idée de base de la méthode décrite dans ce document est d'appliquer de façon récurrente RSVP sur la portion de tunnel du chemin. Dans cette nouvelle session, le point d'entrée de tunnel Rentry envoie des messages PATH et le point de sortie du tunnel Rexit envoie des messages RESV pour réserver des ressources pour les sessions de bout en bout sur le tunnel.

On expose ensuite deux aspects différents du concept : comment améliorer un tunnel IP dans IP avec la capacité RSVP, et comment transposer les sessions RSVP de bout en bout dans une session de tunnel.

#### 2.2.1 Décisions de conception

Pour établir une réservation RSVP sur un tunnel IP dans IP en envoi individuel, on prend les décisions conceptuelles suivantes :

Une ou plusieurs réservations d'envoi individuel de style Filtre fixe seront utilisées entre les deux points d'extrémité du tunnel pour réserver des ressources pour les paquets qui traversent le tunnel. Dans le cas de type 2, ces réservations seront configurées de façon statique pour une interface de gestion. Dans le cas de type 3, ces réservations seront créées et supprimées à la demande, au fil des demandes de réservation de bout en bout.

Les paquets qui n'exigent pas de réservation sont encapsulés de la façon normale, par exemple. en étant enveloppés avec seulement un en-tête IP, spécifiant le point d'entrée du tunnel comme source et le point de sortie comme destination.

Les paquets de données qui exigent une réservation de ressource au sein d'un tunnel doivent avoir des attributs autres que les adresses IP visibles aux routeurs intermédiaires, afin que les routeurs puissent transposer le paquet sur une réservation appropriée. Pour permettre aux routeurs intermédiaires d'utiliser le traitement RSVP filterspec standard, nous choisissons d'encapsuler de tels paquets de données en mettant devant un en-tête IP et un en-tête UDP, et d'utiliser les numéros d'accès UDP pour distinguer les paquets des différentes sessions RSVP. Le numéro de protocole dans l'en-tête IP externe dans ce cas sera celui de UDP.

La Figure 1 montre le fonctionnement de RSVP sur un tunnel. Rentry est le routeur d'entrée du tunnel qui encapsule les données dans le tunnel. Un certain nombre de routeurs intermédiaires transmettent les données à travers le réseau sur la base de l'en-tête IP encapsulant ajouté par Rentry. Rexit est le point d'extrémité du tunnel. Il désencapsule les données et les transmet sur la base de l'en-tête IP "interne", d'origine.

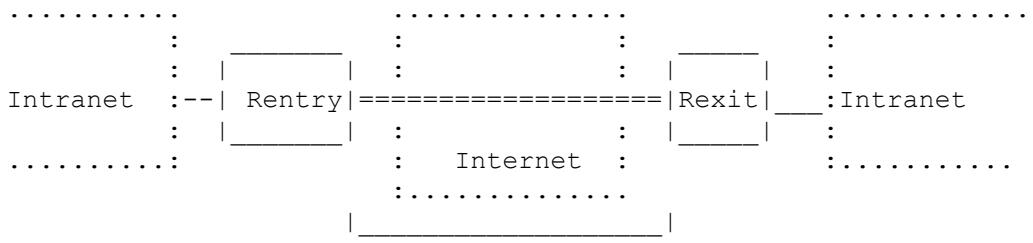



**Figure 1 : Exemple de tunnel IP**

### 2.2.2 Transposition entre sessions de bout en bout et tunnel

La Figure 2 montre une topologie simple avec un tunnel et quelques hôtes. Les hôtes envoyeurs H1 et H3 peuvent être distants de un ou plusieurs bonds IP de Rentry ; les hôtes receveurs H2 et H4 peuvent aussi être à un ou plusieurs bonds IP de Rexit.

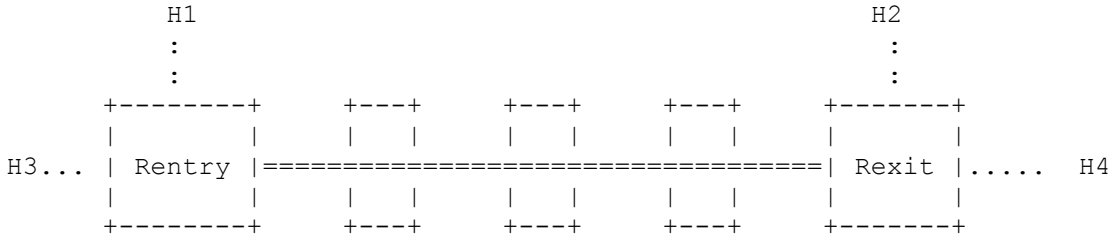



**Figure 2 : Exemple de chemin de bout en bout avec un tunnel au milieu.**

Une session RSVP peut être en place entre les points d'extrémité des hôtes H1 et H2. On se réfère à cette session comme session de "bout en bout" (abrégé en E2E, *pour end to end*) ou "originale", et à ses messages PATH et RESV comme aux messages de bout en bout. Une ou plusieurs sessions RSVP peuvent être en place entre Rentry et Rexit pour fournir des réservations de ressources sur le tunnel. On se réfère à elles comme aux sessions de tunnel RSVP, et à leurs messages PATH et RESV comme aux messages tunnel ou de tunnelage. Une session tunnel RSVP peut exister indépendamment de toute session de bout en bout. Par exemple on peut créer à travers une interface de gestion de réseau une session RSVP sur le tunnel pour assurer la prise en charge de la qualité de service pour les flux de données de H3 à H4, bien qu'il n'y ait pas de session RSVP de bout en bout entre H3 et H4.

Lorsque une session RSVP de bout en bout traverse un tunnel à capacité RSVP, il y a deux cas à considérer pour la conception des mécanismes de prise en charge d'une réservation de bout en bout sur le tunnel : la transposition de la session

de bout en bout dans une session tunnel RSVP existante (tunnel de type 2) et la création de façon dynamique d'une nouvelle session tunnel RSVP pour chaque session de bout en bout (tunnel de type 3). Dans l'un et l'autre cas, on a une application récurrente de RSVP. La session tunnel RSVP voit les deux sessions de bout en bout comme deux hôtes d'extrémité avec entre les deux une réservation de style filtre fixe en envoi individuel. La session RSVP de bout en bout d'origine voit le tunnel comme une seule liaison (logique) sur le chemin entre la ou les sources et la ou les destinations.

