Groupe de travail Réseau
Request for Comments : 4875
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

R. Aggarwal, éd., Juniper Networks
D. Papadimitriou, éd., Alcatel
S. Yasukawa, éd., NTT
mai 2007

Extensions au protocole de réservation de ressource avec ingénierie du trafic (RSVP-TE) pour les chemins de commutation d'étiquettes (LSP) d'ingénierie du trafic en point à multi points

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 "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.
(La présente traduction incorpore les errata 2483, 2485, 2486, 2490, 2493.)

Résumé
Le présent document décrit des extensions au protocole de réservation de ressources – ingénierie du trafic (RSVP-TE, *Resource Reservation Protocol - Traffic Engineering*) pour l'établissement de chemins de commutation d'étiquettes (LSP, *Label Switched Path*) de point à multi points (P2MP, *point-to-multipoint*) d'ingénierie du trafic (TE, *Traffic Engineered*) dans les réseaux de commutation d'étiquettes multi protocoles (MPLS, *Multi-Protocol Label Switching*) et MPLS généralisé (GMPLS, *Generalized MPLS*). La solution repose sur RSVP-TE sans exiger de protocole d'acheminement en diffusion groupée dans le cœur du fournisseur de service. Les éléments de protocole et les procédures pour cette solution sont décrits.

Il peut y avoir diverses applications pour les LSP TE P2MP comme la diffusion groupée IP. La spécification de la façon dont de telles applications vont utiliser un LSP TE P2MP sort du domaine d'application du présent document.

Table des matières

1. Introduction

La [RFC3209] définit un mécanisme pour établir des chemins de commutation d'étiquettes (LSP, *Label Switched Path*) d'ingénierie du trafic (TE, *Traffic Engineered*) en point à point (P2P, *point-to-point*) dans les réseaux de commutation d'étiquettes multi protocoles (MPLS, *Multi-Protocol Label Switching*). La [RFC3473] définit des extensions à la [RFC3209] pour établir des LSP TE P2P dans les réseaux MPLS généralisés (GMPLS, *Generalized MPLS*). Cependant ces spécifications ne fournissent pas de mécanisme pour construire des LSP TE en point à multi points (P2MP).

Le présent document définit des extensions au protocole RSVP-TE ([RFC3209] et [RFC3473]) pour prendre en charge les LSP TE P2MP satisfaisant à l'ensemble d'exigences décrit dans la [RFC4461].

Le présent document s'appuie sur la sémantique du protocole de réservation de ressources (RSVP, *Resource Reservation Protocol*) dont RSVP-TE hérite pour la construction des LSP P2MP. Un LSP P2MP est composé de plusieurs sous LSP de source à feuille (S2L, *source-to-leaf*). Ces sous LSP S2L sont établis entre les LSR d'entrée et de sortie et sont combinés de façon appropriée par les LSR de branche en utilisant la sémantique de RSVP pour résulter en un LSP TE P2MP. Un message Path peut signaler un ou plusieurs sous LSP S2L pour un seul LSP P2MP. Donc le sous LSP S2L appartenant à un LSP P2MP peut être signalé en utilisant un message Path ou en étant partagé sur plusieurs messages Path.

Il y a diverses applications pour les LSP TE P2MP et les techniques de signalisation décrites dans le présent document peuvent être utilisées, parfois en combinaison avec d'autres techniques, pour prendre en charge des applications différentes.

La spécification de comment les applications vont utiliser les LSP TE P2MP et comment les chemins des LSP TE P2MP sont calculés sort du domaine d'application du présent document.

2. Conventions utilisées dans le document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

3. Terminologie

Le présent document utilise les terminologies définies dans les [RFC2205], [RFC3031], [RFC3209], [RFC3473], [RFC4090], et [RFC4461].

4. Mécanisme

Le présent document décrit une solution qui optimise la réplication des données en permettant que les nœuds non d'entrée dans le réseau soient des nœuds de réplication/branche. Un nœud de branche est un LSR qui réplique les données entrantes sur une ou plusieurs interfaces sortantes. La solution s'appuie sur RSVP-TE dans le réseau pour établir un LSP TE P2MP.

Le LSP TE P2MP est établi en associant plusieurs sous LSP S2L et en s'appuyant sur la réplication aux nœuds de branche. Ceci est décrit plus en détails dans les paragraphes qui suivent décrivant les tunnels P2MP et comment ils se rapportent au sous LSP S2L.

4.1 Tunnels P2MP

La caractéristique de définition d'un LSP TE P2MP est l'action requise aux nœuds de branche où se produit la réplication des données. Les données étiquetées MPLS entrantes sont répliquées aux interfaces sortantes qui peuvent utiliser des étiquettes différentes pour les données.

Un tunnel TE P2MP comporte un ou plusieurs LSP P2MP. Un tunnel TE P2MP est identifié par un objet SESSION P2MP. Cet objet contient l'identifiant de la session P2MP, qui inclut l'identifiant P2MP *(P2MP ID)* un identifiant de tunnel *(Tunnel ID)*, et un identifiant de tunnel étendu *(Extended Tunnel ID)*. L'identifiant P2MP est un nombre de quatre octets qui est unique dans la portée du LSR d'entrée.

Le triplet <P2MP ID, Tunnel ID, Extended Tunnel ID>fournit un identifiant pour l'ensemble des destinations du tunnel TE P2MP.

Les champs de l'objet SESSION P2MP sont identiques à ceux de l'objet SESSION défini dans la [RFC3209] excepté que le champ Adresse de point d'extrémité de tunnel est remplacé par le champ Identifiant P2MP. L'objet SESSION P2MP est défini au paragraphe 19.1

4.2 LSP P2MP

Un LSP P2MP est identifié par la combinaison de l'identifiant P2MP, de l'identifiant de tunnel, et de l'identifiant de tunnel étendu qui font partie de l'objet SESSION P2MP, et des champs Adresse d'envoyeur de tunnel et Identifiant de LSP de l'objet P2MP SENDER_TEMPLATE. Le nouvel objet P2MP SENDER_TEMPLATE est défini au paragraphe 19.2.

4.3 Sous groupes

Comme avec tous les autres LSP contrôlés par RSVP, l'état de LSP P2MP est géré en utilisant les messages RSVP. Bien que l'utilisation des messages RSVP soit la même, l'état de LSP P2MP diffère de l'état de LSP P2P d'un certain nombre de

façons. Un LSP P2MP comporte plusieurs sous LSP S2L, et par suite, il peut n'être pas possible de représenter tout l'état dans un seul paquet IP. Il doit aussi être possible d'ajouter et supprimer efficacement des points d'extrémité aux LSP TE P2MP. Un problème supplémentaire est que le LSP P2MP doit aussi traiter le problème de l'état de "re-fusion", voir la [RFC4461] et la section 18.

Ces différences de l'état P2MP sont traitées par l'ajout d'un identifiant de sous groupe *(Sub-Group ID)* et d'un identifiant d'origine de sous groupe *(Sub-Group Originator ID)* aux objets SENDER_TEMPLATE et FILTER_SPEC. Pris ensemble, l'identifiant de sous groupe et l'identifiant d'origine de sous groupe sont appelés les champs de sous groupe.

Les champs de sous groupe, avec le reste des objets SENDER_TEMPLATE et SESSION, sont utilisés pour représenter une portion de l'état d'un LSP P2MP. Cette portion de l'état d'un LSP P2MP se réfère seulement à l'état de signalisation et non à la réplication ou embranchement au plan des données. Par exemple, il est possible à un nœud de "faire un embranchement" de l'état de signalisation pour un LSP P2MP, mais pas de faire un embranchement des données associées au LSP P2MP. Les applications typiques de génération et d'utilisation de plusieurs sous groupes sont (1) l'ajout d'une sortie et (2) la fragmentation sémantique pour assurer qu'un message Path reste dans un seul paquet IP.

4.4 Sous LSP S2L

Un LSP P2MP est constitué d'un ou plusieurs sous LSP S2L.

4.4.1 Représentation d'un sous LSP S2L

Un sous LSP S2L existe au sein du contexte d'un LSP P2MP. Donc, il est identifié par l'identifiant P2MP, l'identifiant de tunnel, et l'identifiant de tunnel étendu, qui font partie des champs P2MP SESSION, Adresse de tunnel envoyeur et Identifiant de LSP de l'objet P2MP SENDER_TEMPLATE, et l'adresse de destination de sous LSP S2L qui fait partie de l'objet S2L_SUB_LSP. L'objet S2L_SUB_LSP est défini au paragraphe 19.3.

Un objet Chemin explicite (ERO, *EXPLICIT_ROUTE Object*) ou un objet Chemin explicite secondaire P2MP (SERO, *P2MP_SECONDARY_EXPLICIT_ROUTE Object*) est utilisé pour spécifier facultativement le chemin explicite d'un sous LSP S2L. Chaque ERO ou SERO signalé correspond à un objet S2L_SUB_LSP particulier. Les détails du codage de chemin explicite sont spécifiés au paragraphe 4.5. L'objet SECONDARY_EXPLICIT_ROUTE est défini dans la [RFC4873], un nouveau type de classe d'objet P2MP SECONDARY_EXPLICIT_ROUTE est défini au paragraphe 19.5, et un type de classe correspondant d'objet P2MP_SECONDARY_RECORD_ROUTE est défini au paragraphe 19.6.

4.4.2 Sous LSP S2L et messages Path

Le mécanisme du présent document permet qu'un LSP P2MP soit signalé en utilisant un ou plusieurs messages Path. Chaque message Path peut signaler un ou plusieurs sous LSP S2L. La prise en charge de plusieurs messages Path est désirable car un message Path peut n'être pas assez grand pour contenir tous les sous LSP S2L; et cela permet aussi des manipulations séparées des sous arborescences de LSP P2MP. La raison de permettre qu'un seul message Path signale plusieurs sous LSP S2L est d'optimiser le nombre de messages de conrôle nécessaires pour établir un LSP P2MP.

4.5 Acheminement explicite

Quand un message Path signale un seul sous LSP S2L (c'est-à-dire, le message Path est seulement ciblé sur une seule feuille dans l'arborescence P2MP) l'objet EXPLICIT_ROUTE code le chemin vers le LSR de sortie. Le message Path inclut aussi l'objet S2L_SUB_LSP pour le sous LSP S2L signalé. Le couple < [<EXPLICIT_ROUTE>], <S2L_SUB_LSP> > représente le sous LSP S2L et est appelé le descripteur de sous LSP. L'absence du ERO devrait être interprétée comme exigeant un acheminement bond par bond pour le sous LSP sur la base du champ Adresse de destination du sous LSP S2L de l'objet S2L_SUB_LSP.

Quand un message Path signale plusieurs sous LSP S2L, le chemin du premier sous LSP S2L vers le LSR de sortie est codé dans l'ERO. Le premier sous LSP S2L est celui qui correspond au premier objet S2L_SUB_LSP dans le message Path. Les sous LSP S2L correspondants aux objets S2L_SUB_LSP qui suivent sont appelés les sous LSP S2L suivants.

Le chemin de chaque sous LSP S2L suivant est codé dans un objet P2MP_SECONDARY_EXPLICIT_ROUTE (SERO). Le format du SERO est le même que celui d'un ERO (comme défini dans les [RFC3209] et [RFC3473]). Chaque sous LSP S2L suivant est représenté par des couples de forme <[<P2MP_SECONDARY_EXPLICIT_ROUTE>],

<S2L_SUB_LSP>>. Un SERO pour un sous LSP S2L particulier inclut seulement le chemin d'un LSR de branche au LSR de sortie de ce sous LSP S2L. La branche DOIT apparaître comme un bond explicite dans l'ERO ou un autre SERO. L'absence d'un SERO devrait être interprétée comme demandant un acheminement bond par bond pour ce sous LSP S2L. Noter que l'adresse de destination est portée dans l'objet Sous LSP S2L. Le codage du SERO et de l'objet S2L_SUB_LSP est décrit en détails à la Section 19.

Afin d'éviter la répétition potentielle des informations de chemin pour les parties de sous LSP S2L qui partagent des bonds, ces informations sont déduites des chemins explicites des autres sous LSP S2L en utilisant une compression de chemin explicite dans les SERO.

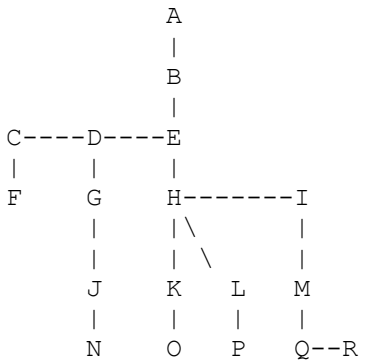


Figure 1 : Compression explicite de chemin

La Figure 1 montre un LSP P2MP avec le LSR A comme LSR d'entrée et six LSR de sortie : (F, N, O, P, Q et R). Quand tous les six sous LSP S2L sont signalés dans un message Path, on suppose que le sous LSP S2L au LSR F est le premier sous LSP S2L, et le reste sont les sous LSP S2L suivants. Le codage qui suit est une façon pour que le LSR d'entrée A code les chemins explicites de sous LSP S2L en utilisant la compression :

sous LSP S2L-F : ERO = {B, E, D, C, F}, <S2L_SUB_LSP> objet-F
sous LSP S2L-N : SERO = {D, G, J, N}, <S2L_SUB_LSP> objet-N
sous LSP S2L-O : SERO = {E, H, K, O}, <S2L_SUB_LSP> objet-O
sous LSP S2L-P : SERO = {H, L, P}, <S2L_SUB_LSP> objet-P
sous LSP S2L-Q : SERO = {H, I, M, Q}, <S2L_SUB_LSP> objet-Q
sous LSP S2L-R : SERO = {Q, R}, <S2L_SUB_LSP> objet-R

Après que le LSR E a traité le message Path entrant provenant du LSR B, il envoie un message Path au LSR D avec les chemins explicites de sous LSP S2L codés comme suit :

sous LSP S2L-F : ERO = {D, C, F}, <S2L_SUB_LSP> objet-F
sous LSP S2L-N : SERO = {D, G, J, N}, <S2L_SUB_LSP> objet-N

Le LSR E envoie aussi un message Path au LSR H, et ce qui suit est une façon de coder les chemins explicites de sous LSP S2L en utilisant la compression :

sous LSP S2L-O : ERO = {H, K, O}, <S2L_SUB_LSP> objet-O
sous LSP S2L-P : SERO = {H, L, P}, S2L_SUB_LSP objet-P
sous LSP S2L-Q : SERO = {H, I, M, Q}, <S2L_SUB_LSP> objet-Q
sous LSP S2L-R : SERO = {Q, R}, <S2L_SUB_LSP> objet-R

Après que le LSR H a traité le message Path entrant provenant de E, il envoie un message Path aux LSR K, L, et I. Le codage pour le message Path au LSR K est comme suit :

sous LSP S2L-O : ERO = {K, O}, <S2L_SUB_LSP> objet-O

Le codage du message Path envoyé par le LSR H au LSR L est comme suit :

sous LSP S2L-P : ERO = {L, P}, <S2L_SUB_LSP> objet-P

Le codage suivant est une façon pour que le LSR H code les chemins explicites de sous LSP S2L dans le message Path

envoyé au LSR I :

sous LSP S2L-Q : ERO = {I, M, Q}, <S2L_SUB_LSP> objet-Q
sous LSP S2L-R : SERO = {Q, R}, <S2L_SUB_LSP> objet-R

Les codages des chemins explicites dans le messages Path envoyé par les LSR D et Q sont laissés comme exercice pour le lecteur.

