

Groupe de travail Réseau
Request for Comments : 3219
 Catégorie : En cours de normalisation
 Traduction Claude Brière de L'Isle

J. Rosenberg, dynamicsoft
 H. Salama, Cisco Systems
 M. Squire, Hatteras Networks
 janvier 2002

Acheminement téléphonique sur IP (TRIP)

Statut de ce mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître 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 322 (06/09/2002) et 2634 (16/11/2010))

Notice de copyright

Copyright (C) The Internet Society (2002). Tous droits réservés.

Résumé

Le présent document présente l'acheminement téléphonique sur IP (TRIP, *Telephony Routing over IP*). TRIP est un protocole inter domaines administratifs fondé sur la politique pour annoncer l'accessibilité des destinations téléphoniques entre serveurs de localisation, et pour annoncer les attributs des chemins vers ces destinations. Le fonctionnement de TRIP est indépendant de tout protocole de signalisation, donc TRIP peut servir de protocole d'acheminement téléphonique pour tout protocole de signalisation.

Le protocole de passerelle frontière (BGP-4, *Border Gateway Protocol*) est utilisé pour distribuer les informations d'acheminement entre les domaines administratifs. TRIP est utilisé pour distribuer les informations d'acheminement téléphonique entre les domaines administratifs de téléphonie. Les similarités entre les deux protocoles sont évidentes, et donc TRIP est modélisé sur BGP-4.

Table des matières

1. Terminologie et définitions.....	2
2. Introduction.....	3
3. Résumé du fonctionnement.....	4
3.1 Établissement et maintenance d'une session d'échange de communications.....	4
3.2 Échanges de bases de données.....	4
3.3 Synchronisation interne ou externe.....	4
3.4 Annonce des chemins TRIP.....	4
3.5 Bases de données d'informations d'acheminement téléphonique.....	5
3.6 Chemins dans TRIP.....	6
3.7 Agrégation.....	6
4. Formats de message.....	6
4.1 Formats d'en-tête de message.....	6
4.2 Format du message OUVERT.....	7
4.3 Format du message METTREÀJOUR.....	10
4.4 Format du message GARDERENVIE.....	14
4.5 Format du message NOTIFICATION.....	14
5. Attributs TRIP.....	15
5.1 CheminsRetirés.....	15
5.2 CheminsAccessibles.....	17
5.3 ServeurDeProchainBond.....	18
5.4 CheminD'Annonce.....	19
5.5 CheminSuivi.....	21
5.6 AgrégatAtomique.....	22
5.7 PréférenceLocale.....	23
5.8 DécouvMultiSorties.....	23
5.9 Communautés.....	24
5.10 Topologie d'ITAD.....	25
5.11 CheminConverti.....	26

5.12 Considérations pour définir de nouveaux attributs TRIP.....	27
6. Détection et traitement d'erreur TRIP.....	27
6.1 Détection et traitement d'erreur d'en-tête de message.....	28
6.2 Détection et traitement d'erreur du message OUVERT.....	28
6.3 Détection et traitement d'erreur du message METTREÀJOUR.....	28
6.4 Détection et traitement d'erreur du message NOTIFICATION.....	29
6.5 Traitement de l'erreur Expiration du temporisateur de garde.....	29
6.6 Traitement de l'erreur de l'automate à états finis.....	29
6.7 Cesser.....	30
6.8 Détection de collision de connexion.....	30
7. Négociation de version TRIP.....	30
8. Négociation de capacité TRIP.....	30
9. Automate à états finis de TRIP.....	30
10. Traitement du message METTREÀJOUR.....	33
10.1 Processus d'arrosage.....	34
10.2 Processus de décision.....	35
10.3 Processus de Mise-à-jour-Envoi.....	37
10.4 Critères de choix de chemin.....	40
10.5 Génération des choix de chemins.....	40
11. Transport TRIP.....	40
12. Topologie d'ITAD.....	40
13. Considérations relatives à l'IANA.....	41
13.1 Capacités TRIP.....	41
13.2 Attributs TRIP.....	41
13.3 Familles d'adresse de destination.....	41
13.4 Protocoles d'applications TRIP.....	41
13.5 Numéros d'ITAD.....	42
14. Considérations pour la sécurité.....	42
Appendice 1. Transitions d'état et actions de l'automate TRIP.....	43
Appendice 2 Recommandations pour la mise en œuvre.....	44
A.2.1 Plusieurs réseaux par message.....	44
A.2.2 Traitement des messages sur un protocole de flux.....	45
A.2.3 Réduction du flottement de chemin.....	45
A.2.4 Temporisateurs TRIP.....	45
A.2.5 Tri sur AP_SET.....	45
Références.....	46
Notice de propriété intellectuelle.....	46
Déclaration complète de droits de reproduction.....	47

1. Terminologie et définitions

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

Un cadre pour l'acheminement de la téléphonie sur IP (TRIP) est décrit dans la [RFC2871]. On suppose que le lecteur est familiarisé avec le cadre et la terminologie de la [RFC2871]. On définit et utilise les termes suivants, en plus de ceux définis dans la [RFC2871].

Base de données d'informations d'acheminement de téléphonie (TRIB, *Telephony Routing Information Base*) : c'est la base de données des destinations de téléphonie accessibles construite et entretenue à un serveur de localisation (LS, *location server*) par suite de sa participation à TRIP.

Domaine administratif de téléphonie IP (ITAD, *IP Telephony Administrative Domain*) : c'est l'ensemble des ressources (passerelles, serveurs de localisation, etc.) sous le contrôle d'une seule autorité administrative. Les utilisateurs finaux sont des clients d'un ITAD.

Chemin moins/plus spécifique : un chemin X est dit être moins spécifique qu'un chemin Y si chaque destination dans Y est aussi une destination dans X, et si X et Y ne sont pas égaux. Dans ce cas, Y est aussi dit être plus spécifique que X.

Agrégation : c'est le processus par lequel plusieurs chemins sont combinés en un seul chemin moins spécifique qui couvre le même ensemble de destinations. L'agrégation est utilisée pour réduire la taille de la TRIB en étant synchronisée avec les LS homologues en réduisant le nombre de chemins TRIP exportés.

Homologues : deux LS qui partagent une association logique (une connexion de transport). Si les LS sont dans le même ITAD, ce sont des homologues internes. Autrement, ils sont des homologues externes. L'association logique entre deux LS homologues est appelée une session d'homologues (*peering session*).

Protocole d'informations d'acheminement téléphonique (TRIP, *Telephony Routing Information Protocol*) : c'est le protocole défini dans la présente spécification. La fonction de TRIP est d'annoncer l'accessibilité des destinations téléphoniques, les attributs associés à la destination, ainsi que les attributs du chemin vers ces destinations.

Destination TRIP : TRIP peut être utilisé pour gérer des tableaux d'acheminement pour plusieurs protocoles (SIP, H323, etc.). Dans TRIP, une destination est la combinaison de (a) un ensemble d'adresses (données par une famille d'adresses et un préfixe d'adresse) et (b) un protocole d'application (SIP, H323, etc.).