Noter qu'en pratique un tunnel peut combiner les caractéristiques de type 2 et de type 3. Certaines sessions RSVP de bout en bout peuvent déclencher la création de nouvelles sessions de tunnel, alors que d'autres peuvent être transposées dans une session tunnel RSVP existante. Le choix de la façon dont une session de bout en bout est traitée dans le tunnel est une affaire de politique locale.

Lorsque une session RSVP de bout en bout traverse un tunnel à capacité RSVP, il est nécessaire de coordonner les actions des deux sessions RSVP, pour déterminer si ou quand la session tunnel RSVP devrait être créée et supprimée, et pour transférer correctement les erreurs et informations ADSPEC entre les deux sessions RSVP. On a fait le choix suivant :

* Les messages de commande RSVP de bout en bout qui sont transmis à travers un tunnel sont encapsulés de la même façon que les paquets IP normaux, par exemple, en étant enveloppés avec seulement l'en-tête IP de tunnel, en spécifiant le point d'entrée du tunnel comme source et le point de sortie comme destination.

### 2.3 Questions majeures

Comme les tunnels IP dans IP sont utilisés de plus en plus largement pour les besoins de la gestion du trafic réseau, il est clair qu'on doit prendre en charge les tunnels de type 2 (réservation de tunnel pour les sessions agrégées de bout en bout). De plus, ces tunnels de type 2 devraient permettre d'utiliser plus d'une réservation (configurable, statique) à la fois, pour prendre en charge différentes classes de trafic au sein du tunnel. Savoir si il est nécessaire de prendre en charge les tunnels de type 3 (réservations de tunnel dynamique par session de bout en bout) est une question de politique qui devrait rester ouverte. Notre concept prend en charge les deux cas.

Si il y a seulement une session RSVP configurée sur un tunnel, toutes les sessions RSVP de bout en bout (qui ont la permission d'utiliser cette session de tunnel) seront alors liées à cette session de tunnel configurée. Cependant, lorsque plus d'une session RSVP est utilisée sur un tunnel, un second problème de conception est celui de la façon dont est créée et convoyée d'une extrémité du tunnel à l'autre l'association, ou le lien entre une réservation RSVP originale et une réservation de tunnel. Le routeur d'entrée Rentry et le routeur de sortie Rexit doivent se mettre d'accord sur ces associations de sorte que les changements dans l'état original de réservation puissent être correctement transposés en changements dans l'état de réservation du tunnel, et que les erreurs rapportées par les routeurs intermédiaires aux points d'extrémité du tunnel puissent être correctement transformées en erreurs rapportées par la session de bout en bout à la session RSVP de bout en bout.

On exige que le même mécanisme d'association fonctionne pour les deux cas de bouquet de réservations sur un tunnel (tunnel de type 2) et dans le cas de transposition biunivoque entre réservation d'origine et tunnel (tunnel de type 3). Dans notre schéma, l'association est créée lorsque un point d'entrée de tunnel voit pour la première fois un message RESV d'une session de bout en bout et établit une nouvelle session de tunnel, ou bien l'ajoute à une session de tunnel existante. Cette nouvelle association doit être convoyée à Rexit, afin que Rexit puisse réserver les ressources pour les sessions de bout en bout à l'intérieur du tunnel. Ces informations incluent l'identifiant et certains paramètres de la session de tunnel, et l'identifiant de la session de bout en bout à laquelle la session du tunnel se lie. Dans notre schéma, toutes les sessions RSVP entre les deux mêmes routeurs Rentry et Rexit vont avoir des valeurs identiques d'adresse IP de source, d'adresse IP de destination, et de numéro d'accès UDP de destination. Une session individuelle est identifiée principalement par la valeur de l'accès de source.

On a identifié trois choix possibles pour le mécanisme de lien :

1. Définir un nouveau message RSVP qui n'est échangé qu'entre les deux points d'extrémité du tunnel pour convoyer les informations de lien.
2. Définir un nouvel objet RSVP à rattacher aux messages PATH de bout en bout à Rentry, associant la session de bout en bout à une des sessions du tunnel. Ce nouvel objet est interprété par Rexit en associant la session de bout en bout à une des sessions de tunnel générée chez Rentry.
3. Appliquer la même encapsulation UDP aux messages PATH de bout en bout qu'aux paquets de données de la session. Lorsque Rexit désencapsule le message PATH, il déduit la relation entre l'accès UDP de source utilisé dans l'encapsulation et la session RSVP qui est spécifiée dans le message PATH original.

La dernière de ces approches n'exige aucun nouveau concept. Cependant, elle exige que des ressources supplémentaires soient réservées pour les messages PATH (car ils sont maintenant sujets à la réservation de tunnel). Elle exige aussi une connaissance à priori de la prise en charge par Rexit de RSVP sur les tunnels par encapsulation UDP. Si Rentry encapsule tous les messages PATH de bout en bout avec l'encapsulation UDP, mais que Rexit ne comprend pas cette encapsulation, les messages PATH encapsulés seront perdus chez Rexit.

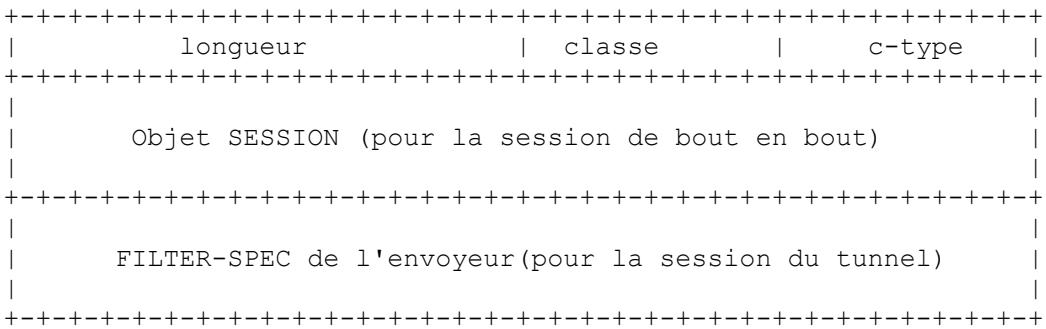
D'un autre côté, les options (1) et (2) peuvent traiter ce cas de façon transparente. Cela permet à Rexit de passer les messages PATH de bout en bout reçus via le tunnel (parce qu'ils sont désencapsulés normalement) tout en éliminant les PATH de tunnel, sans aucune configuration supplémentaire. On choisit l'option (2) parce qu'elle est plus simple. L'objet est décrit au paragraphe suivant.