Ce mécanisme de compression réduit la taille du message Path. Il réduit aussi le traitement supplémentaire qui peut résulter si des chemins explicites sont codés de l'entrée à la sortie pour chaque sous LSP S2L. Aucune hypothèse n'est faite sur l'ordre des sous LSP S2L suivants et donc sur l'ordre des SERO dans le message Path. Tous les LSR ont besoin de traiter le ERO correspondant au premier sous LSP S2L. Un LSR a besoin de traiter un descripteur de sous LSP S2L pour un sous LSP S2L suivant seulement si le premier bond dans le SERO correspondant est une adresse locale de ce LSR. Le LSR de branche qui est le premier bond d'un SERO propage le sous LSP S2L correspondant vers l'aval.

5. Message Path

5.1 Format du message Path

Cette Section décrit les modifications faites au format du message Path comme spécifié dans les [RFC3209] et [RFC3473]. Le message Path est amélioré pour signaler un ou plusieurs sous LSP S2L. Ceci est fait en incluant la liste de descripteurs de sous LSP S2L dans le message Path comme montré ci-dessous.

<Message Path> ::= <En-tête commun> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
[<MESSAGE_ID>]
<SESSION> <RSVP_HOP>
<TIME_VALUES>
[<EXPLICIT_ROUTE>]
<LABEL_REQUEST>
[<PROTECTION>]
[<LABEL_SET> ...]
[<SESSION_ATTRIBUTE>]
[<NOTIFY_REQUEST>]
[<ADMIN_STATUS>]
[<POLICY_DATA> ...]
<descripteur d'envoyeur>
[<liste de descripteurs de sous LSP S2L>]

Voici le format de la liste de descripteurs de sous LSP S2L :

<liste de descripteurs de sous LSP S2L> ::= <descripteur de sous LSP S2L> [<liste de descripteurs de sous LSP S2L>]

<descripteur de sous LSP S2L> ::= <S2L_SUB_LSP> [<P2MP_SECONDARY_EXPLICIT_ROUTE>]

Chaque LSR DOIT utiliser les objets communs dans le message Path et les descripteurs de sous LSP S2L pour traiter chaque sous LSP S2L représenté par la combinaison de l'objet S2L_SUB_LSP et de l'objet SECONDARY-/EXPLICIT_ROUTE.

Selon la définition de <descripteur de sous LSP S2L>, chaque objet S2L_SUB_LSP PEUT être suivi par un SERO correspondant. Le premier objet S2L_SUB_LSP est un cas particulier, et son chemin explicite est spécifié par l'ERO. Donc, le premier objet S2L_SUB_LSP NE DEVRAIT PAS être suivi par un SERO, et si un est présent, il DOIT être ignoré.

Le RRO dans le descripteur de l'envoyeur contient les bonds en amont traversés par le message Path et s'applique à tous les sous LSP S2L signalés dans le message Path.

Un objet IF_ID RSVP_HOP DOIT être utilisé sur les liaisons où il n'y a pas une association bijective d'un canal de contrôle à un canal de données [RFC3471]. Un objet RSVP_HOP défini dans la [RFC2205] DEVRAIT être utilisé autrement.

Le traitement du message Path est décrit dans le paragraphe suivant.

5.2 Traitement du message Path

Le LSR d'entrée initie l'établissement d'un sous LSP S2L à chaque LSR de sortie qui est une destination du LSP P2MP. Chaque sous LSP S2L est associé au même LSP P2MP en utilisant les champs communs d'objet SESSION P2MP et <Adresse d'envoyeur, Identifiant de LSP> dans l'objet P2MP SENDER_TEMPLATE. Donc, il peut être combiné avec d'autres sous LSP S2L pour former un LSP P2MP. Un autre sous LSP S2L appartenant à la même instance de ce sous LSP S2L (c'est-à-dire, le même LSP P2MP) DEVRAIT partager ses ressources avec ce sous LSP S2L. La session correspondant au tunnel TE P2MP est déterminée sur la base de l'objet SESSION P2MP. Chaque sous LSP S2L est identifié en utilisant l'objet S2L_SUB_LSP. L'acheminement explicite pour le sous LSP S2L est réalisé en utilisant le ERO et les SERO.

Comme mentionné précédemment, il est possible de signaler les sous LSP S2L pour un LSP P2MP donné dans un ou plusieurs messages Path, et un certain message Path peut contenir un ou plusieurs sous LSP S2L. Un LSR qui prend en charge les LSP P3MP signalés par RSVP-TE DOIT être capable de recevoir et traiter plusieurs messages Path pour le même LSP P2MP et plusieurs sous LSP S2L dans un message Path. Cela implique qu'un tel LSR DOIT être capable de recevoir et traiter tous les objets mentionnés dans la Section 19.

5.2.1 Messages Path multiples

Comme décrit dans la Section 4, le couple < [<EXPLICIT_ROUTE>] <S2L_SUB_LSP> > ou le couple < [<P2MP SECONDARY_EXPLICIT_ROUTE>] <S2L_SUB_LSP> > est utilisé pour spécifier un sous LSP S2L. Plusieurs messages Path peuvent être utilisés pour signaler un LSP P2MP. Chaque message Path peut signaler un ou plusieurs sous LSP S2L. Si un message Path contient seulement un sous LSP S2L, chaque LSR le long du sous LSP S2L suit les procédures de la [RFC3209] pour traiter le message Path à côté du traitement d'objet S2L_SUB_LSP décrit dans le présent document.

Le traitement des messages Path contenant plus d'un sous LSP S2L est décrit au paragraphe 5.2.2.

Un LSR d'entrée PEUT utiliser plusieurs messages Path pour la signalisation d'un LSP P2MP. Cela peut être parce que un seul message Path ne peut pas être assez grand pour signaler le LSP P2MP. Ou il se peut que quand de nouvelles feuilles sont ajoutées au LSP P2MP, elles soient signalées dans un nouveau message Path. Ou un LSR d'entrée PEUT choisir de casser l'arborescence P2MP en arborescences séparées P2MP plus gérables. Ces arborescences partagent la même racine et peuvent partager le tronc et certaines branches. La portée de cette décomposition de gestion des arborescences P2MP est limitée par une seule arborescence (l'arborescence P2MP) et plusieurs arborescences avec une seule feuille chacune (sous LSP S2L). Selon la [RFC4461], un LSP P2MP DOIT avoir des attributs cohérents à travers toutes les portions d'une arborescence. Cela implique que chaque message Path utilisé pour signaler un LSP P2MP est signalé en utilisant les mêmes attributs de signalisation à l'exception des descripteurs de sous LSP S2L et de l'identifiant de sous groupe.

Les sous LSP résultants provenant de différents messages Path appartenant au même LSP P2MP DEVRAIENT partager les étiquettes et ressources lorsque ils partagent des bonds pour empêcher que plusieurs copies des données soient envoyées.

Dans certains cas, un LSR de transit peut avoir besoin de générer plusieurs messages Path pour signaler l'état correspondant à un seul message Path reçu. Par exemple l'expansion d'ERO peut résulter en une inondation de messages Path résultants. Dans ce cas, le message peut être décomposé en plusieurs messages Path de façon que chaque message porte un sous ensemble de la sous arborescence de X2L portée par le message entrant.

Plusieurs messages Path générés par un LSR qui signale l'état pour le même LSP P2MP sont signalés avec le même objet SESSION et ont la même paire <Adresse de source, Identifiant de LSP> dans l'objet SENDER_TEMPLATE. Afin d'ôter l'ambiguïté de ces messages Path, un couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe> est introduit (aussi appelé les champs de sous groupe) et codé dans l'objet SENDER_TEMPLATE. Plusieurs messages Path générés par un LSR pour signaler l'état pour le même LSP P2MP ont le même identifiant d'origine de sous groupe et ont un identifiant de sous groupe différent. L'identifiant d'origine de sous groupe DOIT être réglé à l'identifiant de routeur TE du LSR qui est à l'origine du message Path. Les cas où un LSR de transit peut changer l'identifiant d'origine de sous groupe d'un message Path entrant sont décrits ci-dessous. L'identifiant d'origine de sous groupe est unique au monde. L'espace d'identifiants de sous groupe est spécifique de l'identifiant d'origine de sous groupe.

5.2.2 Sous LSP S2L multiples dans un message Path

La liste de descripteurs de sous LSP S2L permet la signalisation d'un ou plusieurs sous LSP S2L dans un message Path. Chaque descripteur de sous LSP S2L décrit un seul sous LSP S2L.

Tous les LSR DOIVENT traiter le ERO correspondant au premier sous LSP S2L si le ERO est présent. Si un ou plusieurs SERO sont présents, un ERO DOIT être présent. Le premier sous LSP S2L DOIT être propagé dans un message Path par chaque LSR le long du chemin explicite spécifié par l'ERO, si l'ERO est présent. Autrement, il DOIT être propagé en utilisant l'acheminement bond par bond vers la destination identifiée par l'objet S2L_SUB_LSP.

Un LSR DOIT traiter un descripteur de sous LSP S2L pour un sous LSP S2L suivant comme suit :

Si l'objet S2L_SUB_LSP est suivi par un SERO, le LSR DOIT vérifier le premier bond dans le SERO :
- Si le premier bond du SERO identifie une adresse locale du LSR, et si le LSR est aussi la sortie identifiée par l'objet S2L_SUB_LSP, le descripteur NE DOIT PAS être propagé vers l'aval, mais le SERO peut être utilisé pour le contrôle de sortie conformément à la [RFC4003].
- Si le premier bond du SERO identifie une adresse locale du LSR, et si le LSR n'est pas la sortie identifiée par l'objet S2L_SUB_LSP, le descripteur de sous LSP S2L DOIT être inclus dans un message Path envoyé au prochain bond déterminé d'après le SERO.
- Si le premier bond du SERO n'est pas une adresse locale du LSR, le descripteur de sous LSP S2L DOIT être inclus dans le message Path envoyé au LSR qui est le prochain bond pour atteindre le premier bond dans le SERO. Ce prochain bond est déterminé en utilisant le ERO ou d'autres SERO qui codent le chemin pour le premier bond du SERO.

Si l'objet S2L_SUB_LSP n'est pas suivi par un SERO, le LSR DOIT examiner l'objet S2L_SUB_LSP :
- Si ce LSR est la sortie identifiée par l'objet S2L_SUB_LSP, le descripteur de sous LSP S2L NE DOIT PAS être propagé vers l'aval.
- Si ce LSR n'est pas la sortie identifiée par l'objet S2L_SUB_LSP, le LSR DOIT prendre une décision d'acheminement pour déterminer le prochain bond vers la sortie, et DOIT inclure le descripteur de sous LSP S2L dans un message Path envoyé au prochain bond vers la sortie. Dans ce cas, le LSR PEUT insérer un SERO dans le descripteur de sous LSP S2L.

Donc, un LSR de branche DOIT seulement propager les descripteurs de sous LSP S2L pertinents à chaque bond vers l'aval. Une liste de descripteurs de sous LSP S2L qui est propagée sur une liaison vers l'aval DOIT seulement contenir les sous LSP S2L qui sont acheminés en utilisant de bond. Ce traitement PEUT résulter en ce qu'un sous LSP S2L suivant dans un message Path entrant devienne le premier sous LSP S2L dans un message Path sortant.

Noter que si un ou plusieurs SERO contiennent des bonds lâches, l'expansion de ces bonds lâches PEUT résulter en un débordement de la taille de message Path. Le paragraphe 5.2.3 décrit comment la signalisation de l'ensemble de sous LSP S2L peut être partagé sur plus d'un message Path.

L'objet RECORD_ROUTE (RRO) contient les bonds traversés par le message Path et s'applique à tous les sous LSP S2L signalés dans le message Path. Un LSR de transit DOIT ajouter son adresse dans un RRO entrant et le propager vers l'aval. Un LSR de branche DOIT former un nouveau RRO pour chaque message Path sortant en copiant le RRO du message Path entrant et en ajoutant son adresse. Chacun de ces RRO mis à jour DOIT être formé en utilisant les règles de la [RFC3209] (et mises à jour par la [RFC3473]) comme approprié.

Si un LSR est incapable de prendre en charge un sous LSP S2L dans un message Path (par exemple, il est incapable d'acheminer vers la destination en utilisant le SERO) un message PathErr DOIT être envoyé pour le sous LSP S2L impacté, et le traitement normal du reste du LSP P2MP DEVRAIT continuer. Le comportement par défaut est que le reste du LSP n'est pas impacté (c'est-à-dire, il est permis à toutes les autres branches de s'établir) et les branches défaillantes sont rapportées dans des messages PathErr dans lesquels le fanion Path_State_Removed *(état de chemin supprimé)* NE DOIT PAS être établi. Cependant, le LSR d'entrée peut établir un fanion Intégrité de LSP pour demander que si il y a une défaillance d'établissement sur une branche, l'établissement du LSP entier devrait échouer. Ceci est décrit plus en détails au paragraphe 5.2.4 et à la Section 11.

5.2.3 Fragmentation en transit des informations d'état de chemin

Dans certains cas, un LSR de transit peut avoir besoin de générer plusieurs messages Path pour signaler l'état correspondant à un seul message Path reçu. Par exemple, l'expansion d'ERO peut résulter en un débordement du message Path résultant. RSVP [RFC2205] interdit l'utilisation de la fragmentation IP, et donc la fragmentation IP DOIT être évitée dans ce cas. Afin de réaliser cela, plusieurs messages Path générés par le LSR de transit sont signalés avec l'identifiant d'origine de sous groupe réglé à l'identifiant de routeur TE du LSR de transit et avec un identifiant de sous groupe distinct pour chaque message Path. Donc, chaque message Path distinct qui est généré par le LSR de transit pour le LSP P2MP porte un couple distinct de <Identifiant d'origine de sous groupe, Identifiant de sous groupe>.

Quand plusieurs messages Path sont utilisés par un nœud d'entrée ou de transit, chaque message Path DEVRAIT être identique à l'exception des champs Descripteur de sous LSP S2L qui s'y rapportent (par exemple, SERO) Informations de message et de bond (par exemple, INTEGRITY, MESSAGE_ID, et RSVP_HOP) et Sous groupe, des objets SENDER_TEMPLATE. Sauf quand une opération "faire avant de couper" est effectuée (comme spécifié au paragraphe 14.1) les champs Adresse d'envoyeur de tunnel et Identifiant de LSP DOIVENT être les mêmes dans chaque message. Pour les nœuds de transit, ils DOIVENT être les mêmes que les valeurs dans le message Path reçu.

Comme décrit ci-dessus, un cas dans lequel l'identifiant d'origine de sous groupe d'un message Path reçu est changé est celui d'une fragmentation d'un message Path à un nœud de transit. Un autre cas est quand l'identifiant d'origine de sous groupe d'un message Path reçu peut être changé dans le message Path sortant et réglé à celui du LSR d'origine du message Path sur la base de la politique locale. Par exemple, un LSR peut décider de toujours changer l'identifiant d'origine de sous groupe lorsque il effectue l'expansion d'ERO. L'identifiant de sous groupe NE DOIT PAS être changé si l'identifiant d'origine de sous groupe n'est pas changé.