2. Introduction

Le problème de la localisation de passerelle et de l'acheminement a été introduit par la [RFC2871]. Il est considéré comme un des problèmes les plus difficiles de la téléphonie IP. Le choix d'une passerelle de sortie pour un appel téléphonique, traversant un réseau IP vers sa destination ultime dans le RTPC, est déterminé en grande partie par les politiques des diverses parties le long du chemin, et par les relations établies entre ces parties. À ce titre, un répertoire mondial des passerelles de sortie dans lequel les usagers cherchent les numéros de téléphone de destination n'est pas une solution réalisable. On échange plutôt les informations sur la disponibilité des routeurs de sortie entre les fournisseurs et, sous réserve de la politique, on les rend disponibles localement puis on les propage aux autres fournisseurs dans les autres ITAD, créant ainsi des chemins vers ces routeurs de sortie. Cela va permettre à chaque fournisseur de créer sa propre base de données de numéros de téléphone accessibles et des chemins associés – une telle base de données peut être très différente selon le fournisseur, en fonction de sa politique..

TRIP est un protocole de localisation de routeur et d'acheminement inter domaine (c'est-à-dire, inter-ITAD). La principale fonction d'un locuteur TRIP, qu'on appelle un serveur de localisation (LS, *location server*) est d'échanger des informations avec les autres LS. Ces informations incluent l'accessibilité des destinations téléphoniques, les chemins vers ces destinations, et les informations sur les routeurs vers ces destinations téléphoniques qui résident sur le RTPC. Les exigences de TRIP sont établies dans la [RFC2871].

Les LS échangent des informations d'acheminement suffisantes pour construire un graphe de la connexité des ITAD afin d'empêcher la formation de boucles d'acheminement. De plus, TRIP peut être utilisé pour échanger les attributs nécessaires pour mettre en application les politiques et pour choisir les chemins sur la base des caractéristiques des chemins ou des routeurs. La présente spécification définit les mécanismes de transport et de synchronisation de TRIP, son automate à états finis, et les données de TRIP. La présente spécification définit les attributs de base de TRIP. L'ensemble des attributs de TRIP est extensible, de sorte que des attributs supplémentaires pourront être définis dans des documents futurs.

TRIP est modélisé d'après le protocole 4 de routeur frontière (BGP-4, *Border Gateway Protocol 4*) [RFC1771] et amélioré avec des caractéristiques d'état de liaison, comme dans le protocole Ouvrir le plus court chemin en premier (OSPF, *Open Shortest Path First*) [RFC2328], [IS-IS], et le protocole de synchronisation d'antémémoire de serveur (SCSP, *Server Cache Synchronization Protocol*) [RFC2334]. TRIP utilise le mécanisme de transport inter domaine de BGP, la communication d'homologues de BGP, l'automate à états finis de BGP, et des formats et attributs similaires à ceux de BGP. À la différence de BGP, TRIP permet cependant des topologies génériques de LS intra domaine, ce qui simplifie la configuration et augmente l'adaptabilité ce qui diffère de l'exigence de maillage complet de BGP entre les locuteurs BGP internes. TRIP utilise un mécanisme d'arrosage intra domaine similaire à celui utilisé dans OSPF [RFC2328], [IS-IS], et SCSP [RFC2334].

TRIP permet l'agrégation de chemins lorsque ils sont annoncés à travers le réseau. TRIP ne définit pas d'algorithme spécifique de sélection de chemin.

TRIP fonctionne sur un protocole de transport fiable. Cela élimine le besoin de mettre en œuvre une fragmentation, retransmission, accusé de réception, et séquençage, explicites. Le mécanisme de notification d'erreur utilisé dans TRIP suppose que le protocole de transport prend en charge une fermeture en douceur, c'est-à-dire que toutes les données en instance seront livrées avant que la connexion ne ferme.

Le fonctionnement de TRIP est indépendant de tout protocole de signalisation téléphonique particulier. Donc, TRIP peut

être utilisé comme protocole d'acheminement pour tous ces protocoles, par exemple, [H.323] et SIP [RFC2543].

La topologie de LS communicants est indépendante de la topologie physique du réseau. De plus, les frontières d'un ITAD sont indépendantes de celles des systèmes autonomes d'acheminement de couche 3. Les TRIP communicants internes ou externes n'ont pas besoin d'être physiquement adjacents.

3. Résumé du fonctionnement

La présente section résume le fonctionnement de TRIP. Les détails sont fournis dans les sections ultérieures.

3.1 Établissement et maintenance d'une session d'échange de communications

Deux LS homologues forment entre eux une connexion de protocole de transport. Ils échangent des messages pour ouvrir et confirmer les paramètres de connexion, et pour négocier les capacités de chaque LS ainsi que le type des informations à annoncer sur cette connexion.

Des messages Garder en vie sont envoyés périodiquement pour s'assurer que les homologues adjacents sont opérationnels. Des messages Notification sont envoyés en réponse aux erreurs ou dans des conditions particulières. Si une connexion rencontre une condition d'erreur, un message Notification est envoyé et la connexion est close.

3.2 Échanges de bases de données

Une fois que la connexion des homologues a été établie, le flux de données initial est un catalogue de tous les chemins pertinents vers le nouvel homologue (dans le cas d'un homologue externe, tous les chemins qui sont dans le Adj-TRIB-Out du LS pour cet homologue externe, dans le cas d'un homologue interne, tous les chemins qui sont dans le Ext-TRIB et tous les Adj-TRIBs-In). Noter que les différents TRIB sont définis au paragraphe 3.5.

Des mises à jour incrémentaires sont envoyées lorsque les tableaux d'acheminement TRIP (les TRIB) changent. TRIP n'exige pas de rafraîchissements périodiques des chemins. Donc, un LS doit conserver la version en cours de toutes les entrées d'acheminement.

Si un ITAD particulier a plusieurs LS et fournit un service de transit pour d'autres ITAD, il faut alors veiller à s'assurer d'une vue cohérente de l'acheminement au sein de l'ITAD. Lorsqu'ils sont synchronisés, les tableaux d'acheminement TRIP, c'est-à-dire, les Loc-TRIB, de tous les homologues internes, sont identiques.

3.3 Synchronisation interne ou externe

Comme avec BGP, TRIP distingue entre homologue interne et externe. Au sein d'un ITAD, le TRIP interne utilise des mécanismes d'état de liaison pour arroser une topologie arbitraire avec les mises à jour de base de données. En externe, TRIP utilise des relations d'homologue à homologue en point à point pour échanger les informations de base de données.

Pour réaliser la synchronisation interne, les connexions d'homologues internes sont configurées entre les LS du même ITAD de telle sorte que la topologie intra domaine résultante soit connectée et suffisamment redondante. Ceci est différent de l'approche de BGP qui exige que tous les homologues internes soient connectés en une topologie de maillage complet, ce qui peut résulter en problèmes d'échelle. Lorsque une mise à jour est reçue d'un homologue interne, les chemins qui sont dans la mise à jour sont vérifiés pour déterminer si ils sont plus nouveaux que la version qui est déjà dans la base de données. Les nouveaux chemins sont alors diffusés à tous les autres homologues dans le même domaine.