Les échanges de paquet doivent respecter les contraintes suivantes :

1. Rentry encapsule et envoie les messages PATH de bout en bout sur le tunnel à Rexit où ils sont désencapsulés et transmis vers l'aval.
2. Lorsque un messages RESV de bout en bout correspondant arrive à Rexit, Rexit l'encapsule et l'envoie à Rentry.
3. Sur la base de certaines ou de toutes les informations des messages PATH de bout en bout, de la flowspec dans le message RESV de bout en bout et des politiques locales, Rentry décide si et comment transposer la session de bout en bout dans une session du tunnel.
4. Si la session de bout en bout doit être transposée dans une session de tunnel, Rentry envoie un message PATH pour une nouvelle session de tunnel ou met à jour une session existante.
5. Rentry envoie un PATH de bout en bout contenant un objet SESSION_ASSOC qui associe la session de bout en bout à la session du tunnel ci-dessus. Rexit enregistre l'association et retire l'objet avant de transmettre le message PATH plus loin.
6. Rexit répond au message PATH du tunnel en envoyant un message RESV de tunnel, qui réserve les ressources à l'intérieur du tunnel.
7. Rentry n'encapsule en UDP les paquets qui arrivent que si une réservation de session du tunnel correspondante est en fait en place pour les paquets.

### 2.3.1 Objet SESSION_ASSOC

Le nouvel objet, appelé SESSION_ASSOC, est défini avec le format suivant :




**Objet SESSION_ASSOC**

Longueur : Ce champ contient la taille de l'objet SESSION_ASSOC en octets.

Classe : Devrait être 192.

C-type : Devrait être envoyé à zéro et ignoré à réception.

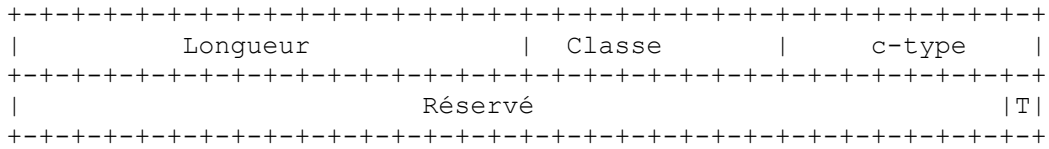
Objet SESSION : La SESSION de bout en bout contenue dans l'objet est à transposer dans la session de tunnel décrite par la FILTER-SPEC de l'envoyeur défini ci-dessous.

FILTER-SPEC de l'envoyeur : C'est la session du tunnel sur laquelle la session de bout en bout mentionnée ci-dessus se transpose sur le tunnel. Comme mentionné ci-dessus, une session de tunnel est identifiée principalement par un accès de source. C'est pourquoi on utilise une Filter-Spec d'envoyeur pour la session de tunnel, à la place d'un objet SESSION.

### 2.3.2 Objet NODE_CHAR

Il faut qu'il y ait une façon (autre que par la configuration) pour que Rexit communique à Rentry le fait qu'il y a un point d'extrémité de tunnel qui prend en charge le schéma décrit dans ce document. Nous avons défini pour cette raison un nouvel objet, appelé NODE_CHAR, qui porte ces informations. Si un nœud reçoit cet objet mais ne le comprend pas, il devrait l'éliminer sans produire de rapport d'erreur. Les objets avec Class-Num = 10bbbbbb (`b' représente un bit), comme défini dans la spécification RSVP [RFC2205], ont les caractéristiques dont nous avons besoin. Bien que pour l'instant cet objet ne porte qu'un seul bit d'information, il peut être utilisé à l'avenir pour décrire les autres caractéristiques d'un nœud à capacité RSVP qui ne fait pas partie de la spécification RSVP d'origine.

L'objet NODE_CHAR a le format suivant :




Longueur : Ce champ contient la taille de l'objet NODE_CHAR en octets. Il devrait être réglé à huit.

Classe : Une valeur appropriée devrait être allouée par l'IANA. On propose que cette valeur soit de 128.

C-type : Devrait être envoyé à zéro et ignoré en réception.

T : Ce bit *(mis à 1)* montre que le nœud a la capacité de tunnel RSVP.

Lorsque Rexit reçoit une réservation de bout en bout, il ajoute un objet NODE_CHAR avec le bit T établi, à l'objet RESV, il l'encapsule et l'envoie à Rentry. Lorsque Rentry reçoit ce message RESV, il en déduit que Rexit met en œuvre le mécanisme décrit ici et donc il crée ou ajuste une session du tunnel et associe la session du tunnel à la session de bout en bout via un objet SESSION_ASSOC. Rentry devrait supprimer l'objet NODE_CHAR avant de transmettre le message RESV vers l'amont. Si de l'autre côté, Rentry ne prend pas en charge le mécanisme de tunnels RSVP, il va simplement ignorer l'objet NODE_CHAR et ne le transmettra pas vers l'amont.

## 3. Mise en œuvre

Dans cette section sont exposés séparément plusieurs cas, en commençant par le scénario le plus simple et en passant ensuite aux plus complexes.

### 3.1 Une seule session RSVP configurée sur un tunnel IP dans IP

En traitant les deux sessions de bout en bout comme un hôte de source et de destination, on peut facilement établir une réservation de style FF entre elles. La question est maintenant de savoir quel type de filterspec utiliser pour la réservation de tunnel, qui se rapporte directement à la façon dont les paquets sont encapsulés sur le tunnel. Deux cas sont exposés ci-dessous.

### 3.1.1 En l'absence de session RSVP de bout en bout

Dans le cas où tous les paquets qui traversent un tunnel utilisent les ressources réservées, on pourrait utiliser l'encapsulation IP dans IP actuelle. La session RSVP sur le tunnel spécifierait simplement une réservation de style FF (avec le numéro d'accès zéro) avec Rentry comme adresse de source et Rexit comme adresse de destination.