5.2.4 Contrôle du partage de sort de branche

Un LSR d'entrée peut contrôler le comportement d'un LSP si il y a une défaillance durant l'établissement de LSP ou après qu'un LSP a été établi. Le comportement par défaut est que seules les branches vers l'aval de la défaillance ne sont pas établies, mais l'entrée peut demander "l'intégrité de LSP"' afin que toute défaillance n'importe où dans l'arborescence de LSP cause l'échec du LSP P2MP entier.

Le LSP d'entrée peut demander "l'intégrité de LSP"' en établissant le bit 3 du TLV Fanions d'attributs. Le bit est établi si l'intégrité du LSP est demandée.

Il est RECOMMANDÉ d'utiliser l'objet LSP_REQUIRED_ATTRIBUTES [RFC4420].

Un LSR de branche qui prend en charge le TLV Fanions d'attributs et reconnaît ce bit DOIT prendre en charge l'intégrité de LSP ou rejeter l'établissement de LSP avec un message PathErr portant l'erreur "Erreur d'acheminement"/"Intégrité de LSP non prise en charge".

5.3 Greffage

L'opération d'ajout de LSR de sortie à un LSP P2MP existant est appelée greffage. Cette opération permet aux nœuds de sortie de se joindre à un LSP P2MP à différents moments.

Il y a deux méthodes pour ajouter des sous LSP S2L à un LSP P2MP. La première est d'ajouter le nouveau sous LSP S2L au LSP P2MP en les ajoutant à un message Path existant et en rafraîchissant le message Path entier. Le traitement de message décrit à la Section 4 résulte en l'ajout de ces sous LSP S2L au LSP P2MP. Noter que en résultat de l'ajout d'un ou plusieurs sous LSP S2L à un message Path, le codage de compression d'ERO peut devoir être recalculé.

La seconde est d'utiliser les mises à jour incrémentaires décrites au paragraphe 10.1. Les LSR de sortie peuvent être ajoutés en signalant seulement le sous LSP S2L impacté dans un nouveau message Path. Donc, les autres sous LSP S2L n'ont pas à être re-signalés.

6. Message Resv

6.1 Format du message Resv

Le message Resv suit le format des [RFC3209] et [RFC3473] :

```
<Message Resv> ::=  <En-tête commun> [ <INTEGRITY> ]
                    [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                    [ <MESSAGE_ID> ]
                    <SESSION> <RSVP_HOP>
                    <TIME_VALUES>
                    [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                    [ <NOTIFY_REQUEST> ]
                    [ <ADMIN_STATUS> ]
```

[<POLICY_DATA> ...]
<STYLE> <liste des descripteurs de flux>

<liste de descripteurs de flux> ::= <liste de descripteurs de flux FF> | <descripteur de flux SE>

<liste de descripteurs de flux FF> ::= <descripteur de flux FF> | <liste de descripteurs de flux FF> <descripteur de flux FF>

<descripteur de flux SE> ::= <FLOWSPEC> <liste spec de filtre SE>

<liste spec de filtre SE> ::= <spec de filtre SE> | <liste spec de filtre SE> <spec de filtre SE>

Le descripteur de flux FF et la spec de filtre SE sont modifiés comme suit pour identifier le sous LSP S2L auquel ils correspondent :

<descripteur de flux FF> ::= [<FLOWSPEC>] <FILTER_SPEC> <LABEL> [<RECORD_ROUTE>]
[<liste des descripteurs de flux de sous LSP S2L>]

<spec de filtre SE> ::= <FILTER_SPEC> <LABEL> [<RECORD_ROUTE>] [<liste des descripteurs de flux de sous LSP S2L>]

<liste des descripteurs de flux de sous LSP S2L> ::= <descripteur de flux de sous LSP S2L> [<liste des descripteurs de flux de sous LSP S2L>]

<descripteur de flux de sous LSP S2L> ::= <S2L_SUB_LSP> [<P2MP_SECONDARY_RECORD_ROUTE>]

FILTER_SPEC est défini au paragraphe 19.4.

Le descripteur de flux de sous LSP S2L a le même format que le descripteur de sous LSP S2L au pararaphe 5.1 avec la différence qu'un objet P2MP_SECONDARY_RECORD_ROUTE est utilisé à la place d'un objet P2MP SECONDARY_EXPLICIT_ROUTE. Les objets P2MP_SECONDARY_RECORD_ROUTE suivent le même mécanisme de compression que les objets P2MP SECONDARY_EXPLICIT_ROUTE. Noter qu'un message Resv peut signaler plusieurs sous LSP S2L qui peuvent appartenir au même objet FILTER_SPEC ou à des objets FILTER_SPEC différents. La même étiquette DEVRAIT être allouée si les champs <Adresse d'envoyeur, Identifiant de LSP> de l'objet FILTER_SPEC sont les mêmes.

Cependant des étiquettes différentes DOIVENT être allouées si le <Adresse d'envoyeur, Identifiant de LSP> de l'objet FILTER_SPEC est différent, car cela implique que le FILTER_SPEC se réfère à un LSP P2MP différent.

6.2 Traitement du message Resv

Le LSR de sortie DOIT suivre les procédures normales de RSVP quand il génère un message Resv. Le format des messages Resv est comme défini au paragraphe 6.1. Comme d'habitude, le message Resv porte l'étiquette allouée par le LSR de sortie.

Un nœud en amont du nœud de sortie DOIT allouer sa propre étiquette et la passer en amont dans le message Resv. Le nœud PEUT combiner plusieurs descripteurs de flux, provenant de différents messages Resv reçus de l'aval, dans un message Resv envoyé vers l'amont. Un message Resv NE DOIT PAS être envoyé vers l'amont tant qu'au moins un message Resv n'a pas été reçu d'un voisin en aval. Quand le bit d'intégrité est établi dans l'objet LSP_REQUIRED_ATTRIBUTE, le message Resv NE DOIT PAS être envoyé en amont tant que tous les messages Resv n'ont pas été reçus des voisins en l'aval.

Chaque descripteur de flux de filtre fixe (FF, *Fixed-Filter*) ou spécification de filtre de partage explicite (SE, *Shared-Explicit*) envoyé vers l'amont dans un message Resv inclut une liste de descripteurs de sous LSP S2L. Chacun de ces descripteur de flux FF ou spécification de filtre SE pour le même LSP P2MP (qu'il soit dans un ou dans plusieurs messages Resv) sur le même Resv DOIT recevoir la même étiquette, et les descripteurs de flux FF ou spécifications de filtre SE DEVRAIENT utiliser la même étiquette sur plusieurs messages Resv.

Le nœud qui envoie le message Resv, pour un LSP P2MP en amont DOIT associer l'étiquette allouée par ce nœud à toutes les étiquettes reçues des messages Resv vers l'aval, pour ce LSP P2MP. Noter qu'un nœud de transit peut devenir un point de réplication à l'avenir quand une branche lui est rattachée. Donc, il en résulte l'établissement d'un LSP P2MP du LSR

d'entrée aux LSR de sortie.

Le LSR d'entrée peut avoir besoin de comprendre quand toutes les sorties désirées ont été atteintes. Ceci est réalisé en utilisant les objets S2L_SUB_LSP.

Chaque nœud de branche PEUT transmettre un seul message Resv vers l'amont pour chaque message Resv reçu d'un receveur vers l'aval. Noter qu'il peut y avoir un grand nombre de messages Resv au LSR d'entrée et à proximité pour un LSP avec de nombreux receveurs. Un LSR de branche DEVRAIT combiner l'état Resv de plusieurs receveurs en un seul message Resv à envoyer en amont (voir le paragraphe 6.2.1). Cependant, noter qu'il peut en résulter un débordement du message Resv, en particulier quand le nombre de receveurs vers l'aval de tout LSR de branche augmente à mesure que le LSR se rapproche du LSR d'entrée. Donc, un LSR de branche PEUT choisir d'envoyer plus d'un message Resv en amont et de partager l'état Resv entre les messages.

Quand un nœud de transit règle le champ Origine de sous groupe dans un message Path, il DOIT remplacer les champs de sous groupe reçus dans les objets FILTER_SPEC de tous les messages Resv associés par la valeur qu'il a reçu à l'origine dans les champs Sous groupe du message Path provenant du voisin en amont.

La génération de message ResvErr n'est pas modifiée. Les nœuds qui propagent un message ResvErr reçu DOIVENT utiliser les valeurs de champs Sous groupe portées dans le message Resv correspondant.

6.2.1 Réduction des messages Resv

Un nœud de branche peut devoir envoyer en amont un message Resv révisé chaque fois qu'il y a un changement dans un message Resv pour un sous LSP S2L reçu d'un des voisins vers l'aval. Il peut en résulter un nombre excessif de messages Resv envoyés en amont, en particulier quand les sous LSP S2L sont établis pour la première fois. Afin d'atténuer cette situation, les nœuds de branche peuvent limiter leur transmission de messages Resv. Précisément, dans le cas où le seul changement envoyé dans un message Resv est dans un ou plusieurs objets P2MP_SECONDARY_RECORD_ROUTE (SRRO) le nœud de branche DEVRAIT transmettre le message Resv seulement après un certain délai passé depuis la transmission du précédent message Resv pour la même session. Ce message Resv retardé DEVRAIT inclure des SRRO pour toutes les branches. Une valeur suggérée pour le délai est trente secondes, et les retards DEVRAIT généralement être de plus d'une seconde. Les mécanismes spécifiques pour le ralentissement des messages Resv et les réglages de temporisateurs de retard dépendent de la mise en œuvre et sortent du domaine d'application du présent document.

6.3 Enregistrement de chemin

6.3.1 Traitement de RRO

Un message Resv pour un P2P LSP contient un chemin enregistré si le LSR d'entrée a demandé l'enregistrement de chemin en incluant un RRO dans le message Path d'origine. La même règle est utilisée durant la signalisation des LSP P2MP. C'est-à-dire, l'inclusion d'un RRO dans le message Path utilisé pour signaler un ou plusieurs sous LSP S2L déclenche l'inclusion d'un chemin enregistré pour chaque sous LSP dans le message Resv.

Le chemin enregistré du premier sous LSP S2L est codé dans le RRO. Les chemins enregistrés supplémentaires pour les sous LSP S2L suivants sont codés dans les objets P2MP_SECONDARY_RECORD_ROUTE (SRRO). Leur format est spécifié au paragraphe 19.5. Chaque objet S2L_SUB_LSP dans un Resv est associé à un RRO ou SRRO. Le premier objet S2L_SUB_LSP (pour le premier sous LSP S2L) est associé au RRO. Les objets S2L_SUB_LSP suivants (pour les sous LSP S2L suivants) sont chacun suivi par un SRRO qui contient le chemin enregistré pour ce sous LSP S2L de la feuille à une branche. Le nœud d'entrée peut alors utiliser le RRO et les SRRO pour déterminer le chemin de bout en bout pour chaque sous LSP S2L.

6.4 Style de réservation

Les considérations sur le style de réservation dans un message Resv s'appliquent comme décrit dans la [RFC3209]. Le style de réservation dans les messages Resv peut être FF ou SE. Tous les LSP P2MP qui appartiennent au même tunnel P2MP DOIVENT être signalés avec le même style de réservation. Sans considération de si le style de réservation est FF ou SE, le sous LSP S2L qui appartient au même LSP P2MP DEVRAIT partager les étiquettes lorsque elles partagent les bonds. Si le sous LSP S2L qui appartient au même LSP P2MP partage les étiquettes, il DOIT alors partager les ressources. Si le style de réservation est FF, alors les sous LSP S2L qui appartiennent à des LSP P2MP différents NE DOIVENT PAS partager de ressources ou d'étiquettes. Si le style de réservation est SE, alors les sous LSP S2L qui appartiennent à des LSP P2MP

différents et au même tunnel P2MP DEVRAIENT partager les ressources lorsque ils partagent les bonds, mais ils NE DOIVENT PAS partager les étiquetttes dans des environnements de paquet.

7. Message PathTear

7.1 Format du message PathTear

Le format du message PathTear est comme suit :

```
<Message PathTear> ::= <En-tête commun> [ <INTEGRITY> ]
                       [ [ <MESSAGE_ID_ACK> |
                          <MESSAGE_ID_NACK> ... ]
                       [ <MESSAGE_ID> ]
                       <SESSION> <RSVP_HOP>
                       [ <descripteur d'envoyeur> ]
                       [ <liste de descripteurs de sous LSP S2L> ]

<liste de descripteurs de sous LSP S2L> ::= <S2L_SUB_LSP> [ <liste de descripteurs de sous LSP S2L> ]
```

La définition de <descripteur d'envoyeur> n'est pas changée par le présent document.

7.2 Élagage

L'opération de suppression de LSR de sortie d'un LSP P2MP existant est appelée élagage. Cette opération permet aux nœuds de sortie d'être retirés d'un LSP P2MP à différents moments. Ce paragraphe décrit les mécanismes pour effectuer l'élagage.

7.2.1 Suppression implicite de sous LSP S2L

La suppression implicite utilise le traitement standard de message RSVP. Selon le traitement standard RSVP, un sous LSP S2L peut être retiré d'un LSP TE P2MP en envoyant un message modifié pour le chemin ou un message Resv qui annonçait précédemment le sous LSP S2L. Ce message DOIT faire la liste de tous les sous LSP S2L qui ne sont pas supprimés. Quand on utilise cette approche, un nœud traitant un message qui supprime un sous LSP S2L d'un LSP TE P2MP DOIT s'assurer que le sous LSP S2L n'est pas inclus dans un autre état de chemin associé à la session avant d'interrompre le chemin de données à cette sortie. Tous le reste du traitement de message est inchangé.

Quand la suppression implicite est utilisée pour supprimer un ou plusieurs sous LSP S2L, en modifiant un message Path, un LSR de transit peut devoir générer un message PathTear vers l'aval pour supprimer un ou plusieurs de ces sous LSP S2L. Cela peut arriver si par suite de la suppression implicite de sous LSP S2L il ne reste plus de sous LSP S2L à envoyer dans le message Path correspondant vers l'aval.

7.2.2 Suppression explicite de sous LSP S2L

La suppression explicite de sous LSP S2L repose sur la génération d'un message PathTear pour le message Path correspondant. Le message PathTear est signalé avec les objets SESSION et SENDER_TEMPLATE correspondant au LSP P2MP et avec le couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe> correspondant au message Path. Cette approche DEVRAIT être utilisée quand toutes les sorties signalées par un message Path doivent être supprimées du LSP P2MP. Les autres sous LSP S2L, provenant d'autres sous groupes signalés en utilisant d'autres messages Path, ne sont pas affectés par le PathTear.

Un LSR de transit qui propage le message PathTear vers l'aval DOIT s'assurer qu'il règle le couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe> dans le message PathTear aux valeurs utilisées dans le message Path qui a été utilisé pour établir le sous LSP S2L à supprimer. Le LSR de transit peut devoir générer plusieurs messages PathTear pour un message PathTear entrant si la fragmentation de transit a été effectuée pour le message Path entrant correspondant.

Quand un LSP P2MP est supprimé par l'entrée, un message PathTear DOIT être généré pour chaque message Path utilisé pour signaler le LSP P2MP.

