Groupe de travail Réseau
Request for Comments : 2205
Catégorie : En cours de normalisation
septembre 1997
Traduction Claude Brière de L'Isle

R. Braden, Ed., ISI
L. Zhang, UCLA
S. Berson, ISI
S. Herzog, IBM Research
S. Jamin, Univ. of Michigan

Protocole de réservation de ressource (RSVP) -- version 1, Spécification fonctionnelle

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.

Résumé

Le présent mémoire décrit la version 1 de RSVP, un protocole d'établissement de réservation de ressource conçu pour un Internet à intégration de services. RSVP fournit un établissement de réservation de ressources à l'initiative du receveur pour les flux de données en diffusion groupée ou en envoi individuel, avec de bonnes propriétés d'adaptabilité et de robustesse.

Table des Matières

Ce qui change

La présente révision contient les changements très mineurs suivants par rapport à la version ID14 :

- Dans un souci de clarté, chaque type de message est maintenant défini séparément au paragraphe 3.1.
- On a ajouté des règles précises et complètes pour l'acceptation des messages Path pour les destinations d'envoi individuel et de diffusion groupée (paragraphe 3.1.3).
- On a ajouté des règles précises et complètes pour le traitement et la transmission des messages PathTear (paragraphe 3.1.5).
- On a ajouté une note disant qu'un objet SCOPE sera ignoré si il apparaît dans un message ResvTear (paragraphe 3.1.6).
- Ajout d'une note disant qu'un objet SENDER_TSPEC ou ADSPEC sera ignoré si il apparaît dans un message PathTear (paragraphe 3.1.5).
- Le code d'erreur obsolète Spécification de filtre ambiguë (09) a été supprimé, et un nouveau nom (plus cohérent) a été donné au code d'erreur 08 (Appendice B).
- Dans l'interface générique au contrôle de trafic, le paramètre Adspec a été ajouté aux appels AddFlow et ModFlow (3.11.2). Cet ajout est nécessaire pour s'accommoder d'un nœud qui met à jour le terme mou (S) de service garanti.
- Un sous-type d'erreur a été ajouté pour une erreur Adspec (Appendice B).
- Des explications supplémentaires ont été ajoutées pour le traitement d'un objet CONFIRM (paragraphe 3.1.4).
- Les règles de transmission des objets qui ont un type de classe inconnu ont été précisées.
- Un exposé supplémentaire est ajouté à l'Introduction et au paragraphe 3.11.2 sur les relations de RSVP avec la couche liaison. (paragraphe 3.10).
- Le paragraphe 2.7 sur la politique et la sécurité a été partagé en deux, et des considérations de sécurité supplémentaires ont été incluses.
- Plus quelques améliorations rédactionnelles mineures.

1. Introduction

Le présent document définit RSVP, un protocole d'établissement de réservation de ressources conçu pour un Internet à intégration de services [RSVP93], [RFC1633]. Le protocole RSVP est utilisé par un hôte pour demander des qualités de service spécifiques de la part du réseau pour des flux de données d'application particuliers. RSVP est aussi utilisé par des routeurs pour livrer des demandes de qualité de service (QS) à tous les nœuds le long du ou des chemins des flux et pour établir et maintenir l'état destiné à fournir le service demandé. Les demandes RSVP vont généralement résulter en la réservation de ressources dans chaque nœud le long du chemin des données.

RSVP demande des ressources pour des flux unidirectionnels *(simplex)*, c'est à dire qu'il demande des ressources seulement dans une direction. Donc, RSVP traite un envoyeur comme logiquement distinct d'un receveur, bien que le même processus d'application puisse agir à la fois comme envoyeur et comme receveur au même moment. RSVP fonctionne par dessus IPv4 ou IPv6, occupant la place d'un protocole de transport dans la pile de protocoles. Cependant, RSVP ne transporte aucune données d'application ; il est plutôt un protocole de contrôle Internet, comme ICMP, IGMP, ou les protocoles d'acheminement. Comme la mise en œuvre de protocoles d'acheminement et de gestion, une mise en œuvre de RSVP va normalement s'exécuter en arrière plan, et non dans le chemin de transmission des données, comme le montre la Figure 1.

RSVP n'est pas lui-même un protocole d'acheminement ; RSVP est conçu pour fonctionner avec les protocoles actuels et futurs d'acheminement en envoi individuel et en diffusion groupée. Un processus RSVP consulte la ou les bases de données d'acheminement locales pour obtenir les routes. Dans le cas de la diffusion groupée, par exemple, un hôte envoie des messages IGMP pour se joindre à un groupe de diffusion groupée puis envoie des messages RSVP pour réserver des ressources le long du ou des chemins de livraison de ce groupe. Les protocoles d'acheminement déterminent si les paquets sont transmis ; RSVP est uniquement concerné par la QS de ces paquets qui sont transmis conformément à ces acheminements.

Afin de s'accommoder efficacement des grands groupes, de l'adhésion dynamique aux groupes, et de l'hétérogénéité des exigences des receveurs, RSVP rend les receveurs responsables de la demande d'une QS spécifique [RSVP93]. Une demande de QS de la part d'une application d'hôte receveur est passée au processus RSVP local. Le protocole RSVP porte alors la demande à tous les nœuds (routeurs et hôtes) le long du ou des chemins inverses des données vers la ou les sources de données, mais seulement jusqu'au routeur où le chemin des données du receveur rejoint l'arbre de distribution de diffusion groupée. Il en résulte que la redondance de réservation de RSVP est en général logarithmique plutôt que linéaire en nombre de receveurs.

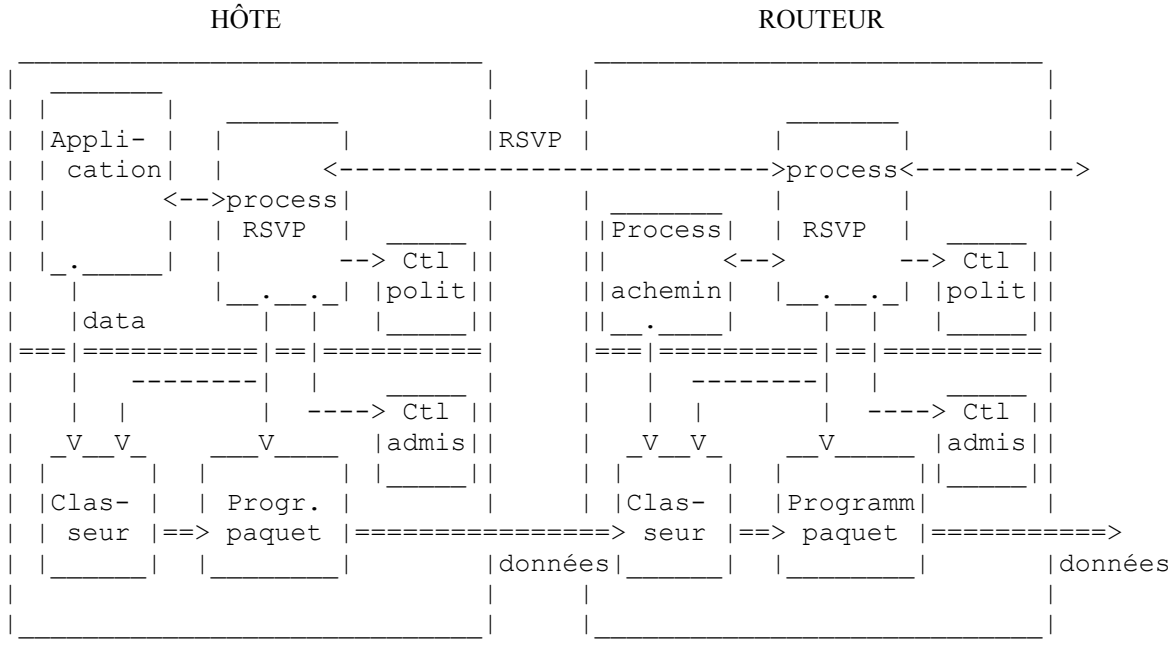


Figure 1 : RSVP dans les hôtes et les routeurs

La qualité de service est mise en œuvre pour un flux de données particulier par des mécanismes collectivement appelés "contrôle de trafic". Ces mécanismes incluent (1) un classeur de paquet, (2) un contrôle d'admission, et (3) un "programmeur de paquet" ou quelque autre mécanisme dépendant de la couche liaison pour déterminer quand des paquets particuliers sont transmis. Le "classeur de paquet" détermine la classe de QS (et peut-être le chemin) pour chaque paquet. Pour chaque interface sortante, le "programmeur de paquet" ou autre mécanisme dépendant de la couche liaison réalise la QS promise. Le contrôle de trafic met en œuvre les modèles de service de QS définis par le groupe de travail Intégration de services.

Durant l'établissement de réservation, une demande de QS RSVP est passée à deux modules de décision locaux, "contrôle d'admission" et "contrôle de politique". Le contrôle d'admission détermines si le nœud a des ressources disponibles suffisantes pour fournir la QS demandée. Le contrôle de politique détermine si l'utilisateur a la permission administrative de faire la réservation. Si les deux vérifications réussissent, les paramètres sont établis dans le classeur de paquet et dans l'interface de couche liaison (par exemple, dans le programmeur de paquet) pour obtenir la QS désirée. Si l'une des vérifications échoue, le programme RSVP retourne une notification d'erreur au processus d'application qui a généré la requête.

Les mécanismes du protocole RSVP fournissent une facilité générale de création et de maintenance d'un état de réservation distribué à travers un maillage de chemins de livraison en diffusion groupée ou en envoi individuel. RSVP lui-même transfère et manipule les paramètres de QS et de contrôle de politique comme des données opaques, les passant aux modules appropriés de contrôle de trafic et de contrôle de politique pour leur interprétation. La structure et le contenu des paramètres de QS sont documentés dans des spécifications développées par le groupe de travail Intégration de services ; voir la [RFC2210]. La structure et le contenu des paramètres de politique sont en cours de développement.

Comme l'adhésion à un grand groupe de diffusion groupée et la topologie résultante de l'arborescence de diffusion groupée vont vraisemblablement changer avec le temps, la conception de RSVP suppose que l'état pour RSVP et l'état pour le contrôle de trafic sont à construire et détruire de façon incrémentaire dans les routeurs et les hôtes. À cette fin, RSVP établit un état "mou" ; c'est-à-dire, RSVP envoie des messages périodiques de rafraîchissement pour maintenir l'état le long du ou des chemins réservés. En l'absence de message de rafraîchissement, l'état se périme automatiquement puis est supprimé.

En résumé, RSVP a les attributs suivants :

- o RSVP fait des réservations de ressource pour les applications aussi bien en envoi individuel qu'en diffusion groupée de plusieurs à plusieurs, s'adaptant de façon dynamique aux changement d'appartenance au groupe ainsi qu'aux changements de chemins.
- o RSVP est unidirectionnel, c'est-à-dire qu'il fait les réservations pour des flux de données unidirectionnels.
- o RSVP est orienté receveur, c'est-à-dire que le receveur d'un flux de données initie et entretient la réservation de ressource utilisée pour ce flux.
- o RSVP maintient un état "mou" dans les routeurs et les hôtes, fournissant une prise en charge gracieuse aux changements dynamiques d'appartenance et une adaptation automatique aux changements d'acheminement.
- o RSVP n'est pas un protocole d'acheminement mais dépend des protocoles d'acheminement présents et futurs.
- o RSVP transporte et maintient les paramètres de contrôle de trafic et de contrôle de politique qui sont opaques pour RSVP.
- o RSVP fournit plusieurs modèles de réservation ou "styles" (définis ci-dessous) pour s'adapter à diverses applications.
- o RSVP fonctionne de façon transparente à travers les routeurs qui ne le prennent pas en charge.
- o RSVP accepte IPv4 et IPv6.

Un exposé plus complet sur les objectifs et les justifications générales du concept de RSVP sont présentés dans [RSVP93] et dans la [RFC1633].

Le reste de cette section décrit les services de réservation de RSVP. La Section 2 présente les généralités des mécanismes du protocole RSVP. La Section 3 contient la spécification fonctionnelle de RSVP, tandis que la Section 4 présente les règles explicites de traitement de message. L'Appendice A définit les objets de données de types de longueur variable utilisés dans le protocole RSVP. L'Appendice B définit les codes d'erreur et les valeurs. L'Appendice C définit une encapsulation UDP des messages RSVP, pour les hôtes dont les systèmes d'exploitation fournissent une prise en charge inadéquate de l'entrée/sortie brute de réseau.

1.1 Flux de données

RSVP définit une "session" comme un flux de données avec une destination particulière et un protocole de couche transport. RSVP traite chaque session de façon indépendante, et le présent document omet souvent la qualification implicite "pour la même session".

Une session RSVP se défini par le triplet (DestAddress, ProtocolId [, DstPort]). Ici DestAddress, l'adresse IP de destination des paquets de données, peut être une adresse d'envoi individuel ou de diffusion groupée. ProtocolId est l'identifiant de protocole IP. Le paramètre facultatif DstPort est un "accès de destination généralisé", c'est-à-dire un point ultérieur de démultiplexage dans la couche transport ou application. DstPort pourrait être défini par un champ d'accès de destination UDP/TCP par un champ équivalent dans un autre protocole de transport, ou par des informations spécifiques de l'application.

Bien que le protocole RSVP soit conçu pour être facilement extensible pour une plus grande généralité, le protocole de base spécifié ici ne prend en charge que les accès UDP/TCP comme accès généralisés. Noter qu'il n'est pas strictement nécessaire d'inclure DstPort dans la définition de session lorsque DestAddress est une diffusion groupée, car différentes sessions peuvent toujours avoir des adresses de diffusion groupée différentes. Cependant, DstPort est nécessaire pour permettre plus d'une session en envoi individuel adressée au même hôte receveur.

La Figure 2 illustre les flux de paquets de données dans une seule session RSVP, en supposant une distribution des données en diffusion groupée. La flèche indique des données s'écoulant des envoyeurs S1 et S2 aux receveurs R1, R2, et R3, et le nuage représente le maillage de distribution créé par l'acheminement de diffusion groupée. La distribution de diffusion groupée transmet une copie de chaque paquet de données provenant d'un envoyeur S_i à chaque receveur R_j ; une session de distribution en envoi individuel a un seul receveur R. Chaque envoyeur S_i peut fonctionner dans un hôte Internet unique, ou un seul hôte peut contenir plusieurs envoyeurs distingués par des "accès de source généralisés".

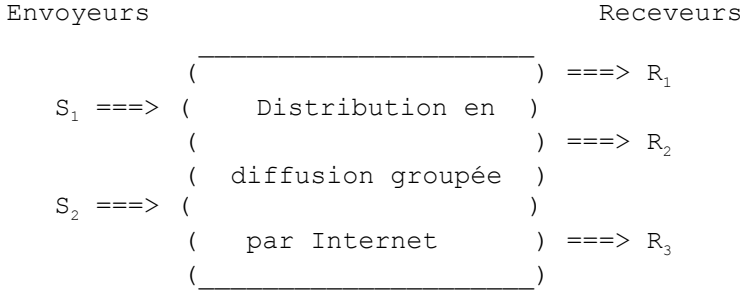


Figure 2 : Session à distribution en diffusion groupée

Pour la transmission en envoi individuel, il y aura un seul hôte de destination mais il peut y avoir plusieurs envoyeurs; RSVP peut établir des réservations pour une transmission. de plusieurs points à un seul point

1.2 Modèle de réservation

Une demande de réservation RSVP élémentaire consiste en une "flowspec" *(spécification de flux)* jointe à une "spécification de filtre" ; cette paire est appelée un "descripteur de flux". La flowspec spécifie une QS désirée. La spec de filtre, jointe à une spécification de session, définit l'ensemble de paquets de données -- le "flux" – qui recevra la QS définie par la flowspec. La flowspec est utilisée pour établir des paramètres dans le programmeur de paquet ou un autre mécanisme de couche liaison du nœud, tandis que la spécification de filtre est utilisée pour régler des paramètres dans le classeur de paquet. Les paquets de données qui sont adressés à une session particulière mais ne satisfont à aucune des spécifications de filtre pour cette session sont traités comme du trafic au mieux.

La flowspec dans une demande de réservation va généralement inclure une classe de service et deux ensembles de paramètres numériques : (1) une "Rspec" (R pour `réserve') qui définit la QS désirée, et (2) une "Tspec" (T pour `trafic') qui décrit le flux de données. Les formats et contenus des Tspec et Rspec sont déterminés par les modèles de service intégré [RFC2210] et sont généralement opaques à RSVP.

Le format exact d'une spec de filtre dépend de l'utilisation de IPv4 ou IPv6 ; voir l'Appendice A. Dans l'approche la plus générale [RSVP93], les spec de filtre peuvent choisir des sous-ensembles arbitraires des paquets dans une session donnée. De tels sous-ensembles peuvent être définis en termes d'envoyeurs (c'est à dire, d'adresse IP d'envoyeur et d'accès de source généralisé), en termes de protocole de niveau supérieur, ou généralement en termes de champs d'en-tête de tout protocole dans le paquet. Par exemple, les spec de filtre peuvent être utilisées pour choisir différents sous flux d'un flux vidéo codé hiérarchiquement en choisissant des champs dans un en-tête de couche application. Dans un but de simplification (et pour minimiser les violations de couches) le format de base de spec de filtre défini dans la présente spécification RSVP a une forme très restreinte : l'adresse IP d'envoyeur et facultativement le numéro d'accès UDP/TCP SrcPort.

Comme les numéros d'accès UDP/TCP sont utilisés pour le classement des paquets, chaque routeur doit être capable d'examiner ces champs. Cela soulève trois problèmes potentiels.

1. Il est nécessaire d'éviter la fragmentation IP d'un flux de données pour lequel une réservation de ressource est désirée. La [RFC2210] spécifie une procédure pour les applications qui utilisent les facilités RSVP pour calculer la MTU minimum sur une arborescence de diffusion groupée et qui retournent le résultat aux envoyeurs.

2. IPv6 insère un nombre variable d'en-têtes de longueur variable de couche Internet avant l'en-tête de transport, ce qui accroît la difficulté et le coût du classement de paquet pour la QS. Un classement efficace des paquets de données IPv6 pourrait être obtenu en utilisant le champ Étiquette de flux de l'en-tête IPv6. Les détails seront fournis à l'avenir.

3. La sécurité de niveau IP, aussi bien dans IPv4 que IPv6, peut chiffrer l'en-tête de transport entier, cachant les numéros d'accès des paquets de données aux routeurs intermédiaires. Une petite extension à RSVP pour la sécurité IP dans IPv4 et IPv6 est décrite séparément dans la [RFC2207].

Les messages RSVP qui portent des demandes de réservation prennent leur origine chez les receveurs et sont passés vers l'amont en direction de ou des envoyeurs. Noter que dans ce document, on définit les termes directionnels "amont"/"aval", "bond précédent"/"prochain bond", et "interface entrante"/"interface sortante" par rapport à la direction du flux des données.

À chaque nœud intermédiaire, une demande de réservation déclanche deux actions générales, comme suit :

1. Faire une réservation sur une liaison
   Le processus RSVP passe la requête au contrôle d'admission et au contrôle de politique. Si l'un ou l'autre essai échoue, la réservation est rejetée et le processus RSVP retourne un message d'erreur au ou aux receveurs appropriés. Si tous deux réussissent, le nœud règle le classeur de paquet pour choisir les paquets de données définis par la spec de filtre, et il interagit avec la couche de liaison appropriée pour obtenir la QS désirée définie par la flowspec.

   Le détail des règles pour satisfaire une demande de QS RSVP dépend de la technologie particulière de couche de liaison utilisée sur chaque interface. Les spécifications sont en cours de développement dans le groupe de travail ISSLL pour transposer les demandes de réservation dans les technologies de couche de liaison courantes. Par exemple, pour une simple liaison louée, la QS désirée sera obtenue du programmeur de paquet dans le pilote de couche de liaison. Si la technologie de couche de liaison met en œuvre sa propre capacité de gestion de la qualité de service, RSVP doit alors négocier avec la couche de liaison pour obtenir la QS demandée. Noter que l'action de contrôle de la QS survient à

l'endroit où les données entrent sur le support de la couche de liaison, c'est-à-dire, à l'extrémité amont de la liaison logique ou physique, bien qu'une demande de réservation RSVP ait pour origine le ou les receveurs vers l'aval.

2. Transmettre la requête vers l'amont
   Une demande de réservation se propage vers l'amont en direction des envoyeurs appropriés. L'ensemble des hôtes envoyeurs à qui une demande de réservation donnée est diffusée est appelée la "portée" de cette demande.

   La demande de réservation qu'un nœud transmet vers l'amont peut différer de la demande qu'il a reçu de l'aval pour deux raisons. Le mécanisme de contrôle de trafic peut modifier la flowspec bond par bond. Plus important, les réservations provenant de différentes branches aval de la ou des arborescences de diffusion groupée du même envoyeur (ou ensemble d'envoyeurs) doivent être "fusionnées" lorsque les réservations voyagent vers l'amont.

Lorsque un receveur génère une demande de réservation, il peut aussi demander un message de confirmation pour indiquer que sa demande était (probablement) installée dans le réseau. Une demande de réservation réussie se propage vers l'amont le long de l'arbre de diffusion groupée jusqu'à ce qu'elle atteigne un point où une réservation existante est égale ou supérieure à celle qui est demandée. À ce point, la demande arrivante est fusionnées avec la réservation en place et n'a pas besoin d'être transmise plus loin ; le nœud peut alors renvoyer un message de confirmation de réservation au receveur. Noter que la réception d'une confirmation est seulement une indication de forte probabilité, et non une garantie, que le service demandé est en place tout le long du chemin jusqu'à l'envoyeur, comme expliqué au paragraphe 2.6.

Le modèle de réservation RSVP de base est à "un passage" : un receveur envoie une demande de réservation vers l'amont, et chaque nœud sur le chemin accepte ou rejette la demande. Ce schéma ne donne pas de moyen facile pour qu'un receveur trouve le service de bout en bout résultant. Donc, RSVP prend en charge une amélioration au service à un passage appelé "un passage avec annonce" (OPWA, *One Pass With Advertising*) [OPWA95]. Avec OPWA, les paquets de contrôle RSVP sont envoyés vers l'aval, suivant les chemins des données, pour rassembler les informations qui peuvent être utilisées pour prédire la QS de bout en bout. Les résultats ("annonces") sont livrés par RSVP aux hôtes receveurs et peut-être aux applications receveuses. Les annonces peuvent alors être utilisées par le receveur pour construire, ou pour ajuster de façon dynamique, une demande de réservation appropriée.

1.3 Styles de réservation

Une demande de réservation comporte un ensemble d'options qui sont collectivement appelées le "style" de réservation.

Une option de réservation concerne le traitement des réservations pour différents envoyeurs au sein de la même session : elle établit une réservation "distincte" pour chaque envoyeur amont, ou autrement fait une seule réservation qui est "partagée" entre tous les paquets des envoyeurs choisis.

Une autre option de réservation contrôle la sélection des envoyeurs ; elle peut être une liste "explicite" de tous les envoyeurs choisis, ou à "caractère générique" qui sélectionne implicitement tous les envoyeurs de la session. Dans une réservation de sélection explicite d'envoyeurs, chaque spec de filtre doit correspondre exactement à un envoyeur, alors que dans une sélection d'envoyeur à caractère générique aucune spec de filtre n'est nécessaire.

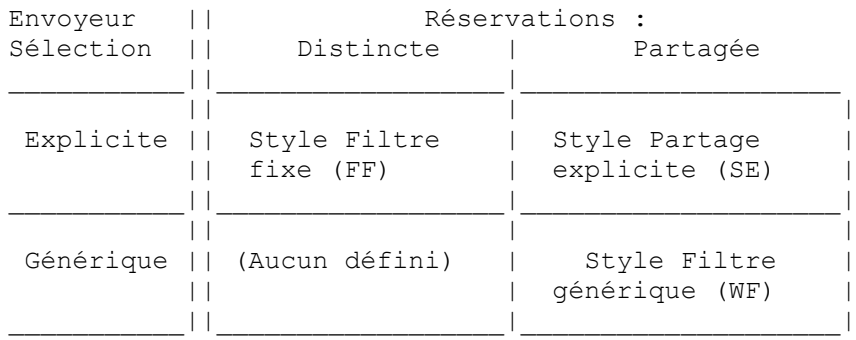


Figure 3 : Attributs et styles de réservation

Les styles suivant sont actuellement définis (voir la Figure 3):

o Style Filtre à caractère générique (WF, *Wildcard-Filter*)
  Le style WF implique les options : réservation "partagée" et sélection de l'envoyeur "à caractère générique". Donc, une réservation de style WF crée une seule réservation partagée par des flux provenant de tous les envoyeurs amont. Cette

réservation peut être vue comme un "tuyau" partagé, dont la "taille" est la plus grande des demandes de ressource de tous les receveurs, indépendamment du nombre des envoyeurs qui l'utilisent. Une réservation de style WF est propagée vers l'amont vers tous les hôtes envoyeurs, et elle s'étend automatiquement à tous les nouveaux envoyeurs à mesure qu'ils apparaissent.

Symboliquement, on peut représenter une demande de réservation de style WF par :

WF(* {Q})

où l'astérisque représente la sélection d'envoyeur à caractère générique et Q représente la flowspec.

o Style de filtre fixe (FF)
Le style FF implique les options : réservations "distinctes" et sélection "explicite" d'envoyeur. Donc, une demande de réservation élémentaire de style FF crée une réservation distincte pour les paquets de données provenant d'un envoyeur particulier, ne les partageant pas avec les paquets d'autres envoyeurs pour la même session.