Cependant si seulement quelques paquets traversant le tunnel devaient bénéficier de la réservation, on doit encapsuler les paquets qualifiés dans IP et UDP. Cela permet aux routeurs intermédiaires d'utiliser le traitement standard de filterspec RSVP, sans avoir à connaître de l'existence des tunnels.

Plutôt que de prendre en charge les deux cas, nous choisissons de simplifier les mises en œuvre en exigeant que tous les paquets de données utilisant les réservations soient encapsulés avec un en-tête IP et UDP externe. Cela réduit les vérifications et traitements de cas particuliers.

### 3.1.2 En présence de session RSVP de bout en bout

Conformément aux politiques de contrôle de tunnel, installées à travers une interface de gestion, certaines (ou toutes) sessions RSVP de bout en bout peuvent être autorisées à se transposer en une seule session RSVP sur le tunnel. Dans ce cas, il n'est pas besoin de fournir des informations de liaison dynamique entre les sessions de bout en bout et la session du tunnel, étant donné que la session du tunnel est unique et pré configurée, et donc, bien connue.

Lier plusieurs sessions de bout en bout à une session du tunnel, soulève cependant une nouvelle question sur quand et comment la taille de la réservation du tunnel devrait être ajustée pour s'accommoder des sessions de bout en bout qui sont transposées sur elle. Là encore, le gestionnaire du tunnel devrait prendre une telle décision de politique. Plusieurs scénarios sont possibles. Dans le premier, la réservation de tunnel n'est jamais ajustée. Cela rend en gros le tunnel équivalent à une liaison matérielle à capacité fixe. Dans le second, la réservation de tunnel est ajustée chaque fois qu'une nouvelle réservation de bout en bout arrive ou qu'une ancienne est supprimée. Dans la troisième, la réservation de tunnel est ajustée à l'occasion vers l'amont ou vers l'aval, chaque fois que le niveau de réservation de bout en bout a suffisamment changé pour garantir l'ajustement. Cela fait un compromis entre l'utilisation de ressources supplémentaires dans le tunnel et une réduction du contrôle du trafic et de la redondance.

On appelle un tunnel dont la réservation ne peut pas être ajustée un "tuyau", par opposition à un "flexible" où la quantité de ressources allouées est réglable. Le paragraphe 5.2 explique comment l'ajustement peut se faire sur des flexibles.

## 3.2 Plusieurs sessions RSVP configurées sur un tunnel IP dans IP

Dans le cas d'une seule session RSVP configurée sur un tunnel on a une construction directe en établissant plusieurs réservations de style FF entre deux sessions de bout en bout utilisant une interface de gestion. Dans ce case Rentry doit encapsuler soigneusement les paquets de données avec les numéros d'accès UDP appropriés, de sorte que les paquets appartenant à des sessions de tunnel différentes soient distingués par les routeurs RSVP intermédiaires. Noter que ce cas et celui décrit auparavant sont ceux de ce qu'on appelle des tunnels de type 2.

### 3.2.1 En l'absence de session RSVP de bout en bout

Rien de plus ne mérite d'être dit sur ce cas. Rentry classe les paquets et les encapsule conformément à ces classes. Les paquets sans réservation sont encapsulés avec seulement un en-tête IP externe, alors que les paquets qualifiés pour les réservations sont encapsulés avec un en-tête UDP ainsi qu'un en-tête IP. La valeur de l'accès de source UDP devrait être correctement réglée pour se transposer dans la réservation de tunnel correspondante que le paquet est supposé utiliser.

### 3.2.2 En présence de session RSVP de bout en bout

Comme dans ce cas, il y a plus d'une session RSVP qui fonctionne sur le tunnel, on doit explicitement lier chaque session RSVP de bout en bout à sa session de tunnel correspondante. Comme exposé précédemment, ce lien sera fourni par le nouvel objet SESSION_ASSOC porté par les messages PATH de bout en bout.

## 3.3 Sessions RSVP avec création dynamique du tunnel

C'est le cas d'un tunnel de type 3. Les seules différences entre ce cas et celui du paragraphe 4.2 sont que :
- La session du tunnel est créée lorsque une nouvelle session de bout en bout apparaît.
- Il y a une transposition biunivoque entre les sessions RSVP de bout en bout et tunnel, par opposition à une possible transposition de plusieurs sur une qui est permise dans le cas décrit au paragraphe 4.2.

# 4. Traitement des messages RSVP sur un tunnel IP dans IP

## 4.1 Messages RSVP pour sessions configurées sur un tunnel

Une ou plusieurs sessions RSVP sont établies sur un tunnel à travers une interface de gestion. Les paramètres de réservation de session ne changent jamais pour un tunnel "tuyau". Les paramètres de réservation peuvent changer pour un tunnel "flexible". Les messages PATH de session de tunnel générés par Rentry sont adressés à Rexit, où ils sont traités puis

détruits.

### 4.2 Traitement des messages RSVP aux points d'extrémité de tunnel

#### 4.2.1 Traitement des messages PATH de bout en bout à Rentry

Lors de la transmission d'un message PATH de bout en bout, un routeur qui agit comme point d'entrée du tunnel, Rentry, prend les mesures suivantes, selon la session de bout en bout mentionnée dans le message PATH. Deux cas sont possibles :

1. il est le rafraîchissement d'une session de bout en bout connue antérieurement ;
2. le message PATH de bout en bout provient d'une nouvelle session de bout en bout.

Si le message PATH est un rafraîchissement d'une session de bout en bout connue précédemment, Rentry rafraîchit alors l'état de chemin de la session de bout en bout et vérifie pour voir si cette session est transposée en une session de tunnel. Si c'est le cas, lorsque Rentry rafraîchit alors la session de bout en bout, il inclut dans le message PATH de bout en bout un objet SESSION_ASSOC qui relie cette session à sa session de tunnel correspondante. Il encapsule alors le message PATH de bout en bout et l'envoie sur le tunnel à Rexit. Si la session du tunnel a été une création dynamique, le message PATH de bout en bout sert de rafraîchissement pour l'état local du tunnel à Rentry ainsi que pour la session de bout en bout.