3.4 Annonce des chemins TRIP

Dans TRIP, un chemin est défini comme la combinaison (a) d'un ensemble d'adresses de destination (données par un indicateur de famille d'adresses et un préfixe d'adresse) et (b) d'un protocole d'application (par exemple SIP, H323, etc.). Généralement, il y a des attributs supplémentaires associés à chaque chemin (par exemple, le serveur de prochain bond).

Les chemins TRIP sont annoncés entre une paire de LS dans des messages MISEÀJOUR (*UPDATE*). Les adresses de destination sont incluses dans l'attribut CheminsAccessibles (*ReachableRoutes*) du message MISEÀJOUR, tandis que d'autres attributs décrivent des choses comme le chemin ou la passerelle de sortie.

Si un LS choisit d'annoncer un chemin TRIP, il peut ajouter ou modifier les attributs du chemin avant de l'annoncer à un homologue. TRIP donne des mécanismes par lesquels un LS peut informer son homologue qu'un chemin précédemment annoncé n'est plus disponible. Il y a trois méthodes par lesquelles un LS donné peut indiquer qu'un chemin a été retiré du service :

- Inclure le chemin dans l'attribut CheminsRetirés (*WithdrawnRoutes*) dans un message MISEÀJOUR, marquant ainsi que les destinations associées ne sont plus disponibles.
- Annoncer un chemin de remplacement avec le même ensemble de destinations dans l'attribut CheminsAccessibles.
- Pour les homologues externes qui ne subissent pas l'arrosage, la connexion de LS à LS homologue peut être close, ce qui retire implicitement du service tous les chemins que s'étaient annoncés l'un à l'autre la paire de LS sur cette session d'homologues. Noter que terminer une session interne d'échange de trafic ne supprime pas nécessairement les chemins annoncés par le LS homologue car les mêmes chemins peuvent avoir été reçus de plusieurs homologues internes à cause de l'arrosage. Si un LS détermine qu'un autre LS interne n'est plus actif (à partir des attributs Topologie ITAD des messages MISEÀJOUR provenant des autres homologues internes) il DOIT alors supprimer tous les chemins générés dans le LS par ce LS et relancer son processus de décision.

3.5 Bases de données d'informations d'acheminement téléphonique

Un LS TRIP traite trois types de chemins :

- Chemin externe : c'est un chemin reçu d'un LS homologue externe.
- Chemin interne : c'est un chemin reçu d'un LS interne dans le même ITAD.
- Chemin local : c'est un chemin injecté localement dans TRIP, par exemple par configuration ou par redistribution de chemin provenant d'un autre protocole d'acheminement.

La base de données d'informations d'acheminement téléphonique (TRIB, *Telephony Routing Information Base*) au sein d'un LS comporte trois parties distinctes :

- Adj-TRIBs-In : elle mémorise les informations d'acheminement qui ont été apprises à partir des messages MISEÀJOUR entrants. Son contenu représente les chemins TRIP qui sont disponibles comme entrées au processus de décision. Ce sont les chemins "non traités" reçus. Les chemins provenant de chaque LS homologue externe et de chaque LS homologue interne sont conservés indépendamment dans cette base de données, afin que les mises à jour provenant d'un homologue n'affectent pas les chemins reçus d'un autre LS. Noter qu'il y a un Adj-TRIB-In pour chaque LS au sein du domaine, même ceux avec lesquels le LS n'échange pas directement de trafic.
- Ext-TRIB : il n'y a qu'une seule base de données Ext-TRIB par LS. Le LS fait fonctionner l'algorithme de choix de chemin sur tous les chemins externes (mémorisés dans le Adj-TRIBs-In des homologues externes) et les chemins locaux (qui peuvent être mémorisés dans un Adj-TRIB-In qui représente le LS local) et choisit le meilleur chemin pour une destination donnée et la mémorise dans la Ext-TRIB. L'utilisation de Ext-TRIB sera mieux expliquée au paragraphe 10.3.1
- Loc-TRIB : elle contient les informations d'acheminement TRIP locales que le LS a choisies en appliquant ses politiques locales aux informations d'acheminement contenues dans sa Adj-TRIBs-In des LS internes et dans la Ext-TRIB.
- Adj-TRIBs-Out : elle mémorise les informations que le LS local a choisies pour les annoncer à ses homologues externes. Les informations d'acheminement mémorisées dans le Adj-TRIBs-Out seront transportées dans les messages MISEÀJOUR du LS local et annoncées à ses homologues.

La Figure 1 illustre les relations entre quatre parties de la base de données d'informations d'acheminement.

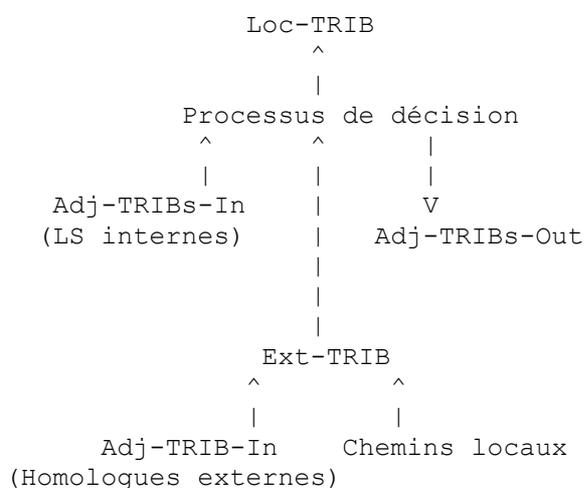


Figure 1 : Relations dans la TRIB

Bien que le modèle conceptuel fasse une distinction entre Adj-TRIBs-In, Ext-TRIB, Loc-TRIB, et Adj-TRIBs-Out, cela

n'implique ni n'exige en aucun cas que les mises en œuvre doivent tenir quatre copies distinctes des informations d'acheminement. Le choix de la mise en œuvre (par exemple, quatre copies des informations contre une copie avec des pointeurs) n'est pas lié par le protocole.

3.6 Chemins dans TRIP

Un chemin dans TRIP spécifie une gamme de nombres qui sont un préfixe de ces nombres (la définition exacte et la syntaxe de chemin figurent au paragraphe 5.1.1). Des gammes de nombres arbitraires ne sont pas représentables de façon atomique par un chemin dans TRIP. Une gamme de préfixe est le seul type de gamme pris en charge de façon atomique. Une gamme arbitraire peut être réalisée en utilisant plusieurs préfixes dans un attribut CheminsAccessibles (voir aux paragraphes 5.1 & 5.2). Par exemple, 222-xxxx jusqu'à 999-xxxx pourrait être représenté en incluant les préfixes 222, 223, 224, ..., 23, 24, ..., 3, 4, ..., 9 dans un attribut CheminsAccessibles.

3.7 Agrégation

L'agrégation est une amélioration d'adaptabilité utilisée par un LS pour réduire le nombre d'entrées d'acheminement à synchroniser avec ses homologues. L'agrégation peut être effectuée par un LS lorsque il y a un ensemble de chemin {R1, R2, ...} dans sa TRIB de telle sorte qu'il existe un chemin R moins spécifique où toute destination valide dans R est aussi une destination valide dans {R1, R2, ...} et vice-versa. La Section 5 comporte une description de la façon de combiner chaque attribut (par type) sur les chemins {R1, R2, ...} en un attribut pour R.