8. Messages Notify et ResvConf

8.1 Messages Notify

L'objet Demande de notification et le message Notify sont décrits dans la [RFC3473]. L'objet et le message DEVRONT tous deux être psis en charge pour la livraison vers l'amont et vers l'aval de la notification. Le traitement non détaillé dans cette section DOIT se conformer à la [RFC3473].

1. Notification vers l'amont

Si un LSR de transit règle l'identifiant d'origine de sous groupe dans l'objet SENDER_TEMPLATE d'un message Path à sa propre adresse, et si le message Path entrant porte un objet Demande de notification, alors ce LSR DOIT changer l'adresse du nœud Notify en l'objet Demande de notification à sa propre adresse dans le message Path qu'il envoie.

Si ce LSR reçoit ensuite un message Notify correspondant d'un LSR aval, il DOIT alors :
- envoyer un message Notify vers l'amont à l'adresse du nœud Notify que le LSR a reçu dans le message Path.
- traiter les champs de sous groupe de l'objet SENDER_TEMPLATE sur le message Notify reçu, et modifier leurs valeurs, dans le message Notify qui est transmis, pour correspondre aux valeurs du champ de sous groupe dans le message Path original reçu de l'amont.

Le receveur d'un message Notify (de l'amont) DOIT identifier l'état référencé dans ce message sur la base des objets SESSION et SENDER_TEMPLATE.

2. Notification vers l'aval

Un LSR de transit règle l'identifiant d'origine de sous groupe dans le ou les objets FILTER_SPEC d'un message Resv à la valeur qui a été reçue dans le message Path correspondant. Si le message Resv entrant porte un objet Demande de notification, alors :
- Si il y a au moins un autre message Resv entrant qui porte un objet Demande de notification, et si le LSR fusionne ces messages Resv en un seul message Resv qui est envoyé vers l'amont, le LSR DOIT régler l'adresse de nœud notifié dans l'objet Demande de notification à son identifiant de routeur.
- Autrement, si le LSR règle l'identifiant d'origine de sous groupe (dans le message Path sortant qui correspond au message Resv reçu) à sa propre adresse, le LSR DOIT régler l'adresse de nœud notifié dans l'objet Demande de notification à son identifiant de routeur.
- Autrement, le LSR DOIT propager l'objet Demande de notification inchangé, dans le message Resv qu'il envoie vers l'amont.

Si ce LSR reçoit ensuite un message Notify correspondant d'un LSR en amont, alors il DOIT :
- traiter les champs de sous groupe de l'objet FILTER_SPEC dans le message Notify reçu, et modifier leurs valeurs, dans le message Notify qui est transmis, pour correspondre aux valeurs du champ de sous groupe du message Path original envoyé vers l'aval par ce LSR,
- envoyer un message Notify vers l'aval à l'adresse du nœud notifié que le LSR a reçu dans le message Resv.

Le receveur d'un message Notify (vers l'aval) DOIT identifier l'état référencé dans le message sur la base des objets SESSION et FILTER_SPEC.

La conséquence de ces règles pour un LSP P2MP est qu'un message Notify vers l'amont généré sur une branche va résulter en un Notify livré à l'adresse du nœud notifié vers l'amont. Le receveur du message Notify NE DOIT PAS supposer que le message Notify s'applique à toutes les sorties vers l'aval, mais DOIT examiner les informations dans le message pour déterminer à quelles sorties le message s'applique.

Les messages Notify vers l'aval DOIVENT être répliqué aux LSR de branches en accord avec les objets Demande de notification reçus sur les messages Resv. Certaines branches vers l'aval pourraient ne pas demander de messages Notify, mais toutes celles qui ont demandé des messages Notify DOIVENT les recevoir.

8.2 Messages ResvConf

Les messages ResvConf sont décrits dans la [RFC2205]. Le traitement de ResvConf dans les [RFC3473] et [RFC3209] est pris directement de la [RFC2205]. Un LSR de sortie PEUT inclure un objet RESV_CONFIRM qui contient l'adresse du LSR de sortie. L'objet et le message DEVRONT être pris en charge pour la confirmation de la réception du message Resv dans les LSP TE P2MP. Le traitement non détaillé dans ce paragraphe DOIT se conformer à la [RFC2205].

Un LSR de transit règle l'identifiant d'origine de sous groupe dans le ou les objets FILTER_SPEC d'un message Resv à la valeur reçue dans le message Path correspondant. Si un message Resv entrant correspondant à un seul message Path porte un objet RESV_CONFIRM, alors le LSR DOIT inclure un objet RESV_CONFIRM dans le message Resv correspondant qu'il envoie vers l'amont et :

- Si il y a au moins un autre message Resv entrant qui porte un objet RESV_CONFIRM, et si le LSR fusionne ces messages Resv en un seul message Resv qui est envoyé vers l'amont, le LSR DOIT régler l'adresse du receveur dans l'objet RESV_CONFIRM à son identifiant de routeur.
- Si le LSR règle l'identifiant d'origine de sous groupe (dans le message Path sortant qui correspond au message Resv reçu) à sa propre adresse, le LSR DOIT régler l'adresse du receveur dans l'objet RESV_CONFIRM à son identifiant de routeur.
- Autrement, le LSR DOIT propager l'objet RESV_CONFIRM inchangé, dans le message Resv qu'il envoie vers l'amont.

Si ce LSR reçoit ensuite un message ResvConf correspondant d'un LSR en amont, alors il DOIT :

- traiter les champs de sous groupe dans l'objet FILTER_SPEC dans le message ResvConf reçu, et modifier leurs valeurs, dans le message ResvConf qui est transmis, pour qu'elles correspondent aux valeurs de champs de sous groupe du message Path original envoyé vers l'aval par ce LSR ;
- envoyer un message ResvConf vers l'aval à l'adresse de receveur que le LSR a reçue dans l'objet RESV_CONFIRM dans le message Resv.

Le receveur d'un message ResvConf DOIT identifier l'état référencé dans ce message sur la base des objets SESSION et FILTER_SPEC.

La conséquence de ces règles pour un LSP P2MP est qu'un message ResvConf généré à l'entrée va résulter en la livraison d'un message ResvConf à la branche et ensuite à l'adresse de receveur dans l'objet RESV_CONFIRM original. Le receveur d'un message ResvConf NE DOIT PAS supposer que le message ResvConf devrait être envoyé à toutes les sorties vers l'aval, mais il DOIT répliquer le message conformément aux objets RESV_CONFIRM reçus dans les messages Resv. Certaines branches aval pourraient ne pas demander de messages ResvConf, et les messages ResvConf NE DEVRAIENT PAS être envoyés sur ces branches. Un tel message DOIT être envoyé à toutes les branches aval qui ont demandé des messages ResvConf.

9. Réduction de rafraîchissement

Les procédures de réduction de rafraîchissement décrites dans la [RFC2961] sont également applicables aux LSP P2MP décrits dans le présent document. La réduction de rafraîchissement s'applique aux messages individuels et à l'état qu'ils installent/maintiennent, et cela continue d'être le cas pour les LSP P2MP.

10. Gestion d'état

L'état signalé par un message Path P2MP est identifié par une mise en œuvre locale en utilisant le triplet <Identifiant P2MP, Identifiant de tunnel, Identifiant étendu de tunnel> au titre de l'objet SESSION et le quadruplet <Adresse d'envoyeur de tunnel, Identifiant de LSP, Identifiant d'origine de sous groupe, Identifiant de sous groupe> au titre de l'objet SENDER_TEMPLATE.

Les informations supplémentaires signalées dans le message Path/Resv font partie de l'état créé par une mise en œuvre locale. Cela inclut les objets PHOP/NHOP et SENDER_TSPEC/FILTER_SPEC.

10.1 Mise à jour incrémentaire d'état

RSVP (comme défini dans la [RFC2205] et comme étendu par RSVP-TE [RFC3209] et GMPLS [RFC3473]) utilise la même approche de base pour l'état de communication et de synchronisation – à savoir, l'état complet est envoyé dans chaque message d'annonce d'état. Selon la [RFC2205], les messages Path et Resv sont idempotents. Aussi, la [RFC2961] catégorise les messages RSVP en deux types (messages de déclenchement et messages de rafraîchissement) et améliore le traitement de message RSVP et adapte les rafraîchissements d'état, mais ne modifie pas la nature de l'annonce d'état complet des messages Path et Resv. La nature de l'annonce d'état complet des messages Path et Resv a de nombreux avantages, mais aussi quelques inconvénients. Un inconvénient notable est quand une modification incrémentaire est faite à un état annoncé précédemment. Dans ce cas, il y a des frais généraux de message à l'envoi de l'état complet et au coût de son traitement. Il est souhaitable de surmonter cet inconvénient et d'ajouter/supprimer le sous LSP S2L de/vers un LSP

P2MP en mettant à jour de façon incrémentaire l'état existant.

Il est possible d'utiliser les procédures décrites dans le présent document pour permettre au sous LSP S2L d'être ajouté ou supprimé de façon incrémentaire au/du LSP P2MP en permettant qu'un message Path ou PathTear change de façon incrémentaire l'état du chemin existant du LSP P2MP.

Comme décrit au paragraphe 5.2, plusieurs messages Path peuvent être utilisés pour signaler un LSP P2MP. Les messages Path sont distingués par des couples <Identifiant d'origine de sous groupe, Identifiant de sous groupe> différents dans l'objet SENDER_TEMPLATE. Afin d'effectuer un ajout incrémentaire d'état de sous LSP S2L, un message Path séparé avec un nouvel identifiant de sous groupe est utilisé pour ajouter le nouveau sous LSP S2L, par le LSR d'entrée. L'identifiant d'origine de sous groupe DOIT être réglé à l'identifiant de routeur TE [RFC3477] du nœud qui établit l'identifiant de sous groupe.

Cela maintient la nature idempotente des messages Path RSVP, évite de garder trace de l'expiration d'état des sous LSP S2L individuels, et donne la capacité d'effectuer des mises à jour incrémentaires d'état de LSP P2MP.

10.2 Combinaison de plusieurs messages Path

Il y a un compromis entre le nombre de messages Path utilisé par l'entrée pour maintenir le LSP P2MP et le traitement imposé par les messages d'état complet quand on ajoute un sous LSP S2L à un message Path existant. Il est possible de combiner les sous LSP S2L précédemment annoncés dans différents messages Path dans un seul message Path afin de réduire le nombre de messages Path nécessaire pour maintenir le LSP P2MP. Ceci peut aussi être fait par un nœud de transit qui a effectué la fragmentation et qui plus tard est capable de combiner plusieurs messages Path qu'il a générés dans un seul message Path. Cela peut arriver quand un ou plusieurs sous LSP S2L sont élagués des états de chemin existants.

Le nouveau message Path est signalé par le nœud qui combine plusieurs messages Path avec tous les sous LSP S2L qui sont combinés dans un seul message Path. Ce message Path PEUT contenir de nouvelles valeurs de champ de sous groupe. Quand un nouveau message Path et Resv qui est signalé pour un sous LSP S2L existant est reçu par un LSR de transit, l'état incluant la nouvelle instance du sous LSP S2L est créé.

Le sous LSP S2L DEVRAIT continuer d'être annoncé dans les anciens et nouveaux messages Path jusqu'à ce qu'un message Resv faisant la liste des sous LSP S2L et correspondant au nouveau message Path soit reçu par le nœud combinant. Donc, jusqu'à ce moment, l'état pour le sous LSP S2L DEVRAIT être maintenu au titre de l'état de chemin pour l'ancien et le nouveau message Path (voir le paragraphe 3.1.3 de la [RFC2205]). À ce moment, le sous LSP S2L DEVRAIT être supprimé de l'ancien état de chemin en utilisant les procédures de la Section 7.

Un message Path avec un identifiant de sous groupe (n) peut signaler un ensemble de sous LSP S2L qui appartiennent partiellement ou entièrement à un identifiant de sous groupe (i) déjà existant, ou à un nouvel ensemble strictement sans chevauchement de sous LSP S2L. Un nouveau message Path reçu avec l'objet SESSION et le triplet <Adresse de tunnel envoyeur, Identifiant de LSP, Identifiant d'origine de sous groupe> correspondant à l'état de chemin existant qui porte un identifiant de sous groupe identique ou différent, appelé l'identifiant de sous groupe (n), est traité comme suit :

1) Si le Sub-Group_ID(i) = Sub-Group_ID(n), alors les sous LSP S2L qui sont dans les deux Sub-Group_ID(i) et Sub-Group_ID(n) sont rafraîchis. Les nouveaux sous LSP S2L sont ajoutés à l'état de chemin du Sub-Group_ID(i) et les sous LSP S2L qui sont dans le Sub-Group_ID(i) mais pas dans le Sub-Group_ID(n) sont supprimés de l'état de chemin du Sub-Group_ID(i).

2) Si le Sub-Group_ID(i) != Sub-Group_ID(n), alors un nouvel état de chemin Sub-Group_ID(n) est créé pour les sous LSP S2L signalés par le Sub-Group_ID(n). Les sous LSP S2L dans l'état de chemin existant des Sub-Group_ID(i) (qui sont ou non dans le Sub-Group_ID(n) du nouveau message Path reçu) sont laissés non modifiés (voir ci-dessus).

11. Traitement des erreurs

Les messages PathErr et ResvErr sont traités selon les procédures de RSVP-TE. Noter qu'un LSR, à réception d'un message PathErr/ResvErr pour un sous LSP S2L particulier, change l'état seulement pour ce sous LSP S2L. Donc les autres sous LSP S2L ne sont pas impactés. Si le nœud d'entrée demande "l'intégrité de LSP", une erreur rapportée sur une branche d'un LSP TE P2MP pour un sous LSP S2L particulier peut changer l'état de tous les autres sous LSP S2L du même LSP TE

P2MP. Ceci est mieux expliqué au paragraphe 11.3.

11.1 Messages PathErr

Le message PathErr va include un ou plusieurs objets S2L_SUB_LSP. Le format résultant modifié pour un message PathErr est :

```
<Message PathErr> ::=  <En-tête commun> [ <INTEGRITY> ]
                       [ [<MESSAGE_ID_ACK> |
                          <MESSAGE_ID_NACK>] ... ]
                       [ <MESSAGE_ID> ]
                         <SESSION> <ERROR_SPEC>
                       [ <ACCEPTABLE_LABEL_SET> ... ]
                       [ <POLICY_DATA> ... ]
                         <descripteur d'envoyeur>
                       [ <liste de descripteurs de sous LSP S2L> ]
```

La génération du message PathErr n'est pas modifiee, mais les nœuds qui établissent le champ Origine de sous groupe et propagent un message PathErr reçu vers l'amont DOIVENT remplacer les champs de sous groupe reçus dans le message PathErr par la valeur qui a été reçue dans les champs de sous groupe du message Path provenant du voisin amont. Noter que le receveur d'un message PathErr est capable d'identifier le message Path erroné sortant, et l'interface sortante, sur la base des champs de sous groupe reçus dans le message PathErr. La liste des descripteurs de sous LSP S2L est définie au paragraphe 5.1.

