

Groupe de travail Réseau
Request for Comments : 2961
 Catégorie : En cours de normalisation

L. Berger, LabN Consulting, LLC
 D. Gan, Juniper Networks, Inc.
 G. Swallow, Cisco Systems, Inc.
 P. Pan, Juniper Networks, Inc.
 F. Tommasi & S. Molendini, University of Lecce
 avril 2001

Traduction Calude Brière de L'Isle

Extensions de réduction de redondance de rafraîchissement pour RSVP

Statut du présent mémoire

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

Notice de Copyright

Copyright (C) The Internet Society (2001). Tous droits réservés.

Résumé

Le présent document décrit un certain nombre de mécanismes qui pourraient être utilisés pour réduire le traitement des exigences de redondance des messages de rafraîchissement, éliminant la latence de synchronisation d'état subie lorsque un message RSVP (Protocole de réservation de ressource) est perdu, et quand désiré, rafraîchissant l'état sans transmission de tous les messages de rafraîchissement. La même extension prend aussi en charge la livraison fiable de message RSVP sur la base du bond. Cette extension ne présente pas de problème de rétrocompatibilité.

Table des Matières

1. Introduction et fondements.....	2
1.1 Messages Trigger et Refresh.....	2
2. Bit de capacité de réduction de rafraîchissement.....	3
3. Message RSVP Bundle.....	3
3.1 En-tête Bundle.....	3
3.2 Formats de message.....	4
3.3 Envoi des messages RSVP Bundle.....	4
3.4 Réception des messages RSVP Bundle.....	5
4. Extension MESSAGE_ID.....	5
4.1 Modification des formats de message standard.....	6
4.2 Objets MESSAGE_ID.....	6
4.3 Objets MESSAGE_ID_ACK et MESSAGE_ID_NACK.....	6
4.4 Format du message ACK.....	7
4.5 Usage de l'objet MESSAGE_ID.....	7
4.6 Objet MESSAGE_ID_ACK et usage de l'objet MESSAGE_ID_NACK.....	8
4.7 Considérations pour la diffusion groupée.....	9
4.8. Compatibilité.....	9
5. Extension Summary Refresh.....	10
5.1 Objets MESSAGE_ID_LIST, SRC_LIST et MCAST_LIST.....	10
5.2 Format du message Srefresh.....	14
5.3 Usage du message Srefresh.....	14
5.4 NACK de Srefresh.....	16
5.5 Préservation de l'état conditionnel RSVP.....	16
5.6 Compatibilité.....	17
6. Procédures de retard exponentiel.....	17
6.1 Fonctionnement général.....	17
6.2 Paramètres de temps.....	17
6.3 Algorithme de retransmission.....	18
6.4 Considérations sur les performances.....	18
7. Remerciements.....	18
8. Considérations pour la sécurité.....	18
9. Références.....	18
10. Adresses des auteurs.....	19
11. Déclaration complète de droits de reproduction.....	19

1. Introduction et fondements

Le protocole RSVP standard [RFC2205] maintient l'état via la génération de messages de rafraîchissement RSVP. Les messages Refresh sont utilisés à la fois pour synchroniser l'état entre les voisins RSVP et pour récupérer des messages RSVP perdus. L'utilisation de messages Refresh pour couvrir de nombreuses défaillances possibles a eu pour conséquence un certain nombre de problèmes de fonctionnement. Un des problèmes se rapporte à l'échelonnement, un autre se rapporte à la fiabilité et à la latence de la signalisation RSVP.

Les problèmes d'échelonnement sont liés aux exigences de ressources (en termes de traitement et de mémoire) pour faire fonctionner RSVP. Les exigences de ressources augmentent proportionnellement au nombre de sessions. Chaque session exige de générer, transmettre, recevoir et traiter les messages RSVP Path et Resv à chaque période de rafraîchissement. La prise en charge d'un grand nombre de sessions, et le volume correspondant de messages de rafraîchissement, pose un problème d'échelle.

Le problème de la fiabilité et de la latence survient lorsque un message RSVP autre que de rafraîchissement est perdu dans la transmission. Le protocole RSVP standard [RFC2205] récupère de la perte d'un message via les messages de rafraîchissement RSVP. Face à la perte de messages RSVP dans la transmission, la latence de bout en bout de la signalisation RSVP est liée à l'intervalle de rafraîchissement du ou des nœuds qui subissent la perte. Lorsque la signalisation de bout en bout est limitée par l'intervalle de rafraîchissement, le délai subi dans l'établissement ou le changement d'une réservation peut sortir de la gamme de ce qui est acceptable par certaines applications.

Une façon de régler ce problème du volume de rafraîchissement est d'augmenter la période de rafraîchissement, "R" comme défini au paragraphe 3.7 de la [RFC2205]. Augmenter la valeur de R donne une amélioration linéaire de la redondance de transmission, mais au prix de l'accroissement du temps mis à synchroniser l'état.

Une façon de traiter la fiabilité et la latence de la signalisation RSVP est de diminuer la période de rafraîchissement R. Diminuer la valeur de R augmente la probabilité que l'état soit installé face à une perte de message, mais au prix de l'augmentation du taux de message de rafraîchissement et des exigences de traitement associées.

Un problème supplémentaire est le temps de désallocation des ressources après la perte d'un message de suppression. RSVP ne retransmet pas les messages ResvTear ou PathTear. Si le seul message de suppression transmis est perdu, les ressources ne seront désallouées qu'une fois passé l'intervalle du "temporisateur de nettoyage". Il peut en résulter que les ressources sont allouées pendant une période de temps inutile. Noter que même si la période de rafraîchissement est ajustée, le "temporisateur de nettoyage" doit quand même arriver à expiration car les messages de suppression ne sont pas retransmis.

Les extensions définies dans le présent document traitent à la fois le problème du volume de rafraîchissement et celui de la fiabilité par des mécanismes autres que le réglage du taux de rafraîchissement. Les extensions sont collectivement désignées sous le nom de "Réduction de la redondance de rafraîchissement" ou d'extensions de "réduction de rafraîchissement". Un message Bundle (*regroupement*) est défini pour réduire la charge globale de traitement de message. Un objet MESSAGE_ID est défini pour réduire le traitement du message de rafraîchissement en permettant au receveur d'identifier plus directement un message inchangé. Un objet MESSAGE_ACK est défini pour être utilisé à détecter la perte de message et prendre en charge la livraison fiable de message RSVP sur la base du bond. Un résumé de message de rafraîchissement est défini pour activer l'état de rafraîchissement sans la transmission de messages de rafraîchissement entiers, tout en maintenant la capacité de RSVP à indiquer quand l'état est perdu et s'ajuster aux changements de l'acheminement.

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

1.1 Messages Trigger et Refresh

Le présent document classe les messages RSVP en deux types : messages de déclenchement, et messages de rafraîchissement. Les messages de déclenchement sont ceux qui annoncent l'état RSVP ou toutes autres informations non transmises précédemment. Les messages de déclenchement incluent les messages qui annoncent un nouvel état, un changement de chemin qui altère un chemin réservé, ou une modification à une session ou réservation RSVP existante. Les messages de déclenchement incluent aussi les messages qui comportent des changements dans des objets traités qui ne sont pas RSVP, tels que des changements dans les objets Policy ou ADSPEC.

Les messages de rafraîchissement représentent des états annoncés précédemment et contiennent exactement les mêmes objets et les mêmes informations qu'un message transmis antérieurement, et ils sont envoyés sur le même chemin. Seuls les messages Path et Resv peuvent être des messages de rafraîchissement. Les messages de rafraîchissement peuvent être identiques au message précédemment transmis correspondant, avec quelques exceptions possibles. Précisément, le champ Somme de contrôle, le champ Fanions et l'objet INTEGRITY peuvent différer dans les messages de rafraîchissement.

2. Bit de capacité de réduction de rafraîchissement

Pour indiquer la prise en charge des extensions de réduction de la redondance de rafraîchissement, un bit de capacité supplémentaire est ajouté à l'en-tête RSVP commun, qui est défini dans la [RFC2205].

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|Fanions|Type de Message| Somme de contrôle RSVP |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TTL_d'envoi | (Réservé) | Longueur RSVP |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Fanions : 4 bits

0x01 : Capacité de réduction de (redondance de) rafraîchissement

Lorsque il est établi, il indique que ce nœud veut et est capable de recevoir tous les messages et objets décrits dans le présent document. Cela inclut le message Bundle décrit à la Section 3, l'objet MESSAGE_ID et les messages Ack décrits à la Section 4, et les objets MESSAGE_ID LIST et les messages Srefresh décrits à la Section 5. Ce bit n'a de signification qu'entre voisins RSVP.

Les nœuds qui acceptent les extensions de réduction de redondance de rafraîchissement doivent aussi veiller à reconnaître quand un prochain bond arrête d'envoyer des messages RSVP avec le bit Capacité de réduction de rafraîchissement établi. Pour traiter ce cas, les nœud qui prennent en charge les extensions de réduction de redondance de rafraîchissement DOIVENT examiner le champ Fanions de chaque message RSVP reçu. Si le fanion change de l'indication de la prise en charge à l'indication de la non prise en charge, alors, sauf configuration contraire, les messages Srefresh (décrits à la Section 5) NE DOIVENT PAS être utilisés pour les rafraîchissement d'état suivants vers ce voisin et les messages Bundle (Section 3) NE DOIVENT PAS être envoyés à ce voisin. Noter qu'un nœud qui prend en charge la livraison fiable de message RSVP (Section 4) mais pas les messages Bundle et Srefresh, ne va pas établir le bit Capacité-de-réduction-de-rafraîchissement.