Noter qu'il n'y a pas de mécanisme au sein de TRIP pour communiquer qu'un préfixe d'adresse particulier n'est pas utilisé ou pas valide au sein d'une famille d'adresses particulière, et donc que ces adresses pourraient être sautées durant l'agrégation. Les LS peuvent utiliser des méthodes en dehors de TRIP pour apprendre les préfixes invalides qui peuvent être ignorés durant l'agrégation.

Un LS n'est pas obligé d'effectuer l'agrégation, cependant elle est recommandée chaque fois qu'il est important de tenir une plus petite TRIB. C'est sur la base de sa politique locale qu'un LS décide d'agréger ou non un ensemble de chemins en un seul chemin agrégé.

Chaque fois qu'un LS agrège plusieurs chemins où le serveur de prochain bond n'est pas identique dans tous les chemins agrégés, l'attribut ServeurDeProchainBond (*NextHopServer*) du chemin agrégé doit être réglé à un serveur de signalisation dans le domaine du LS agrégateur.

Lorsque un LS re-règle l'attribut ServeurDeProchainBond de tout chemin, et cela peut être effectué à cause de l'agrégation ou pour d'autres raisons, cela a pour effet d'ajouter un autre serveur de signalisation le long du chemin de signalisation vers ces destinations. Le résultat final est que le chemin de signalisation entre deux destinations peut consister en plusieurs serveurs de signalisation à travers plusieurs domaines.

4. Formats de message

La présente section décrit les formats de message utilisés par TRIP. Les messages sont envoyés sur une connexion au protocole de transport fiable. Un message DOIT n'être traité qu'après qu'il a été entièrement reçu. La taille maximum de message est 4096 octets. Toutes les mises en œuvre DOIVENT prendre en charge cette taille maximum de message. Le plus petit message qui PEUT être envoyé consiste en un en-tête TRIP sans portion de données, soit 3 octets.

4.1 Formats d'en-tête de message

Chaque message a un en-tête de taille fixe. Il peut y avoir ou non une portion de données à la suite de l'en-tête, selon le type de message. La disposition des champs d'en-tête est indiquée à la Figure 2.

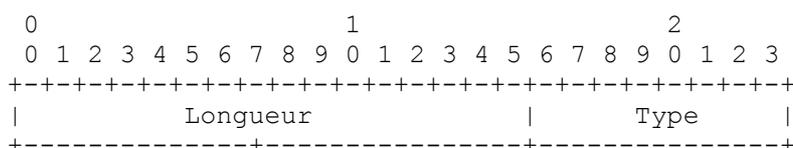


Figure 2 : En-tête TRIP

Longueur : Cet entier non signé de 2 octets indique la longueur totale du message, incluant l'en-tête, en octets. Donc, il permet de localiser, dans le flux de niveau transport, le début du prochain message. La valeur du champ Longueur doit

toujours être d'au moins 3 et n'être pas supérieure à 4096, et peut subir d'autres contraintes selon le type de message. Aucun bourrage n'est permis en plus des données après le message, de sorte que le champ Longueur doit avoir la plus petite valeur possible étant donné le reste du message.

Type : C'est un entier non signé d'un octet qui indique le code de type du message. Les codes de type suivants sont définis :

- 1 - OUVERT
- 2 - MISE A JOUR
- 3 - NOTIFICATION
- 4 - GARDER EN VIE

4.2 Format du message OUVERT

Après l'établissement d'une connexion de protocole de transport, le premier message envoyé par chaque côté est un message OUVERT. Si le message OUVERT est acceptable, un message GARDER EN VIE qui confirme le message OUVERT est renvoyé. Une fois que OUVERT est confirmé, les messages MISE A JOUR, GARDER EN VIE, et NOTIFICATION peuvent être échangés.

La longueur minimum du message OUVERT est de 17 octets (incluant l'en-tête de message). Les messages OUVERT qui ne satisfont pas à cette exigence minimum sont traités comme défini au paragraphe 6.2.

En plus de l'en-tête TRIP de taille fixe, le message OUVERT contient les champs suivants :

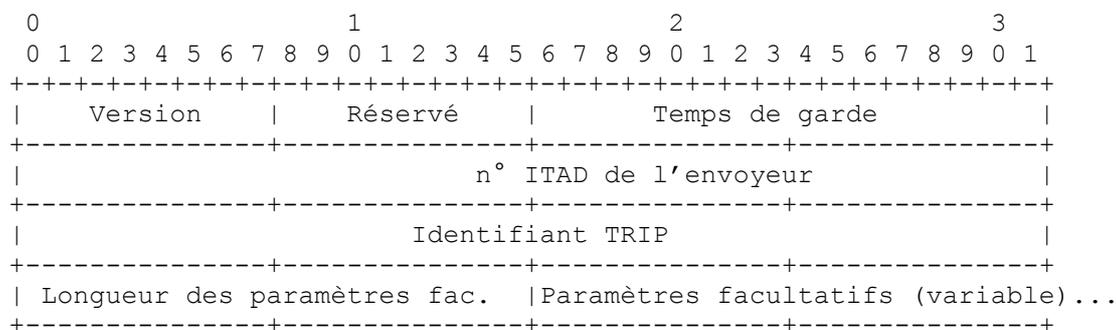


Figure 3 : En-tête de message OUVERT TRIP

Version:

Cet entier non signé de un octet indique la version de protocole du message. Le numéro de version TRIP actuel est 1.

Temps de garde :

Cet entier non signé de deux octets indique le nombre de secondes que l'expéditeur propose pour la valeur du temporisateur de garde. À réception d'un message OUVERT, un LS DOIT calculer la valeur du temporisateur de garde en utilisant le plus petit de ses temps de garde configurés et du temps de garde reçu dans le message OUVERT. Le temps de garde DOIT être zéro ou au moins trois secondes. Une mise en œuvre PEUT rejeter des connexions sur la base du temps de garde. La valeur calculée indique le nombre maximum de secondes qui peut s'écouler entre la réception de messages successifs GARDER EN VIE et/ou MISE A JOUR par l'expéditeur.

n° d'ITAD de l'expéditeur

Cet entier non signé de quatre octets indique le numéro d'ITAD de l'expéditeur. Le numéro d'ITAD doit être unique pour ce domaine au sein de cette confédération de LS coopérants. Les numéros d'ITAD sont alloués par l'IANA comme spécifié à la Section 13. Le présent document réserve le numéro d'ITAD 0. Les numéros d'ITAD de 1 à 255 sont pour utilisation privée.

Identifiant TRIP :

Cet entier non signé de quatre octets indique l'identifiant TRIP de l'expéditeur. L'identifiant TRIP DOIT identifier de façon univoque ce LS au sein de son ITAD. Un LS donné PEUT régler la valeur de son identifiant TRIP à une adresse IPv4 allouée à ce LS. La valeur de l'identifiant TRIP est déterminée au démarrage et DOIT être la même pour toutes les connexions homologues. Lors de la comparaison de deux identifiants TRIP, l'identifiant TRIP est interprété comme un entier non signé numérique de quatre octets.