Symboliquement, on peut représenter une demande de réservation élémentaire FF par :

FF(S{Q})

où S est l'envoyeur sélectionné et Q est la flowspec correspondante ; la paire forme un descripteur de flux. RSVP permet que plusieurs réservations élémentaires de style FF soient demandées en même temps, en utilisant une liste de descripteurs de flux :

FF(S1{Q1}, S2{Q2}, ...)

La réservation totale sur une liaison pour une session donnée est la somme des Q1, Q2, ... pour tous les envoyeurs demandés.

o Style partagé explicite (SE, *Shared Explicit*)
Le style SE implique les options : réservation "partagée" et la sélection "explicite" des envoyeurs. Donc, une réservation de style SE crée une seule réservation partagée par les envoyeurs amont sélectionnés. À la différence du style WF, le style SE permet à un receveur de spécifier explicitement l'ensemble des envoyeurs à inclure.

On peut représenter une demande de réservation SE contenant une flowspec Q et une liste d'envoyeurs S1, S2, ... par :

SE((S1,S2,...){Q})

Les réservations partagées, créées par les styles WF et SE, sont appropriées pour les applications de diffusion groupée dans lesquelles plusieurs sources de données ont peu de chances d'être transmises simultanément. L'audio par paquets est un exemple d'application convenable pour des réservations partagées ; comme un nombre limité de gens parlent en même temps, chaque receveur peut produire une demande de réservation WF ou SE pour deux fois la bande passante requise pour un envoyeur (pour permettre un peu de débordement). D'un autre côté, le style FF, qui crée des réservations distinctes pour les flux provenant d'envoyeurs différents, est approprié pour les signaux vidéo.

Les règles de RSVP ne permettent pas la fusion des réservations partagées avec les réservations distinctes, car ces modes sont fondamentalement incompatibles. Elles interdisent aussi de fusionner une sélection explicite d'envoyeurs avec une sélection d'envoyeur à caractère générique, car cela peut produire un service inattendu pour un receveur qui a spécifié une sélection explicite. Il résulte de cette prohibition que les styles WF, SE, et FF sont tous mutuellement incompatibles.

Il semblerait possible de simuler les effets d'une réservation WF en utilisant le style SE. Lorsque une application a demandé WF, le processus RSVP chez l'hôte receveur pourrait utiliser l'état local pour créer une réservation SF équivalente qui fasse explicitement la liste de tous les envoyeurs. Cependant, une réservation SE force le classeur de paquet dans chaque nœud à choisir explicitement chaque envoyeur dans la liste, alors que WF permet au classeur de paquet de simplement mettre un "caractère générique" pour l'adresse et l'accès de l'envoyeur. Lorsque il y a une longue liste d'envoyeurs, une réservation de style WF peut donc résulter en une charge bien moins considérable qu'une réservation équivalente de style SE. Pour cette raison, SE et WF sont tous deux inclus dans le protocole.

D'autres options et styles de réservation pourront être définis à l'avenir.

1.4 Exemples de styles

Ce paragraphe présente des exemples de chacun des styles de réservation et montre les effets de la fusion.

La Figure 4 illustre un routeur avec deux interfaces entrantes, marquées (a) et (b), à travers lesquelles vont arriver les flux, et deux interfaces sortantes, marquées (c) et (d), à travers lesquelles les données seront transmises. Cette topologie sera supposées dans les exemples qui suivent. Il y a trois envoyeurs en amont ; les paquets de l'envoyeur S1 (S2 et S3) arrivent respectivement à travers les bonds précédents (a) ((b)). Il y a aussi trois receveurs vers l'aval ; les paquets pour R1 (R2 et R3) sont acheminées via l'interface sortante (c) (respectivement, (d)). On suppose de plus que l'interface sortante (d) est connectée à un LAN de diffusion, c'est-à-dire que la réplication survient dans le réseau ; R2 et R3 sont atteints via des routeurs de prochain bond différents (non apparents sur la figure).

On doit aussi spécifier les chemins de diffusion groupée au sein du nœud de la Figure 4. On suppose d'abord que les paquets de données provenant de chaque S_i montré à la Figure 4 sont acheminées aux deux interfaces sortantes. Dans cette hypothèse, les Figures 5, 6, et 7 illustrent respectivement les réservations de filtre à caractère générique, à filtre fixe, et à partage explicite.

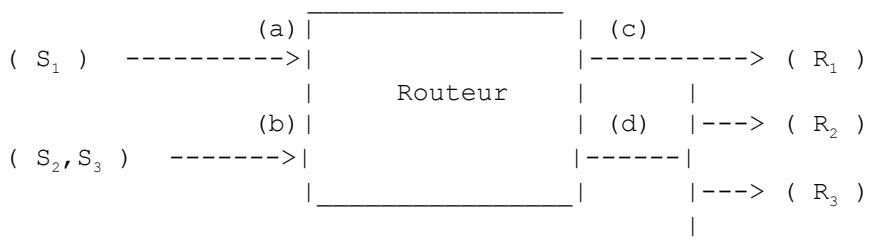


Figure 4 : Configuration de routeur

Pour simplifier, ces exemples montrent les flowspec comme des multiples unidimensionnels d'une certaine quantité B de ressource de base. La colonne "Reçoit" montre les demandes de réservation RSVP reçues sur les interfaces sortantes (c) et (d), et la colonne "Réserve" montre l'état de réservation résultant pour chaque interface. La colonne "Envoi" montre les demandes de réservation qui sont envoyées vers l'amont aux bonds précédents (a) et (b). Dans la colonne "Réserve", chaque boîte représente un "tuyau" réservé sur la liaison sortante, avec le descripteur de flux correspondant.

La Figure 5, qui montre le style WF, illustre deux situations distinctes dans lesquelles la fusion est requise. (1) Chacun des deux prochains bons sur l'interface (d) résulte en une demande de réservation RSVP séparée, comme indiqué ; ces deux demandes doivent être fusionnées en une flowspec effective, 3B, qui est utilisée pour faire la réservation sur l'interface (d). (2) Les réservations sur les interfaces (c) et (d) doivent être fusionnées afin de transmettre les demandes de réservation vers l'amont ; il en résulte que la plus grande flowspec 4B est transmise vers l'amont à chaque bond précédent.

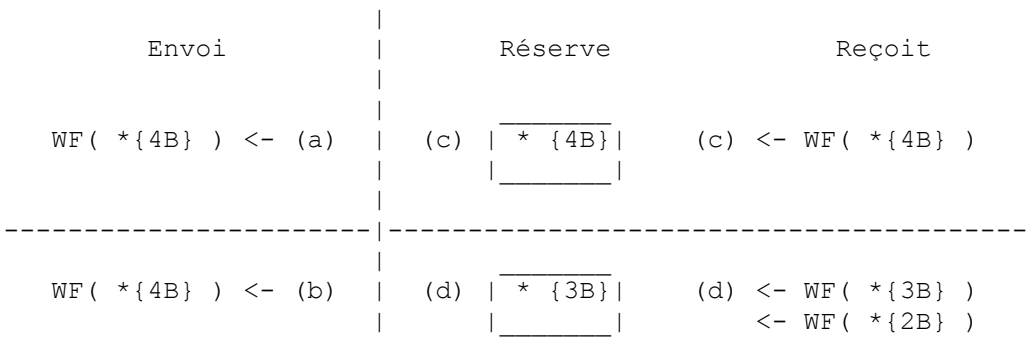


Figure 5 : Exemple de réservation à filtre à caractère générique (WF)

La Figure 6 montre des réservations de style filtre fixe (FF). Pour chaque interface sortante, il y a une réservation distincte pour chaque source demandée, mais cette réservation sera partagée entre tous les receveurs qui ont fait la demande. Les descripteurs de flux pour les envoyeurs S2 et S3, reçus à travers les interfaces sortantes (c) et (d), sont empaquetés (non fusionnés) dans la demande transmise au bond précédent (b). De l'autre côté, les trois différents descripteurs de flux qui spécifient l'envoyeur S1 sont fusionnés en une seule demande FF(S1{4B}) qui est envoyée au bond précédent (a).

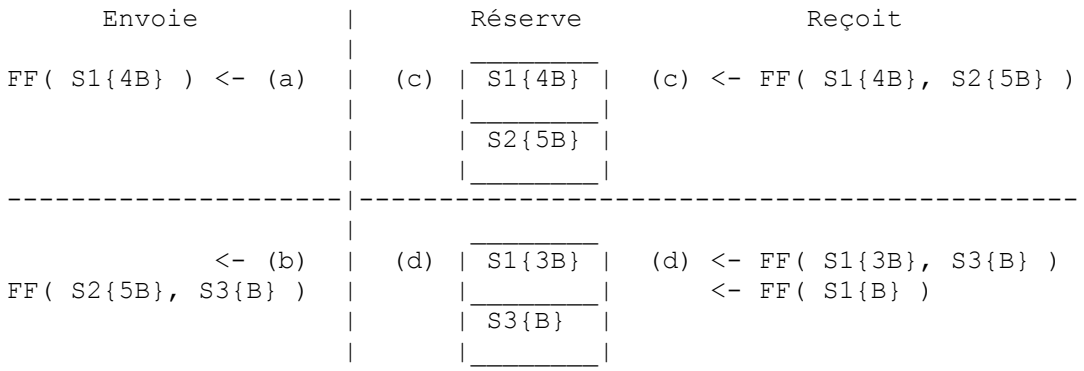


Figure 6 : Exemple de réservation à filtre fixe (FF)

La Figure 7 montre un exemple de réservations de style partage explicite (SE, *Shared-Explicit*). Lorsque les réservations de style SE sont fusionnées, la spec de filtre résultante est l'union des spec de filtre d'origine, et la flowspec résultante est la plus grande flowspec.

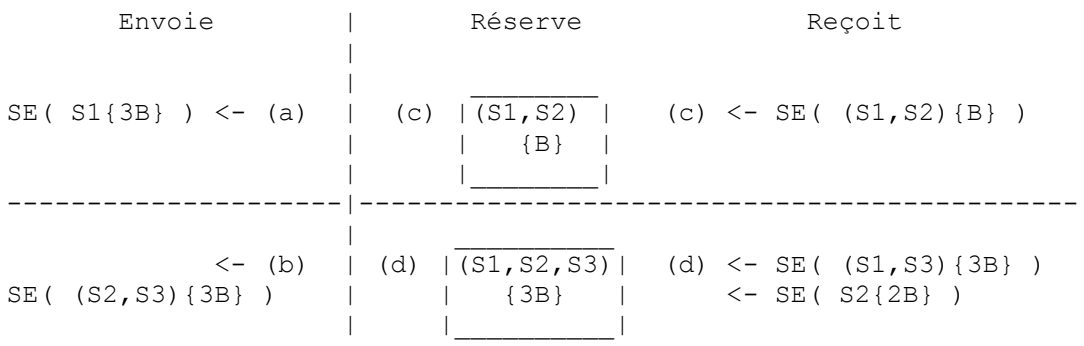
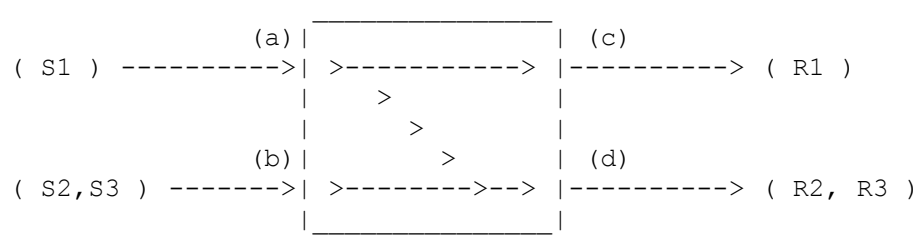


Figure 7 : Exemple de réservation à partage explicite (SE)

Les trois exemples présentés ensuite supposent que les paquets de données provenant de S1, S2, et S3 sont acheminés aux deux interfaces sortantes. La partie supérieure de la Figure 8 montre une autre hypothèse d'acheminement : les paquets de données provenant de S2 et S3 ne sont pas transmis à l'interface (c), par exemple, parce que la topologie du réseau offre un plus court chemin pour ces envoyeurs vers R1, qui ne traverse pas ce nœud. La partie inférieure de la Figure 8 montre le style WF de réservation dans cette hypothèse. Comme il n'y a pas de chemin de (b) à (c), la réservation transmise par l'interface (b) considère seulement la réservation sur l'interface (d).



Configuration de routeur

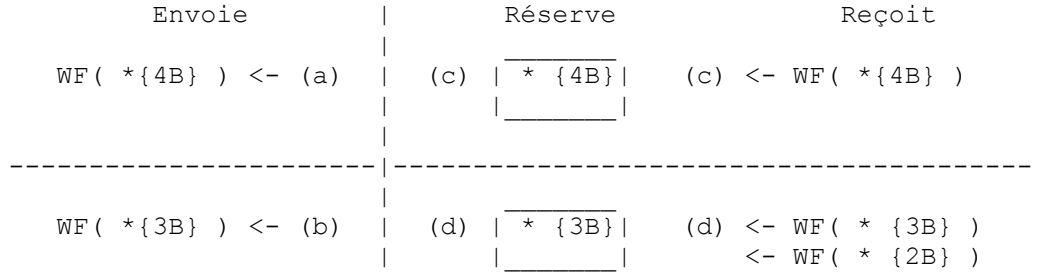


Figure 8 : Exemple de réservation WF – acheminement partiel

2. Mécanismes du protocole RSVP

2.1 Messages RSVP

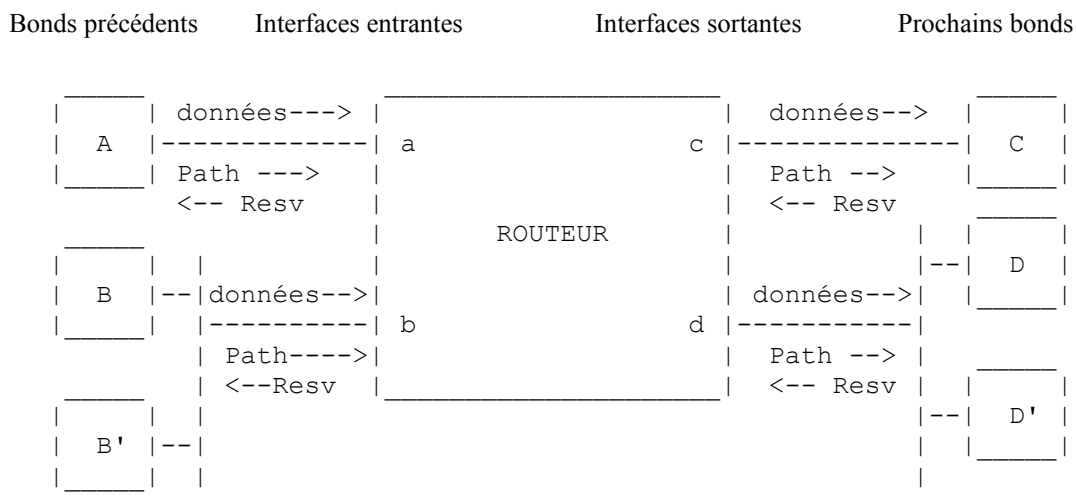


Figure 9 : Routeur utilisant RSVP

La Figure 9 illustre le modèle de nœud routeur de RSVP. Chaque flux de données arrive du "bond précédent" à travers une "interface entrante" correspondante et en part à travers une ou plusieurs "interfaces sortantes". La même interface peut agir dans les deux rôles entrant et sortant pour différents flux de données dans la même session. Plusieurs bonds précédents et/ou prochains bonds peuvent être atteints à travers une interface physique donnée ; par exemple, la figure implique que D et D' sont connectés à (d) par un LAN de diffusion.

Il y a deux types fondamentaux de message RSVP : Resv et Path.

Chaque hôte receveur envoie des messages RSVP de demande de réservation (Resv) en amont vers les envoyeurs. Ces messages doivent suivre exactement l'inverse du ou des chemins que les paquets de données vont utiliser, vers l'amont vers tous les hôtes envoyeurs inclus dans la sélection d'envoyeurs. Ils créent et maintiennent "l'état de réservation" dans chaque nœud le long du ou des chemins. Les messages Resv doivent finalement être livrés aux hôtes envoyeurs eux-mêmes, de sorte que les hôtes puissent établir les paramètres de contrôle de trafic appropriés pour le premier bond. Le traitement des messages Resv a été exposé précédemment au paragraphe 1.2.

Chaque hôte envoyeur RSVP transmet des messages "Path" RSVP vers l'aval le long des chemins d'envoi individuel/diffusion groupée fournis par le ou les protocoles d'acheminement, qui suivent les chemins des données. Ces messages Path mémorisent "l'état de chemin" dans chaque nœud le long du chemin. Cet état de chemin comporte au moins l'adresse IP d'envoi individuel du nœud du bond précédent, qui est utilisée pour acheminer les messages Resv bond par bond dans la direction inverse. (À l'avenir, certains protocoles d'acheminement pourront fournir directement les informations de chemin inverse, remplaçant la fonction d'acheminement inverse de l'état de chemin).

Un message Path contient les informations suivantes en plus de l'adresse du bond précédent :

- Gabarit d'envoyeur
  Un message Path doit porter un gabarit d'envoyeur *(Sender Template)*, qui décrit le format des paquets de données que l'envoyeur va générer. Ce gabarit est sous la forme d'une spec de filtre qui pourra être utilisée pour sélectionner les paquets de cet envoyeur parmi tous les autres dans la même session sur la même liaison.
  Les gabarits d'envoyeur ont exactement le même pouvoir d'expression et format que les spec de filtre qui apparaissent dans les messages Resv. Donc, un gabarit d'envoyeur ne peut spécifier que l'adresse IP de l'envoyeur et facultativement l'accès UDP/TCP de l'envoyeur, et il suppose l'identifiant de protocole spécifié pour la session.

- Tspec d'envoyeur
  Un message Path est obligé de porter une Tspec d'envoyeur, qui définit les caractéristiques de trafic du flux de données que l'envoyeur va générer. Cette Tspec est utilisée par le contrôle du trafic pour empêcher des sur-réservations, et peut-être des défaillances de contrôle d'admission inutiles.

- Adspec
  Un message Path peut porter un paquetage d'informations d'annonce d'OPWA, qu'on appelle une "Adspec". Une Adspec reçue dans un message Path est passée au contrôle de trafic local, qui retourne une Adspec mise à jour ; la version mise à jour est alors transmise dans les messages Path envoyés vers l'aval.

  Les messages Path sont envoyés avec les mêmes adresses de source et de destination que les données, de sorte qu'elles soient acheminées correctement à travers les nuages non RSVP (voir le paragraphe 2.9). D'un autre côté, les messages Resv sont envoyés bond par bond ; chaque nœud qui parle RSVP transmet un message Resv à l'adresse d'envoi individuel d'un bond RSVP précédent.

2.2 Fusion des flowspec

Un message Resv transmis à un bond précédent porte une flowspec qui est la "plus grande" des flowspec demandées par les prochains bonds auxquels le flux de données sera envoyé (cependant, voir au paragraphe 3.5 une règle de fusion différente utilisée dans certains cas). On dit que les flowspec ont été "fusionnées". Les exemples du paragraphe 1.4 illustraient un autre cas de fusion, lorsque il y a plusieurs demandes de réservation provenant de prochains bonds différents pour la même session et avec la même spec de filtre, mais RSVP devrait installer seulement une réservation sur cette interface. Ici encore, la réservation installée devrait avoir une flowspec effective qui est la "plus grande" des flowspec demandées par les différents prochains bonds.

Comme les flowspec sont opaques à RSVP, les règles réelles pour comparer les flowspec doivent être définies et mises en œuvre en dehors de RSVP proprement dit. Les règles de comparaison sont définies dans le document de spécification de service intégré approprié. Une mise en œuvre de RSVP aura besoin d'invoquer des programmes spécifiques des services pour effectuer la fusion des flowspec.

Noter que les flowspec sont généralement des vecteurs pluridimensionnels ; ils peuvent contenir à la fois des composantes Tspec et Rspec, chacune d'elles pouvant être elle-même pluridimensionnelle. Donc, il peut n'être pas possible d'ordonner strictement deux flowspec. Par exemple, si une requête demande une plus forte bande passante et qu'une autre réclame des limites de délai plus serrées, l'une n'est pas "plus grande" que l'autre. Dans un tel cas, au lieu de prendre la plus grande, les programmes de fusion spécifiques du service doivent être capables de retourner une troisième flowspec qui soit, mathématiquement parlant, aussi grande que chacune d'elles. C'est ce qu'on appelle la "moindre limite supérieure" (LUB, *least upper bound*). Dans certains cas, une flowspec au moins aussi petite est nécessaire ; c'est la limite inférieure supérieure (GLB, *greatest lower bound*).

Les étapes suivantes sont utilisées pour calculer la flowspec effective (Re, Te) à installer sur une interface [RFC2210]. Ici, Te est la Tspec effective et Re est la Rspec effective.

1. Une flowspec effective est déterminée pour l'interface sortante. Selon la technologie de la couche liaison, cela peut exiger de fusionner des flowspec provenant de différents prochains bonds ; cela signifie de calculer la flowspec effective comme étant la LUB des flowspec. Noter que les flowspec à fusionner sont déterminées par le support de couche liaison (voir au paragraphe 3.11.2), alors que la façon de les fusionner est déterminée par le modèle de service utilisé [RFC2210].

   Le résultat est une flowspec qui est opaque pour RSVP mais consiste en fait en une paire (Re, Resv_Te), où Re est la Rspec effective et Resv_Te est la Tspec effective.

2. On effectue un calcul spécifique du service de Path_Te, la somme de toutes les Tspecs qui ont été fournies dans les messages Path provenant des différents bonds précédents (par exemple, certains ou tous de A, B, et B' à la Figure 9).

3. (Re, Resv_Te) et Path_Te sont passés au contrôle de trafic. Le contrôle de trafic va calculer la flowspec effective comme étant le "minimum" de Path_Te et Resv_Te, d'une manière qui dépend du service.

Le paragraphe 3.11.6 définit un ensemble générique d'invocations spécifiques du service pour comparer les flowspec, pour calculer la LUB et la GLB des flowspec, et pour comparer et additionner les Tspec.

2.3 État mou

RSVP adopte une approche "d'état mou" pour gérer l'état de réservation dans les routeurs et les hôtes. L'état mou RSVP est créé et périodiquement rafraîchi par les messages Path et Resv. L'état est supprimé si aucun message de rafraîchissement correspondant n'arrive avant l'expiration d'un intervalle de "temporisation de nettoyage". L'état peut aussi être supprimé par

un message explicite "Supprimer", décrit à la section suivante. À l'expiration de chaque période de "fin de temporisation de rafraîchissement" et après un changement d'état, RSVP examine son état pour construire et transmettre des messages de rafraîchissement Path et Resv aux bonds suivants.

Les messages Path et Resv sont idempotents. Lorsque un chemin change, le message Path suivant va initialiser l'état du chemin sur la nouvelle route, et les futurs messages Resv vont y établir l'état de réservation ; l'état sur le segment maintenant non utilisé du chemin va arriver à expiration. Donc, qu'un message soit "nouveau" ou "rafraîchi" est déterminé séparément à chaque nœud, selon l'existence de l'état à ce nœud.

RSVP envoie ses messages comme des datagrammes IP sans amélioration de fiabilité. La transmission périodique des messages de rafraîchissement par les hôtes et routeurs est destinée à traiter la perte occasionnelle d'un message RSVP. Si la temporisation effective de nettoyage est réglée à K fois la période de temporisation de rafraîchissement, RSVP peut alors tolérer K-1 pertes successives de paquet RSVP sans supprimer l'état à tort. Le mécanisme de contrôle du trafic réseau devrait être configuré de façon statique pour accorder une certaine bande passante minimale aux messages Path RSVP pour les protéger contre les pertes dues à l'encombrement.

L'état entretenu par RSVP est dynamique ; pour changer l'ensemble S_i des envoyeurs ou pour changer une demande de QS quelconque, un hôte commence simplement par envoyer des messages Path et/ou Resv révisés. Le résultat sera un ajustement approprié de l'état RSVP dans tous les nœuds le long du chemin ; l'état non utilisé va arriver à expiration si il n'est pas supprimé explicitement.

En régime permanent, l'état est rafraîchi bond par bond pour permettre la fusion. Lorsque l'état reçu diffère de l'état mémorisé, celui-ci est mis à jour. Si cette mise à jour résulte en une modification de l'état à transmettre dans les messages de rafraîchissement, ces messages de rafraîchissement doivent être générés et transmis immédiatement de sorte que ces changements d'état puissent être propagés de bout en bout sans délai. Cependant, la propagation d'un changement s'arrête lorsque et si elle atteint un point où la fusion ne cause pas de changement d'état résultant. Cela minimise le trafic de contrôle RSVP dû aux changements et c'est essentiel pour l'adaptation aux grands groupes de diffusion groupée.

L'état qui est reçu par une interface particulière I* ne devrait jamais être transmis par la même interface. À l'inverse, l'état qui est transmis par l'interface I* doit être calculé en utilisant seulement l'état qui est arrivé sur des interfaces différentes de I*. Un exemple trivial de cette règle est illustré à la Figure 10, qui montre un routeur de transit avec un envoyeur et un receveur sur chaque interface (et suppose un bond prochain/précédent par interface). Les interfaces (a) et (c) servent toutes deux d'interfaces d'entrée et de sortie pour cette session. Les deux receveurs font des réservations de style générique, dans lesquelles les messages Resv sont transmis à tous les bonds précédents pour les envoyeurs dans le groupe, à l'exception du prochain bond duquel ils proviennent. Le résultat est des réservations indépendantes dans les deux directions.
Il n'y a pas d'autre règle pour gouverner la transmission des messages Resv : l'état des messages Resv reçu d'une interface sortante Io ne devrait être transmis à l'interface entrante Ii que si les messages Path provenant de Ii sont transmis à Io.