3. Message RSVP Bundle

Un message RSVP Bundle consiste en un en-tête Bundle suivi d'un corps comportant un nombre variable de messages RSVP standard. Un message Bundle est utilisé pour agréger plusieurs messages RSVP au sein d'une seule PDU. Le terme "bundle" est utilisé pour éviter des confusions avec l'agrégation de réservation RSVP. Les paragraphes qui suivent définissent les formats de l'en-tête Bundle et les règles d'inclusion des messages RSVP standard au titre du message.

3.1 En-tête Bundle

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|Fanions|Type de message| Somme de contrôle RSVP |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TTL d'envoi | (Réservé) | Longueur RSVP |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Le format de l'en-tête Bundle est identique au format de l'en-tête commun RSVP [RFC2205]. Les champs de l'en-tête sont les suivants :

Version : 4 bits

Numéro de version du protocole. Ceci est la version 1.

Fanions : 4 bits

0x01 : Capacité de réduction (de redondance) de rafraîchissementd). Voir la Section 2.

0x02-0x08 : Réserve

Type de message : 8 bits

12 = Bundle

Somme de contrôle RSVP : 16 bits

C'est le complément à un de la somme des compléments à un du message entier, 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. Comme les sous messages individuels peuvent porter leur propre somme de contrôle ainsi que l'objet INTEGRITY pour l'authentification, ce champ PEUT être mis à zéro. Noter que lorsque la somme de contrôle n'est pas calculée, l'en-tête du message Bundle ne sera pas couverte par une somme de contrôle. Si la somme de contrôle est calculée, les sous messages individuels PEUVENT régler leur propre somme de contrôle à zéro.

TTL d'envoi : 8 bits

C'est la valeur du TTL IP avec laquelle le message a été envoyé. C'est utilisé par RSVP pour détecter un bond non RSVP en comparant le TTL d'envoi avec le TTL IP dans un message reçu.

Longueur RSVP : 16 bits

C'est la longueur totale de ce message RSVP Bundle en octets, y compris l'en-tête Bundle et les sous messages qui suivent.

3.2 Formats de message

Un message RSVP Bundle doit contenir au moins un sous message. Un sous message PEUT être de tout type de message sauf un autre message Bundle.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|Fanions|Type de message|  Somme de contrôle RSVP  |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TTL d'envoi | (Réserve) |  Longueur RSVP  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|
//                Premier sous message                //
|
+-----+-----+-----+-----+-----+-----+-----+-----+
|
//                Autres sous messages..                //
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

3.3 Envoi des messages RSVP Bundle

La prise en charge des messages RSVP Bundle est facultative. Bien que le message Bundle aide à la mise à l'échelle de RSVP, en réduisant la redondance du traitement et la consommation de bande passante, un nœud n'est pas obligé de transmettre chaque message RSVP standard dans un message Bundle. Un nœud DOIT toujours être prêt à recevoir les messages RSVP standard.

Les messages RSVP Bundle peuvent seulement être envoyés au voisin RSVP qui accepte le regroupement. Les méthodes pour découvrir de telles informations incluent : (1) la configuration manuelle et (2) l'observation du bit Capacité de réduction de rafraîchissement (voir la Section 2) dans les messages RSVP reçus. Les messages RSVP Bundle NE DOIVENT PAS être utilisés si le voisin RSVP ne prend pas en charge les messages RSVP Bundle.

Les messages RSVP Bundle sont envoyés bond par bond entre les nœuds à capacité RSVP comme des datagrammes IP "bruts" avec le numéro de protocole 46. L'adresse IP de source est une adresse locale du système qui a généré le message Bundle. L'adresse de destination IP est celle du voisin RSVP auquel sont destinés les sous-messages.

Les messages RSVP Bundle NE DEVRAIENT PAS être envoyés avec l'option IP d'alerte de routeur dans leurs en-têtes IP. Cela parce que les messages Bundle sont adressés directement aux voisins RSVP.

Chaque message RSVP Bundle DOIT occuper exactement un datagramme IP, qui fait approximativement 64 k octets. Si il excède la MTU, le datagramme est fragmenté par IP et réassemblé au nœud receveur. Les mises en œuvre peuvent choisir de limiter chaque message RSVP Bundle à la taille de la MTU de la liaison sortante, par exemple, 1500 octets. Les mises en œuvre DEVRAIENT aussi limiter la durée pendant laquelle un message est retardé afin d'être regroupé. Des limites différentes peuvent être utilisées pour les messages de déclenchement et de rafraîchissement standard. Les messages de déclenchement DEVRAIENT être retardés d'une durée minimale. Les messages de rafraîchissement peuvent être retardés d'une durée qui peut aller jusqu'à leur intervalle de rafraîchissement. Noter que les messages qui se rapportent au même état Resv ou Path ne devraient pas être retardés à des intervalles différents afin de préserver leur ordre.

Si le voisin RSVP n'est pas connu ou si des changements dans les prochains bonds ne peuvent pas être identifiés via l'acheminement, les messages Bundle NE DOIVENT PAS être utilisés. Noter que lorsque le prochain bond d'acheminement n'est pas à capacité RSVP, il ne sera normalement pas possible d'identifier les changements dans le prochain bond.

Tout message qui va être traité par le voisin RSVP indiqué dans l'adresse de destination d'un message Bundle peut être inclus dans le même message. Cela inclut tous les messages RSVP qui vont être envoyés sur une liaison point à point. Cela inclut tout message, comme un Resv, adressé à la même adresse de destination. Cela inclut aussi les messages Path et PathTear lorsque le prochain bond est connu comme étant la destination et que les changements dans les prochains bonds peuvent être détectés. Les messages Path et PathTear pour les sessions en diffusion groupée NE DOIVENT PAS être envoyés dans les messages Bundle lorsque la liaison sortante n'est pas en point à point ou lorsque les prochains bonds ne prennent pas en charge les extensions de réduction de redondance de rafraîchissement.

3.4 Réception des messages RSVP Bundle

Si le système local ne reconnaît pas ou ne souhaite pas accepter un message Bundle, les messages reçus devront être éliminés sans autre analyse.

Le receveur compare ensuite le TTL d'envoi à celui avec lequel un message Bundle est envoyé au TTL IP avec lequel il est reçu. Si un bond non RSVP est détecté, le nombre de bonds non RSVP est enregistré. Il est utilisé plus tard dans le traitement des sous-messages.

Ensuite, le receveur vérifie le numéro de version et la somme de contrôle du message RSVP Bundle et élimine le message si une discordance est trouvée.

Le receveur commence alors à désencapsuler les sous-messages individuels. Chaque sous-message a sa propre longueur complète de message et ses informations d'authentification. À l'exception de l'utilisation du TTL d'envoi provenant de l'en-tête du message Bundle, chaque sous-message est traité comme si il avait été reçu individuellement.

4. Extension MESSAGE_ID

Trois nouveaux objets sont définis au titre de l'extension MESSAGE_ID. Les objets sont MESSAGE_ID, MESSAGE_ID_ACK, et MESSAGE_ID_NACK. Les deux premiers objets sont utilisés pour prendre en charge les accusés de réception et la livraison fiable de message RSVP. Le dernier objet est utilisé pour prendre en charge l'extension de résumé de rafraîchissement décrit à la Section 5. L'objet MESSAGE_ID peut aussi être utilisé pour simplement fournir une indication abrégée de quand le message qui porte l'objet est un message de rafraîchissement. De telles informations peuvent être utilisées par le nœud receveur pour réduire les exigences de traitement du rafraîchissement.

L'identification et l'accusé de réception d'un message est faite bond par bond. Tous les types d'objets MESSAGE_ID contiennent un identifiant de message. L'identifiant DOIT être unique pour l'adresse IP du générateur de l'objet. Pas plus d'un objet MESSAGE_ID ne peut être inclus dans un message RSVP. Chaque message contenant un objet MESSAGE_ID peut recevoir un accusé de réception via un objet MESSAGE_ID_ACK, lorsque c'est indiqué. Les objets MESSAGE_ID_ACK et MESSAGE_ID_NACK peuvent être envoyés en portage par des messages RSVP sans relation ou dans des messages d'accusé de réception RSVP. Les messages RSVP qui portent un des trois types d'objet peuvent être inclus dans un message de regroupement. Lorsqu'il en est d'inclus, chaque objet est traité comme si il était contenu dans un message RSVP standard, sans regroupement.

4.1 Modification des formats de message standard

Les objets MESSAGE_ID, MESSAGE_ID_ACK et MESSAGE_ID_NACK peuvent être inclus dans les messages RSVP standard, comme définis dans la [RFC2205]. Lorsque ils sont inclus, un ou plusieurs objets MESSAGE_ID_ACK ou MESSAGE_ID_NACK DOIVENT suivre immédiatement l'objet INTEGRITY. Lorsque aucun objet INTEGRITY n'est présent, les objets MESSAGE_ID_ACK ou MESSAGE_ID_NACK DOIT suivre immédiatement l'en-tête du message ou sous-message. Un seul objet MESSAGE_ID PEUT être inclus dans un message ou sous-message et il DOIT suivre tout objet MESSAGE_ID_ACK ou MESSAGE_ID_NACK présent. Lorsque aucun objet MESSAGE_ID_ACK ou MESSAGE_ID_NACK n'est présent, l'objet MESSAGE_ID DOIT suivre immédiatement l'objet INTEGRITY. Lorsque aucun objet INTEGRITY n'est présent, l'objet MESSAGE_ID DOIT suivre immédiatement l'en-tête du message ou du sous-message.