Longueur de paramètres facultatifs :

Cet entier non signé de deux octets indique la longueur totale du champ Paramètres facultatifs en octets. Si la valeur de ce champ est zéro, aucun paramètre facultatif n'est présent.

Paramètres facultatifs :

Ce champ peut contenir une liste des paramètres facultatifs, où chaque paramètre est codé comme un triplet <Type de paramètre, Longueur de paramètre, Valeur de paramètre>.

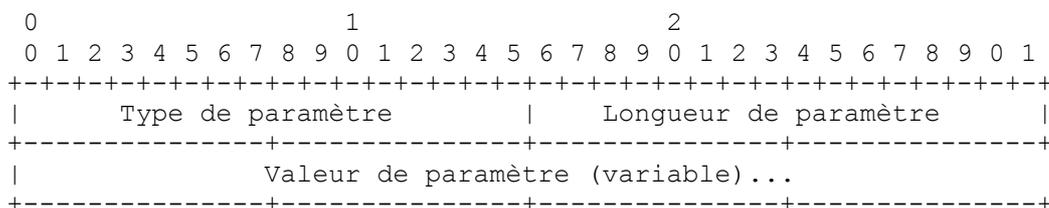


Figure 4 : Codage de paramètre facultatif

Type de paramètre :

C'est un champ de deux octets qui identifie sans ambiguïté les paramètres individuels.

Longueur de paramètre :

C'est un champ de deux octets qui contient la longueur du champ Valeur de paramètre en octets.

Valeur de paramètre :

C'est un champ de longueur variable qui est interprété conformément à la valeur du champ Type de paramètre.

4.2.1 Paramètres facultatifs du message OUVERT

Le présent document définit les paramètres facultatifs suivants pour le message OUVERT.

4.2.1.1 Informations de capacité

Les informations de capacité utilisent le type 1 de paramètre facultatif. C'est un paramètre facultatif utilisé par un LS pour porter à son homologue la liste des capacités prises en charge par le LS. Cela permet à un LS d'apprendre les capacités de ses homologues LS. La négociation des capacités est définie à la Section 8.

Le paramètre contient un ou plusieurs triplets <Code de capacité, Longueur de capacité, Valeur de capacité>, où chaque triplet est codé comme indiqué ci-dessous :

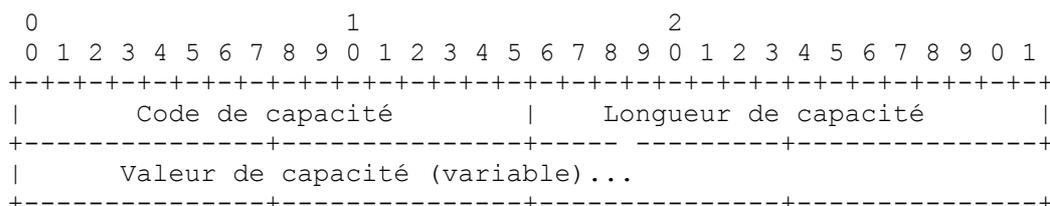


Figure 5 : Paramètre facultatif de capacité

Code de capacité :

C'est un champ de deux octets qui identifie sans ambiguïté les capacités individuelles.

Longueur de capacité :

C'est un champ de deux octets qui contient la longueur du champ Valeur de capacité en octets.

Valeur de capacité :

C'est un champ de longueur variable qui est interprété conformément à la valeur du champ Code de capacité.

Toute capacité particulière, comme identifiée par son code de capacité, peut apparaître plus d'une fois au sein de Paramètres facultatifs.

Le présent document réserve les codes de capacité 32768-65535 pour des applications spécifiques des fabricants (ce sont les codes qui ont le premier bit de la valeur de code égale à 1). Le présent document réserve la valeur 0. Les codes de

capacité (autres que ceux réservés pour utilisation spécifique de fabricant) sont sous le contrôle de l'IANA. Voir à la Section 13 les considérations relatives à l'IANA.

Les codes de capacité suivants sont définis par la présente spécification :

Code	Capacité
1	Types de chemins pris en charge
2	Capacité d'envoi et de réception

4.2.1.1.1 Types de chemins pris en charge

Le code de capacité Types de chemins pris en charge fait la liste des types de chemin pris en charge dans cette session d'échange de trafic par le LS émetteur. Un LS NE DOIT PAS utiliser des types de chemins qui ne sont pas acceptés par le LS homologue dans une session d'échange de trafic particulière. Si les types de chemin pris en charge par un homologue ne sont pas satisfaisants, un LS DEVRAIT terminer la session d'échange de trafic. Le format d'un Type de chemin est :

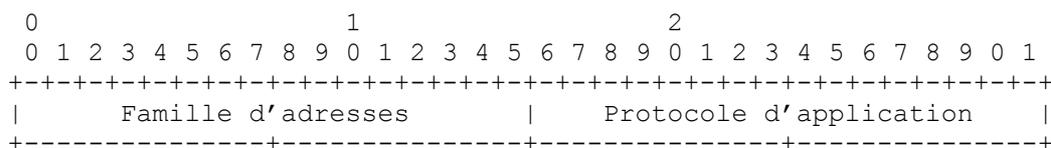


Figure 6 : Capacité Types de chemins pris en charge

La famille d'adresses et le protocole d'application sont définis au paragraphe 5.1.1. Famille d'adresses donne la famille d'adresses qui sont acheminées (au sein de l'attribut CheminsAccessibles). Le protocole d'application fait la liste des applications pour lesquelles les chemins s'appliquent. Par exemple, un type de chemin pour TRIP pourrait être <E.164, SIP>, indiquant un ensemble de destinations E.164 pour le protocole SIP.

La capacité Types de chemins pris en charge PEUT contenir plusieurs types de chemins dans la capacité. Le nombre de types de chemins au sein de la capacité est le nombre maximum qui peut tenir dans la longueur de la capacité. Le code Capacité est 1 et la longueur est variable.

4.2.1.1.2 Capacité d'envoi et réception

Cette capacité spécifie le mode dans lequel va fonctionner le LS avec cet homologue particulier. Les modes possibles sont le mode Envoi seul, le mode Réception seule, ou le mode Envoi-Réception. Le mode par défaut est le mode Envoi-Réception.

Dans le mode Envoi seul, un LS transmet des messages METTREÀJOUR à son homologue, mais l'homologue NE DOIT PAS transmettre de messages METTREÀJOUR à ce LS. Si un LS en mode Envoi seul reçoit un message METTREÀJOUR de son homologue, il DOIT éliminer ce message, mais aucune autre action ne devrait être entreprise.

Les messages METTREÀJOUR envoyés par un LS en mode Envoi seul à son homologue intra-domaine DOIVENT inclure l'attribut Topologie ITAD chaque fois que la topologie change. Une application utile d'un LS en mode Envoi seul avec un homologue externe est d'activer des services d'enregistrement de passerelles.