Autrement, si le message PATH provient d'une nouvelle session de bout en bout qui n'a pas encore été transposée en une session de tunnel, Rentry crée un état de chemin pour cette nouvelle session en réglant l'interface sortante comme interface du tunnel. Après cela, Rentry encapsule le message et l'envoie à Rexit sans ajouter d'objet SESSION_ASSOC.

Lorsque un message PATH TEAR de bout en bout est reçu par Rentry, ce nœud encapsule et transmet le message à Rexit. Si cette session de bout en bout a une transposition biunivoque avec une session de tunnel ou si c'est la dernière de nombreuses sessions de bout en bout se transposant en session de tunnel, Rentry élimine la session du tunnel en envoyant un message PATH TEAR pour cette session à Rexit. Si, d'un autre côté, il reste des transpositions de sessions de bout en bout en session de tunnel, Rentry envoie alors un message PATH de tunnel qui ajuste la Tspec de la session de tunnel.

#### 4.2.2 Traitement des messages PATH de bout en bout à Rexit

Les messages PATH encapsulés de bout en bout sont désencapsulés et traités à Rexit. Selon que le message PATH de bout en bout contient ou non un objet SESSION_ASSOC, Rexit prend les mesures suivantes :

1. Si le message PATH de bout en bout ne contient pas d'objet SESSION_ASSOC, Rentry règle alors le fanion Non_RSVP à l'état de chemin mémorisé pour cet envoyeur de bout en bout, règle le bit de coupure global dans l'ADSPEC et transmet les paquets vers l'aval. Autrement, si les sessions de tunnel existent et qu'aucune d'entre elles n'a le fanion Non_RSVP établi, Rexit peut prendre les paramètres ADSPEC de plus mauvais cas de chemin des sessions de tunnel existantes et mettre à jour l'ADSPEC de bout en bout en utilisant ces valeurs. C'est une estimation prudente de l'ADSPEC composée mais elle présente l'avantage d'éviter d'établir le bit de coupure dans l'ADSPEC de bout en bout avant de transposer les informations disponibles. Dans ce cas, le fanion Non_RSVP n'est pas établi à l'état de chemin de bout en bout.

2. Si le message PATH contient un objet SESSION_ASSOC et qu'aucune association n'existe déjà pour cette session de bout en bout, le Rexit enregistre alors l'association entre la session de bout en bout et la session de tunnel décrite par l'objet. Si le PATH de bout en bout arrive avant que le message PATH de tunnel n'arrive, il crée alors l'état PATH à Rexit pour la session de tunnel. Lorsque le message PATH réel arrive pour la session de tunnel, il est traité comme mise à jour de l'état PATH existant et il met à jour toute information manquante. On pense que cette situation est transitoire avec d'autres qui existent dans RSVP et que cela n'a pas d'effet à long terme sur le fonctionnement correct du mécanisme décrit ici.

Avant de transmettre le message au prochain bond sur le chemin vers la destination, Rexit trouve l'état enregistré pour la session de tunnel correspondante et établit le fanion Non_RSVP dans l'état de chemin de bout en bout si le bit Non_RSVP était établi pour la session du tunnel. Si le message PATH de bout en bout porte un objet ADSPEC, Rexit effectue la composition des paramètres de caractérisation contenus dans l'ADSPEC. Il fait cela en considérant la totalité des paramètres de caractérisation (composés) de la session du tunnel comme les paramètres locaux pour la liaison logique mise en œuvre par le tunnel, et en composant ces paramètres avec ceux de l'ADSPEC de bout en bout en exécutant chaque fonction de composition définie du paramètre. Dans les paramètres de caractérisation de la liaison logique, la latence minimum de chemin peut tenir compte du délai d'encapsulation/désencapsulation et l'estimation de bande passante peut représenter la diminution de bande passante disponible causée par l'ajout de l'en-tête UDP supplémentaire. Les ADSPEC et

les fonctions de composition sont exposées en grand détail dans la [RFC2210].

Si la session de bout en bout a un état de réservation, alors qu'aucun état de réservation n'existe pour la session de tunnel correspondante, Rexit envoie un message RESV de tunnel à Rentry correspondant à la réservation dans la session de bout en bout.

Si Rentry ne prend pas en charge le tunnelage RSVP, Rexit n'aura alors pas d'état PATH pour le tunnel. Dans ce cas, Rexit met simplement le bit de coupure global dans le message PATH désencapsulé de bout en bout et le transmet.

### 4.2.3 Traitement des messages RESV de bout en bout à Rexit

Lors de la transmission d'un message RESV vers l'amont, un routeur qui sert de routeur de sortie, Rexit, peut découvrir qu'une des interfaces vers l'amont est un tunnel. Dans ce cas, le routeur effectue un certain nombre d'essais.

Étape 1 : Rexit doit déterminer si il y a une session de tunnel liée à la session de bout en bout donnée dans le message RESV. Si il n'y en a pas, le tunnel est traité comme une liaison non-RSVP, Rexit ajoute au message RESV un objet NODE_CHAR avec le bit T établi, et le transmet sur l'interface du tunnel (où il est encapsulé comme un datagramme IP normal et transmis vers Rentry).

Étape 2 : Si il trouve une session de tunnel liée, Rexit vérifie pour voir si une réservation est déjà en place pour la session de tunnel liée à la session de bout en bout donnée dans le message RESV. Si le message RESV arrivant de bout en bout est un rafraîchissement d'un état RESV existant, Rexit envoie alors le RESV original à travers l'interface de tunnel (après avoir ajouté l'objet NODE_CHAR). Pour les sessions de tunnel dynamiques, le message RESV de bout en bout agit comme un rafraîchissement pour l'état de réservation de la session du tunnel, alors que pour les sessions de tunnel configurées, l'état de réservation n'arrive jamais à expiration.

Si le message RESV de bout en bout arrivant cause un changement des paramètres de flowspec RESV de bout en bout, il peut aussi déclencher une tentative de changement des paramètres de flowspec de la session de tunnel. Dans ce cas, Rexit envoie un RESV de session de tunnel, incluant un objet RESV_CONFIRM.