L'ordre des objets ACK pour tous les messages RSVP standard est :

```
<En-tête commun> [ <INTEGRITY> ] [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ] [ <MESSAGE_ID> ]
```

4.2 Objets MESSAGE_ID

Classe de MESSAGE_ID = 23

Objet MESSAGE_ID

Classe = Classe de MESSAGE_ID, C_Type = 1

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Fanions  |           Epoque           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Identifiant_de_message           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Fanions : 8 bits

0x01 = Fanion ACK_Désiré

Indique que l'envoyeur demande au receveur d'envoyer un accusé de réception pour le message.

Epoque : 24 bits

C'est une valeur qui indique quand la séquence Identifiant_de_message a été remise à zéro. Elle DEVRAIT être générée de façon aléatoire chaque fois qu'un nœud fait un réamorçage ou quand l'agent RSVP est redémarré. La valeur NE DEVRAIT PAS être la même que celle utilisée lorsque le nœud a été mis en fonctionnement pour la dernière fois. Cette valeur NE DOIT PAS être changée durant le fonctionnement normal.

Identifiant_de_message : 32 bits

Lorsque il est combiné avec l'adresse IP du générateur du message, le champ Identifiant_de_message identifie de façon univoque un message. Les valeurs placées dans ce champ changent de façon incrémentaire et ne diminuent que lorsque le champ Epoque change ou quand la valeur revient à zéro.

4.3 Objets MESSAGE_ID_ACK et MESSAGE_ID_NACK

MESSAGE_ID_ACK Class = 24

Objet MESSAGE_ID_ACK

Classe = Classe de MESSAGE_ID_ACK, C_Type = 1

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Fanions  |           Epoque           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Identifiant_de_message           |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Fanions : 8 bits

Aucun fanion n'est actuellement défini. Ce champ DOIT être à zéro en transmission et ignoré à réception.

Epoque : 24 bits

Le champ Epoque est copié du message dont il est accusé réception.

Identifiant_de_message : 32 bits

Le champ Identifiant_de_message est copié du message dont il est accusé réception.

Objet MESSAGE_ID_NACK

Classe = Classe de MESSAGE_ID_ACK, C_Type = 2

La définition est la même que pour l'objet MESSAGE_ID_ACK.

4.4 Format du message ACK

Les messages Ack portent un ou plusieurs objets MESSAGE_ID_ACK ou MESSAGE_ID_NACK. Ils NE DOIVENT PAS contenir d'objet MESSAGE_ID. Les messages Ack sont envoyés entre des nœuds RSVP voisins. L'adresse de destination IP d'un message Ack est l'adresse d'envoi individuel du nœud qui a généré le ou les messages dont il est accusé réception. Pour les messages avec des objets RSVP_HOP, tels que les messages Path et Resv, l'adresse se trouve dans l'objet RSVP_HOP. Pour les autres messages, tels que ResvConf, l'adresse IP associée est l'adresse de source dans l'en-tête IP. L'adresse IP de source est une adresse du nœud qui envoie le message Ack.

Le format du message Ack est le suivant :

```
<Message ACK> ::= <En-tête commun> [ <INTEGRITY> ] <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>
[ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
```

Pour les messages Ack, le champ Type de message de l'en-tête commun DOIT être réglé à 13.

Le paragraphe 4.6 donne des lignes directrices sur le moment où devrait être utilisé un message Ack et où les objets MESSAGE_ID devraient être envoyés portés dans d'autres messages RSVP.

4.5 Usage de l'objet MESSAGE_ID

L'objet MESSAGE_ID peut être inclus dans tout message RSVP autre que les messages Ack et Bundle. L'objet MESSAGE_ID est toujours généré et traité sur un seul bond entre voisins RSVP. L'adresse IP de l'objet générateur, c'est-à-dire, le nœud qui a créé l'objet, est représentée d'une façon spécifique du type de message RSVP. Pour les messages avec des objets RSVP_HOP, tels que les messages Path et Resv, l'adresse IP du générateur se trouve dans l'objet RSVP_HOP. Pour les autres messages, tels que les messages ResvConf, l'adresse IP du générateur est l'adresse de source dans l'en-tête IP. Noter que les objets MESSAGE_ID ne peuvent être utilisés que dans des sous-messages Bundle, mais pas dans un message Bundle. Comme c'est toujours le cas avec le message Bundle, chaque sous-message est traité comme si il était reçu individuellement. Cela inclut le traitement des objets MESSAGE_ID.

Le champ Epoque contient une valeur choisie par le générateur. La valeur est utilisée pour indiquer quand l'expéditeur réinitialise les valeurs utilisées dans le champ Identifiant_de_message. Au démarrage, un nœud DEVRAIT choisir au hasard une valeur à utiliser dans le champ Epoque. Le nœud DEVRAIT s'assurer que la valeur choisie n'est pas la même que celle utilisée lors du dernier fonctionnement du nœud. La valeur NE DOIT PAS être changée sauf redémarrage du nœud ou de l'agent RSVP.

Le champ Identifiant_de_message contient une valeur choisie par le générateur. Cette valeur, lorsque elle est combinée avec l'adresse IP du générateur, identifie un message RSVP particulier et les informations d'état spécifiques qu'il représente. La combinaison de Identifiant_de_message et de Epoque peut aussi être utilisée pour détecter les messages déclassés. Lorsque un nœud envoie un message de rafraîchissement avec un objet MESSAGE_ID, il DEVRAIT utiliser la même valeur de Identifiant_de_message qu'utilisée dans le message RSVP qui a le premier annoncé l'état à rafraîchir. Lorsque un nœud envoie un message de déclenchement, la valeur du Identifiant_de_message DOIT être supérieure à toute autre valeur utilisée précédemment avec la même valeur de champ Epoque. Une valeur est considérée comme ayant été utilisée lorsque elle a été envoyée dans tout message qui utilise l'adresse IP associée à la même valeur de champ Epoque.

Le fanion ACK_désiré est établi lorsque le générateur de l'objet MESSAGE_ID veut qu'un objet MESSAGE_ID_ACK soit envoyé en réponse au message. Une telle information peut être utilisée pour s'assurer d'une livraison fiable des

messages RSVP face à des pertes par le réseau. Les nœuds qui établissent le fanion ACK_désiré DEVRAIENT retransmettre les messages sans accusé de réception à un intervalle plus rapide que la période standard de rafraîchissement jusqu'à ce que le message soit acquitté ou jusqu'à atteindre une limite à la "rapidité" de retransmission. Le taux de retransmission rapide DOIT se fonder sur les procédures de retard exponentiel définies à la section 6. Le fanion ACK_désiré ne va normalement être établi que dans les messages de déclenchement. Le fanion ACK_désiré PEUT être établi dans les messages de rafraîchissement. Les questions qui se rapportent aux sessions en diffusion groupée sont traitées plus loin.

Les nœuds qui traitent les objets MESSAGE_ID entrant DEVRAIENT vérifier si un message nouvellement reçu est déclassé et peut être ignoré. Les messages déclassés DEVRAIENT être ignorés, c'est-à-dire, éliminés en silence. Les messages déclassés peuvent être identifiés en examinant les valeurs des champs Epoque et Identifiant_de_message. Pour déterminer l'ordre, la valeur de l'Epoque reçue doit correspondre à la valeur reçue précédemment de l'expéditeur du message. Si les valeurs diffèrent, le receveur NE DOIT PAS traiter le message comme s'il était déclassé. Lorsque les valeurs de Epoque correspondent et que la valeur de Identifiant_de_message est inférieure à la plus grande valeur reçue précédemment de l'expéditeur, le receveur DEVRAIT alors vérifier la valeur précédemment reçue pour voir l'état associé au message. Cette vérification devrait être effectuée pour tout message qui installe ou change un état. (Cela inclut au moins : Path, Resv, PathTear, ResvTear, PathErr et ResvErr.) Si aucune information d'état local ne peut être associée au message, le receveur NE DOIT PAS traiter le message comme déclassé. Si un état local peut être associé au message et si la valeur de Identifiant_de_message reçue est inférieure à la valeur la plus récente reçue associée à cet état, le message DEVRAIT être traité comme étant déclassé.

Noter que la valeur sur 32 bits de l'Identifiant_de_message PEUT revenir à zéro. Pour couvrir le cas du retour à zéro, l'expression suivante peut être utilisée pour vérifier si une valeur de Identifiant_de_message nouvellement reçue est inférieure à une valeur précédemment reçue :

```
si ((int) old_id - (int) new_id > 0) { nouvelle valeur inférieure à vieille valeur ; }
```

Les objets MESSAGE_ID des messages qui ne sont pas déclassés DEVRAIENT être utilisés pour aider à déterminer si le message représente un nouvel état ou un rafraîchissement d'état. Noter que l'état n'est rafraîchi que dans les messages Path et Resv. Si les valeurs de Epoque reçues diffèrent de la valeur précédemment reçue de l'expéditeur du message, le message est un message de déclenchement et le receveur DOIT le traiter complètement. Si un message Path ou Resv contient la même valeur de Identifiant_de_message que celle utilisée dans le message le plus récemment reçu pour la même session et, pour les messages Path, SENDER_TEMPLATE, le receveur DEVRAIT alors traiter le message comme un rafraîchissement d'état. Si la valeur de Identifiant_de_message est supérieure à la valeur la plus récemment reçue, le receveur DOIT traiter complètement le message. Lors du traitement complet d'un message Path ou Resv, le receveur DOIT mémoriser la valeur de l'Identifiant_de_message reçu au titre de l'état Path ou Resv locale pour les futures références.