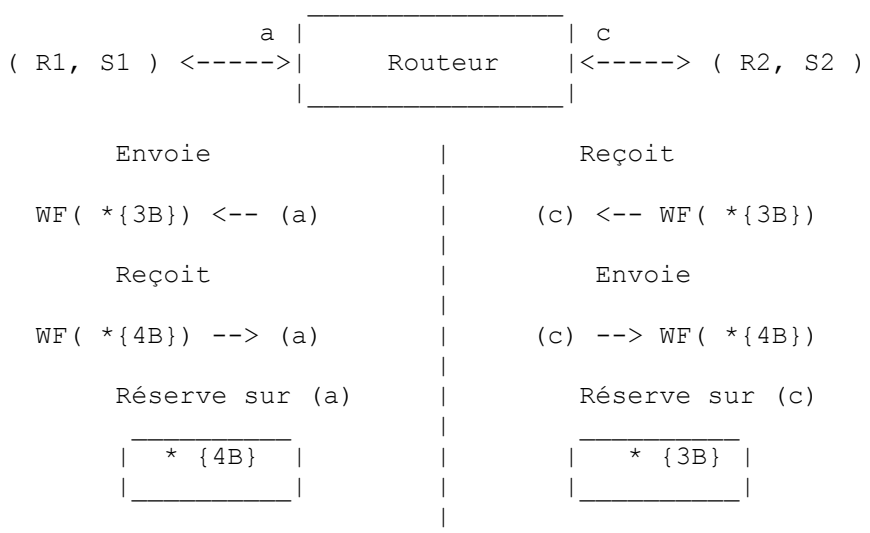


Figure 10 : Réservations indépendantes

2.4 Suppression

Les messages RSVP "Supprimer" retirent immédiatement l'état de chemin ou de réservation. Bien qu'il ne soit pas nécessaire de supprimer explicitement une vieille réservation, on recommande que tous les hôtes d'extrémité envoient une demande de suppression sitôt qu'une application se termine.

Il y a deux types de message de suppression RSVP, PathTear et ResvTear. Un message PathTear voyage vers tous les receveurs en aval de son point d'initiation et supprime l'état de chemin, ainsi que tous les états de réservation dépendants, le long du chemin. Un message ResvTear supprime l'état de réservation et voyage vers tous les envoyeurs en amont de son point d'initiation. Un message PathTear (respectivement ResvTear) peut être vu comme un message Path (respectivement Resv) en sens inverse.

Une demande Suppression peut être à l'initiative d'une application dans un système d'extrémité (envoyeur ou receveur), ou par un routeur par suite de l'expiration de l'état ou de la préemption du service. Une fois initialisée, une demande de suppression doit être transmise bond par bond sans délai. Un message Suppression supprime l'état spécifié dans les nœuds où il est reçu. Comme toujours, ce changement d'état va être propagé immédiatement au prochain nœud, mais seulement si il va y avoir un changement du réseau après la fusion. Il en résulte qu'un message ResvTear va élaguer l'état de réservation vers l'arrière (seulement) aussi loin que possible.

Comme tous les autres messages RSVP, les demandes de suppression ne sont pas livrées de façon fiable. La perte d'un message de demande de suppression ne va pas causer une défaillance de protocole parce que les données non utilisées vont finalement arriver à expiration même si elles ne sont pas explicitement supprimées. Si un message Suppression est perdu, le routeur qui n'a pas reçu ce message va mettre son état en fin de temporisation et initier un nouveau message Suppression au delà du point de perte. En supposant que la probabilité de perte d'un message RSVP est faible, le temps le plus long pour la suppression d'un état va rarement excéder une période de temporisation de rafraîchissement.

Il devrait être possible de supprimer tout sous-ensemble de l'état établi. Pour l'état de chemin, la granularité pour la suppression est d'un seul envoyeur. Pour l'état réservation, la granularité est une spec de filtre individuelle. Par exemple, se référer à la Figure 7. Le receveur R_1 pourrait envoyer un message ResvTear au seul envoyeur S_2 (ou pour tout sous-ensemble de la liste de spec de filtre) laissant S_1 en place.

Un message ResvTear spécifie le style et les filtres ; toute flowspec est ignorée. Toute flowspec en place sera retirée si toutes ses spec de filtre sont supprimées.

2.5 Erreurs

Il y a deux messages d'erreur RSVP, ResvErr et PathErr. Les messages PathErr sont très simples; ils sont simplement envoyés en amont de l'envoyeur qui a créé l'erreur, et ils ne changent pas l'état du chemin dans les nœuds à travers lesquels ils passent. Il y a seulement quelques causes possibles d'erreur de chemin.

Cependant, il y a un certain nombre de façons pour une demande de réservation syntaxiquement valide d'échouer à un nœud quelconque le long du chemin. Un nœud peut aussi décider de préempter une réservation établie. Le traitement des messages ResvErr est assez complexe (paragraphe 3.5). Comme l'échec d'une demande peut être le résultat de la fusion d'un certain nombre de requêtes, une erreur de réservation doit être rapportée à tous les receveurs responsables. De plus, la fusion de requêtes hétérogènes crée une difficulté potentielle connue sous le nom de problème de la "réservation qui tue", dans laquelle une requête peut causer un déni de service à une autre. Il y a en fait deux problèmes de réservation qui tue.

1. Le premier problème de réservation qui tue (KR-I) survient lorsque il y a déjà en place une réservation Q_0. Si un autre receveur fait maintenant une plus grande réservation $Q_1 > Q_0$, le résultat de la fusion de Q_0 et Q_1 peut être rejeté par le contrôle d'admission dans un nœud amont. Cela ne doit pas dénier le service à Q_0.

   La solution à ce problème est simple : lorsque le contrôle d'admission échoue pour une demande de réservation, toute réservation existante reste en place.

2. Le second problème de réservation qui tue (KR-II) est l'inverse : le receveur fait une réservation Q_1 qui persiste bien que le contrôle d'admission ait échoué pour Q_1 dans un nœud. Cela ne doit pas empêcher un receveur différent d'établir maintenant une plus petite réservation Q_0 qui réussirait si elle n'était pas fusionnées avec Q_1.

   Pour résoudre ce problème, un message ResvErr établit un état supplémentaire, appelé "état de blocage", dans chaque nœud à travers lequel il passe. L'état de blocage dans un nœud modifie la procédure de fusion pour omettre de la fusion la flowspec qui pose problème (Q_1 dans l'exemple), permettant à une plus petite demande d'être transmise et établie.

L'état de réservation Q_1 est dit être "bloqué". Des règles détaillées sont présentées au paragraphe 3.5.

Une demande de réservation qui échoue au contrôle d'admission crée un état de blocage mais est laissée en place dans les nœuds en aval du point d'échec. Il a été suggéré que ces réservations en aval de la défaillance représentent des réservations "gâchées" et devraient avoir une temporisation si elles ne sont pas supprimées activement. Cependant, les réservations en aval sont laissées en place, pour les raisons suivantes :

- Il y a deux raisons possibles pour qu'un receveur persiste dans une réservation échouée : (1) il demande la disponibilité de ressources sur la totalité du chemin, ou (2) il veut obtenir la QS désirée sur autant du chemin qu'il est possible. Certainement dans le second cas, et peut-être dans le premier, le receveur voudra maintenir les réservations qu'il a faites en aval de l'échec.

- Si ces réservations vers l'aval n'étaient pas retenues, la réactivité de RSVP à certaines défaillances transitoires serait compromise. Par exemple, supposons qu'une route "saute" au profit d'une autre qui est encombrée, ainsi une réservation existante a une défaillance soudaine, puis elle revient rapidement au chemin original. L'état de blocage dans chaque routeur aval ne doit pas supprimer l'état ou empêcher son rafraîchissement immédiat.

- Si on n'a pas rafraîchi les réservations vers l'aval, elles pourraient arriver en fin de temporisation, pour être restaurées toutes les Tb secondes (où Tb est l'intervalle de temporisation de l'état de blocage). Un tel comportement intermittent pourrait être très perturbant pour les utilisateurs.

2.6 Confirmation

Pour demander une confirmation pour sa demande de réservation, un receveur Rj inclut dans le message Resv un objet demande-de-confirmation contenant l'adresse IP de Rj. À chaque point de fusion, seules les plus grandes flowspec et tout objet demande-de-confirmation qui accompagne est transmis vers l'amont. Si la demande de réservation de Rj est égale ou inférieure à la réservation en place sur un nœud, sa Resv n'est pas transmise plus loin, et si la Resv comporte un objet demande-de-confirmation, un message ResvConf est renvoyé à Rj. Si la demande de confirmation est transmise, elle l'est immédiatement, et pas plus d'une fois pour chaque demande.

Ce mécanisme de confirmation a les conséquences suivantes :

- Une nouvelle demande de réservation avec une flowspec plus grande que toutes celles en place pour une session va normalement résulter en un message ResvErr ou ResvConf en retour au receveur de la part de chaque envoyeur. Dans ce cas, le message ResvConf sera une confirmation de bout en bout.

- La réception d'une ResvConf ne donne aucune garantie. Supposons que les deux premières demandes de réservation provenant des receveurs R1 et R2 arrivent au nœud où elles sont fusionnées. R2, dont la réservation était la seconde à arriver à ce nœud, peut recevoir une ResvConf de ce nœud alors que la demande de R1 n'a pas encore été propagée tout le long du chemin à un envoyeur correspondant et peut encore échouer. Donc, R2 peut recevoir une ResvConf bien qu'il n'y ait pas de réservation de bout en bout en place ; de plus, R2 peut recevoir une ResvConf suivie d'une ResvErr.

2.7 Contrôle de politique

Les demandes de QS qui passent par RSVP permettent à un ou des utilisateurs particuliers d'obtenir un accès préférentiel aux ressources du réseau. Pour prévenir les abus, certaines formes de contraintes seront généralement exercées sur les utilisateurs qui font des réservations. Par exemple, ces contraintes peuvent prendre la forme de politiques administratives d'accès, ou elles peuvent dépendre de certaines formes de rétroactions sur l'usager comme une facturation réelle ou virtuelle du "coût" de la réservation. Dans tous les cas, une identification fiable de l'usager et une admission sélective vont généralement être nécessaires lorsque une réservation est demandée.

Le terme de "contrôle de politique" est utilisé pour les mécanismes requis pour la prise en charge des politiques d'accès et des contraintes de réservations RSVP. Lorsque une nouvelle réservation est demandée, chaque nœud doit répondre à deux questions : " Y a t-il assez de ressources disponibles pour satisfaire cette demande ?" et "L'utilisateur est il autorisé à faire cette réservation ?" Ces deux décisions sont appelées la décision de "contrôle d'admission" et la décision de "contrôle de politique", respectivement, et toutes deux doivent être favorables afin que RSVP fasse une réservation. Différents domaines administratifs de l'Internet peuvent avoir des politiques de réservation différentes.

L'entrée de contrôle de politique est appelée "données de politique", que RSVP porte dans des objets POLICY_DATA. Les données de politique comportent des accréditifs qui identifient les utilisateurs ou classes d'utilisateurs, les numéros de compte, les limites, les quotas, etc. Comme les flowspec, les données de politique sont opaques pour RSVP, qui les passe simplement au contrôle de politique en tant que de besoin. De même, la fusion de données de politique doit être faite par le

mécanisme de contrôle de politique plutôt que par RSVP lui-même. Noter que les points de fusion pour les données de politique seront vraisemblablement aux frontières des domaines administratifs. Il peut donc être nécessaire de porter vers l'amont des données accumulées et non fusionnées de données de politique à travers plusieurs nœuds avant d'atteindre un de ces points de fusion.

Le portage des données de politique fournies par l'utilisateur dans les messages Resv présente un problème potentiel d'échelle. Lorsque un groupe de diffusion groupée a un grand nombre de receveurs, il va être impossible ou indésirable de porter vers l'amont toutes les données de politique des receveurs. Les données de politique devront être fusionnées administrativement à un endroit proche des receveurs, pour éviter un excès de données de politique. Un discussion plus avancée de ces questions et un exemple de schéma de contrôle de politique se trouve dans [PolArch96]. Les spécifications du format des objets de données de politique et les règles de traitement RSVP pour elles sont en cours de développement.

2.8 Sécurité

RSVP soulève les questions de sécurité suivantes.

- Intégrité du message et authentification du nœud
  Les demandes de réservation corrompues ou falsifiées pourraient conduire au vol de service par des parties non autorisées ou à des dénis de service causés par le blocage de ressources du réseau. RSVP protège contre de telles attaques avec un mécanisme d'authentification bond par bond qui utilise une fonction de hachage chiffré. Le mécanisme est pris en charge par des objets INTEGRITY qui peuvent apparaître dans tout message RSVP. Ces objets utilisent une technique de résumé chiffré à clé, qui suppose que les voisins RSVP partagent un secret. Bien que ce mécanisme fasse partie de la spécification RSVP de base, il est décrit dans un document d'accompagnement, la [RFC2747].

  Une utilisation largement répandue du mécanisme d'intégrité RSVP exigera la disponibilité de l'infrastructure de gestion et de distribution de clés pour les routeurs à laquelle on pense depuis longtemps. Jusqu'à ce que cette infrastructure devienne disponible, la gestion de clés manuelle sera nécessaire pour une intégrité du message RSVP sûre.

- Authentification de l'utilisateur
  Le contrôle de politique va dépendre de l'authentification positive de l'usager responsable de chaque demande de réservation. Les données de politique peuvent donc inclure des certificats d'usager protégés cryptographiquement. La spécification de tels certificats est une question en cours.

  Même sans certificats d'usager globalement vérifiables, il est possible de fournir une authentification d'usager pratique dans de nombreux cas en établissant une chaîne de confiance, en utilisant le mécanisme bond par bond INTEGRITY décrit plus haut.

- Flux de données sécurisés
  Les deux premières questions de sécurité concernent le fonctionnement de RSVP. Une troisième question de sécurité concerne les réservations de ressource pour des flux de données sécurisés. En particulier, l'utilisation de IPsec (Sécurité IP) dans le flux des données pose un problème pour RSVP : si les en-têtes de transport et de couche supérieure sont chiffrés, les numéros d'accès généralisés de RSVP ne peuvent pas être utilisés pour définir une session ou un envoyeur.

  Pour résoudre ce problème, une extension RSVP a été définie dans laquelle l'identifiant d'association de sécurité (IPsec SPI) joue un rôle en gros équivalent aux accès généralisés [RFC2207].

2.9 Nuages non RSVP

Il est impossible de déployer RSVP (ou tout nouveau protocole) au même moment dans tout l'Internet. De plus, RSVP peut très bien ne pas être déployé partout. RSVP doit donc fournir un fonctionnement de protocole correct même lorsque deux routeurs à capacité RSVP sont joints par un "nuage" arbitraire de routeurs non RSVP. Bien sûr, un nuage intermédiaire qui ne prend pas en charge RSVP est incapable d'effectuer une réservation de ressource. Cependant, si un tel nuage a des capacités suffisantes, il peut quand même fournir un service en temps réel utile.

RSVP est conçu pour fonctionner correctement à travers un tel nuage non RSVP. Les routeurs RSVP et non RSVP transmettent tous deux les messages Path vers l'adresse de destination en utilisant leur tableau local d'acheminement en envoi individuel/en diffusion groupée. Donc, l'acheminement des messages Path ne sera pas affecté par la présence de routeurs non RSVP dans le chemin. Lorsque un message Path traverse un nuage non RSVP, il porte au prochain nœud à capacité RSVP l'adresse IP du dernier routeur à capacité RSVP avant l'entrée dans le nuage. Un message Resv est alors transmis directement au prochain routeur à capacité RSVP sur le ou les chemins de retour vers la source.

Bien que RSVP fonctionne correctement à travers un nuage non RSVP, les nœuds sans capacité RSVP vont en général

perturber la QS fournie à un receveur. Donc, RSVP passe un bit de fanion "NonRSVP" au mécanisme de contrôle local lorsque il y a des bonds sans capacité RSVP dans le chemin vers un envoyeur donné. Le contrôle du trafic combine ce bit de fanion à ses propres sources d'information, et transmet les informations composées sur la capacité de service intégré le long du chemin vers les receveurs en utilisant les Adspec de la [RFC2210].

Certaines topologies de routeurs RSVP et non RSVP peuvent être cause que les messages Resv arrivent au mauvais nœud à capacité RSVP, ou arrivent à la mauvaise interface sur le nœud correct. Un processus RSVP doit être prêt à traiter l'une et l'autre situation. Si l'adresse de destination ne correspond à aucune interface locale et si le message n'est pas Path ou PathTear, le message doit être transmis sans autre traitement par ce nœud. Pour traiter le cas de la mauvaise interface, un "traitement d'interface logique" (LIH, *Logical Interface Handle*) est utilisé. Les informations de bond précédent incluses dans un message Path ne comportent pas seulement l'adresse IP du nœud précédent mais aussi un LIH qui définit l'interface logique sortante ; les deux valeurs sont mémorisées dans l'état de chemin. Un message Resv qui arrive au nœud visé porte à la fois l'adresse IP et le LIH de l'interface sortante correcte, c'est-à-dire, de l'interface qui devrait recevoir la réservation demandée, sans considération de l'interface sur laquelle elle arrive.

Le LIH peut aussi être utile lorsque les réservations RSVP sont faites sur une couche de liaison complexe pour faire la transposition entre entités de flux de couche IP et de couche de liaison.

2.10 Modèle d'hôte

Avant de pouvoir créer une session, l'identification de session (DestAddress, ProtocolId, [DstPort]) doit être allouée et communiquée à tous les envoyeurs et receveurs par quelque mécanisme hors bande. Lorsque une session RSVP est en cours d'établissement, les événements suivants se produisent dans les systèmes d'extrémité.

H1 Un receveur se joint au groupe de diffusion groupée spécifié par DestAddress, en utilisant IGMP.
H2 Un envoyeur potentiel commence à envoyer des messages Path RSVP à la DestAddress.
H3 Une application de receveur reçoit un message Path.
H4 Un receveur commence à envoyer les messages Resv appropriés, spécifiant les descripteurs de flux désirés.
H5 Une application d'envoyeur reçoit un message Resv.
H6 Un envoyeur commence à envoyer des paquets de données.

Il y a plusieurs considérations qui concernent la synchronisation.

- H1 et H2 peuvent survenir dans l'autre ordre
- Supposons qu'un nouvel envoyeur commence à envoyer des données (H6) mais qu'il n'y a pas de chemin de diffusion groupée parce qu'aucun receveur ne s'est joint au groupe (H1). Les données seront alors abandonnées sur quelque nœud routeur (nœud qui dépend du protocole d'acheminement) jusqu'à ce que des receveurs apparaissent
- Supposons qu'un nouvel envoyeur commence à envoyer simultanément des messages Path (H2) et des données (H6), et qu'il y ait des receveurs mais qu'aucun message Resv n'ait encore atteint l'envoyeur (par exemple, parce que ses messages Path ne se sont pas encore propagés jusqu'aux receveurs).
  Les données initiales peuvent arriver chez les receveurs sans la QS désirée. L'envoyeur pourrait atténuer ce problème en attendant l'arrivée du premier message Resv (H5) ; cependant, les receveurs qui sont plus loin peuvent ne pas encore avoir leurs réservations en place.
- Si un receveur commence à envoyer des messages Resv (H4) avant de recevoir aucun message Path (H3), RSVP va retourner des messages d'erreur au receveur.
  Le receveur peut simplement choisir d'ignorer de tels messages d'erreur, ou il peut les éviter en attendant les messages Path avant d'envoyer les messages Resv.

Une interface de programme d'application (API) spécifique pour RSVP n'est pas définie dans la présente spécification de protocole, car elle peut être hébergée en fonction du système. Cependant le paragraphe 3.11.1 discute des exigences générales et trace les lignes directrices d'une interface générique.

3. Spécification fonctionnelle de RSVP

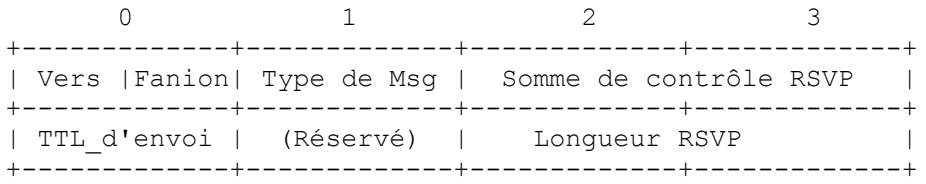
3.1 Formats de message RSVP

Un message RSVP consiste en un en-tête commun, suivi par un corps consistant en un nombre variable d'objets de longueur variable. Les paragraphes suivants définissent les formats de l'en-tête commun, de celui de l'objet standard, et de chacun des types de message RSVP.

Pour chaque type de message RSVP, il y a un ensemble de règles pour les choix permis de types d'objet. Ces règles sont

spécifiées en utilisant le format Backus-Naur augmenté (ABNF) avec des crochets qui entourent les sous séquences facultatives. L'ABNF implique un ordre des objets dans un message. Cependant, dans de nombreux cas (mais pas tous) l'ordre des objets ne fait pas de différence logique. Une mise en œuvre devrait créer des messages avec les objets dans l'ordre indiqué ici, mais accepter les objets dans tout ordre permis.

3.1.1 En-tête commun



Les champs dans l'en-tête commun sont les suivants :

Vers : 4 bits du numéro de version du protocole. Ceci est la version 1.

Fanion : 4 bits : 0x01-0x08 : Réservé ; aucun bit de fanion n'est encore défini.

Type de Msg : 8 bits
1 = Path
2 = Resv
3 = PathErr
4 = ResvErr
5 = PathTear
6 = ResvTear
7 = ResvConf

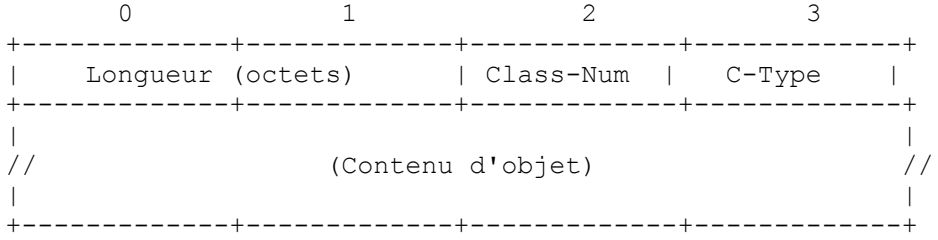
Somme de contrôle RSVP : 16 bits
Le complément à un de la somme des compléments un du message, avec le champ Somme de contrôle remplacé par zéro pour les besoins du calcul de la somme de contrôle. Une valeur toute à zéro signifie qu'aucune somme de contrôle n'a été transmise.

TTL_d'envoi : 8 bits. La valeur du TTL IP avec laquelle le message a été envoyé. Voir au paragraphe 3.8.

Longueur RSVP : 16 bits. La longueur totale de ce message RSVP en octets, y compris l'en-tête commun et les objets de longueur variable qui le suivent.

3.1.2 Formats d'objet

Chaque objet consiste en un ou plusieurs mots de 32 bits avec un en-tête d'un mot, avec le format suivant :



Un en-tête d'objet a les champs suivants :

Longueur : Un champ de 16 bits qui contient la longueur totale de l'objet en octets. Doit toujours être un multiple de 4, et au moins 4.

Class-Num : Identifie la classe de l'objet ; les valeurs de ce champ sont définies à l'Appendice A. Chaque classe d'objet a un nom, qui est toujours en majuscules dans ce document. Une mise en œuvre RSVP doit reconnaître les classes suivantes :

NULL : Un objet NULL a un Class-Num de zéro, et son C-Type est ignoré. Sa longueur doit être au moins 4, mais peut être

tout multiple de 4. Un objet NULL peut apparaître partout dans une séquence d'objets, et son contenu sera ignoré par le receveur.

SESSION : Contient l'adresse de destination IP (DestAddress), l'identifiant de protocole IP, et certaine forme d'accès de destination généralisé, pour définir une session spécifique pour les autres objets qui suivent. Exigé dans tout message RSVP.

RSVP_HOP : Porte l'adresse IP du nœud à capacité RSVP qui envoie ce message et un traitement d'interface logique (LIH, voir au paragraphe 33) sortante. Le présent document se réfère à l'objet RSVP_HOP comme un objet PHOP ("bond précédent") pour les messages vers l'aval ou comme un objet NHOP ("prochain bond") pour les messages vers l'amont.

TIME_VALUES : Contient la valeur de la période de rafraîchissement R utilisée par le créateur du message ; voir au paragraphe 3.7. Exigé dans tout message Path et Resv.

STYLE : Définit le style de réservation plus les informations spécifiques du style qui ne sont pas dans les objets FLOWSPEC ou FILTER_SPEC. Exigé dans tout message Resv.

FLOWSPEC : Définit une QS désirée, dans un message.

FILTER_SPEC : Définit un sous-ensemble de paquets de données d'une session qui devraient recevoir la QS désirée (spécifiée par un objet FLOWSPEC) dans un message Resv.

SENDER_TEMPLATE : Contient l'adresse IP d'envoyeur et peut-être des informations supplémentaires de démultiplexage pour identifier un envoyeur. Exigé dans un message Path.

SENDER_TSPEC : Définit les caractéristiques de trafic du flux de données d'un envoyeur. Exigé dans un message Path.

ADSPEC : Porte les données OPWA dans un message Path.