Dans le cas d'un tunnel "tuyau", une nouvelle réservation de bout en bout ou un changement du niveau des ressources demandées par une réservation existante peut causer le dépassement du niveau de ressources réservées par la réservation du tunnel par le niveau de ressources total nécessaire pour les réservations de bout en bout. Cet évènement devrait être traité comme un échec de contrôle d'admission, identique au cas où les demandes RSVP excèdent le niveau de ressources disponibles sur une liaison matérielle. Un message RESV_ERR avec le code d'erreur réglé à 01 (Échec de contrôle d'admission) devrait être renvoyé à l'origine du message RESV de bout en bout.

Si une réponse RESV CONFIRM arrive, le RESV original est encapsulé et envoyé à travers le tunnel. Si la réservation de tunnel mise à jour échoue, Rexit doit envoyer une RESV ERR à l'origine du message RESV de bout en bout, en utilisant les champs de code d'erreur et de valeur de l'objet ERROR_SPEC du message RESV ERR de session de tunnel reçu. Noter que les réservations préexistantes à travers le tunnel restent en place. Rexit continue de rafraîchir la RESV de tunnel en utilisant l'ancienne flowspec.

L'état de session de tunnel pour un "flexible" peut aussi être ajusté lorsque une réservation de bout en bout est supprimée. La session de tunnel est réduite chaque fois qu'une session de bout en bout utilisant le tunnel s'en va (ou est elle-même réduite). Cependant même lorsque la dernière session de bout en bout liée à ce tunnel s'en va, la session de tunnel configurée reste active, peut-être avec une flowspec configurée minimale.

Noter qu'il sera souvent approprié d'utiliser de l'hystérèse dans l'ajustement des paramètres de réservation du tunnel, plutôt que d'augmenter et diminuer la réservation de tunnel à chaque arrivée ou départ de réservation de bout en bout. Le faire exigera que le routeur de sortie du tunnel garde trace des ressources allouées au tunnel (la flowspec du tunnel) et des ressources réellement utilisées séparément par les réservations de bout en bout (la somme ou la somme statistique des flowspec de réservation de bout en bout).

Lorsque un message RESV TEAR de bout en bout est reçu par Rexit, il encapsule et transmet le message à Rentry. Si la session de bout en bout avait créé une session de tunnel dynamique, une RESV TEAR est alors envoyée par Rexit pour la session de tunnel correspondante.

#### 4.2.4 Traitement des messages RESV de bout en bout à Rentry.

Si le message RESV reçu est un rafraîchissement d'une réservation existante Rentry met alors à jour l'état de réservation et transmet le message vers l'amont. D'un autre côté, si c'est le premier message RESV pour cette session de bout en bout et si un objet NODE_CHAR avec le bit T établi est présent, Rentry devrait initier la transposition entre cette session de bout en bout et quelque autre session (éventuellement nouvelle) du tunnel. Cette transposition se fonde sur certains (ou tous) des contenus du message PATH de bout en bout, sur le contenu du message RESV de bout en bout, et sur les politiques locales. (Par exemple, il pourrait y avoir des sessions de tunnel différentes sur la base des exigences de bande passante ou de délai des sessions de bout en bout.)

Si Rentry décide que cette session de bout en bout devrait être transposée en une session du tunnel configurée existante, il lie cette session de bout en bout à cette session du tunnel.

Si cette session RSVP de bout en bout est admise à établir une nouvelle session de tunnel, Rentry établit un état PATH de session de tunnel comme si il était une source de données en commençant à envoyer des messages PATH de session de tunnel à Rexit, qui est traité comme la destination en envoi individuel des données. La Tspec dans ce nouveau message PATH est calculée à partir du message PATH original en ajustant les paramètres de Tspec pour y inclure la redondance de tunnel de l'encapsulation des paquets de données. Dans ce cas, Rentry devrait aussi envoyer un message PATH à partir de la session de bout en bout contenant cette fois l'objet SESSION_ASSOC reliant les deux sessions. La réception de ce message PATH par Rexit va déclencher une mise à jour de l'état de chemin de bout en bout qui à son tour aura pour effet l'envoi par Rexit d'un message RESV de tunnel, allouant les ressources à l'intérieur du tunnel.

Le dernier cas est quand la session de bout en bout n'est pas admise à utiliser les ressources du tunnel. Dans ce cas, aucune association n'est créée entre cette session de bout en bout et une session de tunnel et aucune nouvelle session de tunnel n'est créée.

Une limitation de notre schéma est que le premier message RESV d'une session de bout en bout détermine la transposition entre cette session de bout en bout et sa session correspondante sur le tunnel. De plus, tant que la réservation est active cette transposition ne peut changer.

## 5. Transmission des données

Lorsque les paquets de données arrivent au point d'entrée du tunnel Rentry, celui-ci doit décider si il transmet les paquets en utilisant l'encapsulation normale de tunnel IP dans IP ou l'encapsulation IP+UDP attendue par la session de tunnel. Cette décision est prise en déterminant si il y a une réservation de ressource (pas seulement l'état PATH) actuellement en place pour la session du tunnel liée au paquet qui arrive, c'est-à-dire, si le paquet correspond à une filterspec active.

Si une réservation est en place, cela signifie que Rentry et Rexit sont tous deux des routeurs à capacité de tunnelage RSVP, et que les données seront correctement désencapsulées à Rexit.

Si aucune réservation de session de tunnel n'est en place, les données devraient être encapsulées dans le format normal du tunnel, sans considération de la présence ou non d'un état PATH de bout en bout couvrant ces données.

## 6. Détails

### 6.1 Choix des numéros d'accès UDP

Il peut y avoir plusieurs sessions RSVP de bout en bout entre les deux points d'extrémité Rentry et Rexit. Ces sessions se distinguent par l'accès UDP de source. Les autres composants de l'identifiant de session, les adresses IP de source et de destination et l'accès UDP de destination, sont identiques pour toutes ces sessions.

L'accès UDP de source est choisi par le point d'entrée du tunnel Rentry lorsque il établit l'état PATH initial pour une nouvelle session de tunnel. L'accès UDP de source associé à la nouvelle session est alors convoyé à Rexit par l'objet SESSION_ASSOC.

L'accès UDP de destination utilisé dans les sessions de tunnel devrait être celui alloué par l'IANA (363).