Les nœuds qui recevront un message non déclassé qui contient un objet MESSAGE_ID avec le fanion ACK_désiré établi DEVRAIENT répondre avec un objet MESSAGE_ID_ACK. Noter que les objets MESSAGE_ID reçus dans des messages qui contiennent des erreurs, c'est-à-dire, qui ne sont pas syntaxiquement valides, NE DOIVENT PAS recevoir d'accusé de réception. Les messages PathErr et ResvErr DEVRAIENT être traités comme des accusés de réception implicites.

4.6 Objet MESSAGE_ID_ACK et usage de l'objet MESSAGE_ID_NACK

L'objet MESSAGE_ID_ACK est utilisé pour accuser réception des messages qui contiennent des objets MESSAGE_ID qui ont été envoyés avec le fanion ACK_Desired établi. Un objet MESSAGE_ID_ACK NE DOIT PAS être généré en réponse à un objet MESSAGE_ID reçu lorsque le fanion ACK_désiré n'est pas établi.

L'objet MESSAGE_ID_NACK est utilisé au titre de l'extension Summary refresh. La génération et le traitement des objets MESSAGE_ID_NACK est décrite plus en détails au paragraphe 5.4.

Les objets MESSAGE_ID_ACK et MESSAGE_ID_NACK PEUVENT être envoyés dans tout message RSVP qui a une adresse de destination IP qui correspond au générateur de l'objet MESSAGE_ID associé. Cela signifie que les objets ne seront pas normalement inclus dans les messages non bond par bond Path, PathTear et ResvConf. Lorsque aucun message approprié n'est disponible, un ou plusieurs objets DEVRAIENT être envoyés dans un message Ack. Les mises en œuvre DEVRAIENT inclure si possible les objets MESSAGE_ID_ACK et MESSAGE_ID_NACK dans les messages RSVP standard.

Les mises en œuvre DEVRAIENT limiter la durée pendant laquelle un objet est retardé pour qu'il soit porté ou envoyé dans un message Ack. Différentes limites peuvent être utilisées pour les objets MESSAGE_ID_ACK et MESSAGE_ID_NACK. Les objets MESSAGE_ID_ACK sont utilisés pour détecter les pertes de transmission de liaison. Si un objet ACK est retardé pendant trop longtemps, le message correspondant sera retransmis. Pour éviter une telle

retransmission, les objets ACK DEVRAIENT être retardés pendant une durée minimale. Un délai égal au temps de transit de la liaison PEUT être utilisé. Les objets MESSAGE_ID_NACK peuvent être retardés pendant un temps indépendant et plus long, bien que des délais supplémentaires augmentent le temps pendant lequel la réservation désirée n'est pas installée.

4.7 Considérations pour la diffusion groupée

Les messages Path et PathTear peuvent être envoyés à des adresses de destination IP en diffusion groupée. Lorsque la destination est une adresse de diffusion groupée, il est possible qu'un seul message contenant un seul objet MESSAGE_ID soit reçu par plusieurs prochains bonds RSVP. Le traitement de l'accusé de réception est plus complexe lorsque le fanion ACK_désiré est établi.

Un certain nombre de problèmes doivent être réglés, y compris celui d'une explosion de ACK, du nombre d'accusés de réception à attendre et du traitement des nouveaux receivers.

Une explosion de ACK survient lorsque chaque receveur répond à l'objet MESSAGE_ID approximativement au même moment. Cela peut conduire à un nombre potentiellement grand d'objets MESSAGE_ID_ACK qui sont livrés simultanément au générateur du message. Pour régler ce cas, le receveur DOIT attendre pendant un intervalle aléatoire avant d'accuser réception d'un objet MESSAGE_ID reçu dans un message destiné à une adresse de diffusion groupée. L'intervalle aléatoire DEVRAIT être entre zéro (0) et un temps maximum configuré. Le maximum configuré DEVRAIT être réglé en proportion de l'intervalle de rafraîchissement de la retransmission "rapide", c'est-à-dire, tel que la durée maximum avant l'envoi d'un accusé de réception ne résulte pas en une retransmission. On notera que l'explosion de ACK est réglée par l'étalement des accusés de réception dans le temps, et non par la suppression des ACK.

Une question plus fondamentale est le nombre d'accusés de réception que le nœud d'amont, c'est-à-dire, le générateur du message, devrait attendre. C'est en fait le même que le nombre de prochains bonds RSVP. Dans le cas de routeur à routeur, le nombre de prochains bonds peut souvent être obtenu de l'acheminement. Lorsque les hôtes sont le nœud amont ou les prochains bonds, le nombre de prochains bonds ne sera normalement pas directement disponible. Un autre cas où le nombre de prochains bonds RSVP ne sera normalement pas connu est lorsque il y a des routeurs non RSVP entre le générateur du message et les prochains bonds RSVP.

Lorsque le nombre de prochains bonds n'est pas connu, le générateur du message DEVRAIT seulement s'attendre à une seule réponse. Le résultat de ce comportement sera un traitement particulier de la retransmission jusqu'à ce que le message soit livré à au moins un prochain bond, suivi ensuite par les rafraîchissements RSVP standard. Les messages de rafraîchissement vont synchroniser l'état avec tous les prochains bonds qui ne reçoivent pas le message original.

4.7.1 Interface de référence RSVP/acheminement

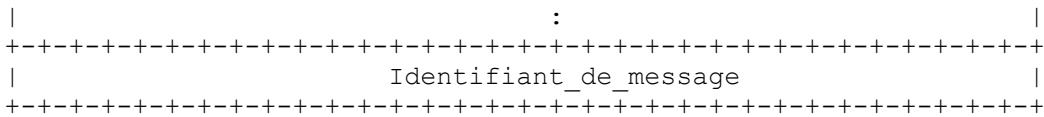
Quand on utilise l'extension MESSAGE_ID avec des sessions en diffusion groupée, il est préférable que RSVP obtienne de l'acheminement le nombre de prochains bonds et qu'il soit notifié des changements de ce nombre. L'interface entre l'acheminement et RSVP est une pure question de mise en œuvre. Comme RSVP [RFC2205] décrit une interface d'acheminement de référence, on présente une version mise à jour de l'interface RSVP/acheminement pour fournir les informations sur le nombre de prochains bonds. Voir la [RFC2205] sur la description des paramètres et fonctions précédemment définis.

- o Interrogation de chemin
Mcast_Route_Query([SrcAddress,] DestAddress, Notify_flag) -> [IncInterface,] OutInterface_list, NHops_list
- o Notification de changement de chemin
Mcast_Route_Change() -> [SrcAddress,] DestAddress, [IncInterface,] OutInterface_list, NHops_list

NHops_list donne le nombre de membres du groupe de diffusion groupée joignables via chaque entrée de OutInterface_list.

4.8. Compatibilité

Tous les nœuds qui envoient des messages avec le bit Capable-de-réduction-de-rafraîchissement établi vont prendre en charge l'extension MESSAGE_ID. Il n'y a pas de problème de rétro-compatibilité soulevé par la classe MESSAGE_ID avec les nœuds qui n'établissent pas le bit Capable-de-réduction-de-rafraîchissement. La classe MESSAGE_ID a une valeur allouée dont la forme est 0bbbbbb. Selon RSVP [RFC2205], les classes avec des valeurs de cette forme doivent être rejetées avec une erreur "Classe d'objet inconnue" par les nœuds qui ne prennent pas en charge la classe. Lorsque le receveur d'un objet MESSAGE_ID ne prend pas en charge la classe, un message d'erreur correspondant sera généré. Le générateur de l'objet MESSAGE_ID va voir l'erreur et DOIT renvoyer le message d'origine sans l'objet MESSAGE_ID.



Fanions : 8 bits

Aucun fanion n'est actuellement défini. Ce champ DOIT être à zéro en émission et ignoré à réception.

Epoque : 24 bits

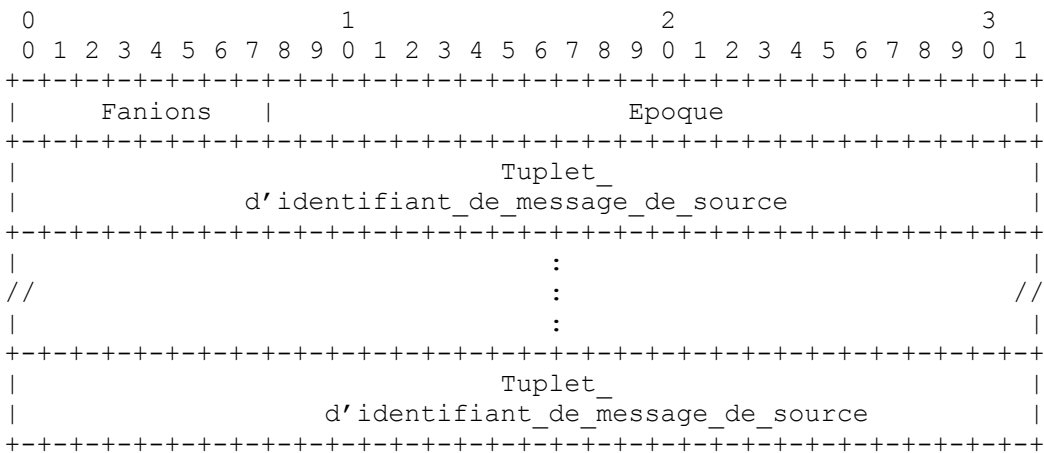
Le champ Epoque tiré de l'objet MESSAGE_ID correspondant au message déclancheur qui a annoncé l'état rafraîchi.