Si un fournisseur de service termine des appels sur un ensemble de passerelles qu'il possède, mais n'initie jamais des appels, il peut régler ses LS à fonctionner en mode Envoi seul, car ils n'ont jamais besoin que de générer des messages METTREÀJOUR, pas de les recevoir. Si un LS en mode Envoi-Réception a une session d'échange de trafic avec un homologue en mode Envoi seul, ce LS DOIT régler sa politique de dissémination de chemins de telle sorte qu'il n'envoie jamais de message METTREÀJOUR à son homologue.

En mode Réception seule, le LS agit comme un écoutant TRIP passif. Il reçoit et traite les messages METTREÀJOUR provenant de son homologue, mais il NE DOIT PAS transmettre de message METTREÀJOUR à son homologue. Ceci est utile pour les stations de gestion qui souhaitent collecter des informations de topologie à des fins d'affichage.

Le comportement d'un LS en mode Envoi-Réception est le fonctionnement TRIP par défaut spécifié dans le présent document.

La capacité Envoi-Réception est une valeur numérique non signée de 4 octets. Elle peut seulement prendre une des trois valeurs suivantes :

- 1 – Mode Envoi-Réception
- 2 – Mode Envoi seul
- 3 – Mode Réception seule

Une session d'échange de trafic NE DOIT PAS être établie entre deux LS si tous deux sont en mode Envoi seul ou si tous les deux sont en mode Réception seule. Si un homologue LS détecte une telle discordance de capacités lors du traitement

d'un message OUVERT, il DOIT répondre par un message NOTIFICATION et clore la session d'échange de trafic. Le code d'erreur dans le message NOTIFICATION doit être réglé à "Discordance de capacités."

Un LS DOIT être configuré dans le même mode Envoi-Réception pour tous les homologues.

4.3 Format du message METTREÀJOUR

Les messages METTREÀJOUR sont utilisés pour transférer les informations d'acheminement entre les LS. Les informations dans le paquet METTREÀJOUR peuvent être utilisées pour construire un graphe décrivant les relations entre les divers ITAD. En appliquant des règles qui seront exposées plus loin, les boucles d'informations d'acheminement et quelques autres anomalies peuvent être empêchées.

Un message METTREÀJOUR est utilisé à la fois pour annoncer et pour retirer des chemins du service. Un message METTREÀJOUR peut simultanément annoncer et retirer des chemins TRIP.

En plus de l'en-tête TRIP, le METTREÀJOUR TRIP contient une liste d'attributs d'acheminement qui sont indiqués à la Figure 7. Il n'y a pas de bourrage entre les attributs d'acheminement.

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Premier attribut de chemin | Second attribut de chemin| ...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 7 : Format de METTREÀJOUR TRIP

La longueur minimale d'un message METTREÀJOUR est trois octets (il n'y a pas d'attribut obligatoire dans TRIP).

4.3.1 Attributs d'acheminement

Une séquence de longueur variable d'attributs d'acheminement est présente dans chaque message METTREÀJOUR. Chaque attribut est un triplet <type d'attribut, longueur d'attribut, valeur d'attribut> de longueur variable.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Fanions d'attr|Code type d'att|          Longueur d'attribut          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Valeur d'attribut (variable)              |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 8 : Format d'attribut d'acheminement

Type d'attribut est un champ de deux octets qui consiste en des fanions d'attribut d'un octet suivi par l'octet du code de type d'attribut.

L'attribut Code de type définit le type de l'attribut. Les codes de type d'attribut de base définis pour TRIP sont exposés plus loin dans la présente section. Les attributs DOIVENT apparaître dans le message METTREÀJOUR dans l'ordre numérique des codes de type d'attribut. Un attribut NE DOIT PAS être inclus plus d'une fois dans le même message METTREÀJOUR. Les fanions d'attribut sont utilisés pour contrôler le traitement des attributs lorsque le type d'attribut est inconnu. Les fanions d'attribut sont mieux définis au paragraphe 4.3.2.

Le présent document réserve les codes de type d'attribut 224 à 255 pour les applications spécifiques de fabricant (ce sont les codes avec les trois premiers bits du code qui sont égaux à 1). Le présent document réserve la valeur 0. Les codes de type d'attribut (autres que ceux réservés pour une utilisation spécifique du fabricant) sont sous le contrôle de l'IANA. Voir la Section 13 sur les considérations relatives à l'IANA.

Le troisième et le quatrième octet de l'attribut de chemin contiennent la longueur du champ Valeur d'attribut en octets.

Les octets restants de l'attribut représentent la valeur de l'attribut et sont interprétés selon les fanions d'attribut et le code de type d'attribut. Les types de base d'attribut pris en charge, leurs valeurs, et leurs utilisations sont définis dans la présente spécification. Ce sont les attributs nécessaires pour un fonctionnement approprié de TRIP sans boucle, à la fois inter-domaine et intra-domaine. Des attributs supplémentaires pourront être définis dans de futurs documents.

4.3.2 Fanions d'attribut

Il est clair que l'ensemble des attributs utilisés par TRIP va évoluer avec le temps. Il est donc essentiel que soient fournis des mécanismes pour traiter les attributs d'un type non reconnu. Le traitement des attributs non reconnus est contrôlé via le champ Fanions de l'attribut. Les attributs reconnus devraient être traités conformément à leur définition spécifique.

Les fanions d'attribut suivants sont définis par la présente spécification :

Bit	Fanion
0	Fanion Bien connu
1	Fanion Transitif
2	Fanion Dépendant
3	Fanion Partiel
4	Fanion Encapsulation d'état de liaison

Le bit de poids fort (bit 0) de l'octet Fanions d'attribut est le bit Bien connu. Il définit si l'attribut n'est pas bien connu (si il est établi à 1) ou bien connu (si il est mis à 0). Les mises en œuvre ne sont pas obligées de prendre en charge les attributs qui ne sont pas bien connus, mais elles DOIVENT prendre en charge les attributs bien connus.

Le second bit de poids fort (le bit 1) de l'octet Fanions d'attribut est le bit Transitif. Il définit si un attribut qui n'est pas bien connu est transitif (si il est réglé à 1) ou non transitif (si il est mis à 0). Pour les attributs bien connus, le bit Transitif DOIT être à zéro à l'émission et DOIT être ignoré à réception.

Le troisième bit de poids fort (bit 2) de l'octet Fanions d'attribut est le bit Dépendant. Il définit si un attribut transitif est dépendant (si il est réglé à 1) ou indépendant (si il est réglé à 0). Pour les attributs bien connus et pour les attributs non transitifs, le bit Dépendant n'est pas pertinent, et DOIT être réglé à zéro à l'émission et DOIT être ignoré à réception.

Le quatrième bit de poids fort (bit 3) de l'octet Fanions d'attribut est le bit Partiel. Il définit si les informations contenues dans l'attribut transitif qui n'est pas bien connu est partiel (si il est réglé à 1) ou complet (si il est réglé à 0). Pour les attributs bien connus et pour les attributs non transitifs, le bit Partiel DOIT être réglé à 0 à l'émission et DOIT être ignoré à réception.