ERROR_SPEC : Spécifie une erreur dans un message PathErr, ResvErr, ou une confirmation dans un message ResvConf.

POLICY_DATA : Porte des informations qui vont permettre à un module de politique locale de décider si une réservation associée est administrativement permise. Peut apparaître dans un message Path, Resv, PathErr, ou ResvErr.
L'utilisation d'objets POLICY_DATA n'est pas pleinement spécifiée pour le moment ; un document va le faire.

INTEGRITY : Porte les données cryptographiques pour authentifier le nœud d'origine et pour vérifier le contenu de ce message RSVP. L'utilisation de l'objet INTEGRITY est décrite dans la [RFC2747].

SCOPE : Porte une liste explicite des hôtes envoyeurs vers lesquels sont à transmettre les informations du message. Peut apparaître dans un message Resv, ResvErr, ou ResvTear. Voir au paragraphe 3.4.

RESV_CONFIRM : Porte l'adresse IP d'un receveur qui a demandé une confirmation. Peut apparaître dans un message Resv ou ResvConf.

C-Type
Le type d'objet, unique au sein du numéro de classe. Les valeurs sont définies dans l'Appendice A.

La longueur maximum de contenu d'objet est de 65 528 octets. Les champs Class-Num et C-Type peuvent être utilisés ensemble comme nombre de 16 bits pour définir un type unique pour chaque objet.

Les deux bits de poids fort de Class-Num sont utilisés pour déterminer quelle action devrait entreprendre un nœud si il ne reconnaît pas le Class-Num d'un objet ; voir au paragraphe 3.10.

3.1.3 Messages Path

Chaque hôte envoyeur envoie périodiquement un message Path pour chaque flux de données qu'il génère. Il contient un objet SENDER_TEMPLATE qui définit le format des paquets de données et un objet SENDER_TSPEC qui spécifie les caractéristiques de trafic du flux. Facultativement, il peut contenir un objet ADSPEC qui porte les données d'annonce (OPWA) pour le flux.

Un message Path voyage d'un envoyeur à un ou des receveurs le long du ou des mêmes chemins utilisés par les paquets de données. L'adresse de source IP d'un message Path doit être une adresse de l'envoyeur qu'elle décrit, alors que l'adresse de

destination doit être la DestAddress pour la session. Ces adresses assurent que le message sera correctement acheminé à travers un nuage non RSVP.

Le format d'un message Path est le suivant :

<Message Path> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <RSVP_HOP> <TIME_VALUES>
[<POLICY_DATA> ...] [<descripteur d'envoyeur>]

<descripteur d'envoyeur> ::= <SENDER_TEMPLATE> <SENDER_TSPEC> [<ADSPEC>]

Si l'objet INTEGRITY est présent, il doit suivre immédiatement l'en-tête commun. Il n'y a pas d'autre exigence sur l'ordre de transmission, bien que l'ordre ci-dessus soit recommandé. N'importe quel nombre d'objets POLICY_DATA peut apparaître.

L'objet PHOP (c'est-à-dire, RSVP_HOP) de chaque message Path contient l'adresse du bond précédent, c'est-à-dire, l'adresse IP de l'interface à travers laquelle le message Path a le plus récemment été envoyé. Il porte aussi un traitement d'interface logique (LIH).

Chaque nœud à capacité RSVP le long du ou des chemins capture un message Path et le traite pour créer un état de chemin pour l'envoyeur défini par les objets SENDER_TEMPLATE et SESSION. Tous les objets POLICY_DATA, SENDER_TSPEC, et ADSPEC sont aussi sauvegardés dans l'état de chemin. Si une erreur se rencontre lors du traitement d'un message Path, un message PathErr est envoyé à l'envoyeur générateur du message Path. Les messages Path doivent satisfaire aux règles sur SrcPort et DstPort du paragraphe 3.2.

Périodiquement, le processus RSVP à un nœud examine l'état de chemin pour créer de nouveaux messages Path à transmettre vers le ou les receveurs. Chaque message contient un descripteur d'envoyeur qui définit un envoyeur, et porte l'adresse IP de l'envoyeur d'origine comme son adresse IP de source. Les messages atteignent finalement les applications chez tous les receveurs ; cependant, ils ne sont pas renvoyés en boucle au receveur qui fonctionne sur le même processus d'application que l'envoyeur.

Le processus RSVP transmet les messages Path et les duplique autant que de besoin pour les sessions de diffusion groupée, en utilisant les informations d'acheminement qu'il obtient du processus approprié d'acheminement en envoi individuel/diffusion groupée. Le chemin dépend de la DestAddress de la session, et pour certains protocoles d'acheminement aussi de l'adresse de source (IP de l'envoyeur). Les informations d'acheminement comportent généralement la liste de zéro, une ou plusieurs interfaces sortantes auxquelles le message Path est à transmettre. Comme chaque interface sortante a une adresse IP différente, les messages Path envoyés sur différentes interfaces contiennent des adresses PHOP différentes. De plus, les objets ADSPEC portés dans les messages Path vont aussi généralement différer pour des interfaces sortantes différentes.

L'état de chemin pour une session et envoyeur donnés n'a pas nécessairement un unique PHOP ou une interface entrante unique. Il y a deux cas, qui correspondent aux sessions de diffusion groupée et d'envoi individuel.

- Sessions en diffusion groupée
  L'acheminement de diffusion groupée permet une arborescence de distribution stable dans laquelle les messages Path provenant du même envoyeur arrivent de plus d'un PHOP, et RSVP doit être prêt à entretenir tous ces états de chemin. Les règles de RSVP pour le traitement de cette situation figurent au paragraphe 3.9. RSVP ne doit pas transmettre (conformément à ces règles) de messages Path qui arrivent sur une interface entrante différente de celle fournie par l'acheminement.

- Sessions en envoi individuel
  Pendant une courte période suivant un changement de chemin d'envoi individuel vers l'amont, un nœud peut recevoir des messages Path provenant de plusieurs PHOP pour une paire (session, envoyeur) donnée. Le nœud ne peut pas déterminer de façon fiable qui est le bon PHOP, bien que le nœud ne va recevoir de données que d'un des PHOP à la fois. Un choix de mise en œuvre pour RSVP est d'ignorer PHOP pour la mise en correspondance des états passés d'envoi individuel, et de permettre que le PHOP bascule entre les candidats. Un autre choix de mise en œuvre est de maintenir l'état de chemin pour chaque PHOP et d'envoyer des messages Resv vers l'amont vers tous ces PHOP. Dans l'un ou l'autre cas, la situation est transitoire ; l'état de chemin non utilisé va arriver en fin de temporisation ou être éliminé (parce que l'état du chemin amont va arriver en fin de temporisation).

3.1.4 Messages Resv

Les messages Resv portent des demandes de réservation bond par bond des receveurs aux envoyeurs, le long des chemins

inverses des flux de données pour la session. L'adresse de destination IP d'un message Resv est l'adresse d'envoi individuel d'un nœud de bond précédent, obtenue de l'état de chemin. L'adresse IP de source est une adresse du nœud qui a envoyé le message.

Le format du message Resv est le suivant :

<Message Resv> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <RSVP_HOP> <TIME_VALUES>
[<RESV_CONFIRM>] [<SCOPE>] [<POLICY_DATA> ...] <STYLE>
<liste de descripteur de flux>

<liste de descripteur de flux> ::= <vide> | <liste de descripteur de flux> <descripteur de flux>

Si l'objet INTEGRITY est présent, il doit suivre immédiatement l'en-tête commun. L'objet STYLE suivi par la liste de descripteur de flux doit survenir à la fin du message, et les objets au sein de la liste de descripteur de flux doivent suivre l'ABNF donné ci-dessous. Il n'y a pas d'autre exigence sur l'ordre de transmission, bien que l'ordre ci-dessus soit recommandé.

L'objet NHOP (c'est-à-dire, le RSVP_HOP) contient l'adresse IP de l'interface à travers laquelle le message Resv a été envoyé et le LIH pour l'interface logique sur laquelle la réservation est demandée.

L'apparition d'un objet RESV_CONFIRM signale une demande d'une confirmation de réservation et porte l'adresse IP du receveur auquel la ResvConf devrait être envoyée. Un nombre quelconque d'objets POLICY_DATA peut apparaître.

Le BNF ci-dessus définit une liste de descripteurs de flux tout simplement comme une liste de descripteurs de flux. Les règles selon le style qui suivent spécifient plus en détail la composition d'une liste de descripteurs de flux valides pour chacun des styles de réservation.

o Style WF :
<liste descripteur de flux> ::= <descripteur de flux WF> <descripteur de flux WF> ::= <FLOWSPEC>

o Style FF :
<liste descripteur de flux> ::= <FLOWSPEC> <FILTER_SPEC> | <liste descripteur de flux> <descripteur de flux FF>
<descripteur de flux FF> ::= [<FLOWSPEC>] <FILTER_SPEC>

Chaque demande élémentaire de style FF est définie par une seule paire (FLOWSPEC, FILTER_SPEC) et plusieurs de ces demandes peuvent être empaquetées dans la liste de descripteurs de flux d'un seul message Resv. Un objet FLOWSPEC peut être omis si il est identique au plus récent des objets qui sont apparus dans la liste ; le premier descripteur de flux FF doit contenir un objet FLOWSPEC.

o Style SE :
<liste descripteur de flux> ::= <descripteur de flux SE> <descripteur de flux SE> ::= <FLOWSPEC> <liste de spec de filtre>
<liste de spec de filtre> ::= <FILTER_SPEC> | <liste de spec de filtre> <FILTER_SPEC>

La portée de réservation, c'est-à-dire, l'ensemble des envoyeurs vers lesquels une réservation particulière est à transmettre (après fusion) est déterminée comme suit :

o Choix explicite de l'envoyeur
La réservation est transmise à tous les envoyeurs dont les objets SENDER_TEMPLATE enregistrés dans l'état de chemin correspondent à un objet FILTER_SPEC dans la réservation. Cette correspondance doit suivre les règles du paragraphe 3.2.

o Choix générique d'envoyeur
Une demande qui comporte le choix générique d'envoyeur va correspondre à tous les envoyeurs qui acheminent sur l'interface sortante concernée.

Chaque fois qu'un message Resv avec le choix générique d'envoyeur est transmis à plus d'un bond précédent, un objet SCOPE doit être inclus dans le message (voir au paragraphe 3.4) ; dans ce cas, la portée pour la transmission de la réservation est restreinte aux seules adresses IP des envoyeurs figurant explicitement sur la liste de l'objet SCOPE.

Un message Resv qui est transmis par un nœud est généralement le résultat de la fusion d'un ensemble de messages Resv entrants (qui ne sont pas bloqués ; voir au paragraphe 3.5). Si un de ces messages fusionnés contient un objet RESV_CONFIRM et si il a une FLOWSPEC supérieure aux FLOWSPEC des autres demandes de réservations fusionnées,

cet objet RESV_CONFIRM est alors transmis dans le message Resv sortant. Un objet RESV_CONFIRM dans une des autres demandes fusionnées (dont les flowspec sont égales ou inférieures, ou non comparables à la flowspec fusionnée, et qui n'est pas bloquée) va déclencher la génération d'un message ResvConf contenant la RESV_CONFIRM. Un objet RESV_CONFIRM dans une demande qui est bloquée ne sera ni transmis ni retourné ; il sera abandonné dans le nœud en cours.

3.1.5 Messages de suppression de chemin

La réception d'un message PathTear (suppression de chemin) supprime l'état de chemin correspondant. L'état correspondant doit avoir correspondu aux objets SESSION, SENDER_TEMPLATE, et PHOP. De plus, un message PathTear pour une session en diffusion groupée peut seulement correspondre à l'état de chemin pour l'interface entrante sur laquelle le PathTear est arrivé. Si il n'y a pas d'état de chemin correspondant, un message PathTear devrait être éliminé et non transmis.

Les messages PathTear sont initiés explicitement par les envoyeurs ou par l'arrivée à expiration de la temporisation de l'état de chemin sur n'importe quel nœud, et ils voyagent vers l'aval vers tous les receveurs. Un PathTear en envoi individuel ne doit pas être transmis si il y a un état de chemin pour la même paire (session, envoyeur) mais un PHOP différent. La transmission de messages PathTear en diffusion groupée est gouvernée par les règles du paragraphe 3.9.

Un message PathTear doit être acheminé exactement comme le message Path correspondant. Donc, son adresse de destination IP doit être la DestAddress de la session, et son adresse IP de source doit être l'adresse de l'envoyeur à partir de l'état de chemin en cours de suppression.

<Message PathTear> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <RSVP_HOP> [<descripteur d'envoyeur>]

<descripteur d'envoyeur> ::= (voir la définition plus haut)

Un message PathTear peut comporter un objet SENDER_TSPEC ou ADSPEC dans son descripteur d'envoyeur, mais ceux-ci doivent être ignorés. Les exigences d'ordre sont celles données précédemment pour un message Path, mais l'ordre ci-dessus est recommandé.

La suppression de l'état de chemin par suite d'un message PathTear ou d'une fin de temporisation doit aussi ajuster l'état de réservation en rapport nécessaire pour maintenir la cohérence dans le nœud local. L'ajustement dépend du style de réservation. Par exemple, supposons qu'un PathTear supprime l'état de chemin pour un envoyeur S. Si le style spécifie un choix explicite d'envoyeur (FF ou SE) toute réservation avec une spec de filtre correspondant à S devrait être supprimée ; si le style a le choix générique d'envoyeur (WF), la réservation devrait être supprimée si S est le dernier envoyeur pour la session. Ces changements de réservation ne devraient pas déclencher un message immédiat de rafraîchissement de Resv, car le message PathTear a déjà fait en amont les changements requis. Ils ne devraient pas déclencher un message ResvErr, car le résultat pourrait être de générer un déluge de tels messages.

3.1.6 Messages de suppression de réservation

La réception d'un message ResvTear (suppression de réservation) supprime l'état de réservation correspondant. L'état de réservation correspondant doit correspondre aux objets SESSION, STYLE, et FILTER_SPEC ainsi qu'au LIH dans l'objet RSVP_HOP. Si il n'y a pas d'état de réservation correspondant, le message ResvTear devrait être éliminé. Un message ResvTear peut supprimer tout sous-ensemble des spec de filtre dans l'état de réservation de style FF ou SE.

Les messages ResvTear sont initiés explicitement par les receveurs ou par tout nœud dans lequel l'état de réservation est arrivé en fin de temporisation, et ils voyagent en amont vers tous les envoyeurs qui correspondent.

Un message ResvTear doit être acheminé comme le message Resv correspondant, et son adresse de destination IP sera l'adresse d'envoi individuel d'un bond précédent.

<Message ResvTear> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <RSVP_HOP> [<SCOPE>] <STYLE>

<liste de descripteur de flux> <liste de descripteur de flux> ::= (voir la définition ci-dessus)

Les objets FLOWSPEC dans la liste de descripteurs de flux d'un message ResvTear seront ignorés et peuvent être omis. Les exigences d'ordre pour l'objet INTEGRITY, le descripteur d'envoyeur, l'objet STYLE, et la liste de descripteurs de flux sont comme celles données plus haut pour un message Resv, mais l'ordre ci-dessus est recommandé. Un message ResvTear peut inclure un objet SCOPE, mais il doit être ignoré.

Un message ResvTear va cesser d'être transmis au nœud où la fusion aurait supprimé la transmission du message Resv

correspondant. Selon le changement d'état résultant dans un nœud, la réception d'un message ResvTear peut causer la transmission d'un message ResvTear, la transmission d'un message Resv modifié, ou aucune transmission de message. Ces trois cas peuvent être illustrés dans le cas des réservations de style FF montrées à la Figure 6.

- Si le receveur R2 envoie un message ResvTear pour sa réservation S3{B}, la réservation correspondante est retirée de l'interface (d) et une ResvTear pour S3{B} est transmise par (b).

- Si le receveur R1 envoie une ResvTear pour sa réservation S1{4B}, la réservation correspondante est retirée de l'interface (c) et un message Resv FF(S1{3B}) modifié est immédiatement transmis de (a).

- Si le receveur R3 envoie un message ResvTear pour S1{B}, il n'y a pas de changement dans la réservation effective S1{3B} sur (d) et aucun message n'est transmis.

3.1.7 Messages d'erreur de chemin

Les messages PathErr (erreur de chemin) rapportent des erreurs dans le traitement des messages Path. Ils voyagent en amont vers les envoyeurs et sont acheminés bond par bond en utilisant l'état de chemin. À chaque bond, l'adresse de destination IP est l'adresse d'envoi individuel d'un bond précédent. Les messages PathErr ne modifient l'état d'aucun nœud à travers lequel ils passent ; ils sont seulement rapportés à l'application de l'envoyeur .

<Message PathErr> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <ERROR_SPEC> [<POLICY_DATA> ...] [
<descripteur d'envoyeur>]

<descripteur d'envoyeur> ::= (voir la définition précédente)

L'objet ERROR_SPEC spécifie l'erreur et comporte l'adresse IP du nœud qui a détecté l'erreur (Adresse de nœud d'erreur). Un ou plusieurs objets POLICY_DATA peuvent être inclus dans le message pour fournir les informations pertinentes. Le descripteur d'envoyeur est copié du message erroné. Les exigences d'ordre des objets sont celles données précédemment pour un message Path, mais l'ordre ci-dessus est recommandé.

3.1.8 Messages d'erreur de réservation

Les messages ResvErr (erreur de réservation) rapportent des erreurs dans le traitement des messages Resv, ou ils peuvent rapporter l'interruption spontanée d'une réservation, par exemple, par préemption administrative.

Les messages ResvErr voyagent en aval vers les receveurs appropriés, acheminés bond par bond en utilisant l'état de réservation. À chaque bond, l'adresse de destination IP est l'adresse d'envoi individuel d'un nœud du prochain bond.

<Message ResvErr> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <RSVP_HOP> <ERROR_SPEC>
[<SCOPE>] [<POLICY_DATA> ...] <STYLE> [<descripteur de flux erroné>]

L'objet ERROR_SPEC spécifie l'erreur et comporte l'adresse IP du nœud qui a détecté l'erreur (Adresse de nœud erroné). Un ou plusieurs objets POLICY_DATA peuvent être inclus dans un message d'erreur pour fournir les informations pertinentes (par exemple, quand une erreur de contrôle de politique est rapportée). L'objet RSVP_HOP contient l'adresse du bond précédent, et l'objet STYLE est copié du message Resv erroné. L'utilisation de l'objet SCOPE dans un message ResvErr est définie au paragraphe 3.4. Les exigences d'ordre des objets sont celles données pour les messages Resv, mais l'ordre ci-dessus est recommandé.

Les règles dépendantes du style suivantes définissent la composition d'un descripteur de flux erroné valide ; les exigences d'ordre des objets sont celles données plus haut pour le descripteur de flux.

- Style WF :
  <descripteur de flux erroné> ::= <descripteur de flux WF>

- Style FF :
  <descripteur de flux erroné> ::= <descripteur de flux FF>

  Chaque descripteur de flux dans un message Resv de style FF doit être traité indépendamment, et un message ResvErr distinct doit être généré pour chacun de ceux qui sont erronés.

- Style SE :
  <descripteur de flux erroné> ::= <descripteur de flux SE>

Un message ResvErr de style SE peut faire la liste du sous-ensemble des spec de filtre dans le message Resv correspondant auquel l'erreur s'applique.

Noter qu'un message ResvErr contient seulement un descripteur de flux. Donc, un message Resv qui contient N > 1 descripteurs de flux (de style FF) peut créer jusqu'à N messages ResvErr distincts.

Généralement parlant, un message ResvErr devrait être transmis vers tous les receveurs qui peuvent avoir causé le rapport de l'erreur. Plus précisément :

o Le nœud qui détecte une erreur dans une demande de réservation envoie un message ResvErr au nœud de prochain bond d'où est venue la réservation erronée.

  Ce message ResvErr doit contenir les informations requises pour définir l'erreur et pour acheminer le message d'erreur dans les bonds ultérieurs. Il inclut donc un objet ERROR_SPEC; une copie de l'objet STYLE, et le descripteur de flux erroné approprié. Si l'erreur est un échec de contrôle d'admission lors d'une tentative d'augmentation d'une réservation existante, la réservation existante doit alors rester en place et le bit fanion InPlace doit être mis à 1 dans l'objet ERROR_SPEC du message ResvErr.

o Les nœuds successifs transmettent le message ResvErr aux prochains bonds qui ont l'état de réservation local. Pour les réservations avec une portée générique, il y a une limitation supplémentaire à la transmission des messages ResvErr, pour éviter les boucles ; voir au paragraphe 3.4. Il y a aussi une règle qui restreint la transmission d'un message Resv après un échec de contrôle d'admission ; voir au paragraphe 3.5.

  Un message ResvErr qui est transmis devrait porter la ou les FILTER_SPEC de l'état de réservation correspondant.

o Lorsque un message ResvErr atteint un receveur, l'objet STYLE, la liste de descripteurs de flux, et l'objet ERROR_SPEC (y compris ses fanions) devraient être livrés à l'application receveuse.

3.1.9 Messages de confirmation

Les messages ResvConf sont envoyés (de façon probabiliste) pour accuser réception des demandes de réservation. Un message ResvConf est envoyé par suite de l'apparition d'un objet RESV_CONFIRM dans un message Resv.

Un message ResvConf est envoyé à l'adresse d'envoi individuel d'un hôte receveur ; l'adresse est obtenue de l'objet RESV_CONFIRM. Cependant, un message ResvConf est transmis au receveur bond par bond, pour s'accommoder du mécanisme de vérification d'intégrité bond par bond.

<message ResvConf> ::= <En-tête commun> [<INTEGRITY>] <SESSION> <ERROR_SPEC> <RESV_CONFIRM>
<STYLE> <liste de descripteur de flux>

<liste de descripteur de flux> ::= (voir la définition antérieure)

Les exigences d'ordre des objets sont les mêmes que celles données plus haut pour un message Resv, mais l'ordre ci-dessus est recommandé.

L'objet RESV_CONFIRM est une copie de cet objet dans le message Resv qui a déclenché la confirmation. L'objet ERROR_SPEC n'est utilisé que pour porter l'adresse IP du nœud d'origine, dans l'adresse de nœud erroné ; le code d'erreur et la valeur sont zéro pour indiquer une confirmation. La liste de descripteurs de flux spécifie les réservations particulières qui sont confirmées ; cela peut être un sous-ensemble de la liste de descripteurs de flux de la Resv qui demandait la confirmation.

3.2 Usage de l'accès

Une session RSVP est normalement définie par le triplet (DestAddress, ProtocolId, DstPort). Ici, DstPort est un champ Accès de destination UDP/TCP (c'est-à-dire, une quantité de 16 bits portée au décalage d'octet + 2 dans l'en-tête de transport). DstPort peut être omis (mis à zéro) si le ProtocolId spécifie un protocole qui n'a pas un champ Accès de destination au format utilisé par UDP et TCP.

RSVP permet toute valeur pour ProtocolId. Cependant, les mises en œuvre de système d'extrémité de RSVP peuvent avoir connaissance de certaines valeurs pour ce champ, et en particulier des valeurs pour UDP et TCP (respectivement 17 et 6,).

Un système d'extrémité peut donner une erreur à une application qui :
- o spécifie un DstPort différent de zéro pour un protocole qui n'a pas d'accès de style UDP/TCP, ou qui
- o spécifie un DstPort à zéro pour un protocole qui a des accès de style UDP/TCP.

Les spec de filtre et les gabarits d'envoyeur spécifient la paire (SrcAddress, SrcPort) où SrcPort est un champ d'accès de source UDP/TCP (c'est-à-dire, une quantité de 16 bits portée à un décalage d'octet de + 0 dans l'en-tête de transport). SrcPort peut être omis (mis à zéro) dans certains cas.

Les règles suivantes sont fixées à l'utilisation des champs DstPort et/ou SrcPort à zéro dans RSVP.

1. Les accès de destination doivent être cohérents.
   L'état de chemin et l'état de réservation pour les mêmes DestAddress et ProtocolId doivent tous deux avoir des valeurs de DstPort qui soient toutes à zéro ou toutes différentes de zéro. La violation de cette condition dans un nœud est une erreur "Accès de destination en conflit".

2. Règles des accès de destination.
   Si DstPort est zéro dans une définition de session, tous les champs SrcPort utilisés pour cette session doivent aussi être zéro. L'hypothèse ici est que le protocole n'a pas d'accès de style UDP/TCP. La violation de cette condition dans un nœud est une erreur "Mauvais accès de source".

3. Les accès de source doivent être cohérents.
   Un hôte envoyeur ne doit pas envoyer d'état de chemin à la fois avec et sans une SrcPort à zéro. La violation de cette condition est une erreur "Accès d'envoyeur en conflit".

Noter que RSVP n'a pas d'accès "générique", c'est-à-dire qu'un accès zéro ne peut jamais correspondre à un accès non zéro.

3.3 Envoi des messages RSVP

Les messages RSVP sont envoyés bond par bond entre routeurs à capacité RSVP comme datagrammes IP "bruts" avec le numéro de protocole 46. Les datagrammes IP bruts sont aussi destinées à être utilisés entre un système d'extrémité et le routeur de premier/dernier bond, bien qu'il soit aussi possible d'encapsuler les messages RSVP comme datagrammes UDP pour les communications de système d'extrémité, comme décrit à l'Appendice C. L'encapsulation UDP est nécessaire pour les systèmes qui ne peuvent pas faire d'entrée/sortie de réseau brute.

Les messages Path, PathTear, et ResvConf doivent être envoyés avec l'option IP Alerte de routeur [RFC2113] dans leurs en-têtes IP. Cette option peut être utilisée dans le chemin de transmission rapide d'un routeur à grande vitesse pour détecter les datagrammes qui requièrent un traitement particulier.