11.2 Messages ResvErr

Le message ResvErr va inclure un ou plusieurs objets S2L_SUB_LSP. Le format résultant modifié pour un message ResvErr est:

```
<Message ResvErr> ::= <En-tête commun> [ <INTEGRITY> ]
                      [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                        [ <MESSAGE_ID> ]
                          <SESSION> <RSVP_HOP>
                          <ERROR_SPEC> [ <SCOPE> ]
                        [ <ACCEPTABLE_LABEL_SET> ... ]
                        [ <POLICY_DATA> ... ]
                          <STYLE> <liste des descripteurs de flux>
```

La génération de message ResvErr n'est pas modifiée, mais les nœuds qui établissent le champ d'origine de sous groupe et propagent un message ResvErr reçu vers l'aval DOIVENT remplacer les champs de sous groupe reçus dans le message ResvErr par la valeur qui était établie dans les champs de sous groupe du message Path envoyé au voisin en aval. Noter que le receveur d'un message ResvErr est capable d'identifier le message Resv erroné sortant, et l'interface sortante, sur la base des champs de sous groupe reçus dans le message ResvErr. La liste des descripteurs de flux est définie au paragraphe 6.1.

11.3. Traitement d'une défaillance de branche

Durant l'établissement et durant le fonctionnement normal, des messages PathErr peuvent être reçus à un nœud de branche. Dans tous les cas, un message PathErr reçu est d'abord traité selon les règles de traitement standard. C'est-à-dire, le message PathErr est envoyé bond par bond au LSR d'entrée/de branche pour ce message Path. Les nœuds intermédiaires jusqu'à ce LSR d'entrée/de branche PEUVENT inspecter ce message mais pas effectuer d'action sur lui. Le comportement d'un LSR de branche qui génère un message PathErr est sous le contrôle du LSR d'entrée.

Le comportement par défaut est que le message PathErr n'a pas le fanion Path_State_Removed établi. Cependant, si le LSR d'entrée a établi le fanion "intégrité de LSP" sur le message Path (voir l'objet LSP_REQUIRED_ATTRIBUTE au paragraphe 5.2.4) et si le fanion Path_State_Removed est pris en charge, le LSR qui génère une PathErr pour rapporter la défaillance d'une branche du LSP P2MP DEVRAIT établie le fanion Path_State_Removed.

Un LSR de branche qui reçoit un message PathErr durant l'établissement de LSP avec le fanion Path_State_Removed établi DOIT agir en accord avec les souhaits du LSR d'entrée. Le comportement par défaut est que le LSR de branche mette à

zéro le fanion Path_State_Removed sur le message PathErr et l'envoie plus loin vers l'amont. Il ne supprime aucune autre branche du LSP. Cependant, si le fanion "intégrité de LSP" est établi sur le message Path, le LSR de branche DOIT envoyer un PathTear sur toutes les autres branches vers l'aval et envoyer le message PathErr vers l'amont avec le fanion Path_State_Removed établi.

Un LSR de branche qui reçoit un message PathErr avec le fanion Path_State_Removed à zéro DOIT agir en accord avec les souhaits du LSR d'entrée. Le comportement par défaut est que le LSR de branche transmette le PathErr vers l'amont et ne fasse rien d'autre. Cependant, si le fanion "intégrité de LSP" est établi sur le message Path, le LSR de branche DOIT envoyer le PathTear sur toutes les branches vers l'aval et envoyer le PathErr vers l'amont avec le fanion Path_State_Removed établi (conformément à la [RFC3473]).

Dans tous les cas, le message PathErr transmis par un LSR de branche DOIT contenir l'identification du sous LSP S2L et les chemins explicites de toutes les branches qui sont rapportées par les messages PathErr reçus et toutes les branches qui sont explicitement supprimées par le LSR de branche.

12. Changement d'état administratif

Un nœud de branche qui reçoit un objet ADMIN_STATUS le traite normalement et aussi le relaie dans un message Path sur toutes les branches. Tous les messages Path peuvent être envoyés concurremment aux voisins vers l'aval.

Les nœuds en aval traitent le changement de l'objet ADMIN_STATUS conformément à la [RFC3473], incluant la génération de messages Resv. Quand le dernier objet ADMIN_STATUS reçu vers l'amont a le bit R établi, les nœuds de branche attendent que soit reçu un message Resv avec un objet ADMIN_STATUS correspondant (ou un message PathErr ou ResvTear correspondant) sur toutes les branches avant de relayer un message Resv correspondant vers l'amont.

13. Allocation d'étiquette sur des LAN avec plusieurs nœuds en aval

Un LSR de branche d'un LSP P2MP sur un segment de LAN Ethernet DEVRAIT envoyer une copie du trafic de données à chaque LSR connecté vers l'aval sur ce LAN pour ce LSP P2MP. Les procédures pour empêcher la réplication du trafic étiqueté MPLS dans un tel cas sortent du domaine d'application du présent document.

14. Réoptimisation de LSP et sous LSP P2MP

Il est possible de changer le chemin utilisé par les LSP P2MP pour atteindre les destinations du tunnel P2MP. Deux méthodes peuvent être utilisées pour ce faire. La première est "faire avant de couper" *(make-before-break)*, définie dans la [RFC3209], et la seconde utilise les sous groupes définis ci-dessus.

14.1 Faire avant de couper

Dans ce cas, tous les sous LSP S2L sont signalés avec un identifiant différent de LSP par le LSR d'entrée et suivent la procédure de "faire avant de couper" définie dans la [RFC3209]. Donc, un nouveau LSP P2MP est établi. Chaque sous LSP S2L est signalé avec un identifiant de LSP différent, correspondant au nouveau LSP P2MP. Après le déplacement du trafic au nouveau LSP P2MP, l'entrée peut supprimer l'ancien LSP P2MP. Cette procédure peut être utilisée pour ré-optimiser le chemin du LSP P2MP entier ou les chemins sur un sous ensemble des destinations du LSP P2MP. Quand on modifie juste une portion du LSP P2MP, cette approche exige que le LSP P2MP entier soit re-signalé.

14.2 Réoptimisation fondée sur le sous groupe

Tout nœud peut initier la ré-optimisation d'un ensemble de sous LSP S2L en utilisant la mise à jour incrémentaire d'état, et ensuite de combiner facultativement plusieurs messages Path.

Pour modifier le chemin pris par un ensemble particulier de sous LSP S2L, le nœud initiateur du changement de chemin initie un ou plusieurs messages Path séparés pour le même LSP P2MP, chacun avec un nouvel identifiant de sous groupe. La génération de ces messages Path, chacun avec un ou plusieurs sous LSP S2L, suit les procédures du paragraphe 5.2.

Comme c'est le cas au paragraphe 10.2, une sortie particulière continue d'être annoncée dans les deux messages Path, ancien et nouveau, jusqu'à ce qu'un message Resv faisant la liste des sorties et correspondant au nouveau message Path soit reçu par le nœud qui ré-optimise. À ce point, la sortie DEVRAIT être supprimée de l'ancien état de chemin en utilisant les procédures de la Section 7. La ré-optimisation de sous arborescence est alors achevée.

La ré-optimisation fondée sur le sous groupe peut résulter en une duplication transitoire des données car les nouveaux messages Path pour un ensemble de sous LSP S2L peuvent transiter par un ou plusieurs nœuds avec le vieil état de chemin pour le même ensemble de sous LSP S2L.

Comme c'est toujours le cas, un nœud peut choisir de combiner plusieurs messages Path comme décrit au paragraphe 10.2.

15. Réacheminement rapide

Les extensions de la [RFC4090] peuvent être utilisées pour effectuer le réacheminement rapide pour le mécanisme décrit dans le présent document quand il est appliqué dans les réseaux de paquets. GMPLS introduit d'autres techniques de protection qui peuvent être appliquées aux environnements de paquet et non de paquet [RFC4873], mais qui ne sont pas discutés plus avant dans le présent document. Cette Section s'applique seulement aux LSR qui prennent en charge la [RFC4090].

Cette Section utilise la terminologie définie dans la [RFC4090], et les procédures de réacheminement rapide définies dans la [RFC4090] DOIVENT être suivies sauf mention contraire ci-dessous. Les LSR d'extrémité de tête et de transit DOIVENT suivre le traitement des objets SESSION_ATTRIBUTE et FAST_REROUTE comme spécifié dans la [RFC4090] pour chaque message Path et sous LSP S2L d'un LSP P2MP. Chaque sous LSP S2L d'un LSP P2MP DOIT avoir les mêmes caractéristiques de protection. Le traitement de RRO DOIT aussi s'appliquer au SRRO sauf modification ci-après.

Les paragraphes qui suivent décrivent comment le réacheminement rapide peut être appliqué aux LSP TE P2MP de MPLS dans tous les principaux scénarios de fonctionnement. Le présent document ne décrit pas le détail des étapes de traitement pour chaque cas d'utilisation imaginable, et ils pourraient être décrits dans de futurs documents, en tant que de besoin.

15.1 Sauvegarde de facilités

La sauvegade de facilités peut être utilisée pour la protection de liaison ou de nœud des LSR sur le chemin d'un LSP P2MP. Les étiquettes vers l'aval DOIVENT être apprises par le point de réparation local (PLR, *Point of Local Repair*), comme spécifié dans la [RFC4090], de l'étiquette correspondant au sous LSP S2L dans le message RESV. Le traitement des SERO signalés dans un tunnel de sauvegarde DOIT suivre le traitement d'ERO de tunnel de sauvegarde décrit dans la [RFC4090].

15.1.1 Protection de liaison

Si la protection de liaison est désirée, un tunnel de contournement DOIT être utilisé pour protéger la liaison entre le PLR et le prochain bond. Donc tous les sous LSP S2L qui utilisent la liaison DEVRAIENT être protégés en cas de défaillance de la liaison. Noter que tous ces sous LSP S2L appartenant à une instance particulière d'un tunnel P2MP DEVRAIENT partager la même étiquette sortante sur la liaison entre le PLR et le prochain bond conformément au paragraphe 5.2.1. C'est l'étiquette de LSP P2MP sur la liaison. La mise en pile des étiquettes est utilisée pour envoyer les données pour chaque LSP P2MP dans le tunnel de contournement. L'étiquette interne est l'étiquette de LSP P2MP allouée par le prochain bond.

Durant une défaillance, les messages Path pour chaque sous LSP S2L affecté DOIVENT être envoyés au point de fusion (MP, *Merge Point*) par le PLR. Il est RECOMMANDÉ que le PLR utilise la méthode spécifique du gabarit de l'envoyeur pour identifier ces messages Path. Donc, le PLR va établir l'adresse de source dans le gabarit d'envoyeur à une adresse de PLR local.

Le MP DOIT utiliser l'identifiant de LSP pour identifier le sous LSP S2L correspondant. Le MP NE DOIT PAS utiliser le couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe> lorsque il identifie le sous LSP S2L correspondant. Afin de poursuivre le traitement d'un sous LSP S2L, le MP DOIT déterminer le sous LSP S2L protégé en utilisant l'identifiant de LSP et l'objet S2L_SUB_LSP.

15.1.2 Protection de nœud

Si la protection de nœud est désirée, le PLR DEVRAIT utiliser un ou plusieurs tunnels de contournement P2P pour protéger l'ensemble de sous LSP S2L qui transitent par le nœud protégé. Chacun de ces tunnels de contournement P2P DOIT couper le chemin du sous LSP S2L qu'ils protègent sur un LSR qui est en aval du nœud protégé. Cela contraint l'ensemble de sous LSP S2L à être sauvegardés via ce tunnel de contournement pour les sous LSP S2L qui passent à travers un MP commun vers l'aval. Ce MP est la destination du tunnel de contournement. Quand le PLR transmet les données entrantes pour un LSP P2MP dans le tunnel de contournement, l'étiquette externe est l'étiquette du tunnel de contournement et l'étiquette interne est l'étiquette allouée par le MP de l'ensemble de sous LSP S2L appartenant à ce LSP P2MP.

Après la détection d'une défaillance du nœud protégé, le PLR DOIT envoyer un ou plusieurs messages Path pour tous les sous LSP S2L protégés au MP du sous LSP S2L protégé. Il est RECOMMANDÉ que le PLR utilise la méthode spécifique du gabarit de l'envoyeur pour identifier ces messages Path. Donc le PLR va établir l'adresse de source dans le gabarit d'envoyeur à une adresse de PLR local. Le MP DOIT utiliser l'identifiant de LSP pour identifier le sous LSP S2L correspondant. Le MP NE DOIT PAS utiliser le couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe> lorsque il identifie le sous LSP S2L correspondant parce que l'identifiant d'origine de sous groupe pourrait être changé par un LSR qui est outrepassé par le tunnel de contournement. Afin de poursuivre le traitement d'un sous LSP S2L le MP DOIT déterminer le sous LSP S2L protégé en utilisant l'identifiant de LSP et l'objet S2L_SUB_LSP.

Noter que la protection de nœud PEUT exiger que le PLR soit capable de faire une branche dans le plan des données, car plusieurs tunnels de contournement peuvent être nécessaires pour sauvegarder l'ensemble de sous LSP S2L qui passent à travers le nœud protégé. Si le PLR n'est pas capable de faire un embranchement, le mécanisme de protection de nœud décrit ici n'est applicable qu'aux seuls cas où tous les sous LSP S2L passant à travers le nœud protégé passent aussi à travers un seul MP qui est vers l'aval du nœud protégé. Un PLR DOIT établir le fanion Protection de nœud dans le RRO/SRRO, comme spécifié dans la [RFC4090]. Si un PLR n'est pas capable de faire un embranchement, et si un ou plusieurs sous LSP S2L sont ajoutés à une arborescence en P2MP, et si ces sous LSP S2L ne transitent pas par le MP existant vers l'aval du nœud protégé, alors le PLR DOIT rétablir ce fanion.

Il est à noter que les procédures de ce paragraphe exigent des tunnels de contournement P2P. Les procédures pour utiliser des tunnels de contournement P2MP feront l'objet d'études ultérieures.

15.2 Sauvegarde biunivoque

La sauvegarde biunivoque, telle que décrite dans la [RFC4090], peut être utilisée pour protéger un sous LSP S2L particulier contre une défaillance de liaison et de prochain bond. La protection peut être utilisée pour un ou plusieurs sous LSP S2L entre le PLR et le prochain bond. Tous les sous LSP S2L correspondant à la même instance de tunnel P2MP entre le PLR et le prochain bond DEVRAIENT partager la même étiquette de LSP P2MP, conformément au paragraphe 5.2.1. Tous ces sous LSP S2L appartenant à un LSP P2MP DOIVENT être protégés.

Le sous LSP S2L de sauvegarde peut traverser des prochains bonds différents au PLR. Donc, l'ensemble d'étiquettes sortantes et de prochains bonds pour un LSP P2MP, au PLR, peut changer une fois que la protection est déclenchée. Si on considère un LSP P2MP qui utilise un seul prochain bond et une seule étiquette entre le PLR et le prochain bond du PLR, cela ne peut plus être le cas une fois que la protection est déclenchée. Cela PEUT exiger qu'un PLR soit capable d'embranchement dans le plan des données. Si le PLR n'est pas capable d'embranchement, les mécanismes de sauvegarde biunivoque décrits ici sont seulement applicables aux cas où tous les sous LSP S2L de sauvegarde passent par le même prochain bond vers l'aval du PLR. Les procédures pour la sauvegarde biunivoque quand un PLR n'est pas capable d'embranchement et quand tous les sous LSP S2L de sauvegarde ne passent pas à travers le même prochain bond vers l'aval feront l'objet d'études ultérieures.