### 6.2 Rapports d'erreur

Lorsque un message PATH de session de tunnel rencontre une erreur, elle est rapportée à Rentry. Rentry doit relayer le rapport d'erreur à la source originale de la session de bout en bout.

Lorsque une demande RESV de session de tunnel échoue, un message d'erreur est retourné à Rexit. Rexit doit traiter cela comme une erreur dans la traversée de la liaison logique (le tunnel) et retransmettre le message d'erreur à l'hôte d'extrémité.

### 6.3 Découverte de MTU

Comme les paquets encapsulés par UDP ne devraient pas être fragmentés, les routeurs d'entrée de tunnel doivent prendre en charge la découverte de MTU de tunnel comme exposé au paragraphe 5.1 de la [RFC2003]. Autrement, le mécanisme de découverte de la MTU de chemin exposé dans la [RFC2210] peut être utilisé.

### 6.4 Calculs des Tspec et Flowspec

Comme plusieurs sessions de bout en bout peuvent être transposées en une seule session de tunnel, il est nécessaire de calculer la Tspec agrégée de tous les envoyeurs de ces sessions de bout en bout. Cette Tspec agrégée sera la Tspec de la session de tunnel représentative. Il est nécessaire d'effectuer la même opération pour les flowspec des réservations de bout en bout qui arrivent à Rexit.

La sémantique de ces opérations n'est pas traitée ici. La façon la plus simple de les effectuer est de calculer une somme des Tspec de bout en bout, comme défini dans les spécifications des services Charge contrôlée et Garantie de service (respectivement [RFC2211] et [RFC2212]). Cependant, il peut aussi être approprié de calculer le niveau agrégé de réservation pour le tunnel en utilisant un calcul fondé sur la statistique ou des mesures plus sophistiquées.

## 7. Tunnels IPSEC

Dans le cas où le tunnel IP dans IP prend en charge IPSEC (en particulier ESP en mode tunnel avec ou sans AH) la session de tunnel utilise alors les objets GPI SESSION et GPI SENDER_TEMPLATE/FILTER_SPEC comme défini dans la [RFC2207] pour les messages PATH et RESV.

Les paquets de données ne sont pas encapsulés avec un en-tête UDP car le SPI *(Security Parameter Index, indice de paramètre de sécurité)* peut être utilisé par les nœuds intermédiaires pour les besoins du classement. Remarquer que le chiffrement orienté utilisateur doit être utilisé entre Rentry et Rexit, afin que des SPI différents soient alloués aux paquets de données qui ont une réservation et aux paquets "au mieux", ainsi qu'aux paquets qui appartiennent à des sessions de tunnel différentes si celles-ci sont prises en charge.

## 8. Prise en charge de RSVP pour les tunnels en diffusion groupée et multipoints

Les mécanismes décrits ci-dessus sont utiles pour les tunnels en envoi individuel. Les tunnels en envoi individuel fournissent des liaisons logiques en point à point dans l'infrastructure IP, bien qu'ils puissent encapsuler et transporter du trafic aussi bien en envoi individuel qu'en diffusion groupée entre ces points.

Deux autres types de tunnels peuvent être imaginés. Le premier d'entre eux est un tunnel "en diffusion groupée". Dans ce type de tunnel, les paquets qui arrivent à un point d'entrée sont encapsulés et transportés (en diffusion groupée) à tous les points de sortie. Cette sorte de tunnel peut se révéler utile pour la mise en œuvre d'un réseau hiérarchisé de distribution de diffusion groupée, ou pour émuler efficacement certaines portions d'une arborescence native de distribution de diffusion groupée.

Un second type de tunnel possible est le tunnel "multipoint". Dans ce type de tunnel, les paquets qui arrivent au point d'entrée sont normalement encapsulés et transportés à un des points de sortie, conformément à un algorithme de choix de chemin.

Ce type de tunnel diffère de tous les précédents types en ce que la "forme" du chemin de distribution usuel des données ne

correspond pas à la "forme" du tunnel. La topologie du tunnel ne définit pas par elle-même la fonction de transmission des données effectuée par le tunnel. Au lieu de cela, le tunnel devient un moyen d'exprimer des propriétés partagées par l'ensemble de la session de bout en bout connectée. Par exemple, le "tunnel" peut être utilisé pour créer et incorporer un réseau logique de diffusion partagé au sein de quelque réseau plus grand. Dans ce cas, la session de bout en bout est constituée par les nœuds connectés au réseau logique de diffusion partagé. Le trafic des données peut être en envoi individuel entre deux de ces nœuds, diffusé à tous les nœuds connectés, ou envoyés en diffusion groupée entre un sous-ensemble des nœuds connectés. Le tunnel lui-même est utilisé pour définir un domaine dans lequel gérer l'acheminement et la gestion de ressources – essentiellement un réseau virtuel privé (VPN).

Noter qu'alors qu'un VPN de cette forme peut toujours être mis en œuvre en utilisant un tunnel en diffusion groupée pour "émuler" le support de diffusion, cette approche sera inefficace dans le cas de VPN de large zone, et un tunnel multipoint avec des mécanismes de contrôle appropriés sera préférable.

Les paragraphes qui suivent fournissent un bref commentaire sur l'utilisation de RSVP dans ces situations. Les futures versions de ce document donneront des détails et des spécifications plus concrètes.

L'utilisation de RSVP pour faire de la gestion de ressource sur un tunnel en diffusion groupée est relativement directe. Comme dans le cas de l'envoi individuel, une ou plusieurs sessions RSVP peuvent être utilisées, et les sessions RSVP de bout en bout peuvent être transposées en sessions de tunnel RSVP de plusieurs à une ou en biunivoque. À la différence du cas de l'envoi individuel, la transposition est cependant compliquée par l'hétérogénéité sémantique de RSVP. Si différents receveurs ont fait des demandes de réservation différentes, il se peut que les messages RESV arrivant au tunnel transposent logiquement les demandes de receveurs sur des sessions de tunnel différentes. Comme en fait les données ne peuvent être placées que dans une seule session, le choix de la session doit être fait (fusionné) pour sélectionner celle qui va satisfaire aux besoins de toutes les applications. Cela exige une extension relativement simple au mécanisme de transposition de la session.