À l'arrivée d'un message RSVP M qui change l'état, un nœud doit transmettre immédiatement la modification d'état. Cependant, cela ne doit pas déclencher l'envoi d'un message par l'interface à travers laquelle M est arrivé (ce qui pourrait arriver si la mise en œuvre déclenche simplement un rafraîchissement immédiat de tout état pour la session). Cette règle est nécessaire pour empêcher les tempêtes de paquets sur les LAN de diffusion.

Dans la présente version de la spécification, chaque message RSVP doit occuper exactement un datagramme IP. Si il excède la MTU, un tel datagramme sera fragmenté par IP et réassemblé au nœud receveur. Cela a plusieurs conséquences :

- o Un seul message RSVP ne peut pas excéder la taille maximum de datagramme IP, d'approximativement 64 k octets.

- o Un nuage non RSVP encombré pourrait perdre des fragments de message individuels, et tout fragment perdu perd le message entier.

Des versions futures du protocole fourniront des solutions à ces problèmes si ils se révélaient un fardeau trop lourd à porter. La direction la plus vraisemblable sera d'effectuer une "fragmentation sémantique", c'est-à-dire, de couper le chemin ou l'état de réservation à transmettre en plusieurs messages d'un seul datagramme, chacun d'une taille acceptable.

RSVP utilise ses mécanismes de rafraîchissement périodique pour récupérer de pertes de paquet occasionnelles. Cependant, en cas de surcharge du réseau, des pertes substantielles de messages RSVP pourraient causer une défaillance de la réservation de ressource. Pour contrôler le délai de mise en file d'attente et l'abandon des paquets RSVP, les routeurs devraient être configurés de façon à leur offrir une classe de service préférentielle. Si les paquets RSVP subissent des pertes notables lors de la traversée d'un nuage non RSVP encombré, une valeur plus élevée peut être utilisée pour le facteur K de temporisation (voir au paragraphe 3.7).

Certains protocoles d'acheminement de diffusion groupée fournissent des "tunnels de diffusion groupée", qui font l'encapsulation IP des paquets de diffusion groupée à transmettre à travers des routeurs qui n'ont pas de capacité de diffusion groupée. Un tunnel de diffusion groupée ressemble à une interface logique sortante qui est transposée en une interface physique. Un protocole d'acheminement de diffusion groupée qui prend en charge les tunnels décrira un chemin utilisant une liste d'interfaces logiques plutôt que physiques. RSVP peut fonctionner à travers de tels tunnels de diffusion groupée de la façon suivante :

1. Lorsque un nœud N transmet un message Path sur une interface logique sortante L, il inclut dans le message un codage de l'identité de L, appelé le "traitement d'interface logique" (LIH). Les valeurs de LIH sont portées dans l'objet RSVP_HOP.

2. Le nœud N' de prochain bond mémorise la valeur de LIH dans son état de chemin.

3. Lorsque N' envoie un message Resv à N, il inclut la valeur de LIH tirée de l'état de chemin (aussi dans l'objet RSVP_HOP).

4. Lorsque le message Resv arrive à N, sa valeur de LIH donne les informations nécessaires pour rattacher la réservation à l'interface logique appropriée. Noter que N crée et interprète le LIH ; c'est une valeur opaque pour N'.

Noter que ceci ne résout que le problème d'acheminement posé par les tunnels. Le tunnel apparaît à RSVP comme un nuage non RSVP. Pour établir des réservations RSVP au sein du tunnel, une machinerie supplémentaire sera nécessaire, qui sera définie plus tard.

3.4 Éviter les messages RSVP en boucle

La transmission des messages RSVP doit éviter les boucles. En état permanent, les messages Path et Resv sont transmis une seule fois sur chaque bond par période de rafraîchissement. Cela évite les paquets en boucle, mais il y a toujours la possibilité d'une boucle "d'auto rafraîchissement", rythmée par la période de rafraîchissement. De telles boucles d'auto rafraîchissement gardent l'état actif "pour toujours", même si les nœuds d'extrémité ont cessé de le rafraîchir, jusqu'à ce que les receveurs quittent le groupe de diffusion groupée et/ou que les envoyeurs arrêtent d'envoyer des messages Path. D'un autre côté, les messages d'erreur et de suppression sont transmis immédiatement et sont donc sujets à des mises en boucle directes.

Considérons chaque type de message.

- Messages Path
  Les messages Path sont transmis exactement de la même façon que les paquets de données IP. Donc, il ne devrait pas y avoir de boucle de messages Path (excepté peut-être des boucles d'acheminement transitoires, qui sont ignorées ici), même dans une topologie avec des cycles.

- Messages PathTear
  Les messages PathTear utilisent le même acheminement que les messages Path et ne peuvent donc pas être en boucle.

- Messages PathErr
  Comme les messages Path ne se mettent pas en boucle, ils créent un état de chemin qui définit une chemin inverse sans boucle sur chaque envoyeur. Les messages PathErr sont toujours dirigés sur des envoyeurs particuliers et ne peuvent donc pas être en boucle.
- Messages Resv
  Les messages Resv dirigés sur des envoyeurs particuliers (c'est-à-dire, avec sélection explicite de l'envoyeur) ne peuvent pas être en boucle. Cependant, les messages Resv avec un choix générique de l'envoyeur (style WF) ont un potentiel de boucle d'auto rafraîchissement.

- Messages ResvTear
  Bien que les messages ResvTear soient acheminés de la même façon que les messages Resv, durant le second passage dans une boucle, il n'y aura pas d'état, de sorte que tout message ResvTear sera abandonné. Et donc il n'y a pas là de problème de boucle.

- Messages ResvErr
  Les messages ResvErr pour les réservations de style WF peuvent se mettre en boucle essentiellement pour les mêmes raisons que les messages Resv.

o Messages ResvConf
  Les messages ResvConf sont transmis vers une adresse fixe de receveur en envoi individuel et ne peuvent pas se mettre en boucle.

Si la topologie n'a pas de boucle, la mise en boucle des messages Resv et ResvErr avec la sélection générique d'envoyeur peut être évitée en mettant simplement en application la règle donnée plus haut : l'état qui est reçu à travers une interface particulière ne doit jamais être transmis par la même interface. Cependant, lorsque la topologie a des cycles, d'autres efforts sont nécessaires pour empêcher les boucles d'auto rafraîchissement de messages Resv génériques et des boucles rapides de messages ResvErr génériques. La solution à ce problème adoptée par cette spécification est pour de tels messages de porter une liste explicite d'adresses d'envoyeurs dans un objet SCOPE.

Lorsque un message Resv de style WF est à envoyer à un bond précédent particulier, un nouvel objet SCOPE est calculé à partir des objets SCOPE qui ont été reçus dans les messages Resv correspondants. Si l'objet SCOPE calculé est vide, le message n'est pas transmis au bond précédent ; autrement, le message est envoyé, contenant le nouvel objet SCOPE. Les règles de calcul d'un nouvel objet SCOPE pour un message Resv sont les suivantes :

1. L'union est formée des ensembles des adresses IP d'envoyeur figurant dans tous les objets SCOPE dans l'état de réservation pour la session en question.

   Si des états de réservation de certains NHOP ne contiennent pas d'objet SCOPE, une liste d'envoyeurs de substitution doit être créée et incluse dans l'union. Pour un message qui est arrivé sur l'interface sortante OI, la liste de substitution est l'ensemble des envoyeurs qui acheminent vers OI.

2. Tous les envoyeurs locaux (c'est-à-dire, toutes les applications d'envoyeur sur ce nœud) sont retirés de cet ensemble.

3. Si l'objet SCOPE est à envoyer à PHOP, retirer de l'ensemble tous les envoyeurs qui ne venaient pas de PHOP.

La Figure 11 donne un exemple de message Resv à portée générique (style WF). Les listes d'adresses au sein des objets SCOPE sont montrées entre crochets. Noter qu'il peut y avoir des connexions supplémentaires parmi les nœuds, créant une topologie de boucle qui n'est pas montrée.

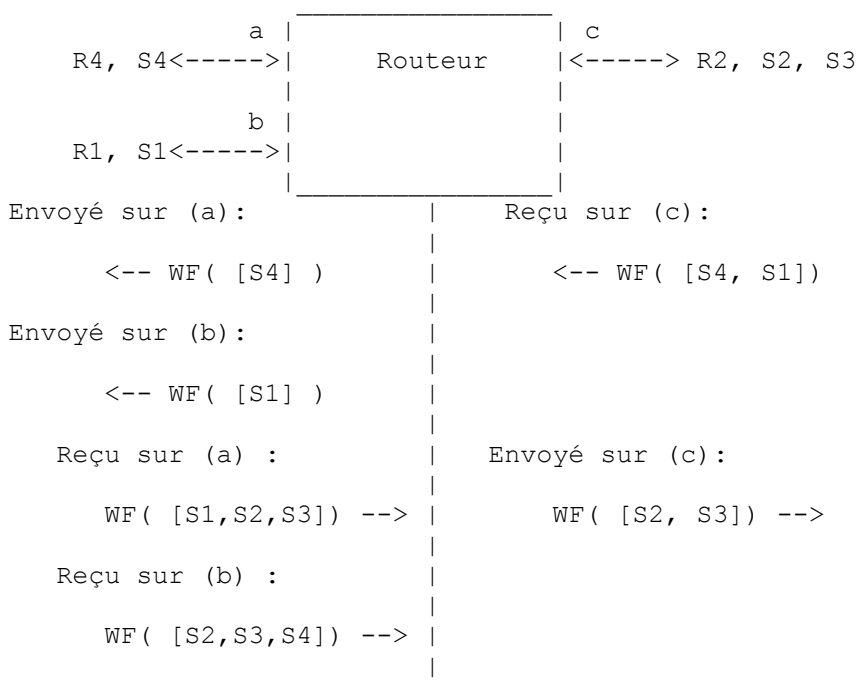


Figure 11 : Objets SCOPE dans des réservations à portée générique

Les objets SCOPE ne sont pas nécessaires si l'acheminement en diffusion groupée utilise des arborescences partagées ou si le style de réservation a la sélection explicite d'envoyeur. De plus, rattacher un objet SCOPE à une réservation devrait être réservé à un nœud qui a plus d'un bond précédent pour l'état de réservation.

Les règles suivantes sont utilisées pour les objets SCOPE dans les messages ResvErr de style WF :

1. Le nœud qui a détecté l'erreur initie un message ResvErr contenant une copie de l'objet SCOPE associé à l'état de réservation ou au message erroné.

2. Supposons qu'un message ResvErr de style générique arrive à un nœud avec un objet SCOPE contenant la liste d'adresse L d'hôtes envoyeurs. Le nœud transmet le message ResvErr en utilisant les règles du paragraphe 3.1.8. Cependant, le message ResvErr transmis de OI doit contenir un objet SCOPE déduit de L en incluant seulement les envoyeurs qui acheminent sur OI. Si cet objet SCOPE est vide, le message ResvErr ne devrait pas être envoyé de OI.

3.5 État de blocage

La règle de base pour créer un message Resv de rafraîchissement est de fusionner les flowspec de demandes de réservation en place dans le nœud, en calculant leur LUB. Cependant, cette règle est modifiée par l'existence de "l'état de blocage" résultant des messages ResvErr, pour résoudre le problème KR-II (voir au paragraphe 2.5). L'état de blocage entre aussi dans l'acheminement des messages ResvErr pour les défaillances de contrôle d'admission.

Lorsque un message ResvErr est reçu pour une défaillance de contrôle d'admission, sa flowspec Qe est utilisée pour créer ou rafraîchir un élément de l'état de blocage local. Chaque élément d'état de blocage consiste en un blocage de la flowspec Qb tirée de la flowspec du message ResvErr, et en un temporisateur de blocage associé Tb. Lorsque un temporisateur de blocage arrive à expiration, l'état de blocage correspondant est supprimé.

La granularité de l'état de blocage dépend du style du message ResvErr qui l'a créé. Pour un style explicite, il peut y avoir un élément d'état de blocage (Qb(S),Tb(S)) pour chaque envoyeur S. Pour un style générique, l'état de blocage est selon le bond précédent P.

Un élément d'état de blocage avec la flowspec Qb est dit "bloquer" une réservation avec la flowspec Qi si Qb n'est pas (strictement) supérieur à Qi. Par exemple, supposons que le LUB de deux flowspec soit calculé en prenant le maximum de chacun de leurs composants correspondants. Qb bloque alors Qi si pour un composant j, $Qb[j] \leq Qi[j]$.

Supposons qu'un nœud reçoive un message ResvErr d'un bond précédent P (ou, si le style est explicite, de l'envoyeur S) par suite d'une défaillance du contrôle d'admission en amont. Alors :

1. Un élément d'état de blocage est créé pour P (ou S) si il n'existe pas.

2. Qb(P) (ou Qb(S)) est réglé égal à la flowspec Qe du message ResvErr.

3. Un temporisateur de blocage correspondant Tb(P) (ou Tb(S)) est lancé ou relancé pour une durée Kb*R. Ici Kb est un multiplicateur fixe et R est l'intervalle de rafraîchissement pour l'état de réservation. Kb devrait être configurable.

4. Si il y a un état de réservation local qui n'est pas bloqué (voir ci-dessous) un rafraîchissement immédiat de réservation est généré pour P (ou S).

5. Le message ResvErr est transmis aux prochains bonds de la façon suivante. Si le bit InPlace est à zéro, le message ResvErr est transmis à tous les prochains bonds pour lesquels il y a un état de réservation. Si le bit InPlace est à un, le message ResvErr est transmis seulement aux prochains bonds dont Qi est bloqué par Qb.

Finalement, on présente la règle modifiée pour la fusion des flowspec pour créer un message de rafraîchissement de réservation.

- Si il y a des demandes de réservation locales Qi qui ne sont pas bloquées, elles sont fusionnées en calculant leur LUB. Les réservations bloquées sont ignorées ; cela permet la transmission d'une plus petite réservation qui n'a pas échoué et va peut-être réussir, après l'échec d'une plus grosse réservation.

- Autrement (toutes les demandes locales Qi sont bloquées) elles sont fusionnées en prenant le GLB (limite inférieure supérieure) des Qi.

(L'utilisation d'une définition de "minimum" améliore les performances en mettant entre guillemets le niveau d'échec entre la plus grande qui réussit et la plus petite qui échoue. En particulier le choix de la GLB a été fait parce qu'il est simple à définir et mettre en œuvre, et on ne voit aucune raison d'utiliser une définition différente de "minimum" ici).

Cet algorithme de fusion des rafraîchissements est appliqué séparément à chaque flux (à chaque envoyeur ou PHOP) qui

contribue à une réservation partagée (style WF ou SE).

La Figure 12 montre un exemple de l'application de l'état de blocage pour une réservation partagée (style WF). Il y a deux bonds précédents marqués (a) et (b), et deux prochains bonds marqués (c) et (d). La plus grande réservation 4B est arrivée de (c) en premier, mais elle a échoué quelque part en amont via PHOP (a), mais pas via PHOP (b). La figure montre "l'état permanent" final après l'arrivée de la plus petite réservation 2B en provenance de (d). L'état permanent est perturbé en gros toutes les Kb*R secondes, lorsque l'état de blocage arrive en fin de temporisation. Le prochain rafraîchissement envoie alors 4B au bond précédent (a) ; cela va vraisemblablement échouer, en envoyant un message ResvErr qui va rétablir l'état de blocage, retournant à la situation montrée sur la figure. Au même moment, le message ResvErr va être transmis au prochain bond (c) et à tous les receveurs en aval responsables des réservations de 4B.

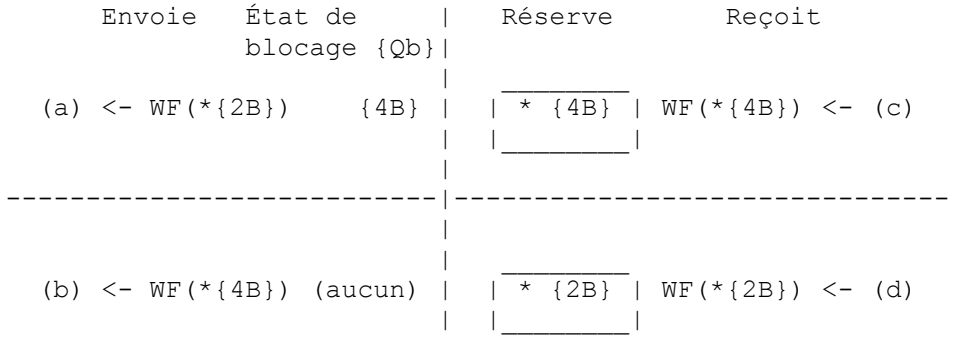


Figure 12 : Blocage avec style partagé

3.6 Réparation locale

Lorsque un chemin change, le prochain message de rafraîchissement de Path ou Resv va établir (respectivement) un chemin ou un état de réservation le long du nouveau chemin. Pour fournir une adaptation rapide aux changements d'acheminement sans la redondance des périodes de rafraîchissement courtes, le module local de protocole d'acheminement peut notifier au processus RSVP des changements de chemin pour des destinations particulières. Le processus RSVP devrait utiliser ces informations pour déclencher un rafraîchissement rapide de l'état pour ces destinations, en utilisant la nouvelle route.

Les règles spécifiques sont les suivantes :

o Lorsque l'acheminement détecte un changement de l'ensemble des interfaces sortantes pour la destination G, RSVP devrait mettre à jour l'état de chemin, attendre une courte période W, puis envoyer des rafraîchissements de Path pour toutes les sessions G/* (c'est-à-dire, pour toute session avec la destination G, sans considération de l'accès de destination).

  La courte période d'attente avant l'envoi des rafraîchissements de Path est destinée à permettre au protocole d'acheminement de s'établir, et la valeur pour W devrait être choisie en conséquence. Actuellement W = 2 s est suggéré, cependant, cette valeur devrait être configurable interface par interface.

o Lorsque un message Path arrive avec une adresse de bond précédent qui diffère de celle mémorisée dans l'état de chemin, RSVP devrait envoyer des rafraîchissements de Resv immédiats à cette PHOP.

3.7 Paramètres horaires

Il y a deux paramètres horaires pertinents pour chaque élément de chemin RSVP ou d'état de réservation dans un nœud : la période de rafraîchissement R entre la génération de rafraîchissements successifs pour l'état par le nœud voisin, et la durée de vie de l'état local L. Chaque message Path ou Resv RSVP peut contenir un objet TIME_VALUES qui spécifie la valeur R qui a été utilisée pour générer ce message (de rafraîchissement). Cette valeur R est alors utilisée pour déterminer la valeur de L lorsque l'état est reçu et mémorisé. Les valeurs pour R et L peuvent varier d'un bond à l'autre.

Plus précisément :

1. Floyd et Jacobson [FJ94] ont montré que les messages périodiques générés par des nœuds du réseau indépendants peuvent devenir synchronisés. Cela peut conduire à des interruptions dans les services du réseau lorsque les messages périodiques entrent en concurrence avec d'autre trafic du réseau pour les liaisons et les ressources de transmission.

Comme RSVP envoie des messages de rafraîchissement périodiques, il doit éviter la synchronisation de messages et s'assurer que toute synchronisation qui pourrait survenir n'est pas stable.

Pour cette raison, le temporisateur de rafraîchissement devrait être réglé de façon aléatoire à une valeur dans la gamme [0,5R, 1,5R].

2. Pour éviter la perte prématurée d'état, L doit satisfaire $L \geq (K + 0,5)*1,5*R$, où K est un petit entier. Dans le pire des cas, K-1 messages successifs peuvent être perdus sans que l'état ne soit supprimé. Pour calculer une durée de vie L pour une collection d'état avec des valeurs R différentes R0, R1, ..., remplacer R par max(Ri).

   Actuellement, K = 3 est suggéré comme valeur par défaut. Cependant, il peut être nécessaire de fixer une plus grande valeur de K pour les bonds avec de forts taux de perte. K peut être établi par configuration manuelle interface par interface, ou par quelque technique adaptative qui n'a pas encore été spécifiée.

3. Chaque message Path ou Resv porte un objet TIME_VALUES qui contient la durée de rafraîchissement R utilisée pour générer des rafraîchissements. Le nœud receveur utilise ces R pour déterminer la durée de vie L de l'état mémorisé créé ou rafraîchi par le message.

4. La durée de rafraîchissement R est choisie localement par chaque nœud. Si le nœud ne met pas en œuvre la réparation locale des réservations perturbées par les changements de chemin, un R plus petit accélère l'adaptation aux changements d'acheminement, tout en accroissant la redondance de RSVP. Avec la réparation locale, un routeur peut être plus détendu avec R car le rafraîchissement périodique devient seulement un mécanisme de robustesse de sauvegarde. Un nœud peut donc ajuster le R effectif de façon dynamique pour contrôler la quantité de redondance due aux messages de rafraîchissement

   La valeur suggérée actuelle de R par défaut est de 30 secondes. Cependant, la valeur par défaut Rdef devrait être configurable interface par interface.

5. Lorsque R est changée de façon dynamique, il y a une limite à la vitesse de sa croissance. Précisément, le ratio de deux valeurs successives R2/R1 ne doit pas excéder 1 + Slew.Max.

   Actuellement, Slew.Max est 0,30. Avec K = 3, un paquet peut être perdu sans péremption d'état alors que R est augmenté de 30 pour cent par cycle de rafraîchissement.

6. Pour améliorer la robustesse, un nœud peut temporairement envoyer des rafraîchissements plus souvent que R après un changement d'état (y compris l'établissement d'état initial).

7. Les valeurs de Rdef, K, et Slew.Max utilisées dans une mise en œuvre devraient être facilement modifiables interface par interface, car l'expérience peut conduire à des valeurs différentes. La possibilité d'une adaptation dynamique de K et/ou Slew.Max en réponse à des taux de perte mesurés sera étudiée prochainement.

3.8 Régulation du trafic et bonds à service non intégré

Certains services de QS peuvent exiger une régulation du trafic à certaines ou toutes (1) les bordures du réseau, (2) point de fusion des données de plusieurs envoyeurs, et/ou (3) un point d'embranchement où le flux de trafic provenant de l'amont peut être supérieur aux réservation demandées vers l'aval. RSVP sait où de tels points surviennent et doit l'indiquer au mécanisme de contrôle du trafic. D'un autre côté, RSVP n'interprète pas le service incorporé dans la flowspec et ne sait donc pas si la régulation va réellement être appliquée dans un cas en particulier.

Le processus RSVP passe au contrôle de trafic un fanion de régulation distinct pour chacune de ces trois situations.

- E_Police_Flag – régulation d'entrée
  Ce fanion est mis dans le nœud RSVP de premier bond qui met en œuvre le contrôle de trafic (et est donc capable de réguler).
  Par exemple, les hôtes envoyeurs doivent mettre en œuvre RSVP mais actuellement beaucoup d'entre eux ne mettent pas en œuvre le contrôle du trafic. Dans ce cas, le fanion E_Police_Flag devrait être à zéro chez l'hôte envoyeur, et il ne devrait être établique quand le premier nœud capable de contrôle de trafic est atteint. Ceci est contrôlé par le fanion E_Police dans les objets SESSION.

- M_Police_Flag – régulation des fusions
  Ce fanion devrait être établi pour une réservation qui utilise un style partagé (WF ou SE) lorsque des flux provenant de

plus d'un envoyeur sont fusionnés.

o B_Police_Flag – régulation d'embranchement
Ce fanion devrait être mis lorsque la flowspec installée est plus petite que, ou non comparable à, la FLOWSPEC en place sur toute autre interface, pour les mêmes FILTER_SPEC et SESSION.

RSVP doit aussi vérifier la présence de bonds non RSVP dans le chemin et passer cette information au contrôle de trafic. D'après ce bit fanion que fournit le processus RSVP et d'après sa propre connaissance locale, le contrôle du trafic peut détecter la présence dans le chemin d'un bond qui n'est pas capable de contrôle de la QS, et il passe cette information aux receveurs dans les Adspec [RFC2210].

Avec la transmission IP normale, RSVP peut détecter un bond non RSVP en comparant le TTL IP avec lequel un message Path est envoyé au TTL avec lequel il est reçu ; à cette fin, le TTL de transmission est placé dans l'en-tête commun. Cependant, le TTL n'est pas toujours un indicateur fiable des bonds non RSVP, et d'autres moyens doivent parfois être utilisés. Par exemple, si le protocole d'acheminement utilise des tunnels d'encapsulation d'IP, le protocole d'acheminement doit alors informer RSVP lorsque des bonds non RSVP sont inclus. Si aucun mécanisme automatique ne fonctionne, la configuration manuelle sera nécessaire.

3.9 Hôtes multi rattachements

S'accommoder des hôtes multi rattachement exige quelques règles particulières dans RSVP. On utilise le terme "multi rattachement" pour parler aussi bien des hôtes (systèmes d'extrémité) avec plus d'une interface réseau que des routeurs qui prennent en charge des programmes d'application locale.