Il est recommandé que la méthode spécifique du chemin soit utilisée pour identifier un sous LSP S2L de sauvegarde. Donc, l'objet DETOUR DEVRAIT être inséré dans le message Path de sauvegarde. Un sous LSP S2L de sauvegarde DOIT être traité comme appartenant à une instance de tunnel P2MP différente de celle spécifiée par l'identifiant de LSP. De plus plusieurs sous LSP S2L de sauvegarde DOIVENT être traités comme faisant partie de la même instance de tunnel P2MP si ils ont le même identifiant de LSP et le même objet DETOUR. Noter que, comme spécifié à la Section 4, les sous LSP S2L entre des instances différentes de tunnel P2MP utilisent des étiquettes différentes.

Si il y a seulement un sous LSP S2L dans le message Path, l'objet DETOUR s'applique à ce sous LSP. Si il y a plusieurs sous LSP S2L dans le message Path, l'objet DETOUR s'applique à tous les sous LSP S2L.

16. Prise en charge des LSR qui n'ont pas de capacité P2MP

Il se peut que certains des LSR dans un réseau soient capables de traiter les extensions P2MP décrites dans le présent document, mais ne prennent pas en charge l'embranchement P2MP dans le plan des données. Si il est demandé à un tel LSR de devenir un LSR de branche par un message Path reçu, il DOIT répondre par un message PathErr portant le code d'erreur "Erreur d'acheminement" et la valeur d'erreur "Incable d'embranchement".

Il est aussi concevable que certains des LSR, dans un réseau qui déploie une capacité P2MP, ne puissent pas prendre en charge les extensions décrites dans le présent document. Si un message Path pour l'établissement d'un LSP P2MP atteint un tel LSR, celui ci va le rejeter avec une PathErr parce qu'il ne va pas reconnaître le C-Type de l'objet SESSION P2MP.

Les LSR qui ne prennent pas en charge les extensions P2MP du présent document peuvent être inclus comme des LSR de transit par l'utilisation du raccordement de LSP [RFC5150] et de la hiérarchie de LSP [RFC4206]. Noter que les LSR dont il est exigé qu'ils jouent un autre rôle dans le réseau (entrée, branche ou sortie) DOIVENT prendre en charge les extensions définies dans le présent document.

L'utilisation du raccordement de LSP et de la hiérarchie de LSP [RFC4206] permet aux LSP P2MP d'être construits dans un tel environnement. Un segment de LSP P2P est signalé à partir du dernier bond capable de P2MP qui est à l'amont d'un LSR traditionnel jusqu'au premier bond capable de P2MP qui est en aval de lui. Cela suppose que les LSR intermédiaires traditionnels sont des LSR de transit : ils ne peuvent pas agir comme des points d'embranchement P2MP. Les LSR de transit le long de ce segment de LSP ne traitent pas les messages de plan de contrôle associés aux LSP P2MP. De plus, ces LSR de transit n'ont pas besoin d'avoir des capacités de plan des données P2MP car ils ont seulement besoin de traiter les données qui appartiennent au segment de LSP P2P. Donc, ces LSR de transit n'ont pas besoin de prendre en charge MPLS P2MP. Ce segment de LSP P2P est raccordé au LSP P2MP entrant. Après que le segment de LSP P2P est établi, le message Path P2MP est envoyé au prochain LSR capable de P2MP comme un message Path dirigé. Le nouveau LSR à capacité P2MP raccorde le segment de LSP P2P au LSP P2MP sortant.

Dans les réseaux de paquets, le sous LSP S2L peut être incorporé dans le LSP P2P externe. Donc, la mise en pile d'étiquettes peut être utilisée pour permettre l'utilisation du même segment de LSP pour plusieurs LSP P2MP. Les considérations et procédures de raccordement et d'incorporation sont décrites dans les [RFC5150] et [RFC4206].

Il peut y avoir des frais généraux pour un opérateur à configurer à l'avance les segments de LSP P2P quand il désire prendre en charge les LSR traditionnels. Il peut être souhaitable de le faire de façon dynamique. L'entrée peut utiliser les extensions d'IGP pour déterminer les LSR capables de P2MP [RFC5073). Il peut utiliser ces informations pour calculer les chemins de sous LSP S2L de telle façon qu'ils évitent les LSR traditionnels sans capacité P2MP. L'objet Chemin explicite d'un chemin de sous LSP S2L peut contenir des bonds lâches si il y a des LSR traditionnels le long du chemin. Le chemin explicite correspondant contient une liste des objets jusqu'à un LSR à capacité P2MP qui est adjacent à un LSR traditionnel suivi par un objet lâche avec l'adresse du prochain LSR à capacité P2MP. Le LSR à capacité P2MP étend le bond lâche en utilisant sa base de données d'ingénierie du trafic (TED, *Traffic Engineering Database*). En faisant cela, il détermine si l'expansion du bond lâche exige qu'un LSP P2P tunnelle à travers le LSR traditionnel. Si il existe un tel LSP P2P, il utilise ce LSP P2P. Autrement, il établit le LSP P2P. Le message Path P2MP est envoyé au prochain LSR à capacité P2MP en utilisant une signalisation de non adjacence.

Le LSR à capacité P2MP qui initie le message de signalisation de non adjacence au prochain LSR à capacité P2MP peut devoir employer un mécanisme de détection rapide (comme celui de la [RFC5880) ou de la [RFC5884]) vers le prochain LSR à capacité P2MP. Cela peut être nécessaire que pour le message Path dirigé vers l'extrémité de tête utilise le réacheminement rapide de protection de nœud quand le nœud protégé est la queue du message Path dirigé.

Noter que les LSR traditionnels le long d'un segment de LSP P2P ne peuvent pas effectuer la protection de nœud de la queue du segment de LSP P2P.

17. Réduction dans le traitement du plan de contrôle avec hiérarchie de LSP

Il est possible de tirer parti de la hiérarchie de LSP [RFC4206] lors de l'établissement de LSP P2MP, comme décrit dans la Section précédente, pour réduire le traitement de plan de contrôle le long des LSR de transit qui sont à capacité P2MP. Ceci est applicable seulement dans les environnements où la hiérarchie de LSP peut être utilisée. Les LSR de transit le long d'un segment de LSP P2P utilisé par un LSP P2MP ne traitent pas les messages de plan de contrôle associés aux LSP P2MP. En fait, ils ignorent ces messages car ils sont tunnelés sur le segment de LSP P2P. Cela réduit la quantité de traitement de plan

de contrôle requis sur ces LSR de transit.

Noter que les LSP P2P peuvent être établis dynamiquement comme décrit à la Section précédente ou préconfigurés. Par exemple, dans la Figure 2 de l'Appendice A, PE1 peut établir un LSP P2P à P1 et l'utiliser comme un segment de LSP. Les messages Path pour PE3 et PE4 peuvent alors être tunnelés sur le segment de LSP. Donc, P3 ignore le LSP P2MP et ne traite pas les messages de contrôle P2MP.

18. Re-fusion et chevauchement de LSP P2MP

Cette Section détaille les procédures pour détecter et traiter la re-fusion et le croisement. Le terme de "re-fusion" se réfère au cas d'un nœud d'entrée ou de transit qui crée une branche d'un LSP P2MP, une branche de re-fusion, qui fait une intersection du LSP P2MP à un autre nœud plus loin dans l'arborescence. Cela peut arriver à cause d'événements comme une erreur de calcul de chemin, une erreur de configuration manuelle, ou un changement de la topologie de réseau durant l'établissement du LSP P2MP. Si les procédures détaillées dans cette Section ne sont pas suivies, la duplication des données va en résulter.

Le terme de "croisement" se réfère au cas d'un nœud d'entrée ou de transit qui crée une branche d'un LSP P2MP, une branche de croisement, qui coupe le LSP P2MP à un autre nœud plus loin dans l'arborescence. Il est improbable qu'il re-fusionne parce que, au nœud d'intersection, la branche de croisement a une interface sortante différente ainsi qu'une interface entrante différente. Cela peut être nécessaire dans certaines combinaisons de topologie et de technologie ; par exemple, dans un réseau optique transparent dans lequel des longueurs d'onde différentes sont nécessaires pour atteindre des nœuds d'extrémité différents.

Normalement, un LSP P2MP a une seule interface entrante sur laquelle sont reçues toutes les données pour le LSP P2MP. L'interface entrante est identifiée par l'objet IF_ID RSVP_HOP, si il est présent, et par l'interface sur laquelle le message Path a été reçu si l'objet IF_ID RSVP_HOP n'est pas présent. Cependant, dans le cas de réacheminement dynamique de LSP, l'interface entrante peut changer.

De façon similaire, dans les deux cas de re-fusion et de croisement, un nœud va recevoir un message Path pour un certain LSP P2MP identifiant une interface entrante différente pour les données, et le nœud doit être capable de distinguer entre le réacheminement dynamique de LSP et la re-fusion/croisement.

Le "faire avant de couper" représente encore un autre cas similaire mais différent, en ce que l'interface entrante associée au LSP P2MP "faire avant de couper" peut être differente de celle associée au LSP P2MP original. Cependant, les deux LSP P2MP vont être traités comme des LSP distincts (mais en rapport) parce que ils vont avoir des valeurs de champ Identifiant de LSP différentes dans leur objet SENDER_TEMPLATE.

18.1 Procédures

Quand un nœud reçoit un message Path, il DOIT vérifier si il a un état correspondant pour le LSP P2MP. L'état correspondant est identifié en comparant les objets SESSION et SENDER_TEMPLATE dans le message Path reçu avec les objets SESSION et SENDER_TEMPLATE de chaque état de chemin de LSP P2MP conservé localement. L'identifiant P2MP, l'identifiant de tunnel, et l'identifiant étendu de tunnel dans l'objet SESSION et l'adresse d'envoyeur et l'identifiant de LSP dans l'objet SENDER_TEMPLATE sont utilisés pour la comparaison. Si le nœud a un état correspondant, et si l'interface entrante pour le message Path reçu est différente de l'interface entrante de l'état de chemin du LSP P2MP correspondant, alors le nœud DOIT déterminer si il traite un réacheminement dynamique de LSP ou une re-fusion/croisement.

Le réacheminement dynamique de LSP est identifié en vérifiant si il y a une intersection entre l'ensemble des objets S2L_SUB_LSP associés à l'état de chemin de LSP P2MP correspondant et l'ensemble des objets S2L_SUB_LSP dans le message Path reçu. Si il y a une intersection, un réacheminement dynamique s'est alors produit. Si il n'y a pas d'intersection entre les deux ensembles d'objets S2L_SUB_LSP, alors soit une re-fusion, soit un croisement s'est produit. (Noter que dans le cas de réacheminement dynamique de LSP, les messages Path pour les membres non interséquants de l'ensemble de S2L_SUB_LSP associés à l'état de chemin de LSP P2MP correspondant vont être reçus ensuite sur la nouvelle interface entrante.)

Afin d'identifier le cas de re-fusion, le nœud qui traite le message Path reçu DOIT identifier les interfaces sortantes associées à l'état de chemin P2MP correspondant. La re-fusion s'est produite si il y a une intersection entre l'ensemble des

interfaces sortantes associées à l'état de chemin de LSP P2MP correspondant et l'ensemble des interfaces sortantes dans le message Path reçu.

18.1.1 Procédures de re-fusion

Il y a deux approches pour traiter le cas de re-fusion. Dans la première, le nœud qui détecte le cas de re-fusion, c'est-à-dire, le nœud de re-fusion, permet au cas de re-fusion de persister, mais les données provenant de toutes les interfaces entrantes sauf une sont éliminées au nœud de re-fusion. Dans la seconde, le nœud de re-fusion initie la suppression des branches de re-fusion via la signalisation. L'approche utilisée est une affaire de politique locale.

Un nœud DOIT prendre en charge les deux approches et DOIT permettre une configuration par l'utilisateur de l'approche utilisée.

Quand il est configuré à permettre à un cas de re-fusion de persister, le nœud de re-fusion DOIT valider la cohérence entre les objets inclus dans le message Path reçu et l'état de chemin de LSP P2MP correspondant. Toute incohérence DOIT résulter en un message PathErr envoyé au bond précédent du message Path reçu. Le code d'erreur est réglé à "Problème d'acheminement", et la valeur d'erreur est réglée à "Discordance de paramètre de re-fusion P2MP".

Si il n'y a pas d'incohérence, le nœud fusionne logiquement, du point de vue vers l'aval, l'état de contrôle du message Path entrant avec l'état de chemin de LSP P2MP correspondant. Spécifiquement, les procédures relatives au traitement des messages reçus de l'amont NE DOIVENT PAS être modifiées du point de vue de l'amont ; cela inclut le traitement relatif aux temporisations de rafraîchissement et d'état. En plus des procédures standard vers l'amont, le nœud DOIT s'assurer que chaque objet reçu de l'amont est représenté de façon appropriée dans l'ensemble des messages Path envoyés vers l'aval. Par exemple, la <liste des descripteurs de sous LSP S2L> reçue DOIT être incluse dans l'ensemble des messages Path sortants. Si des objets NOTIFY_REQUEST sont présents, alors les procédures définies à la Section 8 DOIVENT être suivies pour tous les messages Path et Resv. Un traitement particulier est aussi requis pour Resv. Spécifiquement, tout message Resv reçu de l'aval DOIT être transposé en un message Resv sortant qui est envoyé au bond précédent du message Path reçu. En pratique, ceci se traduit par la décomposition complète de la <liste des descripteurs de sous LSP S2L> en sous ensembles qui correspondent aux messages Path entrants, et ensuite en la construction d'un message Resv sortant pour chaque message Path entrant.

Quand il est configuré à permettre au cas de re-fusion de persister, le nœud de re-fusion reçoit les données associées au LSP P2MP sur plusieurs interfaces entrantes, mais il DOIT seulement envoyer les données d'une de ces interfaces à ses interfaces sortantes. C'est-à-dire, le nœud DOIT éliminer les données provenant de toutes les interfaces entrantes sauf une. Cela assure que des données dupliquées ne sont pas envoyées sur une interface sortante. Le mécanisme utilisé pour choisir l'interface entrante est spécifique de la mise en œuvre et sort du domaine d'application du présent document.

Quand il est configuré à corriger la branche de re-fusion via la signalisation, le nœud de re-fusion DOIT envoyer un message PathErr correspondant au message Path reçu. Le message PathErr DOIT inclure tous les objets normalement inclus dans un message PathErr, ainsi que un ou plusieurs objets S2L_SUB_LSP provenant de l'ensemble de sous LSP associés à l'état de chemin du LSP P2MP correspondant. Un minimum de trois objets S2L_SUB_LSP est RECOMMANDÉ. Cela va permettre au nœud qui a causé la re-fusion d'identifier l'état de chemin sortant associé à la portion valide du LSP P2MP. L'ensemble d'objets S2L_SUB_LSP dans le message Path reçu DOIT aussi être inclus. Le message PathErr DOIT inclure le code d'erreur "Problème d'acheminement" et la valeur d'erreur de "Détection de re-fusion P2MP". Le nœud PEUT établir le fanion Path_State_Removed [RFC3473]. Comme c'est toujours le cas, le message PathErr est envoyé au bond précédent du message Path reçu.