L'utilisation de RSVP pour prendre en charge les tunnels multipoints est un peu plus difficile. Dans ce cas, l'objectif est de donner au tunnel pris dans son ensemble un niveau spécifique de ressources. Par exemple, on peut souhaiter émuler un "Ethernet à 10 mégabits à partage logique " plutôt qu'un "Ethernet à partage logique". Cependant, le problème est compliqué par le fait que dans ce type de tunnel les données ne vont pas toujours à toutes les sessions de bout en bout. Cela implique qu'on ne peut pas utiliser l'adresse de destination des paquets encapsulés au titre du filtre de classement de paquet, parce que l'adresse de destination va varier pour les différents paquets dans le tunnel.

Cela implique le besoin d'une extension à la sémantique actuelle de la session RSVP dans laquelle l'identifiant de session (adresse IP de destination) est utilisée – seulement - pour identifier l'état de la session au sein des nœuds du réseau, mais n'est pas utilisé pour classer les paquets. Autrement, l'utilisation de RSVP pour les tunnels multipoint suit celles des tunnels en diffusion groupée. Un groupe de diffusion groupée est créé pour représenter l'ensemble des nœuds qui sont de session de bout en bout, et un ou plusieurs tunnels de session RSVP sont créés pour réserver des ressources pour les paquets encapsulés. Dans le cas d'un tunnel qui met en œuvre un simple VPN, il est très vraisemblable qu'il y aura une session pour réserver les ressources pour le VPN tout entier. Chaque point d'extrémité de tunnel va participer à la fois comme source de messages PATH et comme source de messages RESV (FF ou SE) pour cette seule session, créant effectivement une seule réservation partagée pour la totalité du support logique partagé. Les points d'extrémité de tunnel NE DOIVENT PAS faire de réservations génériques sur des tunnels multipoints.

## 9. Extensions à l'interface d'acheminement RSVP

La spécification RSVP [RFC2205] déclare qu'à travers l'interface d'acheminement RSVP, le démon RSVP doit être capable d'apprendre la liste des interfaces locales avec leurs adresses IP. Dans le cas des tunnels RSVP, le démon RSVP a aussi besoin d'apprendre la ou lesquelles des interfaces locales sont des tunnels IP dans IP qui ont les capacités décrites ici. Le démon RSVP peut acquérir ces informations, soit en interrogeant directement les couches réseau et physique sous-jacentes, soit en utilisant toute interface existante entre RSVP et le protocole d'acheminement étendu comme il convient pour fournir ces informations.

## 10. Considérations pour la sécurité

L'introduction des tunnels RSVP n'introduit aucune nouvelle question de sécurité autre que celles associées à l'utilisation de RSVP et des tunnels. En ce qui concerne RSVP, la question principale est la nécessité de contrôler et authentifier l'accès

aux qualités de service améliorées. Cette exigence est exposée plus en détails dans la [RFC2205]. La [RFC2747] décrit le mécanisme utilisé pour protéger l'intégrité des messages RSVP qui portent les informations décrites ici. Les questions de sécurité associées avec les tunnels IP dans IP sont discutées dans les [RFC2003] et [RFC2473].

## 11. Considérations relatives à l'IANA

L'IANA devrait allouer un numéro de classe pour l'objet NODE_CHAR défini au paragraphe 3.3.2. Ce numéro devrait être dans la gamme 10bbbbbb. La valeur suggérée est 128.

## 12. Remerciements

Merci à Bob Braden dont les commentaires éclairés nous ont aidé à produire cette version mise à jour du document.

## 13. Références

*(Les liens sur les numéros pointent sur la version anglaise, ceux du corps du titre sur la traduction française)*

[RFC1701] S. Hanks, T. Li, D. Farinacci et P. Traina, "Encapsulation générique d'acheminement (GRE)", octobre 1994. *(Information)*

[RFC1702] S. Hanks, T. Li, D. Farinacci et P. Traina, "Encapsulation générique d'acheminement sur réseaux IPv4", octobre 1994. *(Info.)*

[RFC1827] R. Atkinson, "Encapsulation dans IP de charge utile de sécurité (ESP)", août 1995. *(Obsolète, voir RFC2406)*

[RFC1933] R. Gilligan, E. Nordmark, "Mécanismes de transition pour hôtes et routeurs IPv6", avril 1996. (*Obsolète, voir* RFC2893) *(P.S.)*

[RFC2003] C. Perkins, "Encapsulation de IP dans IP", octobre 1996.

[RFC2004] C. Perkins, "Encapsulation minimale au sein de IP", octobre 1996. *(P.S.)*

[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.)*

[RFC2207] L. Berger, T. O'Malley, "Extensions RSVP pour flux de données IPSEC", septembre 1997. *(P.S.)*

[RFC2210] J. Wroclawski, "Utilisation de RSVP avec les services intégrés de l'IETF", septembre 1997. *(P.S.)*

[RFC2211] J. Wroclawski, "Spécification du service d'élément de réseau à charge contrôlée", septembre 1997. *(P.S.)*

[RFC2212] S. Shenker, C. Partridge, R. Guerin, "Spécification de la qualité de service garantie", septembre 1997. *(P.S.)*

[RFC2473] A. Conta, S. Deering, "Spécification du tunnelage générique de paquet dans IPv6", décembre 1998. *(P.S.)*

[RFC2747] F. Baker, B. Lindell, M. Talwar, "Authentification cryptographique RSVP", janvier 2000. (*MàJ par* RFC3097) *(P.S.)*

## 14. Adresse des auteurs

John Krawczyk
ArrowPoint Communications
50 Nagog Park
Acton, MA 01720
téléphone : 978-206-3027
mél : jj@arrowpoint.com

John Wroclawski
MIT Laboratory for Computer Science
545 Technology Sq.
Cambridge, MA 02139
téléphone : 617-253-7885
Fax : 617-253-2673
mél : jtw@lcs.mit.edu

Lixia Zhang
UCLA
4531G Boelter Hall
Los Angeles, CA 90095
téléphone : 310-825-2695
mél : lixia@cs.ucla.edu

Andreas Terzis
UCLA
4677 Boelter Hall
Los Angeles, CA 90095
téléphone : 310-267-2190
mél : terzis@cs.ucla.edu

## 15. Déclaration complète de droits de reproduction



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