Le cinquième bit de poids fort (bit 4) de l'octet Fanions d'attribut est le bit Encapsulation d'état de liaison. Ce bit n'est applicable qu'à certains attributs (CheminsAccessibles et CheminsRetirés) et détermine l'encapsulation du chemin au sein de ces attributs. Si ce bit est établi, l'encapsulation d'état de liaison est utilisée au sein de l'attribut. Autrement, l'encapsulation standard est utilisée au sein de l'attribut. La technique de l'encapsulation d'état de liaison est décrite au paragraphe 4.3.2.4. Ce fanion n'est valide que sur les attributs CheminsAccessibles et CheminsRetirés. Il DOIT être supprimé à l'émission et DOIT être ignoré à réception pour tous les autres attributs.

Les autres bits de l'octet Fanions d'attribut ne sont pas utilisés. Ils DOIVENT être mis à zéro à l'émission et ignorés à réception.

4.3.2.1 Fanions d'attribut et choix de chemin

Tout attribut reconnu peut être utilisé comme entrée au processus de choix de chemin, bien que l'utilité de certains attributs soit minimale dans le choix de chemin.

4.3.2.2 Fanions d'attribut et dissémination de chemin

TRIP connaît deux variantes de transitivité du fait que les LS intermédiaires ne sont pas obligés de modifier le serveur de prochain bond lorsque ils propagent les chemins. Les attributs peuvent être non transitifs, transitifs dépendants, ou transitifs indépendants. Un attribut ne peut pas être à la fois transitif dépendant et transitif indépendant.

Les attributs transitifs indépendants non reconnus peuvent être propagés par tout LS intermédiaire. Les attributs transitifs dépendants reconnus ne PEUVENT être propagés que si le LS NE CHANGE PAS le serveur de prochain bond. Les variations de transitivité permettent que certains attributs non reconnus soient portés de bout en bout (transitif indépendant) que certains soient portés entre des serveurs de prochain bond adjacents (transitif dépendant) et que d'autres soient restreints aux LS homologues (non transitif).

Un LS qui passe un attribut transitif non reconnu à un homologue DOIT régler le fanion Partiel sur cet attribut. Tout LS le long d'un chemin PEUT insérer un attribut transitif dans un chemin. Si un LS, sauf le LS d'origine, insère un nouvel attribut transitif indépendant dans un chemin, il DOIT alors établir le fanion Partiel sur cet attribut. Si un LS, sauf un LS qui modifie le ServeurDeProchainBond, insère un nouvel attribut transitif dépendant dans un chemin, il DOIT alors établir le fanion Partiel sur cet attribut. Le fanion Partiel indique que tous les LS le long du chemin pertinent n'ont pas traité et

compris l'attribut. Pour les attributs transitifs indépendants, le "chemin pertinent" est le chemin donné dans l'attribut CheminD'Annonce. Pour les attributs transitifs dépendants, le chemin pertinent ne consiste qu'en les domaines à travers lesquels cet objet a passé depuis que le serveur de prochain bond a été modifié. Le fanion Partiel dans un attribut transitif indépendant NE DOIT être modifié par aucun autre LS le long du chemin. Le fanion Partiel dans un attribut transitif dépendant DOIT être rétabli chaque fois que le ServeurDeProchainBond est changé, mais NE DOIT être désétabli par aucun LS qui ne change pas le ServeurDeProchainBond.

Les règles qui gouvernent l'ajout de nouveaux attributs non transitifs sont définies indépendamment pour chaque attribut non transitif. Tout attribut PEUT être mis à jour pour un LS sur le chemin.

4.3.2.3 Fanions d'attribut et agrégation de chemin

Chaque attribut définit comment il doit être traité durant l'agrégation de chemin.

Les règles qui gouvernent le traitement des attributs inconnus sont guidées par les fanions d'attribut. Les attributs transitifs non reconnus sont abandonnés durant l'agrégation. Il ne devrait pas y avoir d'attribut non transitif non reconnu durant l'agrégation parce que les attributs non transitifs doivent être traités par le LS local afin d'être propagés.