Identifiant_de_message : 32 bits

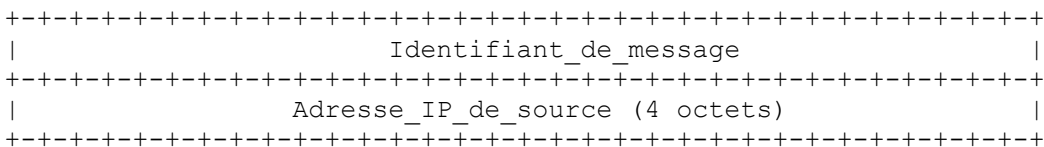
Le champ Identifiant_de_message tiré de l'objet MESSAGE_ID correspondant au message déclancheur qui a annoncé l'état rafraîchi. Un ou plusieurs Identifiant_de_message peuvent être inclus.

Objet IPv4 MESSAGE_ID_SRC_LIST

Classe = classe MESSAGE_ID_LIST, C_Type = 2

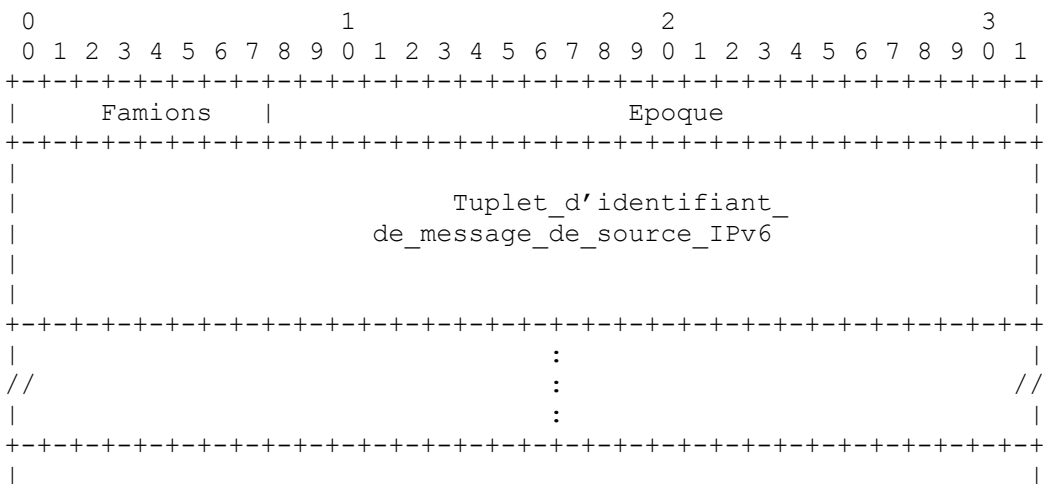


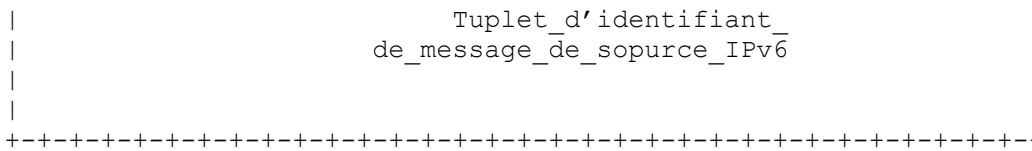
Où un Tuplet_d'identifiant_de_message_de_source consiste en :



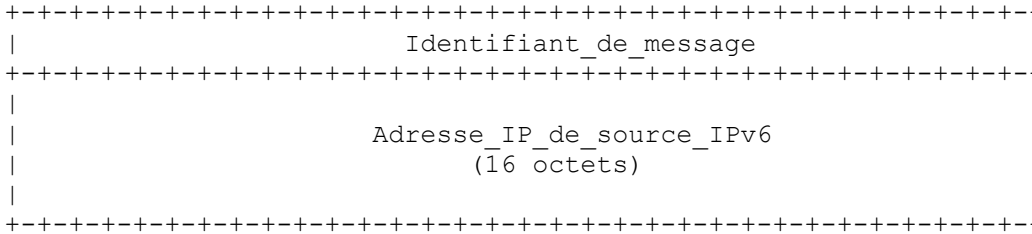
Objet IPv6 MESSAGE_ID_SRC_LIST

Classe = classe MESSAGE_ID_LIST, C_Type = 3





Où un Tuplet_d'identifiant_de_message_de_source_IPv6 consiste en :



Fanions : 8 bits

Aucun fanion n'est défini actuellement. Ce champ DOIT être à zéro en émission et ignoré à réception.

Epoque : 24 bits

Le champ Epoque tiré de l'objet MESSAGE_ID qui correspond au message déclancheur qui a annoncé l'état rafraîchi.

Identifiant_de_message

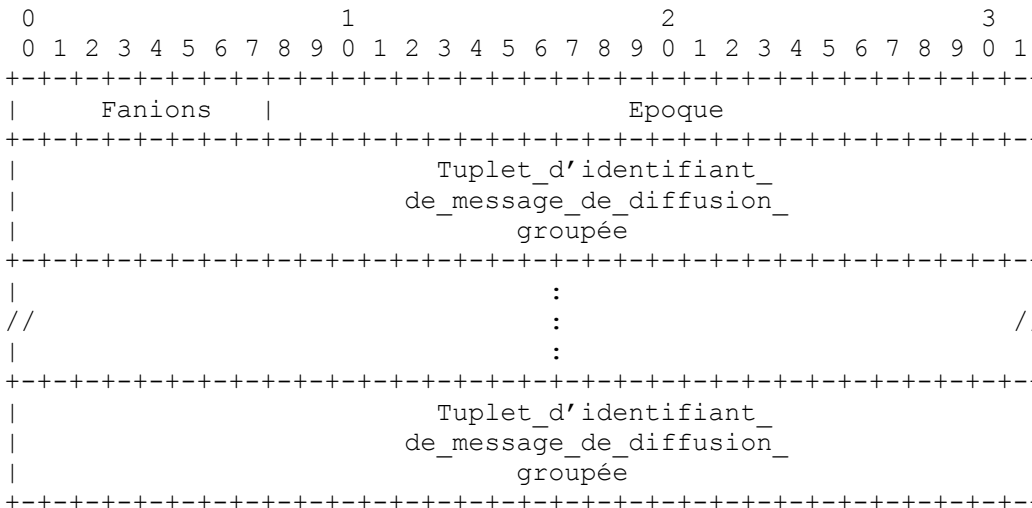
Le champ Identifiant_de_message tiré de l'objet MESSAGE_ID qui correspond au message déclancheur qui a annoncé l'état Path rafraîchi. Un ou plusieurs Identifiant_de_message peuvent être inclus. Chaque Identifiant_de_message DOIT être suivi par l'adresse IP de source qui correspond à l'envoyeur décrit dans l'état Path qui est rafraîchi.

Adresse_IP_de_source

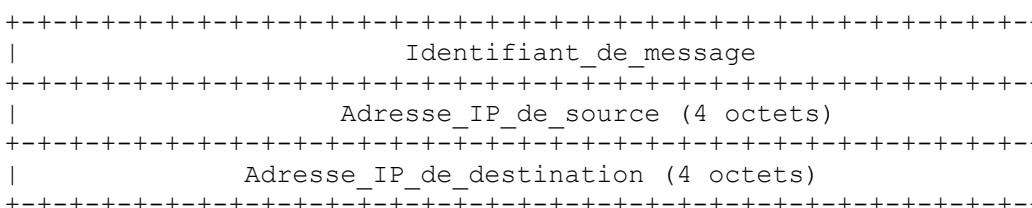
C'est l'adresse IP qui correspond à l'envoyeur de l'état Path qui est rafraîchi.

Objet IPv4 MESSAGE_ID_MCAST_LIST

Classe = classe MESSAGE_ID_LIST, C_Type = 4

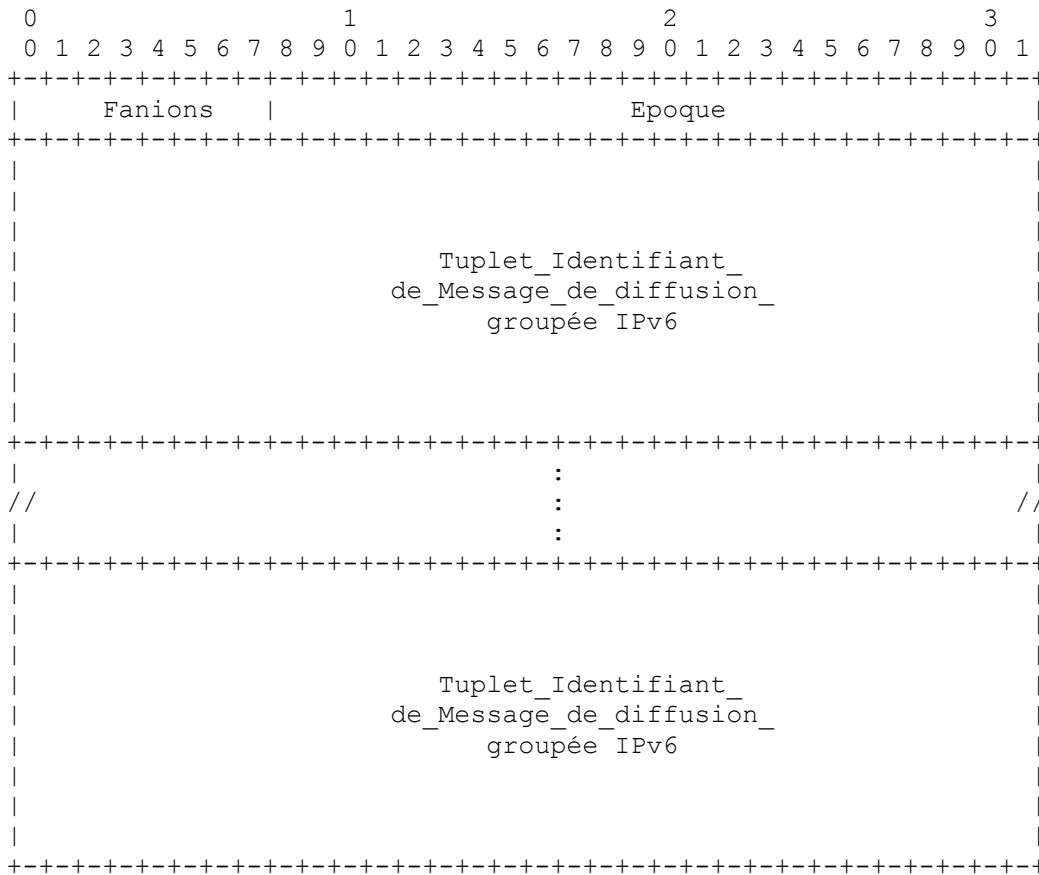


Où un Tuplet_d'identifiant_de_message_de_diffusion_groupée consiste en :