Un nœud qui reçoit un message PathErr contenant la valeur d'erreur "Problème d'acheminement/Détection de re-fusion P2MP" DOIT déterminer si il est le nœud qui a créé le cas de re-fusion. Ceci est fait en vérifiant si il y a une intersection entre l'ensemble d'objets S2L_SUB_LSP associés à l'état de chemin du LSP P2MP correspondant et l'ensemble des objets S2L_SUB_LSP d'autres branches dans le message PathErr reçu. Si il y en a une, alors le nœud a créé le cas de re-fusion. Les objets S2L_SUB_LSP d'autres branches sont ceux qui sont inclus dans le message PathErr par le nœud qui détecte le cas de re-fusion, qui ont été pris dans l'état de chemin du LSP P2MP correspondant. De tels objets S2L_SUB_LSP sont identifiables car il ne sont pas être inclus dans le message Path associé au message PathErr reçu. Voir au paragraphe 11.1 les détails de la façon dont une telle association est identifiée.

Le nœud DEVRAIT supprimer le cas de re-fusion en déplaçant les objets S2L_SUB_LSP inclus dans le message Path associé au message PathErr reçu à l'interface sortante associée à l'état de chemin du LSP P2MP correspondant. Un message Path déclencheur pour les objets S2L_SUB_LSP déplacés est alors envoyé via cette interface sortante. Si le message PathErr reçu n'avait pas le fanion Path_State_Removed établi, le nœud DEVRAIT envoyer un PathTear via l'interface

sortante associée à la branche de re-fusion.

Si l'utilisation d'une nouvelle interface sortante viole une ou plusieurs contraintes de SERO, alors un message PathErr contenant les sorties associées et tout objet S2L_SUB_LSP identifié DEVRAIT être généré avec le code d'erreur "Problème d'acheminement" et la valeur d'erreur de "L'ERO a résulté en une re-fusion".

Le seul cas où ce traitement ca échouer est quand tous les objets S2L_SUB_LSP de la liste sont supprimés avant que le message PathErr soit propagé à l'entrée. Dans ce cas, tout le traitement va être corigé sur la prochaine transmission (rafraîchissement ou déclenchement) du message Path en cause.

19. Objets de message nouveaux et mis à jour

Cette Section présente les formats des objets RSVP tels que modifiés par le présent document.

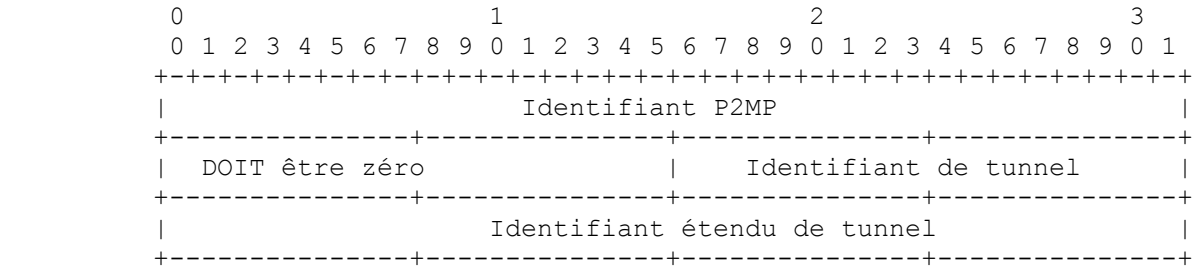
19.1 Object SESSION

Un objet SESSION de LSP P2MP est utilisé. Cet object utilise le C-Num *(numéro de classe)* de SESSION existant. De nouveaux C-Type *(type de classe)* sont définis pour traiter un identifiant de destination logique P2MP de tunnel P2MP. Cet objet SESSION a une structure similaire à celle de l'objet SESSION RSVP-TE point à point existant. Cependant l'adresse de destination est réglée à l'identifiant de P2MP au lieu de l'adresse de point d'extrémité de tunnel rn envoi individuel. Tous les sous LSP S2L qui font partie du même LSP P2MP partagent le même objet SESSION. Cet objet SESSION identifie le tunnel P2MP.

La combinaison de l'objet SESSION, de l'objet SENDER_TEMPLATE et de l'objet S2L_SUB_LSP identifie chaque sous LSP S2L. Ceci suit la notion existante de P2P RSVP-TE en utilisant l'objet SESSION pour identifier un tunnel P2P, qui à son tour peut contenir plusieurs LSP, chacun distingué par un unique objet SENDER_TEMPLATE.

19.1.1 Objet SESSION de Tunnel IPv4 de LSP P2MP

Classe = SESSION, C-Type P2MP_LSP_TUNNEL_IPv4 = 13



Identifiant P2MP : identifiant de 32 bits utilisé dans l'objet SESSION, qui reste constant sur la vie du tunnel P2MP. Il code l'identifiant de P2MP qui est unique sur la portée du LSR d'entrée.

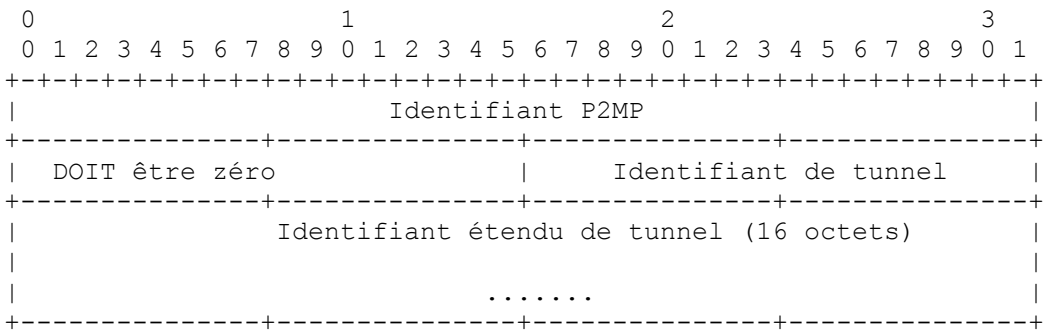
Identifiant de tunnel : identifiant de 16 bits utilisé dans l'objet SESSION qui reste constant sur la vie du tunnel P2MP.

Identifiant étendu de tunnel : identifiant de 32 bits utilisé dans l'objet SESSION, qui reste constant sur la vie du tunnel P2MP. Les LSR d'entrée qui souhaitent avoir un identifiant unique au monde pour le tunnel P2MP DEVRAIT placer ici leur adresse d'envoyeur de tunnel. Une combinaison de cette adresse, de l'identifiant P2MP, et de l'identifiant de tunnel donne un identifiant unique au monde pour le tunnel P2MP.

19.1.2 Objet SESSION de Tunnel IPv6 de LSP P2MP

C'est le même que l'objet SESSION de LSP P2MP IPv4 avec la différence que l'identifiant étendu de tunnel peut être réglé à un identifiant de 16 octets [RFC3209].

Classe = SESSION, C-Type P2MP_LSP_TUNNEL_IPv6 = 14



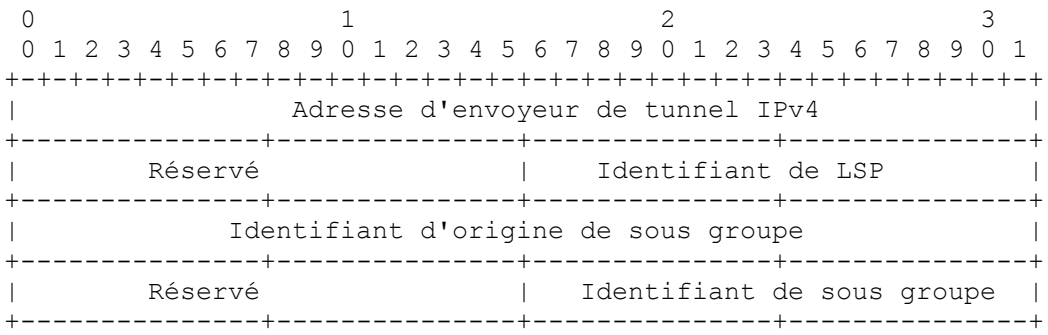
19.2 Objet SENDER_TEMPLATE

L'objet SENDER_TEMPLATE contient l'adresse de source du LSR d'entrée. L'identifiant de LSP peut être changé pour permettre à un envoyeur de partager des ressources avec lui-même. Donc, plusieurs instances du tunnel P2MP peuvent être créées, chacune avec un identifiant de LSP différent. Les instances peuvent partager des ressources avec chaque autre. Les sous LSP S2L correspondant à une instance particulière utilisent le même identifiant de LSP.

Comme décrit au paragraphe 4.2, il est nécessaire de distinguer les différents messages Path qui sont utilisés pour signaler l'état pour le même LSP P2MP en utilisant un couple <Identifiant d'origine de sous groupe, Identifiant de sous groupe>. L'objet SENDER_TEMPLATE est modifié pour porter cette information comme montré ci-dessous.

19.2.1 Objet SENDER_TEMPLATE de tunnel IPv4 de LSP P2MP

Classe = SENDER_TEMPLATE, C-Type P2MP_LSP_TUNNEL_IPv4 = 12



Adresse d'envoyeur de tunnel IPv4 : voir la [RFC3209].

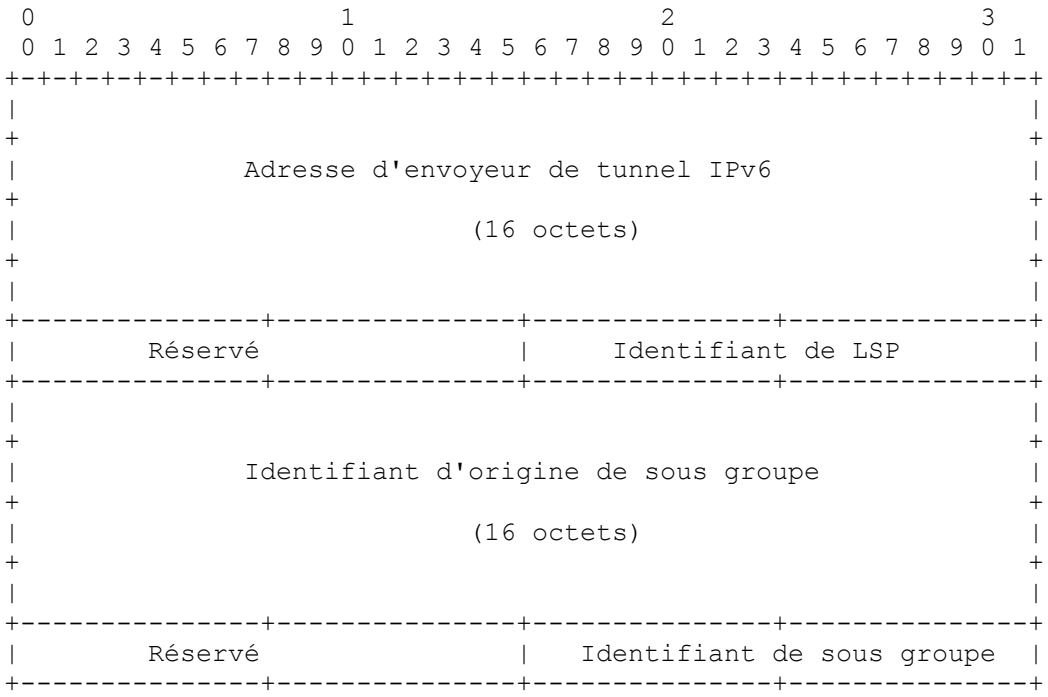
Identifiant de LSP : voir la [RFC3209].

Identifiant d'origine de sous groupe : réglé à l'identifiant de routeur TE du LSR qui génère le message Path. C'est soit le LSR d'entrée soit un LSR qui regénère le message Path avec son propre identifiant d'origine de sous groupe.

Identifiant de sous groupe : un identifiant d'un message Path utilisé pour différentier plusieurs messages Path qui signalent l'état pour le même LSP P2MP. Cela peut être vu comme identifiant un groupe d'un ou plusieurs nœuds de sortie ciblés par ce message Path.

19.2.2 Objet SENDER_TEMPLATE de tunnel IPv6 de LSP P2MP

Classe = SENDER_TEMPLATE, C-Type P2MP_LSP_TUNNEL_IPv6 = 13



Adresse d'envoyeur de tunnel IPv6 : voir la [RFC3209].

Identifiant de LSP : voir la [RFC3209].

Identifiant d'origine de sous groupe : l'identifiant d'origine de sous groupe est réglé à l'identifiant de routeur TE IPv6 du LSR qui génère le message Path. C'est soit le LSR d'entrée, soit un LSR qui regénère le message Path avec son propre identifiant d'origine de sous groupe.

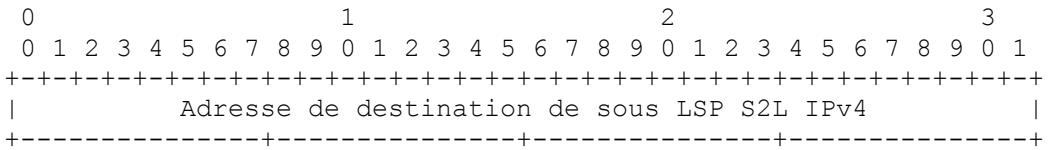
Identifiant de sous groupe : comme ci-dessus au paragraphe 19.2.1.

19.3 Objet S2L_SUB_LSP

Un objet S2L_SUB_LSP identifie un sous LSP S2L particulier appartenant au LSP P2MP.

19.3.1 Objet S2L_SUB_LSP IPv4

Classe de S2L_SUB_LSP = 50, C-Type S2L_SUB_LSP_IPv4 = 1

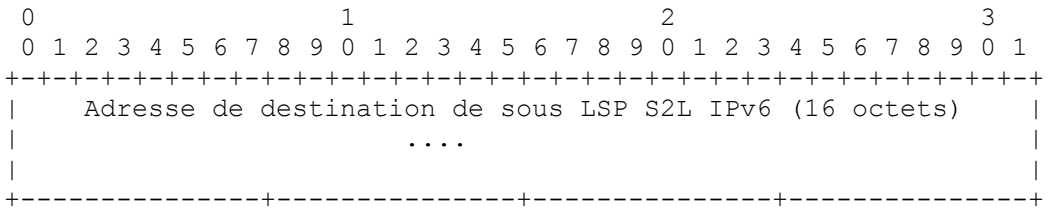


Adresse de destination de sous LSP IPv4 : adresse IPv4 du sous LSP S2L de destination.

19.3.2 Objet S2L_SUB_LSP IPv6

Classe de S2L_SUB_LSP = 50, C-Type S2L_SUB_LSP_IPv6 = 2

C'est le même objet que le sous LSP S2L IPv4, avec la différence que l'adresse de destination est une adresse IPv6 de 16 octets.



19.4 Objet FILTER_SPEC

L'objet FILTER_SPEC est canonique de l'objet P2MP SENDER_TEMPLATE.

19.4.1 Objet LSP P2MP_IPv4 FILTER_SPEC

Classe = FILTER_SPEC, C-Type LSP P2MP_IPv4 = 12

Le format de l'objet de LSP P2MP_IPv4 FILTER_SPEC est identique à celui l'objet de LSP P2MP_IPv4 SENDER_TEMPLATE.

19.4.2 Objet LSP P2MP_IPv6 FILTER_SPEC

Classe = FILTER_SPEC, C-Type LSP P2MP_IPv6 = 13

Le format de l'objet LSP P2MP_IPv6 FILTER_SPEC est identique à celui de l'objet LSP P2MP_IPv6 SENDER_TEMPLATE.