Une application qui s'exécute sur un hôte multi rattachement peut explicitement spécifier quelle interface va utiliser un flux donné pour envoyer et/ou recevoir des paquets de données, pour outrepasser l'interface par défaut spécifiée par le système. Le processus RSVP doit connaître les valeurs par défaut, et si une application fixe une interface spécifique, elle doit aussi passer cette information à RSVP.

o Envoi des données
Une application envoyeuse utilise un appel d'API (SENDER au paragraphe 3.11.1) pour déclarer au RSVP les caractéristiques du flux de données qu'elle va générer. Cet appel peut facultativement inclure l'adresse IP locale de l'envoyeur. Si il est réglé par l'application, ce paramètre doit être l'adresse de l'interface pour l'envoi des paquets de données ; autrement, l'interface par défaut du système est implicite.
Le processus RSVP chez l'hôte envoie alors les messages Path pour cette application par l'interface spécifiée (et elle seule).

o Faire des réservations
Une application receveuse utilise un appel d'API (RESERVE au paragraphe 3.11.1) pour demander une réservation à RSVP. Cet appel peut facultativement inclure l'adresse IP locale du receveur, c'est-à-dire, l'adresse d'interface pour les paquets de données à recevoir. Dans le cas de sessions en diffusion groupée, c'est l'interface sur laquelle le groupe a été rejoint. Si le paramètre est omis, on utilise l'interface par défaut du système.

En général, le processus RSVP devrait envoyer des messages Resv pour une application par l'interface spécifiée. Cependant, lorsque l'application s'exécute sur un routeur et que la session est en diffusion groupée, c'est une situation plus complexe. Supposons dans ce cas qu'une application receveuse se joigne au groupe sur une interface Iapp qui diffère de Isp, l'interface de plus court chemin vers l'envoyeur. Il y a alors deux chemins possibles pour que l'acheminement de diffusion groupée livre les paquets de données à l'application. Le processus RSVP doit déterminer quel cas l'emporte en examinant l'état du chemin, pour décider quelle interface entrante utiliser pour l'envoi des messages Resv.

1. Le protocole d'acheminement de diffusion groupée peut créer un embranchement distinct dans l'arborescence de la distribution de diffusion groupée pour livrer à Iapp. Dans ce cas, il y aura un état de chemin pour les deux interfaces Isp et Iapp. L'état de chemin sur Iapp devrait ne correspondre à une réservation que de l'application locale ; il doit être marqué "Local_seul" par le processus RSVP. Si l'état de chemin "Local_seul" existe pour Iapp, le message Resv devrait être envoyé d'Iapp.

   Noter qu'il est possible que les blocs d'état de chemin pour Isp et Iapp aient le même prochain bond si intervient un nuage non RSVP.

2. Le protocole d'acheminement de diffusion groupée peut transmettre des données au sein du routeur de Isp à Iapp.

Dans ce cas, Iapp va apparaître dans la liste des interface sortantes de l'état de chemin pour Isp, et le message Resv devrait être envoyé de Isp.

3. Lorsque les messages Path et PathTear sont transmis, l'état de chemin marqué "Local_Seul" doit être ignoré.

3.10 Compatibilité future

On peut s'attendre à ce qu'à l'avenir de nouveaux objets C-Type seront définis pour des classes d'objet existantes, et peut-être que de nouvelles classes d'objet seront définies. Il faudra employer de tels nouveaux objets au sein de l'Internet en utilisant des mises en œuvre plus anciennes qui ne les reconnaissent pas. Malheureusement, cela n'est possible que dans une mesure limitée avec une complexité raisonnable. Les règles sont les suivante (`b' représente un bit).

1. Classe inconnue
   Il y a trois façons possibles pour une mise en œuvre RSVP de traiter un objet d'une classe inconnue. Ce choix est déterminé par les deux bits de plus fort poids de l'octet Class-Num, comme suit .

   o Class-Num = 0bbbbbbb
   Le message entier devrait être rejeté et une erreur "Classe d'objet inconnue" retournée.

   o Class-Num = 10bbbbbb
   Le nœud devrait ignorer l'objet, et ne pas le transmettre ni envoyer de message d'erreur.

   o Class-Num = 11bbbbbb
   Le nœud devrait ignorer l'objet mais le transmettre, non examiné ni modifié,dans tous les messages résultant de ce message.

   Les règles plus détaillées sont valables pour les objets de classe inconnue avec un Class-Num de forme 11bbbbbb :

   1 De tels objets de classe inconnue reçus dans les messages PathTear, ResvTear, PathErr, ou ResvErr devraient être transmis immédiatement dans les mêmes messages.

   2. De tels objets de classe inconnue reçus dans les messages Path ou Resv devraient être sauvegardés avec l'état correspondant et transmis dans tout message de rafraîchissement résultant de cet état.

   3. Lorsque un rafraîchissement de Resv est généré par la fusion de plusieurs demandes de réservation, le message de rafraîchissement devrait inclure l'union des objets de classe inconnue des demandes composantes *(de la fusion)*. Une seule copie de chaque objet de classe inconnue devrait être incluse dans cette union.

   4. L'ordre d'origine de tels objets de classe inconnue n'a pas besoin d'être conservé ; cependant, le message qui est transmis doit obéir aux exigences générales d'ordre pour son type de message.

   Bien que les objets de classe inconnue ne puissent pas être fusionnés, ces règles vont transmettre de tels objets jusqu'à ce qu'ils atteignent un nœud qui sait comment les fusionner. La transmission d'objets de classe inconnue permet un déploiement incrémentaire de nouveaux objets ; cependant, les limitations d'échelle d'une telle pratique doivent être examinées avec soin avant de déployer une nouvelle classe d'objet avec les deux bits de poids forts établis.

2. C-Type inconnu pour une classe connue
   On peut s'attendre à ce que le Class-Num connu fournisse des informations qui pourraient permettre un traitement intelligent d'un tel objet. Cependant, en pratique, un tel traitement dépendant de la classe est complexe, et dans de nombreux cas, n'est pas utile.

   Généralement, l'apparition d'un objet d'un C-Type inconnu devrait déboucher sur le rejet du message entier et générer un message d'erreur (ResvErr ou PathErr selon le cas). Le message d'erreur va inclure le Class-Num et le C-Type qui ont échoué (voir l'Appendice B) ; le système d'extrémité qui a généré le message en échec peut être capable d'utiliser cette information pour réessayer la demande en utilisant un objet C-Type différent, répétant ce procès jusqu'à ce qu'il réussisse ou qu'il n'ait plus de solution de remplacement.

   Les objets de certaines classes (FLOWSPEC, ADSPEC, et POLICY_DATA) sont opaques pour RSVP, qui les passe simplement au contrôle de trafic ou aux modules de régulation. Selon ses règles internes, l'un ou l'autre de ces derniers modules peut rejeter un C-Type et informer le processus RSVP ; RSVP devrait alors rejeter le message et envoyer une erreur, comme décrit au paragraphe précédent.

3.11 Interfaces RSVP

RSVP sur un routeur a des interfaces pour acheminer et pour contrôler le trafic. RSVP sur un hôte a une interface avec les applications (c'est-à-dire, une API) et aussi une interface avec le contrôle de trafic (si il existe sur l'hôte).

3.11.1 Interface application/RSVP

Ce paragraphe décrit une interface générique entre une application et un processus de contrôle RSVP. Les détails d'une interface réelle peuvent dépendre du système d'exploitation ; ce qui suit suggère seulement les fonctions de base à effectuer. Certains de ces appels causent un retour asynchrone des informations.

o Enregistrer une session
Appel : SESSION(DestAddress , ProtocolId, DstPort [, SESSION_object] [, Upcall_Proc_addr]) -> Session-id

Cet appel initie un traitement RSVP pour une session, définie par une DestAddress ainsi qu'un ProtocolId et éventuellement un numéro d'accès DstPort. S'il réussit, l'appel SESSION retourne immédiatement avec un identifiant de session locale Session-id, qui peut être utilisé dans les appels ultérieurs.

Le paramètre Upcall_Proc_addr définit l'adresse d'un rappel de procédure pour recevoir une notification asynchrone d'erreur ou d'événement ; voir ci-dessous. Le paramètre SESSION_object est inclus comme mécanisme d'échappement pour la prise en charge d'une définition plus générale de la session ("accès de destination généralisé") si cela devait devenir nécessaire à l'avenir. Normalement, SESSION_object sera omis.

o Définir l'envoyeur
Appel : SENDER(Session-id [, Source_Address] [, Source_Port] [, Sender_Template] [, Sender_Tspec] [, Adspec] [, Data_TTL] [, Policy_data])

Un envoyeur utilise cet appel pour définir, ou modifier la définition des attributs d'un flux de données. Le premier appel SENDER pour la session enregistré comme "Session-id" va causer l'envoi de messages Path par RSVP pour cette session ; les appels ultérieurs vont modifier les informations de chemin.

Les paramètres SENDER sont interprétés comme suit :

- Source_Address
C'est l'adresse de l'interface d'où les données seront envoyées. Si il est omis, on utilisera une interface par défaut. Ce paramètre est nécessaire seulement sur un hôte envoyeur à rattachements multiples.

- Source_Port
C'est l'accès UDP/TCP à partir duquel les données sont envoyées.

- Sender_Template
Ce paramètre est inclus dans un mécanisme d'échappement pour prendre en charge une définition plus générale de l'envoyeur ("accès de source généralisé"). Normalement, ce paramètre peut être omis.

- Sender_Tspec
Ce paramètre décrit le flux de trafic à envoyer, voir la [RFC2210].

- Adspec
Ce paramètre peut être spécifié pour initialiser le calcul des propriétés de QS le long du chemin ; voir [RFC2210].

- Data_TTL
C'est le paramètre Durée de vie IP (pas par défaut) qui est fourni sur les paquets de données. Il est nécessaire pour s'assurer que les messages Path n'ont pas une portée plus grande que celle des paquets de données en diffusion groupée.

- Policy_data
Ce paramètre facultatif passe les données de politique pour l'envoyeur. Ces données peuvent être fournies par un service système, l'application le traitant comme opaque.

o Réservé

Appel : RESERVE(session-id, [receveur_adress ,] [CONF_flag,] [Policy_data,] style, style-dependent-parms)

Un receveur utilise cet appel pour faire ou modifier une réservation de ressource pour la session enregistrée comme "session-id". Le premier appel RESERVE va initier la transmission périodique de messages Resv. Un appel RESERVE ultérieur peut être fait pour modifier les paramètres de l'appel antérieur (mais noter que changer les réservations existantes peut résulter en des défaillances du contrôle d'admission).

Le paramètre facultatif "receveur_adresse" peut être utilisé par un receveur sur un hôte (ou routeur) multi rattachement ; c'est l'adresse IP de l'une des interfaces du nœud. Le fanion CONF_flag devrait être établi si une confirmation de réservation est désirée, et à zéro autrement. Le paramètre "Policy_data" spécifie les données de politique pour le receveur, alors que le paramètre "style" indique le style de la réservation. Le reste des paramètres dépend du style ; ce seront généralement les flowspec et spec de filtre appropriées.

L'appel RESERVE est retourné immédiatement. À la suite d'un appel RESERVE, une invocation asynchrone ERROR/EVENT peut survenir à tout moment.

- Libération
  Appel : RELEASE(session-id)
  Cet appel supprime l'état RSVP pour la session spécifiée par session-id. Le nœud envoie alors les messages de suppression appropriés et cesse d'envoyer des messages de rafraîchissement pour ce session-id.

- Rappels d'erreur/événement
  La forme générale d'un rappel est la suivante :
  Rappel : <Upcall_Proc>( ) -> session-id, Info_type, information_parameters

  Ici, "Upcall_Proc" représente la procédure de rappel dont l'adresse a été fournie dans l'appel SESSION. Ce rappel peut survenir en asynchrone à tout moment après un appel SESSION et avant un appel RELEASE, pour indiquer une erreur ou un événement.

  Actuellement, il y a cinq types de rappel, distingués par le paramètre Info_type. Le choix des paramètres d'information dépend du type.

  1. Info_type = PATH_EVENT
     Un rappel d'événement de chemin résulte de la réception du premier message Path pour cette session, indiquant à une application receveuse qu'il y a au moins un envoyeur actif, ou si l'état du chemin change.

     Rappel : <Upcall_Proc>( ) -> session-id, Info_type=PATH_EVENT, Sender_Tspec, Sender_Template [, Adspec] [, Policy_data]

     Ce rappel présente le Sender_Tspec, le Sender_Template, le Adspec, et toutes données de politique d'un message Path.

  2. Info_type = RESV_EVENT
     Un rappel d'événement de réservation est déclenché par la réception du premier message RESV, ou par la modification d'un état de réservation précédent, pour cette session.

     Rappel : <Upcall_Proc>( ) -> session-id, Info_type=RESV_EVENT, Style, Flowspec, Filter_Spec_list [, Policy_data]

     Ici, "Flowspec" va être la QS effective qui a été reçue. Noter qu'un message Resv de style FF peut résulter en plusieurs rappels RESV_EVENT, un pour chaque descripteur de flux.

  3. Info_type = PATH_ERROR
     Un événement Erreur de chemin indique une erreur dans les informations d'envoyeur qui étaient spécifiées dans un appel SENDER.

     Rappel : <Upcall_Proc>( ) -> session-id, Info_type=PATH_ERROR, Error_code , Error_value , Error_Node , Sender_Template [, Policy_data_list]

     Le paramètre Error_code va définir l'erreur, et Error_value peut fournir des données supplémentaires (peut-être spécifiques du système) sur l'erreur. Le paramètre Error_Node va spécifier l'adresse IP du nœud qui a détecté l'erreur. Le paramètre Policy_data_list, s'il est présent, contiendra tous les objets POLICY_DATA du message Path en échec.

4. Info_type = RESV_ERR
   Un événement Erreur de réservation indique une erreur dans un message de réservation auquel cette application a contribué.

   Rappel : <Upcall_Proc>( ) -> session-id, Info_type=RESV_ERROR, Error_code , Error_value , Error_Node , Error_flags , Flowspec, Filter_spec_list [, Policy_data_list]

   Le paramètre Error_code va définir l'erreur et Error_value peut fournir des données supplémentaires (peut-être spécifiques du système). Le paramètre Error_Node va spécifier l'adresse IP du nœud qui a détecté l'événement rapporté.

   Il y a deux Error_flags :

   - InPlace
     Ce fanion peut être mis pour une défaillance de contrôle d'admission, pour indiquer qu'il y avait, et qu'il y a toujours, une réservation en place au nœud défaillant. Ce fanion est mis au point de défaillance et transmis dans les messages ResvErr.

   - NotGuilty
     Ce fanion peut être mis pour une défaillance de contrôle d'admission, pour indiquer que la flowspec demandée par ce receveur était strictement inférieure à la flowspec qui a eu l'erreur. Ce fanion est mis à l'API receveuse.

     Filter_spec_list et Flowspec contiennent les objets correspondants du descripteur de flux erroné (voir au paragraphe 3.1.8). List_count spécifie le nombre de FILTER_SPECS dans Filter_spec_list. Le paramètre Policy_data_list contient tous les objets POLICY_DATA du message ResvErr.

5. Info_type = RESV_CONFIRM
   Un événement Confirmation indique qu'un message ResvConf a été reçu.

   Rappel : <Upcall_Proc>( ) -> session-id, Info_type=RESV_CONFIRM, Style, List_count, Flowspec, Filter_spec_list [, Policy_data]

   Les paramètres sont interprété comme le rappel Erreur de réservation.

   Bien que les messages RSVP indiquant des événements de chemin ou de réservation puissent être reçus périodiquement, l'API ne devrait faire le rappel asynchrone correspondant pour l'application que sur la première occurrence ou lorsque les informations à rapporter changent. Tous les événements d'erreur et de confirmation devraient être rapportés à l'application.

3.11.2 Interface de contrôle RSVP/trafic

Il est difficile de présenter une interface générique au contrôle de trafic, parce que les détails de l'établissement d'une réservation dépendent fortement de la technologie particulière de la couche liaison utilisée sur une telle interface.

La fusion des réservations RSVP est nécessaire à cause de la livraison des données en diffusion groupée, qui duplique les paquets de données pour leur livraison sur les différents nœuds de prochain bond. À chacun de ces points de réplication, RSVP doit fusionner les demandes de réservation provenant des prochains bonds correspondants en calculant le "maximum" de leur flowspec. Chez un routeur ou hôte donné, une ou plusieurs des trois localisations de réplication suivantes peuvent être utilisées.

1. Couche IP
   La transmission IP en diffusion groupée effectue la réplication dans la couche IP. Dans ce cas, RSVP doit fusionner les réservations qui sont en place sur les interfaces sortantes correspondantes dans l'ordre pour transmettre une demande vers l'amont.

2. "Le réseau"
   La réplication peut avoir lieu en aval du nœud, par exemple, dans un LAN de diffusion, dans des commutateurs de couche liaison, ou dans un maillage de routeurs sans capacité RSVP (voir au paragraphe 2.8). Dans ces cas, RSVP doit fusionner les réservations provenant des différents prochains bonds afin de faire la réservation sur la seule interface sortante. Il doit aussi fusionner les demandes de réservations provenant de toutes les interfaces sortantes afin de transmettre une demande vers l'amont.

3. Pilote de couche liaison
   Pour une technologie multi accès, la réplication peut survenir dans le pilote de couche liaison ou la carte d'interface. Par exemple, ce cas peut se produire lorsqu'il y a un circuit virtuel ATM point à point distinct vers chaque prochain bond. RSVP peut avoir besoin d'appliquer le contrôle de trafic indépendamment à chaque circuit virtuel, sans fusionner les demandes des différents prochains bonds.

   En général, ces situations complexes n'ont pas d'impact sur le traitement du protocole qui est exigé par RSVP, excepté pour déterminer exactement quelles demandes de réservation doivent être fusionnées. Il peut être souhaitable d'organiser une mise en œuvre RSVP en deux parties : un cœur qui effectue le traitement indépendant de la couche de liaison, et une couche d'adaptation dépendante de la couche de liaison. Cependant, nous présentons ici une interface générique qui suppose que la réplication ne peut survenir qu'à la couche IP ou dans "le réseau".

o Faire une réservation
  Appel : TC_AddFlowspec(Interface, TC_Flowspec, TC_Tspec, TC_Adspec, Police_Flags) -> RHandle [, Fwd_Flowspec]

  Le paramètre TC_Flowspec définit la QS désirée effective pour le contrôle d'admission ; sa valeur est calculée comme le maximum sur la flowspecs des différents prochains bonds (voir l'appel Compare_Flowspecs ci-dessous). Le paramètre TC_Tspec définit le Path_Te de Tspec effective d'envoyeur (voir au paragraphe 2.2). Le paramètre TC_Adspec définit l'Adspec effective. Le paramètre Police_Flags porte les trois fanions E_Police_Flag, M_Police_Flag, et B_Police_Flag ; voir au paragraphe 3.8.

  Si cet appel est réussi, il établit un nouveau canal de réservation correspondant à RHandle ; autrement, il retourne un code d'erreur. Le nombre opaque RHandle est utilisé par l'appelant pour les références ultérieures à cette réservation. Si le service de contrôle de trafic met à jour la flowspec, l'appel va aussi retourner l'objet mis à jour comme Fwd_Flowspec.

o Modifier une réservation
  Appel : TC_ModFlowspec(Interface, RHandle, TC_Flowspec, TC_Tspec, TC_Adspec, Police_flags) [-> Fwd_Flowspec]

  Cet appel est utilisé pour modifier une réservation existante. La TC_Flowspec est passée au contrôle d'admission ; si elle est rejetée, la flowspec actuelle reste en vigueur. Les spec de filtre correspondantes, s'il en est, ne sont pas affectées. Les autres paramètres sont définis comme dans TC_AddFlowspec. Si le service met à jour la flowspec, l'appel va aussi retourner l'objet mis à jour comme Fwd_Flowspec.

o Supprimer une Flowspec
  Appel : TC_DelFlowspec(Interface, RHandle)

  Cet appel va supprimer une réservation existante, y compris la flowspec et toutes les spec de filtre associées.

o Ajouter une spec de filtre
  Appel : TC_AddFilter(Interface, RHandle, Session , FilterSpec) -> FHandle

  Cet appel est utilisé pour associer une spec de filtre supplémentaire à la réservation spécifiée par le RHandle donné, suite à un appel TC_AddFlowspec réussi. Cet appel retourne un traitement de filtre FHandle.

o Supprimer une spec de filtre
  Appel : TC_DelFilter(Interface, FHandle)

  Cet appel est utilisé pour retirer un filtre spécifique, spécifié par FHandle.

o Mise à jour d'OPWA
  Appel : TC_Advertise(Interface, Adspec, Non_RSVP_Hop_flag) -> New_Adspec

  Cet appel est utilisé pour que l'OPWA calcule l'annonce sortante New_Adspec pour une interface spécifiée. Le bit de fanion Non_RSVP_Hop_flag devrait être établi chaque fois que le démon RSVP détecte que le précédent bond RSVP comporte un ou plusieurs routeurs sans capacité RSVP. TC_Advertise va insérer cette information dans New_Adspec pour indiquer qu'il a été trouvé un bond sans intégration de service ; voir au paragraphe 3.8.

o Rappel de préemption
  Rappel : TC_Preempt() -> RHandle, Reason_code

Afin d'accorder une nouvelle demande de réservation, les modules de contrôle d'admission et/ou de contrôle de politique peuvent préempter une ou plusieurs réservations existantes. Cela va déclencher un rappel TC_Preempt() à RSVP pour chaque réservation préemptée, qui passe le RHandle de la réservation et un sous code qui en indique la raison.

3.11.3 Interface RSVP/contrôle de politique

Cette interface sera spécifiée dans un document à paraître.

3.11.4 Interface RSVP/acheminement

Une mise en œuvre RSVP doit avoir le soutien des mécanismes d'acheminement du nœud suivants.

o Interrogation de chemin
Pour transmettre les messages Path et PathTear, un processus RSVP doit être capable d'interroger le ou les processus d'acheminement sur les chemins.

Ucast_Route_Query([SrcAddress,] DestAddress, Notify_flag) -> OutInterface Mcast_Route_Query([SrcAddress,] DestAddress, Notify_flag) -> [IncInterface,] OutInterface_list

Selon le protocole d'acheminement, l'interrogation peut ne pas dépendre de SrcAddress, c'est-à-dire, de l'adresse IP de l'hôte envoyeur, qui est aussi l'adresse IP de source du message. Ici, IncInterface est l'interface à travers laquelle le paquet est supposé arriver ; certains protocoles d'acheminement de diffusion groupée peuvent ne pas le fournir. Si le Notify_flag est Vrai, l'acheminement va conserver l'état nécessaire pour produire des rappels non sollicités de notification de changement de chemin (voir ci-dessous) chaque fois que le chemin spécifié change.

Une interrogation de chemin de diffusion groupée peut retourner une OutInterface_list vide si il n'y a pas de receveur en aval d'un routeur particulier. Une interrogation de chemin peut aussi retourner une erreur "Pas de tel chemin", probablement par suite d'une incohérence transitoire de l'acheminement (parce qu'un message Path ou PathTear est arrivé à ce nœud pour le chemin demandé). Dans l'un et l'autre cas, l'état local devrait être mis à jour comme demandé par le message, qui ne peut pas être transmis plus loin. La mise à jour de l'état local va rendre l'état de chemin immédiatement disponible pour un nouveau receveur local, ou il va supprimer immédiatement l'état de chemin.

o Notification de changement de chemin
Si c'est demandé par une interrogation de chemin avec le fanion Notify_flag Vrai, le processus d'acheminement peut fournir un rappel asynchrone au processus RSVP qui indique qu'un chemin spécifié a changé.

Ucast_Route_Change( ) -> [SrcAddress,] DestAddress, OutInterface Mcast_Route_Change( ) -> [SrcAddress,] DestAddress, [IncInterface,] OutInterface_list

o Découverte de liste d'interfaces
RSVP doit être capable d'apprendre quelles interfaces réelles et virtuelles sont actives, avec leur adresse IP.

Il devrait être possible de désactiver logiquement une interface pour RSVP. Lorsque une interface est désactivée pour RSVP, un message Path ne devrait jamais être transmis de cette interface, et si un message RSVP est reçu sur cette interface, le message devrait être éliminé en silence (peut être avec un enregistrement de journalisation local).

3.11.5 Interface d'entrée/sortie RSVP/paquet

Une mise en œuvre RSVP a besoin du soutien des mécanismes d'entrée/sortie de paquets et de transmission du nœud suivants.

o Mode de réception de proximité pour les messages RSVP
Les paquets reçus pour le protocole IP 46 mais non adressés au nœud doivent être déviés sur le programme RSVP pour traitement, sans être transmis. Les messages RSVP à dévier de cette manière incluront les messages Path, PathTear, et ResvConf. Ces types de message portent l'option IP Alerte de routeur qui peut être utilisée pour les extraire d'un chemin de transmission à grande vitesse. Autrement, le nœud peut intercepter tous les paquets du protocole 46.

Sur un routeur ou hôte multi rattachement, l'identité de l'interface (réelle ou virtuelle) sur laquelle un message dévié est reçu, ainsi que l'adresse IP de source et le TTL IP avec lesquels il est arrivé, doivent aussi être disponibles au processus RSVP.

- Spécification de liaison sortante
  RSVP doit être capable de forcer l'envoi d'un datagramme (de diffusion groupée) sur une liaison sortante réelle ou virtuelle spécifique, en outrepassant le mécanisme d'acheminement normal. Une liaison virtuelle peut être un tunnel de diffusion groupée, par exemple. La spécification de la liaison sortante est nécessaire pour envoyer des versions différentes d'un message Path sortant sur des interfaces différentes, et pour éviter des boucles d'acheminement dans certains cas.

- Spécification d'adresse de source et TTL
  RSVP doit être capable de spécifier l'adresse IP de source et la TTL IP à utiliser lors de l'envoi de messages Path.

- Alerte de routeur
  RSVP doit être capable de causer l'envoi de messages Path, PathTear, et ResvConf avec l'option IP Alerte de routeur.