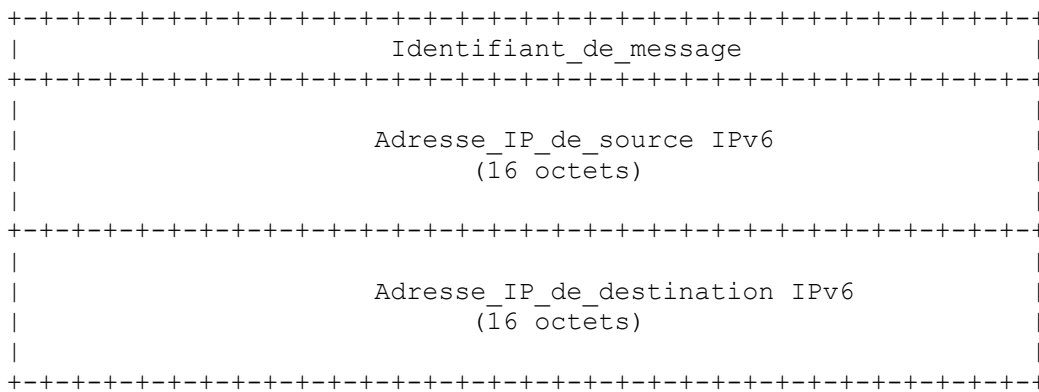


Objet IPv6 MESSAGE_ID_MCAST_LIST

Classe = classe MESSAGE_ID_LIST, C_Type = 5



Où un Tuplet_Identifiant_de_Message_de_diffusion_groupe Ipv6 consiste en :



Fanions : 8 bits
Aucun fanion n'est actuellement défini. Ce champ DOIT être à zéro en émission et ignoré à réception.

Epoque : 24 bits
Le champ Epoque tiré de l'objet MESSAGE_ID qui correspond au message déclancheur qui annonçait l'état qui est rafraîchi.

Identifiant_de_message : 32 bits
Le champ Identifiant_de_message tiré de l'objet MESSAGE_ID qui correspond au message déclancheur qui annonçait l'état Path qui est rafraîchi. Un ou plusieurs Identifiant_de_message peuvent être inclus. Chaque Identifiant_de_message DOIT être suivi par l'adresse IP de source qui correspond à l'envoyeur de l'état Path qui est rafraîchi, et à l'adresse IP de destination de la session.

Adresse_IP_de_source

L'adresse IP qui correspond à l'expéditeur de l'état Path qui est rafraîchi.

Adresse_IP_de_destination

L'adresse IP de destination qui correspond à la session de l'état Path qui est rafraîchi.

5.2 Format du message Srefresh

Les messages Srefresh portent un ou plusieurs objets MESSAGE_ID_LIST, MESSAGE_ID_SRC_LIST, et MESSAGE_ID_MCAST_LIST. Les objets MESSAGE_ID_LIST et MESSAGE_ID_MCAST_LIST PEUVENT être portés dans le même message Srefresh. MESSAGE_ID_SRC_LIST ne peut pas être combiné dans les messages Srefresh avec les autres objets. Un seul message Srefresh PEUT rafraîchir les deux états Path et Resv.

Les messages Srefresh qui portent un champ Identifiant_de_message qui correspond à l'état Path sont normalement envoyés avec une adresse IP de destination égale à l'adresse portée dans les objets SESSION correspondants. L'adresse IP de destination PEUT être réglée au prochain bond RSVP lorsque le prochain bond est connu comme ayant la capacité RSVP et que soit (a) la session est en envoi individuel, soit (b) l'interface sortante est une liaison en point à point. Les messages Srefresh qui portent un champ Identifiant_de_message qui correspond à l'état Resv DOIVENT être envoyés avec une adresse IP de destination réglée au bond précédent de l'état Resv.

Les messages Srefresh envoyés à l'adresse IP de destination d'une session de diffusion groupée DOIVENT contenir l'objet MESSAGE_ID_SRC_LIST et NE DOIVENT PAS inclure d'objet MESSAGE_ID_LIST ou MESSAGE_ID_MCAST_LIST. Les messages Srefresh envoyés au prochain bond RSVP PEUVENT contenir l'un ou/et l'autre objet MESSAGE_ID_LIST et MESSAGE_ID_MCAST_LIST, mais NE DOIVENT PAS inclure d'objet MESSAGE_ID_SRC_LIST.

L'adresse IP de source d'un message Srefresh est une adresse du nœud qui génère le message. L'adresse IP de source DOIT correspondre à l'adresse associée aux objets MESSAGE_ID lorsque ils étaient inclus dans un message RSVP standard. Comme mentionné précédemment, l'adresse de source associée à un message MESSAGE_ID est représentée d'une façon spécifique du type de message RSVP. Pour les messages avec des objets RSVP_HOP, tels que les messages Path et Resv, l'adresse se trouve dans l'objet RSVP_HOP. Pour les autres messages, tels que le message ResvConf, l'adresse IP associée est l'adresse de source dans l'en-tête IP.

Les messages Srefresh qui sont adressés à l'adresse de destination IP d'une session DOIVENT être envoyés avec l'option IP Alerte de routeur dans leur en-tête IP. Les messages Srefresh adressés directement aux voisins RSVP NE DEVRAIENT PAS être envoyés avec l'option IP Alerte de routeur dans leur en-tête IP.

Chaque message Srefresh DOIT occuper exactement un datagramme IP. Si il excède la MTU, le datagramme est fragmenté par IP et réassemblé au nœud receveur. Les messages Srefresh PEUVENT être envoyés au sein d'un message RSVP Bundle. Bien que ceci ne soit pas attendu car les messages Srefresh peuvent porter une liste de champs Identifiant_de_message au sein d'un seul objet. Les mises en œuvre peuvent choisir de limiter chaque message Srefresh à la taille de la MTU de la liaison sortante, par exemple, à 1500 octets.

Le format du message Srefresh est :

```
<Message Srefresh> ::= <En-tête commun> [ <INTEGRITY> ] [ [ <MESSAGE_ID_ACK> |
<MESSAGE_ID_NACK> ] ... ] [ <MESSAGE_ID> ] <liste srefresh> | <liste srefresh de source>
```

```
<liste srefresh> ::= <MESSAGE_ID LIST> | <MESSAGE_ID MCAST_LIST> [ <liste srefresh> ]
```

```
<liste srefresh de source> ::= <MESSAGE_ID SRC_LIST> [ <liste srefresh de source> ]
```

Pour les messages Srefresh, le champ Type de message de l'en-tête commun DOIT être réglé à 15.

5.3 Usage du message Srefresh

Un message Srefresh peut être généré pour rafraîchir un état Resv et Path. Si un message Srefresh est utilisé pour rafraîchir un état particulier, la génération d'un message de rafraîchissement standard pour cet état particulier DEVRAIT être supprimée. L'intervalle de rafraîchissement d'un état n'est pas affecté par l'utilisation des rafraîchissements fondés sur le message Srefresh.

En générant un message Srefresh, un nœud DEVRAIT rafraîchir autant d'états Path et Resv qu'il est possible en incluant les informations provenant d'autant d'objets MESSAGE_ID que dans le même message Srefresh. Seules les informations provenant des objets MESSAGE_ID qui satisfont aux restrictions d'adresse IP de source et de destination, comme décrit au paragraphe 5.2, peuvent être incluses dans le même message Srefresh. Identifier l'état Resv qui peut être rafraîchi en utilisant le même message Srefresh est très direct. Identifier quel état Path peut être inclus est un peu plus complexe.

Seul l'état qui a été annoncé précédemment dans les messages Path et Resv qui contiennent des objets MESSAGE_ID peuvent être rafraîchis via un message Srefresh. Les messages Srefresh fondés sur des rafraîchissements doivent préserver les propriétés de synchronisation d'état des messages Path ou Resv fondés sur les rafraîchissements. Précisément, l'utilisation des messages Srefresh NE DOIT PAS résulter en la préemption de l'état au prochain bond RSVP. La période à laquelle l'état est rafraîchi quand on utilise les messages Srefresh PEUT être plus courte que la période qui aurait été subie en utilisant le message Path ou Resv sur la base des rafraîchissements, mais elle NE DOIT PAS être plus longue.

L'approche particulière utilisée pour déclencher un message Srefresh sur la base des rafraîchissements est spécifique de la mise en œuvre. Certaines possibilités sont de déclencher la génération du message Srefresh sur la base de la période de rafraîchissement de chaque état, ou selon l'interface, en générant périodiquement des messages Srefresh pour rafraîchir tout état qui n'a pas été rafraîchi pendant l'intervalle de rafraîchissement de l'état. D'autres approches sont aussi possibles. Un intervalle par défaut de génération de message Srefresh de 30 secondes est suggéré pour les nœuds qui ne calculent pas de façon dynamique un intervalle de génération.