19.5 Objet P2MP SECONDARY_EXPLICIT_ROUTE (SERO)

L'objet P2MP SECONDARY_EXPLICIT_ROUTE (SERO) est défini comme identique à l'ERO. La classe du SERO P2MP est la même que celle du SERO défini dans la [RFC4873]. Le SERO P2MP utilise un nouveau C-Type = 2. Les sous objets sont identiques à ceux définis pour l'ERO.

19.6 Objet P2MP SECONDARY_RECORD_ROUTE (SRRO)

L'objet P2MP SECONDARY_RECORD_ROUTE (SRRO) est défini comme identique à l'ERO. La classe du SERO P2MP est la même que celle du SRRO défini dans la [RFC4873]. Le SERO P2MP utilise un nouveau C-Type = 2. Les sous objets sont identiques à ceux défini pour le RRO.

20. Considérations relatives à l'IANA

20.1 Nouveaux numéros de classes

L'IANA a alloué les numéros de classe suivants aux nouvelles classes d'objets introduites. Les types de classe pour chacune d'elles sont à allouer par action de normalisation. Les types de sous objet pour P2MP SECONDARY_EXPLICIT_ROUTE et P2MP_SECONDARY_RECORD_ROUTE suivent les mêmes considérations relatives à l'IANA que ceux de ERO et RRO dans la [RFC3209].

Nom de classe 50 = S2L_SUB_LSP
C-Type :
1 S2L_SUB_LSP_IPv4
2 S2L_SUB_LSP_IPv6

20.2 Nouveaux types de classes

L'IANA a alloué les valeurs de C-Type suivantes :

Nom de classe = SESSION

C-Type :
13: P2MP_LSP_TUNNEL_IPv4
14 : P2MP_LSP_TUNNEL_IPv6

Nom de classe = SENDER_TEMPLATE
C-Type :
12 : P2MP_LSP_TUNNEL_IPv4
13 : P2MP_LSP_TUNNEL_IPv6

Nom de classe = FILTER_SPEC
C-Type :
12 : LSP P2MP_IPv4
13 : LSP P2MP_IPv6

Nom de classe = SECONDARY_EXPLICIT_ROUTE (défini dans la [RFC4873])
C-Type = 2 P2MP SECONDARY_EXPLICIT_ROUTE

Nom de classe = SECONDARY_RECORD_ROUTE (défini dans la [RFC4873])
C-Type = 2 P2MP_SECONDARY_RECORD_ROUTE

20.3 Nouvelles valeurs d'erreur

Cinq nouvelles valeurs d'erreur sont définies avec le code d'erreur "Problème d'acheminement". L'IANA a alloué leurs valeurs comme suit :

La valeur d'erreur "Incapable de branchement" indique qu'une branche de P2MP ne peut pas être formée par le LSR qui fait rapport. L'IANA a alloué la valeur 23 à cette valeur d'erreur.

La valeur d'erreur "Intégrité de LSP non prise en charge" indique qu'une branche de P2MP ne prend pas en charge la fonction d'intégrité de LSP demandée. L'IANA a alloué la valeur 24 à cette valeur d'erreur.

La valeur d'erreur "Re-fusion de P2MP détectée" indique qu'un nœud a détecté une re-fusion. L'IANA a alloué la valeur 25 à cette valeur d'erreur.

La valeur d'erreur "Discordance de paramètre de re-fusion P2MP" est décrite à la Section 18. L'IANA a alloué la valeur 26 à cette valeur d'erreur.

La valeur d'erreur "L'ERO a résulté en une re-fusion" est décrite à la Section 18. L'IANA a alloué la valeur 27 à cette valeur d'erreur.

20.4 Fanions d'attributs de LSP

Il a été demandé à l'IANA de gérer l'espace de fanions dans le TLV Fanions d'attributs porté dans les objets LSP_REQUIRED_ATTRIBUTES [RFC4420]. Le présent document définit un nouveau fanion comme suit :

Numéro de bit : 3
Signification : Intégrité de LSP exigée
Utilisé dans les fanions d'attributs sur Path : Oui
Utilisé dans les fanions d'attributs sur Resv : Non
Utilisé dans les fanions d'attributs sur RRO : Non
Paragraphe de référence dans le présent document : 5.2.4

21. Considérations sur la sécurité

En principe, le présent document n'introduit aucun nouveau problème de sécurité au delà de ceux identifiés dans les [RFC3209], [RFC3473], et [RFC4206]. La [RFC2205] spécifie les mécanismes d'intégrité de message pour la signalisation RSVP bond par bond. Ces mécanismes s'appliquent à la signalisation RSVP-TE P2MP bond par bond dans le présent document. De plus, les [RFC3473] et [RFC4206] spécifient les mécanismes de sécurité pour la signalisation RSVP-TE

P2MP non bond par bond. Ces mécanismes s'appliquent à la signalisation RSVP-TE P2MP non bond par bond spécifiée dans le présent document, en particulier aux sections 16 et 17.

Une administration peut souhaiter limiter le domaine sur lequel les tunnels TE P2MP peuvent être établis. Cela peut être accompli en établissent des filtres sur divers accès pour refuser l'action sur un message Path RSVP avec un objet SESSION de type P2MP_LSP_IPv4 ou P2MP_LSP_IPv6.

Le LSR d'entrée d'un LSP TE P2MP détermine les feuilles du LSP TE P2MP sur la base de l'application du LSP TE P2MP. La spécification de comment de telles applications vont utiliser un LSP TE P2MP sort du domaine d'application du présent document. Les applications DOIVENT fournir un mécanisme pour notifier au LSR d'entrée les feuilles appropriées pour le LSP P2MP. Les spécifications d'applications dans l'IETF DOIVENT spécifier ce mécanisme dans un détail suffisant pour qu'un LSR d'entrée d'un fabricant puisse être utilisé avec une mise en œuvre d'application fournie par un autre fabricant. La configuration manuelle des paramètres de sécurité quand d'autres paramètres sont auto découverts est généralement insuffisante pour satisfaire aux exigences de sécurité et d'interopérabilité des spécifications de l'IETF.

22. Remerciements

Le présent document est le produit du travail de nombreuses personnes. La liste des contributeurs est à l'Appendice B.

Merci à Yakov Rekhter, Der-Hwa Gan, Arthi Ayyanger, et Nischal Sheth de leurs suggestions et commentaires. Merci aussi à Dino Farninacci et Benjamin Niven de leurs commentaires.

23. Références

23.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[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, RFC6780)) *(P.S.)*

[RFC2961] L. Berger et autres, "Extensions de réduction de redondance de rafraîchissement pour RSVP", avril 2001. (*MàJ par* RFC5063) *(P.S.)*

[RFC3031] E. Rosen, A. Viswanathan, R. Callon, "Architecture de commutation d'étiquettes multi protocoles", janvier 2001. *(P.S.)* (*MàJ par la* RFC6790)

[RFC3209] D. Awduche, et autres, "RSVP-TE : Extensions à RSVP pour les tunnels LSP", décembre 2001. (*Mise à jour par* RFC3936, RFC4420, RFC4874, RFC5151, RFC5420, RFC6790)

[RFC3471] L. Berger, éd., "Commutation d'étiquettes multi-protocoles généralisée (GMPLS) : description fonctionnelle de la signalisation", janvier 2003. (*MàJ par* RFC4201, RFC4328, RFC4872, RFC8359) *(P.S.)*

[RFC3473] L. Berger, "Extensions d'ingénierie de protocole - trafic de signalisation de réservation de ressource (RSVP-TE) de commutation d'étiquettes multi-protocoles généralisée (GMPLS)", janvier 2003. *(P.S., MàJ par 4003, 4201, 4420, 4783, 4784, 4873, 4974, 5063, 5151,* 8359)

[RFC3477] K. Kompella, Y. Rekhter, "Signalisation des liaisons non numérotées dans le protocole de réservation de ressource – ingénierie du trafic (RSVP-TE)", janvier 2003. *(P.S.)*

[RFC4090] P. Pan et autres, "Extensions de réacheminement rapide à RSVP-TE pour les tunnels de LSP", mai 2005. *(P.S. ; MàJ par* RFC8271, RFC8537, RFC8796)

[RFC4206] K. Kompella, Y. Rekhter, "Hiérarchie de chemins commutés par étiquettes (LSP) avec l'ingénierie de trafic (TE) de la commutation généralisée d'étiquettes multi-protocoles (GMPLS)", octobre 2005. *(P.S.)*

[RFC4420] A. Farrel et autres, "Codage des attributs pour l'établissement de chemin à commutation d'étiquettes (LSP) de la commutation d'étiquettes multiprotocoles (MPLS) en utilisant le protocole de réservation de ressources avec extensions d'ingénierie de trafic (RSVP-TE)", février 2006. (*MàJ* RFC3209, RFC3473) *(P.S. : Obsolète, voir RFC5420,)*

[RFC4873] L. Berger et autres, "Récupération de segment GMPLS", mai 2007. (*MàJ* RFC3473, RFC4872) *(P.S.)*

23.2 Références pour information

[RFC4003] L. Berger, "Procédure de signalisation GMPLS pour contrôle de sortie", février 2005. *(P.S.)*

[RFC4461] S. Yasukawa, éd., "Exigences de signalisation pour les chemins à commutation d'étiquettes (LPS) de MPLS à ingénierie de trafic en point à multipoint", avril 2006. *(Information)*

[RFC5880] D. Katz, D. Ward, "Détection de transmission bidirectionnelle (BFD)", juin 2010. *(P. S. ; MàJ par* RFC7880)

[RFC5884] R. Aggarwal, K. Kompella, T. Nadeau, G. Swallow, "Détection de transmission bidirectionnelle (BFD) pour chemins à commutation d'étiquette (LSP) MPLS", juin 2010. (*MàJ* RFC1122). *(P.S.)*

[RFC5150] A. Ayyangar et autres, "Raccordement de chemin à commutation d'étiquette avec la commutation généralisée d'étiquettes multiprotocoles à ingénierie de trafic (GMPLS-TE)", février 2008. *(P.S.)*

[RFC5073] J.P. Vasseur et J.L. Le Roux, éd., "Extensions au protocole d'acheminement IGP pour la découverte de capacités de nœud d'ingénierie de trafic", décembre 2007. *(P.S.)*

Appendice A. Exemple d'établissement de LSP P2MP

Voici un exemple d'établissement d'un LSP P2MP en utilisant les procédures décrites dans le présent document.

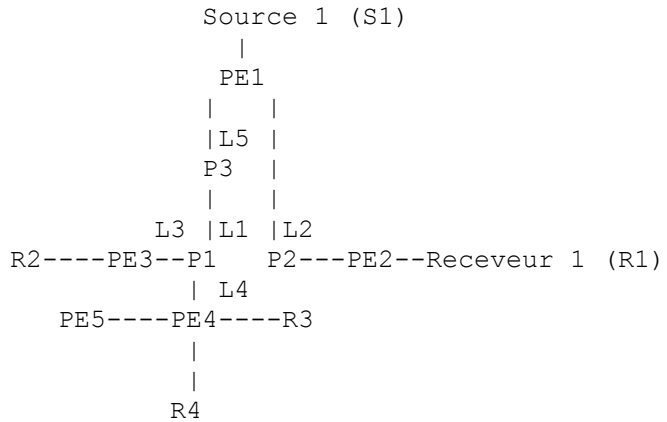


Figure 2.

Le mécanisme est expliqué en utilisant la Figure 2. PE1 est le LSR d'entrée. PE2, PE3, et PE4 sont les LSR de sortie.

a) PE1 apprend que PE2, PE3, et PE4 sont intéressés à se joindre à une arborescence P2MP avec un identifiant de P2MP de P2MP ID1. On suppose que PE1 apprend les LSR de sortie à des moments différents.

b) PE1 calcule le chemin P2P pour atteindre PE2.

c) PE1 établit le sous LSP S2L pour PE2 le long de <PE1, P2, PE2>.

d) PE1 calcule le chemin P2P pour atteindre PE3 quand il découvre PE3. Ce chemin est calculé pour partager les mêmes liaisons lorsque possible avec le sous LSP pour PE2 car ils appartiennent à la même session P2MP.

e) PE1 établit le sous LSP S2L pour PE3 le long de <PE1, P3, P1, PE3>.

f) PE1 calcule le chemin P2P pour atteindre PE4 quand il découvre PE4. Ce chemin est calculé pour partager les mêmes liaisons lorsque possible avec les sous LSP pour PE2 et PE3 car ils appartiennent à la même session P2MP.

g) PE1 signale le message Path pour le sous LSP PE4 le long de <PE1, P3, P1, PE4>.

h) P1 reçoit un message Resv de PE4 avec l'étiquette L4. Il avait précédemment reçu un message Resv de PE3 avec l'étiquette L3. Il a alloué une étiquette L1 pour le sous LSP à PE3. Il utilise la même étiquette et envoie les messages Resv à P3. Noter qu'il peut seulement envoyer un message Resv avec plusieurs descripteurs de flux dans la liste des descripteurs de flux. Si c'est le cas, et si le style FF est utilisé, le descripteur de flux FF va contenir la liste des descripteurs de sous LSP S2L avec deux entrées : une pour PE4 et l'autre pour PE3. Pour le style SE, la spécification de filtre SE va contenir cette liste des descripteurs de sous LSP S2L. P1 crée aussi une transposition d'étiquette de (L1 -> {L3, L4}). P3 utilise l'étiquette L5 et envoie le message Resv à PE1, avec l'étiquette L5. Il réutilise la transposition d'étiquette de {L5 -> L1}.

Appendice B. Contributeurs

John Drake
Boeing
mél : john.E.Drake2@boeing.com

Alan Kullberg
Motorola Computer Group
mél : alan.kullberg@motorola.com

Lou Berger
LabN Consulting, L.L.C.
mél : lberger@labn.net

Liming Wei
Redback Networks
mél : lwei@redback.com

Adrian Farrel
Old Dog Consulting
mél : adrian@olddog.co.uk

George Apostolopoulos
Redback Networks
mél : georgeap@redback.com

Kireeti Kompella
Juniper Networks
mél : kireeti@juniper.net

George Swallow
Cisco Systems, Inc.
mél : swallow@cisco.com

JP Vasseur
Cisco Systems, Inc.
mél : jpv@cisco.com

Markus Jork
Avici Systems
mél : mjork@avici.com

Dean Cheng
Cisco Systems Inc.
mél : dcheng@cisco.com

Hisashi Kojima
NTT Corporation
mél : kojima.hisashi@lab.ntt.co.jp

Andrew G. Malis
Tellabs
mél : Andy.Malis@tellabs.com

Koji Sugisono
NTT Corporation
mél : sugisono.koji@lab.ntt.co.jp

Masanori Uga
NTT Corporation
mél : uga.masanori@lab.ntt.co.jp

Igor Bryskin
Movaz Networks, Inc.
mél : ibryskin@movaz.com

Jean-Louis Le Roux
France Telecom
mél : jeanlouis.leroux@francetelecom.com

Adresse des éditeurs

Rahul Aggarwal
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
mél : rahul@juniper.net

Seisho Yasukawa
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-Shi, Tokyo 180-8585 Japan
mél :yasukawa.seisho@lab.ntt.co.jp

Dimitri Papadimitriou
Alcatel
Francis Wellesplein 1,
B-2018 Antwerpen, Belgium
mél : Dimitri.Papadimitriou@alcatel-lucent.be

Déclaration complète de droits de reproduction

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