3.11.6 Manipulations dépendantes du service

Les Flowspec, Tspec et Adspec sont des objets opaques pour RSVP ; leur contenu est défini dans des documents de spécification de service. Pour manipuler ces objets, le processus RSVP doit avoir les programmes dépendant du service suivant à sa disposition.

- Comparer les Flowspec
  Compare_Flowspecs(Flowspec_1, Flowspec_2) -> result_code

  Les result_code possibles indiquent : les flowspec sont égales, la Flowspec_1 est supérieure, la Flowspec_2 est supérieure, les flowspec sont incomparables mais le LUB peut être calculé, ou les flowspec sont incompatibles

  Noter que comparer deux flowspec compare implicitement les Tspec qui sont contenues. Bien que le processus RSVP ne puisse pas par lui-même analyser une flowspec pour extraire la Tspec, il peut utiliser l'invocation Compare_Flowspecs pour calculer implicitement Resv_Te (voir au paragraphe 2.2).

- Calculer le LUB des Flowspec
  LUB_of_Flowspecs(Flowspec_1, Flowspec_2) -> Flowspec_LUB

- Calculer le GLB des Flowspec
  GLB_of_Flowspecs(Flowspec_1, Flowspec_2) -> Flowspec_GLB

- Comparer les Tspec
  Compare_Tspecs(Tspec_1, Tspec_2) -> result_code
  Les result_codes possibles indiquent : les Tspec sont égales, ou les Tspec sont inégales.

- Faire la somme des Tspec
  Sum_Tspecs(Tspec_1, Tspec_2) -> Tspec_sum
  Cette invocation est utilisée pour calculer Path_Te (voir au paragraphe 2.2).

4. Remerciements

La conception de RSVP se fonde sur des recherches effectuées en 1992-1993 par la collaboration de Lixia Zhang (UCLA), Deborah Estrin (USC/ISI), Scott Shenker (Xerox PARC), Sugih Jamin (USC/Xerox PARC) et Daniel Zappala (USC). Sugih Jamin a développé le premier prototype de mise en œuvre de RSVP et en a donné une démonstration réussie en mai 1993.
Shai Herzog, et plus tard Steve Berson, ont continué le développement des prototypes de RSVP.

Depuis 1993, de nombreux membres de la communauté Internet de la recherche ont contribué à la conception et au développement de RSVP ; parmi eux (en ordre alphabétique) Steve Berson, Bob Braden, Lee Breslau, Dave Clark, Deborah Estrin, Shai Herzog, Craig Partridge, Scott Shenker, John Wroclawski, Daniel Zappala, et Lixia Zhang.De plus, un certain nombre de fabricants d'hôtes et de routeurs ont fait des contributions précieuses aux documents RSVP, en particulier Fred Baker (Cisco), Mark Baugher (Intel), Lou Berger (Fore Systems), Don Hoffman (Sun), Steve Jakowski (NetManage), John Krawczyk (Bay Networks), et Bill Nowicki (SGI), ainsi que de nombreux autres.

Appendice A. Définitions des objets

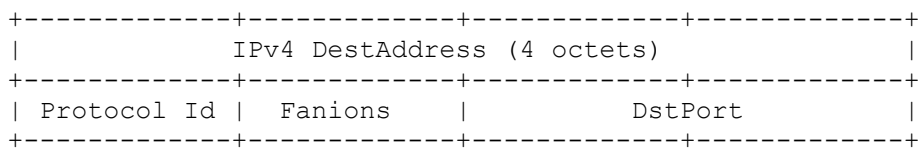
Les types de classe (C-Type) sont définis pour les deux familles d'adresse Internet IPv4 et IPv6. Pour s'accommoder d'autres familles d'adresses, des C-Types supplémentaires pourraient facilement être définis. Ces définitions sont contenues en Appendice, pour en faciliter la mise à jour.

Tous les champs non utilisés devraient être envoyés à zéro et ignorés à réception.

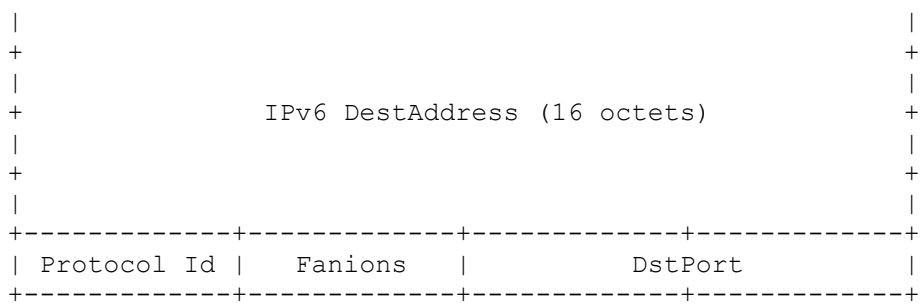
A.1 Classe SESSION

Classe SESSION = 1.

o Objet IPv4/UDP SESSION : Classe = 1, C-Type = 1



o Objet IPv6/UDP SESSION : Classe = 1, C-Type = 2



DestAddress
C'est l'adresse IP de destination d'envoi individuel ou de diffusion groupée de la session. Ce champ ne doit pas être à zéro.

Protocol Id
C'est l'identifiant de protocole IP pour le flux de données. Ce champ ne doit pas être à zéro.

Fanions
0x01 = Fanion E_Police
Le fanion E_Police est utilisé dans les messages Path pour déterminer le "bord" effectif du réseau, pour contrôler la régulation du trafic. Si l'hôte envoyeur n'est pas lui-même capable de réguler le trafic, il va régler ce bit à un dans les messages Path qu'il envoie. Le premier nœud dont le RSVP est capable de réguler le trafic va le faire (si c'est approprié pour le service) et ôter le fanion *(le mettre à zéro)*.

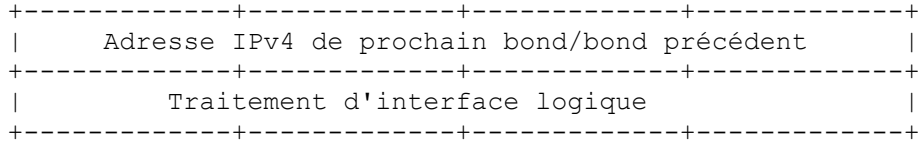
DstPort
C'est l'accès UDP/TCP de destination pour la session. Zéro peut être utilisé pour indiquer "aucun".

D'autres C-Type SESSION pourront être définis à l'avenir pour prendre en charge d'autres conventions de démultiplexage dans la couche transport ou la couche application.

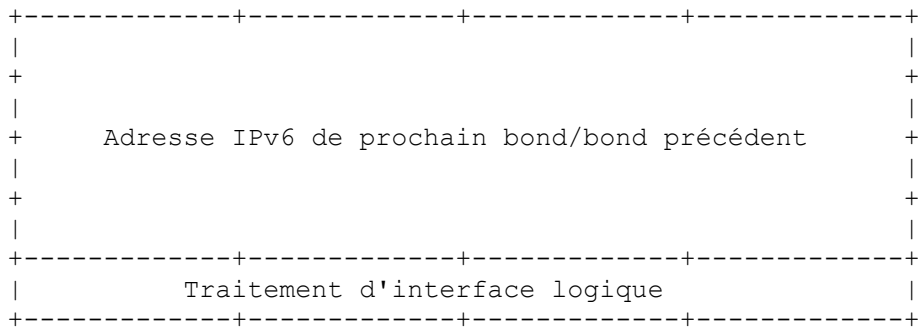
A.2 Classe RSVP_HOP

Classe RSVP_HOP = 3.

o Objet IPv4 RSVP_HOP : Classe = 3, C-Type = 1



o Objet IPv6 RSVP_HOP : Classe = 3, C-Type = 2



Cet objet porte l'adresse IP de l'interface à travers laquelle le dernier bond à capacité RSVP a transmis ce message. Le traitement logique d'interface (LIH, *Logical Interface Handle*) est utilisé pour distinguer les interfaces sortantes logiques, comme exposé aux paragraphes 3.3 et 3.9. Un nœud qui reçoit un LIH dans un message Path sauvegarde sa valeur et la retourne dans les objets HOP des messages Resv suivants envoyés au nœud qui a généré le LIH. Le LIH devrait être entièrement de zéros si il n'y a pas de traitement d'interface logique.

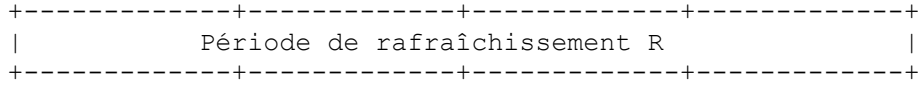
A.3 Classe INTEGRITY

Classe INTEGRITY = 4. Voir la [RFC2747].

A.4 Classe TIME_VALUES

Classe TIME_VALUES = 5.

o Objet TIME_VALUES : Classe = 5, C-Type = 1

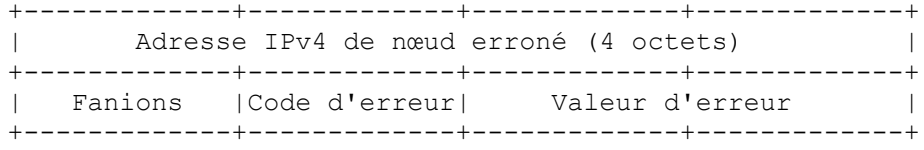


Période de rafraîchissement
C'est la période de temporisation de rafraîchissement R utilisée pour générer ce message; en millisecondes.

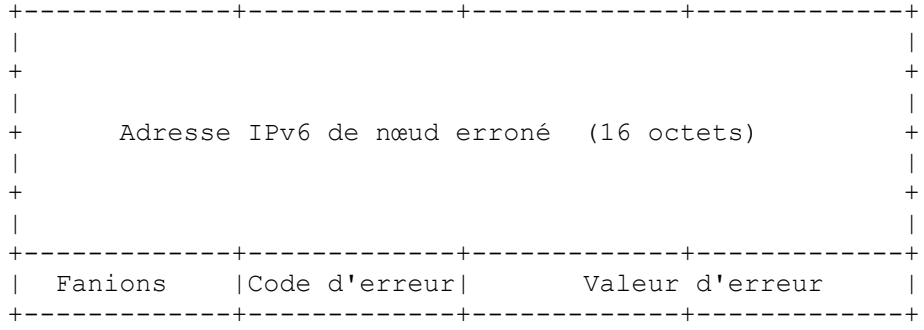
A.5 Classe ERROR_SPEC

Classe ERROR_SPEC = 6.

o Objet IPv4 ERROR_SPEC : Classe = 6, C-Type = 1



o Objet IPv6 ERROR_SPEC : Classe = 6, C-Type = 2



Adresse de nœud erroné
C'est l'adresse IP du nœud dans lequel l'erreur a été détectée.

Fanions
0x01 = InPlace
Ce fanion n'est utilisé que pour un objet ERROR_SPEC dans un message ResvErr. S'il est à un, le fanion indique qu'il y avait, et qu'il y a toujours, une réservation en place au point de défaillance.

0x02 = NotGuilty
Ce fanion est utilisé seulement pour un objet ERROR_SPEC dans un message ResvErr, et il n'est mis que dans l'interface avec l'application receveuse. S'il est mis, ce fanion indique que la FLOWSPEC défaillante était strictement supérieure à la FLOWSPEC demandée par ce receveur.

Code d'erreur
Description d'erreur d'un octet.

Valeur d'erreur
Champ de deux octets qui contient des informations supplémentaires sur l'erreur. Son contenu dépend du type d'erreur.

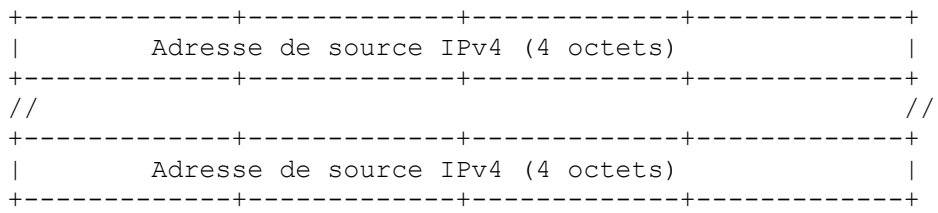
Les valeurs de code d'erreur et les valeurs d'erreur sont définies à l'Appendice B.

A.6 Classe SCOPE

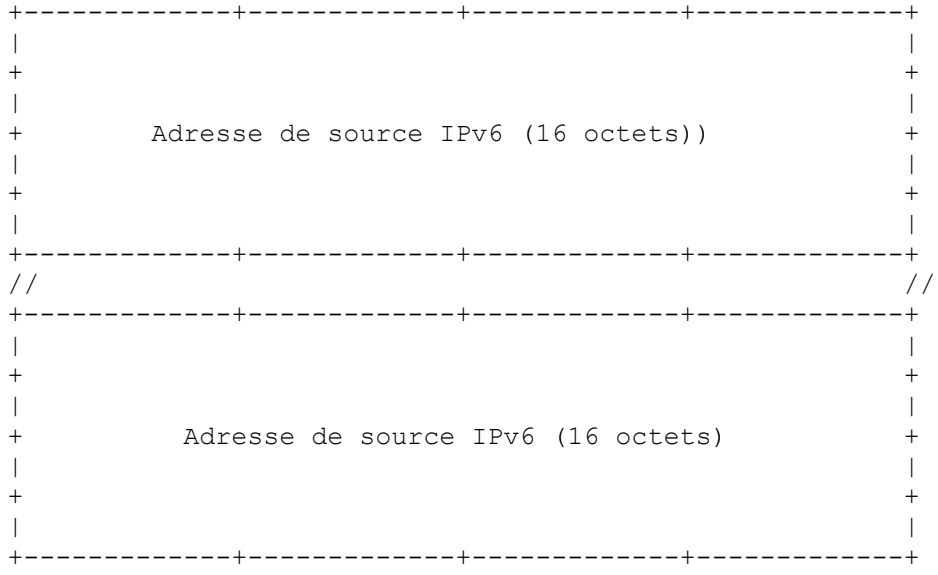
Classe SCOPE = 7.

Cet objet contient une liste d'adresses IP, utilisées pour acheminer les messages avec une portée générique sans boucle. Les adresses doivent être classées par ordre numérique croissant.

o Objet IPv4 SCOPE List : Classe = 7, C-Type = 1



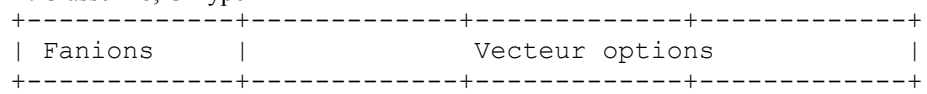
o Objet IPv6 SCOPE List : Classe = 7, C-Type = 2



A.7 Classe STYLE

Classe STYLE = 8.

o Objet STYLE : Classe = 8, C-Type = 1



Fanions : 8 bits (aucun n'est alloué pour l'instant)

Vecteur options : 24 bits
Ensemble de champs de bits qui donnent des valeurs pour les options de réservation. Si de nouvelles options sont ajoutées à l'avenir, les champs correspondants dans le vecteur d'options seront alloués à partir de l'extrémité de moindre poids. Si un nœud ne reconnaît pas un identifiant de style, il peut interpréter autant du vecteur d'options qu'il peut, et ignorer les nouveaux champs qui peuvent avoir été définis.

Les bits du vecteur d'options sont alloués (à partir de la gauche) comme suit :
19 bits : Réservés
2 bits : Contrôle partagé
- 00b : Réservé
- 01b : Réservations distinctes
- 10b : Réservations partagées
- 11b : Réservé

3 bits : Contrôle de sélection d'envoyeur
- 000b : Réservé
- 001b : Générique
- 010b : Explicite
- 011b à 111b : Réservé

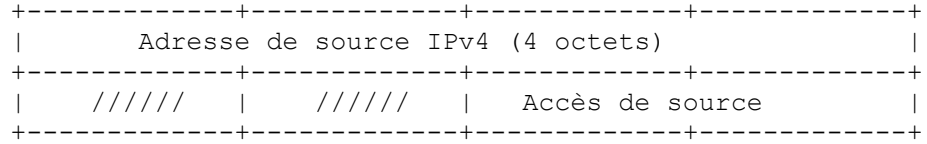
Les bits de moindre poids du vecteur d'options sont déterminés par le style, comme suit :
WF : 10001b
FF : 01010b
SE : 10010b

A.8 Classe FLOWSPEC

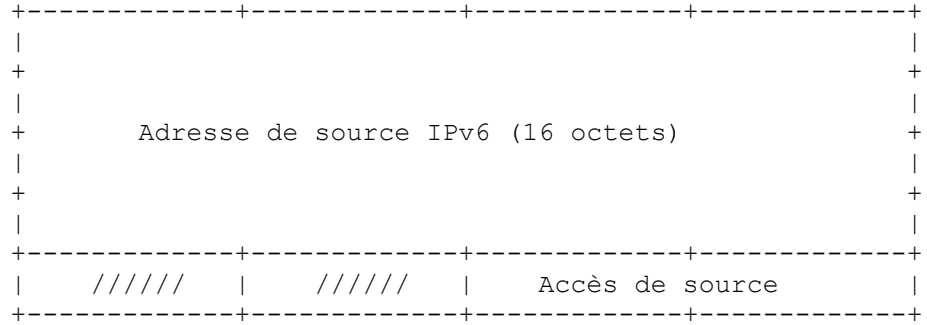
Classe FLOWSPEC = 9.
o Objet flowspec réservé (obsolète) : Classe = 9, C-Type = 1
o Objet flowspec Inv-serv : Classe = 9, C-Type = 2
Le contenu et les règles de codage pour cet objet sont spécifiés dans les documents préparés par le groupe de travail int-serv [RFC2210].

A.9 Classe FILTER_SPEC

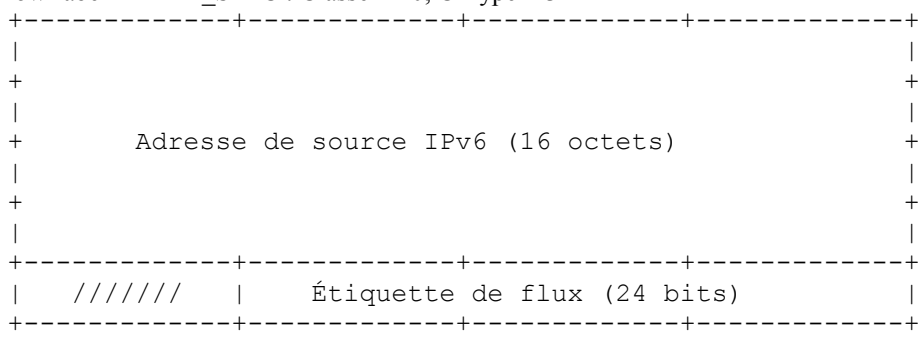
Classe FILTER_SPEC = 10.
o Objet IPv4 FILTER_SPEC : Classe = 10, C-Type = 1



o Objet IPv6 FILTER_SPEC : Classe = 10, C-Type = 2



o Objet IPv6 Flow-label FILTER_SPEC : Classe = 10, C-Type = 3



Adresse de source
Adresse IP de source de l'hôte envoyeur. Ne doit pas être à zéro.

Accès de source
Accès de source UDP/TCP pour un envoyeur, ou à zéro pour indiquer "aucun".

Étiquette de flux
Une étiquette de flux de 24 bits, définie dans IPv6. Cette valeur peut être utilisée par le classeur de paquet pour identifier efficacement les paquets qui appartiennent à un flux de données particulier (envoyeur->destination).

A.10 Classe SENDER_TEMPLATE

Classe SENDER_TEMPLATE = 11.

o Objet IPv4 SENDER_TEMPLATE : Classe = 11, C-Type = 1
Même définition que pour l'objet IPv4/UDP FILTER_SPEC.

o Objet IPv6 SENDER_TEMPLATE : Classe = 11, C-Type = 2
Même définition que pour l'objet IPv6/UDP FILTER_SPEC.

o Objet IPv6 Flow-label SENDER_TEMPLATE : Classe = 11, C-Type = 3

A.11 Classe SENDER_TSPEC

Classe SENDER_TSPEC = 12.

o Objet Intserv SENDER_TSPEC : Classe = 12, C-Type = 2
Le contenu et les règles de codage de cet objet sont spécifiés dans la [RFC2210].

A.12 Classe ADSPEC

Classe ADSPEC = 13.

o Objet Intserv ADSPEC : Classe = 13, C-Type = 2
Le contenu et les règles de codage de cet objet sont spécifiés dans la [RFC2210].

A.13 Classe POLICY_DATA

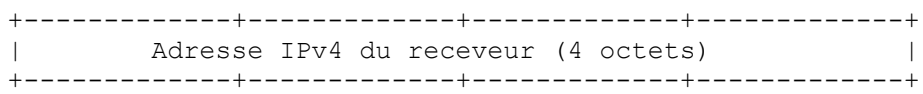
Classe POLICY_DATA = 14.

o Objet Type 1 POLICY_DATA : Classe = 14, C-Type = 1
Le contenu de cet objet fera l'objet d'un complément d'étude.

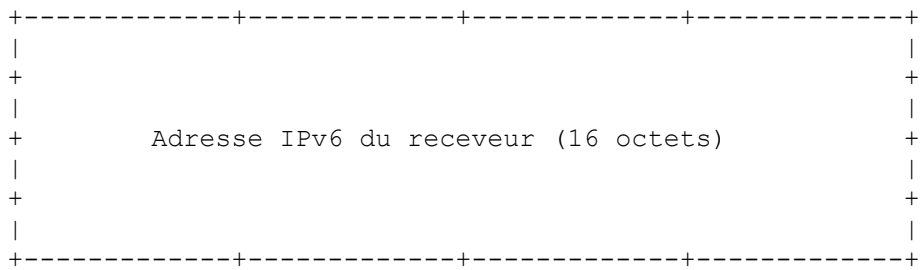
A.14 Classe Resv_CONFIRM

Classe RESV_CONFIRM = 15.

o Objet IPv4 RESV_CONFIRM : Classe = 15, C-Type = 1



o Objet IPv6 RESV_CONFIRM : Classe = 15, C-Type = 2



Appendice B Codes d'erreur et valeurs

Les codes d'erreur suivants peuvent apparaître dans les objets ERROR_SPEC et être passés aux systèmes d'extrémité. Sauf note contraire, ces codes d'erreur ne peuvent apparaître que dans les messages ResvErr.

o Code d'erreur = 00 : Confirmation
Ce code est réservé à l'utilisation dans l'objet ERROR_SPEC d'un message ResvConf. La valeur d'erreur sera aussi zéro.

o Code d'erreur = 01 : Échec du contrôle d'admissionLa demande de réservation a été rejetée par le contrôle d'admission à cause de ressources indisponibles.

Pour ce Code d'erreur, les 16 bits du champ valeur d'erreur sont :

ssur cccc cccc cccc

où les bits sont :
ss = 00 : Les 12 bits de moindre poids contiennent un sous-code à définition mondiale (les valeurs figurent ci-dessous).
ss = 10 : Les 12 bits de moindre poids contiennent un sous-code spécifique d'une organisation. RSVP n'est pas supposé capable d'interpréter cela, sauf comme valeur numérique.
ss = 11 : Les 12 bits de moindre poids contiennent un sous-code spécifique du service. RSVP n'est pas supposé capable d'interpréter cela, sauf comme valeur numérique.
Comme le mécanisme de contrôle du trafic peut substituer un service différent, ce codage peut comporter des représentations du service utilisé.
u = 0 : RSVP rejette le message sans mettre à jour l'état local.
u = 1 : RSVP peut utiliser le message pour mettre à jour l'état local et transmettre le message. Cela signifie que le message est pour information.
r : Bit réservé, devrait être zéro.
cccc cccc cccc : code de 12 bits.
Les sous-codes à définition mondiale suivants peuvent apparaître dans les 12 bits de moindre poids lorsque ssur = 0000:
- sous-code = 1 : la limite de délai ne peut être respectée
- sous-code = 2 : la bande passante demandée est indisponible
- sous-code = 3 : la MTU dans la flowspec est plus grande que la MTU de l'interface.

o Code d'erreur = 02 : Échec de contrôle de politique
Le message de réservation ou de chemin a été rejeté pour des raisons administratives, par exemple, les accréditifs demandés ne sont pas fournis, quota ou équilibre insuffisant, ou préemption administrative. Ce code d'erreur peut apparaître dans un message PathErr ou ResvErr.

Le contenu du champ Valeur d'erreur sera déterminé ultérieurement.

o Code d'erreur = 03 : Pas d'information de chemin pour ce message Resv.

Pas d'état de chemin pour cette session. Le message Resv ne peut pas être transmis.

- Code d'erreur = 04 : Pas d'informations d'envoyeur pour ce message Resv.
  Il y a un état de chemin pour cette session, mais il ne comporte pas l'envoyeur correspondant à un descripteur de flux contenu dans le message Resv. Le message Resv ne peut pas être transmis.

- Code d'erreur = 05 : Conflit de style de réservation
  Le style de réservation est en conflit avec le ou les styles des états de réservation existants. Le champ Valeur d'erreur contient les 16 bits de moindre poids du vecteur d'options du style existant avec lequel le conflit est survenu. Ce message Resv ne peut pas être transmis.

- Code d'erreur = 06 : Style de réservation inconnu
  Le style de la réservation est inconnu. Ce message Resv ne peut pas être transmis.

- Code d'erreur = 07 : Conflit d'accès de destination
  Des sessions pour les mêmes adresses de destination et de protocole sont apparues avec des champs d'accès de destination à la fois à zéro et non zéro. Ce code d'erreur peut apparaître dans un message PathErr ou ResvErr.