Lors de la génération d'un message Srefresh, il y a deux méthodes pour identifier quel état Path peut être rafraîchi dans un message spécifique. Dans les deux cas, l'intervalle de rafraîchissement mentionné précédemment et les restrictions d'adresse IP de source doivent être suivis. La méthode principale est de n'inclure que les sessions qui partagent la même adresse IP de destination dans le même message Srefresh.

La méthode secondaire pour l'identification de l'état Path qui peut être rafraîchi au sein d'un seul message Srefresh est une optimisation. Cette méthode PEUT être utilisée lorsque le prochain bond est connu pour prendre en charge RSVP et quand soit (a) la session est en envoi individuel, soit (b) l'interface sortante est une liaison point à point. Cette méthode NE DOIT PAS être utilisée lorsque le prochain bond n'est pas connu pour prendre en charge RSVP ou quand l'interface sortante est un réseau multi accès et que la session est une adresse de diffusion groupée. L'utilisation de cette méthode PEUT être configurée administrativement. Lorsque on utilise cette méthode, l'adresse de destination dans l'en-tête IP du message Srefresh est normalement l'adresse du prochain bond. Lorsque l'utilisation de cette méthode est configurée administrativement, l'adresse de destination devrait être l'adresse de groupe bien connue 224.0.0.14. Lorsque l'interface sortante est une liaison point à point, tout état Path associé aux sessions annoncées de l'interface DEVRAIT être inclus dans le même message Srefresh. Lorsque l'interface sortante n'est pas une liaison point à point, tout état Path de session en envoi individuel DEVRAIT être inclus dans le même message Srefresh.

Identifier quel état Resv peut être rafraîchi au sein d'un seul message Srefresh se fonde simplement sur les adresses IP de source et de destination. Tout état qui a été annoncé précédemment dans des messages Resv avec les mêmes adresses IP qu'un message Srefresh PEUT être inclus.

Après avoir identifié les états Path et Resv qui peuvent être inclus dans un message Srefresh particulier, le générateur du message ajoute au message les informations de MESSAGE_ID qui correspondent à un objet utilisé précédemment de chaque état identifié. Pour tout état Resv et pour tout état Path de sessions en envoi individuel, les informations sont ajoutées au message dans un objet MESSAGE_ID_LIST qui a une valeur Epoque correspondante. (Noter que une seule valeur Epoque sera utilisée en fonctionnement normal.) S'il n'existe aucun objet qui corresponde, un nouvel objet MESSAGE_ID_LIST est créé.

L'état Path des sessions en diffusion groupée peut être ajouté au même message lorsque l'adresse de destination du message Srefresh est le prochain bond RSVP et que l'interface sortante est une liaison point à point. Dans ce cas, les informations sont ajoutées au message dans un objet MESSAGE_ID_MCAST_LIST qui a une valeur Epoque qui correspond. Si aucun objet qui corresponde n'existe, un nouvel objet MESSAGE_ID_MCAST_LIST est créé. Lorsque l'adresse de destination du message est une adresse de diffusion groupée, les informations identifiées sont ajoutées au message dans un objet MESSAGE_ID_SRC_LIST qui a une valeur Epoque qui correspond. Si il n'existe aucun objet qui corresponde, un nouvel objet MESSAGE_ID_SRC_LIST est alors créé. Une fois que le message Srefresh est composé, le générateur du message le transmet sur l'interface appropriée.

À réception d'un message Srefresh, le nœud DOIT tenter d'identifier l'état Path ou Resv installé qui correspond. La correspondance est faite sur la base de l'adresse de source dans l'en-tête IP du message Srefresh, du type d'objet et de chaque champ Identifiant_de_message. Si l'état correspondant peut être trouvé, le nœud receveur DOIT alors mettre à jour les informations d'état correspondantes comme si un message de rafraîchissement standard avait été reçu. Si l'état correspondant ne peut pas être identifié, un NACK Srefresh DOIT alors être généré, correspondant au champ Identifiant_de_message pour lequel la correspondance n'a pas été trouvée. Les champs Identifiant_de_message reçus dans

des objets MESSAGE_ID_LIST peuvent correspondre à tout état Resv ou Path de sessions en envoi individuel. Les champs Identifiant_de_message reçus dans des objets MESSAGE_ID_SRC_LIST ou MCAST_LIST correspondent à l'état Path des sessions de diffusion groupée.

Une vérification supplémentaire doit être effectuée pour déterminer si un NACK devrait être généré pour les champs Identifiant_de_message sans correspondance associés à l'état Path de sessions de diffusion groupée, c'est-à-dire, les champs qui étaient portés dans les objets MESSAGE_ID_SRC_LIST ou MCAST_LIST. Le nœud receveur doit vérifier si le nœud devrait transmettre des paquets de données originaires de la source correspondant aux champs sans correspondance. Cette vérification, couramment appelée une vérification RPF, est effectuée sur la base des informations de source et de groupe portées dans les objets MESSAGE_ID_SRC_LIST et MCAST_LIST. Dans les deux objets, l'adresse IP de la source figure immédiatement après le champ Identifiant_de_message correspondant. L'adresse de groupe figure immédiatement après l'adresse IP de source dans les objets MESSAGE_ID_MCAST_LIST. L'adresse de groupe est l'adresse IP de destination du message lorsque l'objet MESSAGE_ID_SRC_LIST est utilisé. Le nœud receveur ne génère qu'un NACK Srefresh quand le nœud devrait transmettre des paquets au groupe identifié à partir de l'expéditeur qui figure sur la liste. Si le nœud devait transmettre des paquets de données en diffusion groupée à partir d'un expéditeur qui figure sur la liste et si il y a un champ Identifiant_de_message correspondant qui n'est pas satisfait, un NACK Srefresh approprié DOIT alors être généré. Si le nœud ne devait pas transmettre de paquets au groupe identifié à partir d'un expéditeur figurant sur la liste, le champ correspondant Identifiant_de_message non satisfait est ignoré en silence.

5.4 NACK de Srefresh

Les NACK Srefresh sont utilisés pour indiquer qu'un champ Identifiant_de_message reçu porté dans un objet MESSAGE_ID_LIST, SRC_LIST, ou MCAST_LIST ne satisfait aucun des états installés. Cela peut survenir pour un certain nombre de raisons, y compris, par exemple, un changement de chemin. Un NACK Srefresh est codé dans un objet MESSAGE_ID_NACK. En générant un NACK Srefresh, les champs Epoque et Identifiant_de_message de l'objet MESSAGE_ID_NACK DOIVENT avoir la même valeur que celle reçue. Les objets MESSAGE_ID_NACK sont transmis comme décrit au paragraphe 4.6.

Les objets MESSAGE_ID_NACK reçus indiquent que le générateur de l'objet n'a aucun état installé qui corresponde à l'objet. À réception d'un objet MESSAGE_ID_NACK, le receveur effectue une recherche d'état Path ou Resv installé sur la base des valeurs de Epoque et d'Identifiant_de_message contenues dans l'objet. Si un état correspondant est trouvé, le receveur DOIT alors transmettre l'état correspondant via un messages Path ou Resv standard. Si le receveur ne peut pas identifier d'état installé, aucune action n'est requise.

5.5 Préservation de l'état conditionnel RSVP

Comme exposé dans la [RFC2205], RSVP utilise un état conditionnel pour traiter une large classe d'erreurs potentielles. RSVP le fait en envoyant périodiquement une représentation complète de l'état installé dans les messages Resv et Path. Les messages Srefresh sont utilisés à la place de l'envoi périodique des messages standard de rafraîchissement de Path et Resv. Alors que cela donne un avantage d'adaptabilité et protège des événements courants du réseau comme les pertes de paquet ou les changements d'acheminement, cela ne donne pas exactement les mêmes propriétés de récupération sur erreur. Un exemple d'erreur dont il serait possible de se récupérer via les messages standard mais pas avec les messages Srefresh est la corruption interne d'état. Ce paragraphe recommande deux méthodes qui peuvent être utilisées pour mieux préserver le mécanisme de récupération d'erreur en état conditionnel de RSVP. Les deux mécanismes sont pris en charge en utilisant les messages de protocole existants.

Le premier mécanisme utilise une somme de contrôle ou un autre algorithme pour détecter un changement non remarqué auparavant dans l'état interne. Ce mécanisme ne protège pas contre la corruption interne de l'état. Il couvre juste le cas où un message déclencheur aurait dû être envoyé, mais ne l'a pas été. Lors de l'envoi d'un message déclencheur Path ou Resv, un nœud devrait faire une somme de contrôle ou un autre algorithme, tel que [MD5], sur l'état interne et mémoriser le résultat. Le choix de l'algorithme est une décision administrative. Périodiquement, le nœud devrait relancer l'algorithme et comparer le nouveau résultat avec celui mémorisé. Si les valeurs diffèrent, un message standard de rafraîchissement Path ou Resv correspondant devrait être envoyé et la nouvelle valeur devrait être mémorisée. La période de recalcul devrait être établie sur la base des ressources de calcul du nœud et des exigences de fiabilité du réseau.

Le second mécanisme est simplement d'envoyer périodiquement des messages de rafraîchissement standard Path et Resv. Comme ce mécanisme utilise des messages de rafraîchissement standard, il peut récupérer du même ensemble d'erreurs que le RSVP standard. Quand on utilise ce mécanisme, la période d'envoi des messages de rafraîchissement standard doit être plus longue que l'intervalle où sont générés les messages Srefresh afin de gagner le bénéfice de l'utilisation de l'extension Summary refresh. Lorsque le message de rafraîchissement standard est envoyé, un Summary refresh correspondant NE DEVRAIT PAS être envoyé durant la même période de rafraîchissement. Lorsque un nœud prend en

charge la génération périodique des messages standard de rafraîchissement alors que des Srefresh sont utilisés, la fréquence de génération des messages de rafraîchissement standard par rapport à la génération des Summary refresh DEVRAIT être configurable par l'administrateur de réseau.

5.6 Compatibilité

Les nœuds qui acceptent l'extension Summary refresh annoncent leur prise en charge via le bit Capacité-de-réduction-de-rafraîchissement dans l'en-tête de message RSVP. Cela permet aux nœuds qui prennent en charge l'extension de se détecter les uns les autres. Lorsque on ne sait pas si un prochain bond accepte l'extension, le message standard Path et Resv fondé sur les rafraîchissements DOIT être utilisé. Noter que quand le prochain bond de l'acheminement ne prend pas en charge RSVP, il ne sera pas toujours possible de détecter si le prochain bond RSVP accepte l'extension Summary refresh. Donc, lorsque le prochain bond d'acheminement n'a pas la capacité RSVP, le message Srefresh fondé sur le rafraîchissement NE DEVRAIT PAS être utilisé. Un nœud PEUT être configuré administrativement pour utiliser les messages Srefresh dans tous les cas où tous les nœuds RSVP d'un réseau sont connus pour prendre en charge l'extension Summary refresh. C'est utile en fonctionnement dans ce mode, l'extension s'ajuste correctement au cas des prochains bonds non RSVP et de changement d'acheminement.

Selon la section 2, les nœuds qui acceptent l'extension Summary refresh doivent aussi veiller à reconnaître quand un prochain bond cesse d'envoyer des messages RSVP avec le bit Capacité-de-réduction-de-rafraîchissement établi.

6. Procédures de retard exponentiel

Cette section se fonde sur [Pan] et donne des procédures pour mettre en œuvre un retard exponentiel pour la retransmission des messages qui attendent un accusé de réception, voir au paragraphe 4.5. Les mises en œuvre DOIVENT utiliser les procédures décrites ou leur équivalent.

6.1 Fonctionnement général

Ce qui suit est un mécanisme possible pour le retard exponentiel de retransmission d'un message RSVP dont il n'a pas été accusé réception: Lors de l'envoi d'un tel message, un nœud insère un objet MESSAGE_ID avec le fanion ACK_Désiré établi. Le nœud d'envoi va retransmettre le message jusqu'à ce qu'un message d'accusé de réception soit reçu ou que le message ait été transmis un nombre maximum de fois. À réception, un nœud receveur accuse réception de l'arrivée du message par le renvoi d'un message d'accusé de réception (qui est un objet MESSAGE_ID_ACK correspondant.) Lorsque le nœud expéditeur reçoit l'accusé de réception, la retransmission du message s'arrête. L'intervalle entre les retransmissions est gouverné par un temporisateur de retransmission rapide. Le temporisateur de retransmission rapide commence à un faible intervalle et augmente exponentiellement jusqu'à atteindre un seuil.

6.2 Paramètres de temps

Les procédures décrites utilisent les paramètres temporels suivants. Tous les paramètres sont par interface.

Intervalle de retransmission rapide (Rf) :

Rf est l'intervalle initial de retransmission pour les messages dont il n'a pas été accusé réception. Après l'envoi du message pour la première fois, le nœud d'envoi va programmer la retransmission après Rf secondes. La valeur de Rf peut être aussi brève que le délai d'aller-retour (RTT, *round trip time*) entre le nœud d'envoi et le nœud receveur, s'il est connu.

Limite de réessai rapide (Rl) :

Rl est le nombre maximum de fois qu'un message sera transmis sans recevoir d'accusé de réception.

Valeur d'incrément (Delta) :

Delta indique la vitesse à laquelle l'expéditeur augmente l'intervalle de retransmission. Le rapport de deux intervalles successifs de retransmission est $(1 + \text{Delta})$.

Les valeurs par défaut suggérées sont une temporisation initiale de retransmission (Rf) de 500 ms, un retard exponentiel de puissance 2 ($\text{Delta} = 1$) et une limite de réessai (Rl) de 3.

6.3 Algorithme de retransmission

Après qu'un nœud d'envoi a transmis un message contenant un objet MESSAGE_ID avec le fanion ACK_Désiré établi, il devrait immédiatement programmer une retransmission après Rf secondes. Si un objet MESSAGE_ID_ACK correspondant est reçu avant Rf secondes, la retransmission DEVRAIT alors être annulée. Autrement, il va retransmettre le message après $(1 + \Delta) * R_f$ secondes. La retransmission programmée va continuer jusqu'à ce qu'un objet MESSAGE_ID_ACK approprié soit reçu, ou que la limite de réessai rapide, Rl, soit atteinte.

Un nœud d'envoi peut utiliser l'algorithme suivant lors de la transmission d'un message contenant un objet MESSAGE_ID avec le fanion ACK_Désiré établi :

Avant la première transmission, initialiser : $R_k = R_f$ et $R_n = 0$

```
Alors que ( $R_{n++} < R_l$ ) {
  transmettre le message ;
  relancer après  $R_k$  secondes ;
   $R_k = R_k * (1 + \Delta)$ ;
}
/* accusé réception ou pas de réponse du receveur pendant trop longtemps :
*/ aucun nettoyage nécessaire ; sortie ;
```

En asynchrone, lorsque un nœud d'envoi reçoit un objet MESSAGE_ID_ACK correspondant, il va changer le compte d'essais, R_n , en R_l .

Noter que le nœud émetteur n'annonce pas l'utilisation des procédures décrites de retard exponentiel via l'objet TIME_VALUE.

6.4 Considérations sur les performances

L'utilisation du retard exponentiel de retransmission est un ajout nouveau et significatif à RSVP. Il sera important de revoir à la lumière de l'expérience le fonctionnement et les performances qui s'y rapportent avant que ce document n'avance sur la voie de la normalisation. Il sera particulièrement important de revoir le fonctionnement avec la diffusion groupée, et tous les problèmes d'explosion d'accusés de réception rencontrés actuellement.

7. Remerciements

Le présent document représente les idées et commentaires de l'équipe de conception de MPLS-TE et des participants à la réunion intermédiaire du groupe de travail RSVP. Merci à Bob Braden, Lixia Zhang, Fred Baker, Adrian Farrel, Roch Guerin, Kireeti Kompella, David Mankins, Henning Schulzrinne, Andreas Terzis, Lan Wang et Masanobu Yuhara pour leurs réactions spécifiques sur les diverses versions du document.

Des portions de ce travail se fondent sur les travaux de Masanobu Yuhara et Mayumi Tomikawa [Yuhara].

8 Considérations pour la sécurité

Aucun nouveau problème de sécurité n'est soulevé dans ce document. Voir dans la [RFC2205] une discussion générale sur les problèmes de sécurité dans RSVP.

9. Références

[Pan] P. Pan, H. Schulzrinne, "Staged Refresh Timers for RSVP," Global Internet'97, Phoenix, AZ, novembre 1997. <http://www.cs.columbia.edu/~pingpan/papers/timergi.pdf>

[MD5] R. Rivest, "Algorithme de [résumé de message](#) MD5", RFC1321, avril 1992. (*Information*)

[RFC2119] S. Bradner, "[Mots clés](#) à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

- [RFC2205] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Protocole de réservation de ressource ([RSVP](#)) -- version 1, spécification fonctionnelle", septembre 1997. (MàJ par [RFC2750](#), [RFC3936](#), [RFC4495](#)) (P.S.)
- [Yuhara] M. Yuhara et M. Tomikawa, "Extensions RSVP pour des rafaichissements fondés sur l'identifiant", Travail en cours.

10. Adresses des auteurs

Lou Berger
LabN Consulting, LLC
téléph. : +1 301 468 9228
mél : lberger@labn.net

Der-Hwa Gan
Juniper Networks, Inc.
1194 N. Mathilda Avenue,
Sunnyvale, CA 94089
téléph. : +1 408 745 2074
mél : dhg@juniper.net

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
téléph. : +1 978 244 8143
mél : swallow@cisco.com

Ping Pan
Juniper Networks, Inc.
1194 N. Mathilda Avenue,
Sunnyvale, CA 94089
téléph. : +1 408 745 3704
mél : pingpan@juniper.net

Franco Tommasi
University of Lecce, Fac. Ingegneria
Via Monteroni 73100 Lecce, ITALY
mél : franco.tommasi@unile.it

Simone Molendini
University of Lecce, Fac. Ingegneria
Via Monteroni 73100 Lecce, ITALY
mél : molendini@ultra5.unile.it

11. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2001). Tous droits réservés.

Ce document et ses traductions peuvent être copiés et diffusés, et les travaux dérivés qui commentent ou expliquent autrement ou aident à sa mise en œuvre peuvent être préparés, copiés, publiés et distribués, partiellement ou en totalité, sans restriction d'aucune sorte, à condition que l'avis de droits de reproduction ci-dessus et ce paragraphe soit inclus sur toutes ces copies et œuvres dérivées. Toutefois, ce document lui-même ne peut être modifié en aucune façon, par exemple en supprimant la notice de droits d'auteur ou les références à l'Internet Society ou à d'autres organisations Internet, sauf si c'est nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les procédures des normes pour l'Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

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

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

Remerciement

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