- Code d'erreur = 08 : Conflit d'accès d'envoyeur
  L'accès d'envoyeur est à la fois à zéro et non zéro dans les messages Path pour la même session. Ce code d'erreur peut apparaître seulement dans un message PathErr.

- Code d'erreur = 09, 10, 11 : (réservé)

- Code d'erreur = 12 : Service préempté
  La demande de service définie par l'objet STYLE et le descripteur de flux a été préemptée administrativement.
  Pour ce code d'erreur, les 16 bits du champ Valeur d'erreur sont :

  ssur cccc cccc cccc

  Ici, les bits de poids fort ssur sont comme défini sous le code d'erreur 01. Les sous-codes à définition mondiale qui peuvent apparaître dans les 12 bits de moindre poids ssur = 0000 seront définis prochainement.

- Code d'erreur = 13 : Classe d'objet inconnue
  La valeur d'erreur contient une valeur de 16 bits composée de (Class-Num, C-Type) d'un objet inconnu. Cette erreur ne devrait être envoyée que si RSVP va rejeter le message, comme déterminé par les bits de poids fort de Class-Num. Ce code d'erreur peut apparaître dans un message PathErr ou ResvErr.

- Code d'erreur = 14 : Objet C-Type inconnu
  La valeur d'erreur contient une valeur de 16 bits composée du (Class-Num, C-Type) de l'objet.

- Code d'erreur = 15-19 : (réservé)

- Code d'erreur = 20 : Réservé pour l'API
  Le champ Valeur d'erreur contient un code d'erreur d'API, pour une erreur d'API qui a été détectée en asynchrone et doit être rapportée via un rappel.

- Code d'erreur = 21 : Erreur de contrôle de trafic
  L'appel de contrôle de trafic a échoué à cause du format ou du contenu des paramètres de la requête. Le message Resv ou Path qui a causé l'appel ne peut pas être transmis, et répéter l'appel serait inutile.
  Pour ce code d'erreur, les 16 bits du champ Valeur d'erreur sont :

  ss00 cccc cccc cccc

  Ici, les bits ss de poids fort sont comme défini sous le code d'erreur 01.
  Les sous-codes à définition mondiale suivants peuvent apparaître dans les 12 bits de moindre poids (cccc cccc cccc) lorsque ss = 00 :
  - Sous-code = 01 : conflit de service. Essaye de fusionner deux demandes de service incompatibles.
  - Sous-code = 02 : Service non pris en charge. Le contrôle de trafic ne peut fournir ni le service demandé ni un remplacement acceptable.
  - Sous-code = 03 : Mauvaise valeur de Flowspec. Demande mal formée ou déraisonnable.
  - Sous-code = 04 : Mauvaise valeur de Tspec. Demande mal formée ou déraisonnable.
  - Sous-code = 05 : Mauvaise valeur de Adspec. Demande mal formée ou déraisonnable.

- Code d'erreur = 22 : Erreur du système de contrôle de trafic
  Une erreur système a été détectée et rapportée par les modules de contrôle du trafic. La valeur d'erreur va contenir une valeur spécifique du système qui donne plus d'informations sur l'erreur. RSVP n'est pas supposé être capable

d'interpréter cette valeur.

- Code d'erreur = 23 : Erreur du système RSVP
  Le champ Valeur d'erreur va donner des informations dépendantes de la mise en œuvre sur l'erreur. RSVP n'est pas supposé être capable d'interpréter cette valeur.

En général, chaque message RSVP est reconstruit à chaque bond, et le nœud qui crée un message RSVP est responsable de sa construction correcte. De même, il est exigé de chaque nœud qu'il vérifie la construction correcte de chaque message RSVP qu'il reçoit. Si une erreur de programmation permettait à un RSVP de créer un message mal formé, l'erreur ne serait généralement pas rapportée aux systèmes d'extrémité dans un objet ERROR_SPEC ; à la place, l'erreur est simplement enregistrée localement, et peut-être rapportée par des mécanismes de gestion du réseau.

Les seules erreurs de formatage de message qui soient rapportées aux systèmes d'extrémité sont celles qui peuvent refléter des discordances de version, que le système pourrait être capable de circonvenir, par exemple, en revenant à un C-Type précédent pour un objet ; voir les codes 13 et 14 ci-dessus.

Le choix des erreurs de formatage de message que RSVP peut détecter et enregistrer localement est spécifique de la mise en œuvre, mais il va normalement inclure celles qui suivent :

- Mauvaise longueur de message : le champ Longueur RSVP ne correspond pas à la longueur du message.
- Version RSVP inconnue ou non prise en charge.
- Mauvaise somme de contrôle RSVP.
- Défaillance de INTEGRITY.
- Type de message RSVP illégal.
- Longueur d'objet illégale : n'est pas un multiple de 4, ou est inférieure à 4.
- L'adresse du prochain bond/bond précédent dans l'objet HOP est illégale.
- Mauvais accès de source : l'accès de source est différent de zéro dans une spec de filtre ou un gabarit d'envoyeur pour une session dont l'accès de destination est zéro.
- Il manque la classe d'objet (spécifier) exigée.
- Classe d'objet illégale (spécifier) dans ce type de message.
- Violation de l'ordre d'objets exigé.
- Mauvais compte de descripteur de flux pour le style ou le type de message
- Traitement d'interface logique invalide.
- Objet Class-Num invalide.
- Adresse de destination du message ResvConf ne correspondant pas à l'adresse de receveur dans l'objet RESV_CONFIRM qu'elle contient.

Appendice C Encapsulation UDP

Une mise en œuvre RSVP va généralement exiger la capacité d'effectuer des entrées/sorties "brutes" de réseau, c'est-à-dire, d'envoyer et recevoir des datagrammes IP en utilisant le protocole 46. Cependant, certaines importantes classes de systèmes d'hôtes peuvent ne pas prendre en charge l'entrée/sortie brute de réseau. Pour utiliser RSVP, de tels hôtes doivent encapsuler les messages RSVP dans UDP.

Le schéma de base d'encapsulation UDP fait deux hypothèses :

1. Tous les hôtes sont capables d'envoyer et recevoir des paquets de diffusion groupée si les destinations de diffusion groupée doivent être prises en charge.

2. Les routeurs de premier/dernier bond sont à capacité RSVP.

Une méthode pour assouplir la seconde hypothèse sera donnée plus loin.

Soit Hu un hôte "UDP seul" qui exige l'encapsulation UDP, et Hr un hôte qui peut faire des entrées/sorties réseau brutes. Le schéma d'encapsulation UDP doit permettre l'interopération de RSVP parmi une topologie arbitraire d'hôtes Hr, d'hôtes Hu, et de routeurs.

Des messages Resv, ResvErr, ResvTear, et PathErr sont envoyés à des adresses d'envoi individuel apprises du chemin ou de l'état de réservation dans le nœud. Si le nœud garde trace des bonds précédents et des interfaces qui ont besoin de l'encapsulation UDP, ces messages peuvent être envoyés en utilisant l'encapsulation UDP lorsque elle est nécessaire. D'autre part, les messages Path et PathTear sont envoyés à l'adresse de destination pour la session, qui peut être en envoi individuel ou en diffusion groupée.

Les tableaux des Figures 13 et 14 montrent les règles de base de l'encapsulation UDP des messages Path et PathTear, respectivement pour une DestAddress en envoi individuel et une DestAddress en diffusion groupée. Les autres types de message, qui sont en envoi individuel, devraient suivre les règles de l'envoi individuel de la Figure 13. Dans ces figures, sous la colonne "Envoi RSVP", la notation est "mode(destaddr, destport)" ; destport est omis pour les paquets bruts. La colonne "Reçoit" montre le groupe qui est rejoint et, le cas échéant, l'accès d'écoute UDP.

Il est utile de définir deux sortes d'encapsulation UDP, une qui est envoyée par Hu et l'autre qui est envoyée par Hr et R, pour éviter un double traitement par le receveur. En pratique, ces deux types sont distingués par des numéros d'accès UDP différents Pu et Pu'.

Les symboles suivants sont utilisés dans les tableaux :
- o D est l'adresse de destination pour cette session particulière,
- o G* est une adresse de groupe bien connue de la forme 224.0.0.14, c'est-à-dire, un groupe qui est limité au réseau à connexion locale.
- o Pu et Pu' sont deux accès UDP bien connus pour l'encapsulation UDP de RSVP, avec les valeurs 1698 et 1699.
- o Ra est l'adresse IP de l'interface de routeur "a".
- o L'interface de routeur "a" est sur le réseau local connecté à Hu et Hr.

Les notes suivantes s'appliquent à ces figures :

[Note 1] Hu envoie un message Path en envoi individuel à l'adresse de destination D, si D est local, ou à l'adresse Ra du routeur de premier bond. Ra est présumé connu de l'hôte.
[Note 2] Ici, D est l'adresse de l'interface locale à travers laquelle le message est arrivé.
[Note 3] Cela suppose que l'application a rejoint le groupe D.

Destination d'envoi individuel D :

Nœud	RSVP envoie	RSVP reçoit
Hu	UDP(D/Ra,Pu) [Note 1]	UDP(D,Pu) et UDP(D,Pu') [Note 2]
Hr	Raw(D) et si (UDP) alors UDP(D,Pu')	Raw() et UDP(D, Pu) [Note 2] (Ignore Pu')
R (Interface a) :	Raw(D) et si (UDP) alors UDP(D,Pu')	Raw() et UDP(Ra, Pu) (Ignore Pu')

Figure 13 : Règles d'encapsulation UDP pour les messages Path et Resv en envoi individuel

Destination de diffusion groupée D :

Nœud	RSVP envoie	RSVP reçoit
Hu	UDP(G*,Pu)	UDP(D,Pu') [Note 3] et UDP(G*,Pu)
Hr	Raw(D,Tr) et si (UDP) alors UDP(D,Pu')	Raw() et UDP(G*,Pu) (Ignore Pu')
R (Interface a) :	Raw(D,Tr) et si (UDP) alors UDP(D,Pu')	Raw() et UDP(G*,Pu) (Ignore Pu')

Figure 14 : Règles UDP d'encapsulation pour les messages Path en diffusion groupée

Un routeur peut déterminer si son interface X a besoin de l'encapsulation UDP en écoutant les messages Path encapsulés dans UDP qui ont été envoyés à G* (D en diffusion groupée) ou à l'adresse de l'interface X (D en diffusion groupée). Il y a un mode défaillant pour ce schéma : si aucun hôte sur le réseau connecté n'agit comme envoyeur RSVP, il n'y aura pas de message Path pour déclencher l'encapsulation UDP. Dans ce cas (peu vraisemblable) il sera nécessaire de configurer explicitement l'encapsulation UDP sur l'interface réseau locale du routeur.

Lorsque un paquet encapsulé UDP est reçu, la TTL IP n'est pas disponible à l'application sur la plupart des systèmes. Le processus RSVP qui reçoit un message Path ou PathTear encapsulé dans UDP devrait donc utiliser le champ Send_TTL de l'en-tête commun RSVP comme TTL effective de réception. Cela peut être outrepassé par configuration manuelle.

Nous avons supposé que le routeur de premier bond à capacité RSVP R est sur le réseau directement connecté. Plusieurs approches sont possibles si ce n'est pas le cas.

1. Hu peut envoyer des sessions en envoi individuel aussi bien qu'en diffusion groupée à UDP(Ra,Pu) avec TTL=Ta

   Ici, Ta doit être la TTL pour atteindre exactement R. Si Ta est trop petit, le message Path n'atteindra pas R. Si Ta est trop grand, R et les routeurs successifs peuvent transmettre le paquet UDP jusqu'à ce que son compte de dernier bond arrive à expiration. Cela va déclencher l'encapsulation UDP entre les routeurs au sein de l'Internet, causant peut-être du trafic UDP erroné. L'hôte Hu doit être explicitement configuré avec Ra et Ta.

2. Un hôte particulier sur le LAN connecté à Hu pourrait être désigné comme "hôte relais RSVP". Un hôte relais

écouterait sur (G*,Pu) et transmettrait tous les messages Path directement à R, bien qu'il ne soit pas sur le chemin des données. L'hôte relais devrait être configuré avec Ra et Ta.

Appendice D Glossaire

- Contrôle d'admission
  C'est une fonction de contrôle du trafic qui décide si le programmeur de paquets dans les nœuds peut fournir la qualité de service demandée tout en continuant de fournir la qualité de service exigée par les demandes précédemment admises. Voir aussi "contrôle de politique" et "contrôle du trafic".

- Adspec
  Une Adspec est un élément de données (objet) dans un message Path qui porte un paquetage d'OPWA annonçant des informations. Voir "OPWA".

- Boucle d'autorafraîchissement
  Une boucle d'autorafraîchissement est une condition d'erreur qui survient lorsque une boucle topologique de routeurs continue de rafraîchir l'état de réservation existant bien que tous les receveurs aient arrêté de demander ces réservations. Voir d'autres informations au paragraphe 3.4.

- État de blocage
  L'état de blocage aide à résoudre un problème de "réservation qui tue". Voir les paragraphes 2.5 et 3.5, et "réservation qui tue".

- Politique d'embranchement
  Régulation du trafic à un point d'embranchement de diffusion groupée sur une interface sortante qui a "moins" de ressources réservées qu'une autre interface sortante pour le même flux. Voir "régulation du trafic".

- C-Type
  Type de classe d'un objet ; unique au sein d'un nom de classe. Voir "nom de classe".

- Nom de classe
  Classe d'un objet. Voir "objet".

- DestAddress
  Adresse de destination IP ; partie de l'identification d'une session. Voir "session".

- Style distinct
  Attribut de style (d'une réservation) ; des ressources distinctes sont réservées pour chaque envoyeur différent. Voir aussi "style partagé".

- Vers l'aval
  Vers le ou les receveurs de données.

- DstPort
  Accès de destination IP (généralisé) utilisé au titre d'une session. Voir "accès de destination généralisé".

- Régulation d'entrée
  Régulation du trafic faite au premier routeur à capacité RSVP (et à capacité de régulation) sur un chemin de données.

- ERROR_SPEC
  Objet qui porte le rapport d'erreur dans un message PathErr ou ResvErr.

- Sélection explicite de l'envoyeur
  Attribut de style (de réservation) ; tous les envoyeurs réservés sont énumérés explicitement dans le message de réservation. Voir aussi "sélection générique d'envoyeur".

- Style FF
  Style de réservation à filtre fixe, qui a les attributs Sélection explicite de l'envoyeur et Distinct.

- FilterSpec
  Avec les informations de session, il définit l'ensemble des paquets de données qui reçoivent la QS spécifiée dans une

flowspec. La filterspec est utilisée pour régler les paramètres dans la fonction de classeur de paquet. Une filterspec peut être portée dans un objet FILTER_SPEC ou SENDER_TEMPLATE.

- Descripteur de flux
  Combinaison d'une flowspec et d'une filterspec.

- Flowspec
  Définit la QS à fournir à un flux. La flowspec est utilisée pour régler les paramètres dans la fonction de programmation de paquet pour fournir la qualité de service demandée. Une flowspec est portée dans un objet FLOWSPEC. Le format de flowspec est opaque pour RSVP et est défini par le groupe de travail Services intégrés.

- Accès de destination généralisé
  Composant d'une définition de session qui fournit le démultiplexage de couche de protocole transport ou application au delà de la DestAddress. Voir "session".

- Accès de source généralisé
  Composant d'une spec de filtre qui fournit le démultiplexage de couche de protocole transport ou application au delà de l'adresse de l'envoyeur.

- GLB (Greatest Lower Bound)
  Limite inférieure supérieure

- Interface entrante
  L'interface sur laquelle les paquets de données sont supposés arriver, et sur laquelle sont envoyés les messages Resv.

- INTEGRITY
  Objet d'un message de contrôle RSVP qui contient les données cryptographiques pour authentifier le nœud générateur et pour vérifier le contenu d'un message RSVP.

- Problème de la réservation qui tue
  Le problème de la réservation qui tue décrit un cas où un receveur qui tente vainement de faire une grande réservation de QS empêche l'établissement de plus petites réservations de QS. Voir les précisions aux paragraphes 2.5 et 3.5.

- LIH (Logical Interface Handle)
  Le traitement logique d'interface est utilisé pour faciliter le traitement des nuages non RSVP. Voir les précisions au paragraphe 2.9.

- Réparation locale
  Permet à RSVP d'adapter rapidement ses réservations aux changements d'acheminement. Voir les précisions au paragraphe 3.6.

- LPM (Local Policy Module)
  Module de politique locale, c'est la fonction qui exerce le contrôle de politique.

- LUB (Least Upper Bound)
  Limite supérieure inférieure.

- Politique de fusion
  Régulation du trafic qui a lieu au point de fusion des données pour une réservation partagée.

- Fusion
  C'est le processus qui prend le maximum (ou plus généralement la limite supérieure inférieure) des réservations qui arrivent sur les interfaces sortantes, et transmet ce maximum sur l'interface entrante. Voir les précisions au paragraphe 2.2.

- MTU (Maximum Transmission Unit)
  Unité de transmission maximum.

- Prochain bond
  Le routeur suivant dans la direction du flux de trafic.

- NHOP
  Objet qui porte les informations de prochain bond dans les messages de contrôle RSVP.

- Nœud
  Un routeur ou un système hôte.

- Nuage non RSVP
  Groupe d'hôtes et routeurs qui ne fonctionnent pas avec RSVP. Le traitement des nœuds qui ne prennent pas en charge RSVP est important pour la rétro compatibilité. Voir au paragraphe 2.9.

- Objet
  Un élément d'un message de contrôle RSVP ; c'est un triplet type, longueur, valeur.

- OPWA (One Pass With Advertising)
  Un passage avec annonce. Décrit un modèle d'établissement de réservation dans lequel les messages (Path) envoyés vers l'aval rassemblent des informations que le ou les receveurs peuvent utiliser pour prédire le service de bout en bout. Les informations rassemblées sont appelées une annonce. Voir aussi "Adspec".

- Interface sortante
  Interface à travers laquelle sont transmis les paquets de données et messages Path.

- Classeur de paquets
  Fonction de contrôle du trafic dans le principal chemin de transmission des paquets qui choisit une classe de service pour chaque paquet, conformément à l'état de réservation établi par RSVP. Le classeur de paquet peut être combiné à la fonction d'acheminement. Voir aussi "contrôle du trafic".

- Programmeur de paquet
  Fonction de contrôle du trafic dans le principal chemin de transmission des paquets de données qui met en œuvre la QS pour chaque flux, en utilisant un des modèles de service définis par le groupe de travail Services intégrés. Voir aussi "contrôle de trafic".

- État de chemin
  Informations conservées dans les routeurs et les hôtes sur tous les envoyeurs RSVP.

- PathErr
  Message de contrôle RSVP indiquant une erreur de chemin.

- PathTear
  Message de contrôle RSVP ordonnant la suppression du chemin.

- PHOP
  Objet qui porte les informations sur le bond précédent dans les message de contrôle RSVP.

- Régulation
  Voir à régulation du trafic.

- Contrôle de politique
  Fonction qui détermine si une nouvelle demande de qualité de service a la permission administrative de faire la réservation demandée. Le contrôle de politique peut aussi effectuer la comptabilité (retour sur l'utilisation) pour une réservation.

- Données de politique
  Données portées dans un message Path ou Resv et utilisées comme entrées de contrôle de politique pour déterminer l'autorisation et/ou le retour sur l'utilisation pour le flux en question.

- Bond précédent
  Le routeur précédent dans la direction du flux de trafic. Les messages Resv s'écoulent vers les bonds précédents.

- ProtocolId
  Le composant d'identification de session qui spécifie le numéro de protocole IP utilisé par le flux de données.

- QS
  Qualité de service.

- État de réservation

Informations conservées dans les nœuds à capacité RSVP sur les demandes de réservation RSVP réussies.

- Style de réservation
  Décrit un ensemble d'attributs pour une réservation, y compris les attributs de partage et les attributs de sélection de l'envoyeur. Voir les détails au paragraphe 1.3.

- Message ResvMessage de contrôle RSVP de demande de réservation.

- ResvConf
  Message de contrôle RSVP de confirmation de réservation qui confirme la réussite de l'installation d'une réservation à un nœud en amont.

- ResvErr
  Message de contrôle RSVP d'erreur de réservation qui indique qu'une demande de réservation a échoué ou qu'une réservation active a été préemptée.

- ResvTear
  Message de contrôle RSVP de suppression de réservation qui supprime l'état de réservation.

- Rspec
  Composant d'une flowspec qui définit une QS désirée. Le format de Rspec est opaque pour RSVP et est défini par le groupe de travail Integrated Services de l'IETF.

- RSVP_HOP
  Objet d'un message de contrôle RSVP qui porte l'adresse de PHOP ou NHOP de la source du message.

- Portée
  L'ensemble des hôtes envoyeurs auxquels une demande de réservation donnée est à propager.

- Style SE
  Style de réservation à partage explicite, qui a les attributs Sélection explicite de l'envoyeur et l'attribut Partage.

- Fragmentation sémantique
  Méthode de fragmentation d'un grand message RSVP en utilisant les informations sur la structure et le contenu du message, de sorte que chaque fragment soit un message RSVP logiquement complet.

- Gabarit d'envoyeur
  Paramètre dans un message Path qui définit un envoyeur ; porté dans un objet SENDER_TEMPLATE. Il a la forme d'une spec de filtre qui peut être utilisée pour sélectionner les paquets de cet envoyeur parmi les autres paquets dans la même session sur la même liaison.

- Tspec d'envoyeur
  Paramètre dans un message Path, c'est une Tspec qui caractérise les paramètres de trafic pour le flux de données provenant de l'envoyeur correspondant. Il est porté par un objet SENDER_TSPEC.

- Session
  Une session RSVP définit un flux de données unidirectionnel en envoi individuel ou en diffusion groupée pour lequel des réservations sont demandées. Une session est identifiée par l'adresse de destination, le protocole de couche transport, et un accès de destination (généralisé) facultatif.

- Style partagé
  Attribut de style (de réservation) : tous les envoyeurs réservés partagent les mêmes ressources réservées. Voir aussi "style distinct".

- État mou *(Soft state)*
  État de contrôle chez les hôtes et les routeurs qui va arriver à expiration si il n'est pas rafraîchi dans un délai spécifié.

- STYLE
  Objet d'un message RSVP qui spécifie le style de réservation désiré.

- Style
  Voir "style de réservation"

- TIME_VALUES
  Objet dans un message de contrôle RSVP qui spécifié la durée utilisée pour rafraîchir l'état dans ce message.

- Contrôle de trafic
  Ensemble entier de la mécanique d'un nœud qui fournit la QS demandée aux flux de données. Le contrôle du trafic comporte les fonctions de classeur de paquet, de programmeur de paquet, et de contrôle d'admission.

- Régulation du trafic
  C'est la fonction, effectuée par le contrôle de trafic, qui force un flux de données donné à la conformité avec les paramètres de trafic impliqués par la réservation. Elle peut impliquer, par exemple, l'abandon des paquets non conformes ou leur envoi avec une priorité inférieure.

- TSpec
  Ensemble de paramètres de trafic qui décrivent un flux. Le format d'une Tspec est opaque pour RSVP et est défini par le groupe de travail Intégration de service.

- Encapsulation UDP
  Façon pour les hôtes qui ne peuvent pas utiliser les prises brutes de participer à RSVP en encapsulant les paquets (bruts) de protocole RSVP dans des paquets UDP ordinaires. Voir des précisions à l'Appendice C.

- Vers l'amont
  Vers la source du trafic. Les messages RSVP Resv s'écoulent vers l'amont.

- Style WF
  Style de réservation à filtre générique, qui a les attributs Sélection générique d'envoyeur et Partagé.

- Sélection générique d'envoyeur
  Attribut de style (de réservation) : le trafic provenant de tout envoyeur à une session spécifique reçoit la même QS. Voir aussi "sélection explicite d'envoyeur".

Références

[FJ94] Floyd, S. et V. Jacobson, "Synchronization of Periodic Routing Messages", IEEE/ACM Transactions on Networking, Vol. 2, n° 2, avril 1994.

[OPWA95] Shenker, S. et L. Breslau, "Two Issues in Reservation Establishment", Proc. ACM SIGCOMM '95, Cambridge, MA, août 1995.

[PolArch96] Herzog, S., "Policy Control for RSVP: Architectural Overview". Travail en cours.

[RFC1633] R. Braden, D. Clark et S. Shenker, "Intégration de services dans l'architecture l'Internet : généralités", juin 1994. *(Info.)*

[RFC2113] D. Katz, "Option d'alerte de routeur IP", février 1997. (*MàJ par* RFC5350, RFC6398) *(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.)*

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

[RSVP93] Zhang, L., Deering, S., Estrin, D., Shenker, S., et D. Zappala, "RSVP: A New Resource ReSerVation Protocol", IEEE Network, septembre 1993.

Considérations pour la sécurité

Voir au paragraphe 2.8.

Adresse des auteurs

Bob Braden
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
téléphone : (310) 822-1511
mél : Braden@ISI.EDU

Lixia Zhang
UCLA Computer Science Department
4531G Boelter Hall
Los Angeles, CA 90095-1596 USA
téléphone : 310-825-2695
mél : lixia@cs.ucla.edu

Steve Berson
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
téléphone : (310) 822-1511
mél : Berson@ISI.EDU

Shai Herzog
IBM T. J. Watson Research Center
P.O Box 704
Yorktown Heights, NY 10598
téléphone : (914) 784-6059
mél : Herzog@WATSON.IBM.COM

Sugih Jamin
University of Michigan
CSE/EECS
1301 Beal Ave.
Ann Arbor, MI 48109-2122
téléphone : (313) 763-1583
mél : jamin@EECS.UMICH.EDU