4.3.2.4 Fanions d'attribut et encapsulation

Normalement, les attributs ont le simple format décrit au paragraphe 4.3.1. Si le fanion Encapsulation d'état de liaison est établi, les deux champs supplémentaires sont ajoutés à l'en-tête d'attribut comme le montre la Figure 9.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Fanions d'attr|Code type d'att|          Longueur d'attribut          |
+-----+-----+-----+-----+-----+-----+-----+
|                               Identifiant de générateur TRIP                               |
+-----+-----+-----+-----+-----+-----+-----+
|                               Numéro de séquence                               |
+-----+-----+-----+-----+-----+-----+-----+
|                               Valeur d'attribut (variable)                               |
+-----+-----+-----+-----+-----+-----+-----+

```

Figure 9 : Encapsulation d'état de liaison

L'identifiant de générateur TRIP et le numéro de séquence sont utilisés pour contrôler l'arrosage des mises à jour d'acheminement au sein d'une collection de serveurs. Ces champs sont utilisés pour détecter les chemins dupliqués et périmés afin qu'ils ne soient plus propagés aux autres LS. L'utilisation de ces champs est définie au paragraphe 10.1.

4.3.3 Attributs obligatoires

Il n'y a pas d'attribut obligatoire dans TRIP. Cependant, il y a des attributs conditionnellement obligatoires. Un attribut conditionnellement obligatoire est un attribut qui DOIT être inclus dans un message METTREÀJOUR si un autre attribut est inclus dans ce message. Par exemple, si un message METTREÀJOUR inclut un attribut CheminsAccessibles, il DOIT inclure aussi un attribut CheminD'Annonce.

Les trois attributs de base dans TRIP sont CheminsRetirés, CheminsAccessibles, et Topologie ITAD. Leur présence dans un message METTREÀJOUR est entièrement facultative et indépendante de tout autre attribut.

4.3.4 Attributs TRIP METTREÀJOUR

Ce paragraphe récapitule les attributs qui peuvent être portés dans un message METTREÀJOUR. Les attributs DOIVENT apparaître dans le message METTREÀJOUR en ordre croissant de code de type d'attribut. Des détails complémentaires sont fournis à la section 5.

4.3.4.1 CheminsRetirés

Cet attribut fait la liste d'un ensemble de chemins qui sont retirés du service. Le LS transmetteur a déterminé que ces chemins ne devraient plus être annoncés, et il propage cette information à ses homologues.

4.3.4.2 CheminsAccessibles

Cet attribut fait la liste d'un ensemble de chemins qui sont ajoutés au service. Ces chemins pourront être insérés dans le Adj-TRIBs-In du LS receveur et le processus de choix du chemin leur sera appliqué.

4.3.4.3 ServeurDeProchainBond

Cet attribut donne l'identité de l'entité à laquelle les messages devraient être envoyés le long de ce chemin. Il spécifie l'identité du serveur de prochain bond comme un nom de domaine d'hôte ou une adresse IP. Il PEUT facultativement spécifier le numéro d'accès UDP/TCP pour le serveur de signalisation du prochain bond. Si il n'est pas spécifié, alors l'accès par défaut DEVRAIT être utilisé. Le ServeurDeProchainBond est spécifique de l'ensemble de destinations et de protocoles d'application définis dans l'attribut CheminsAccessibles. Noter que ceci N'EST PAS nécessairement l'adresse à laquelle les supports (voix, vidéo, etc.) devraient être transmis ; c'est seulement pour le protocole d'application tel que donné dans l'attribut CheminsAccessibles.

4.3.4.4 CheminD'Annonce

Le CheminD'Annonce est analogue à l'AS_PATH dans BGP4 [RFC1771]. L'attribut enregistre la séquence des domaines à travers lesquels cette annonce est passée. L'attribut est utilisé pour détecter quand l'annonce d'acheminement fait une boucle. Cet attribut NE reflète PAS le chemin à travers lequel passent les messages qui suivent ce chemin. Comme le prochain bond n'a pas besoin d'être modifié par chaque LS, le chemin réel pour la destination peut n'avoir pas à traverser chacun des domaines du CheminD'Annonce.

4.3.4.5 CheminSuivi

L'attribut CheminSuivi est analogue à l'attribut CheminD'Annonce, sauf qu'il enregistre le chemin réel (donné par la liste des domaines) *vers* la destinations. À la différence de CheminD'Annonce, qui est modifié chaque fois que le chemin est propagé, CheminSuivi n'est modifié que lorsque l'attribut ServeurDeProchainBond change. Donc, il enregistre le sous-ensemble du CheminD'Annonce que vont traverser les messages de signalisation qui suivent ce chemin particulier.

4.3.4.6 AgrégatAtomique

L'attribut AgrégatAtomique indique qu'un chemin peut en fait inclure des domaines non énumérés dans l'attribut CheminSuivi. Si un LS, lorsque il est présenté avec un ensemble de chemins qui se chevauchent avec ceux d'un LS homologue, choisit un chemin moins spécifique sans choisir le chemin le plus spécifique, le LS DOIT alors inclure l'attribut AgrégatAtomique avec le chemin. Un LS qui reçoit un chemin avec un attribut AgrégatAtomique NE DOIT PAS rendre l'ensemble de destinations plus spécifique quand il l'annonce aux autres LS.

4.3.4.7 PréférenceLocale

L'attribut PréférenceLocale est un attribut intra-domaine utilisé pour informer les autres LS de la préférence du LS local pour un certain chemin. La préférence d'un chemin est calculée à l'entrée d'un domaine et passée comme un attribut avec ce chemin à travers le domaine. Les autres LS au sein du même ITAD utilisent cet attribut dans leur processus de choix de chemin. Cet attribut n'a pas de signification entre les domaines.

4.3.4.8 DécouvMultiSorties

Il peut y avoir plus d'une relation de LS homologue entre des domaines voisins. L'attribut DécouvMultiSorties est utilisé par un LS pour exprimer une préférence pour une liaison entre les domaines par rapport à une autre liaison entre ces domaines. L'utilisation de l'attribut DécouvMultiSorties est contrôlée par la politique locale.

4.3.4.9 Communautés

L'attribut Communautés n'est pas un attribut bien connu. Il est utilisé pour faciliter et simplifier le contrôle des informations d'acheminement en groupant les destinations en communautés.

4.3.4.10 Topologie d'ITAD

L'attribut Topologie d'ITAD est un attribut intra domaine qui est utilisé par les LS pour indiquer leur topologie intra

domaine aux autres LS dans le domaine.

4.3.4.11 CheminConverti

L'attribut CheminConverti indique qu'un LS intermédiaire a altéré le chemin en changeant le protocole d'application du chemin.

4.4 Format du message GARDERENVIE

TRIP n'utilise aucun mécanisme de garde en vie fondé sur le transport pour déterminer si les homologues sont accessibles. À la place, des messages GARDERENVIE sont échangés entre les homologues assez souvent pour que le temporisateur de garde n'arrive pas à expiration. Un délai maximum raisonnable entre les messages GARDERENVIE serait un tiers de l'intervalle de temps de garde. Les messages GARDERENVIE NE DOIVENT PAS être envoyés plus d'une fois toutes les trois secondes. Une mise en œuvre DEVRAIT ajuster le taux d'envoi des messages GARDERENVIE en fonction de l'intervalle négocié de temps de garde.

Si l'intervalle négocié de temps de garde est zéro, alors les messages périodiques GARDERENVIE NE DOIVENT PAS être envoyés.

Le message GARDERENVIE consiste seulement en un en-tête de message et il a une longueur de trois octets.

4.5 Format du message NOTIFICATION

Un message NOTIFICATION est envoyé lorsque une condition d'erreur est détectée. La connexion de transport TRIP est fermée immédiatement après l'envoi d'un message NOTIFICATION.

En plus de l'en-tête TRIP de taille fixe, le message NOTIFICATION contient les champs suivants :

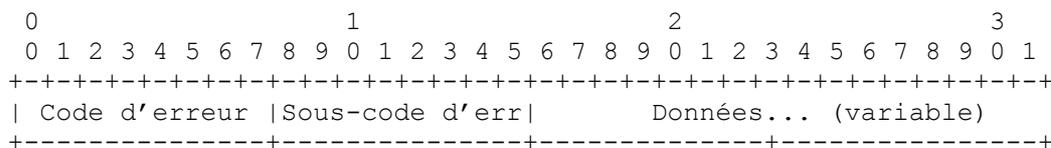


Figure 10 : Format du message TRIP NOTIFICATION

Code d'erreur :

Cet entier non signé d'un octet indique le type de NOTIFICATION. Les codes d'erreur suivants ont été définis :

Code d'erreur	Nom symbolique	Référence
1	Erreur d'en-tête de message	paragraphe 6.1
2	Erreur de message OUVERT	paragraphe 6.2
3	Erreur de message METTREÀJOUR	paragraphe 6.3
4	Expiration du temporisateur de garde	paragraphe 6.5
5	Erreur de l'automate à états finis	paragraphe 6.6
6	Cesser	paragraphe 6.7

Sous-code d'erreur :

Cet entier non signé d'un octet fournit des informations plus spécifiques sur la nature de l'erreur en cause. Chaque code d'erreur peut avoir un ou plusieurs sous-codes d'erreur qui lui sont associés. Si aucun sous-code d'erreur approprié n'est défini, une valeur de zéro (non spécifique) est utilisée pour le champ Sous-code d'erreur.

Sous-codes d'erreur d'en-tête de message :

- 1 - Mauvaise longueur de message.
- 2 - Mauvais type de message.

Sous-code d'erreur de message OUVERT :

- 1 - Numéro de version non accepté.
- 2 - Mauvais ITAD homologue.
- 3 - Mauvais identifiant TRIP.
- 4 - Paramètre facultatif non accepté.

- 5 - Temps de garde non acceptable.
- 6 - Capacité non acceptée.
- 7 - Incompatibilité de capacités.

Sous-code d'erreur de message METTREÀJOUR :

- 1 – Liste d'attributs mal formée.
- 2 – Attribut bien connu non reconnu.
- 3 – Attribut obligatoire bien connu manquant.
- 4 – Erreur de fanions d'attribut.
- 5 – Erreur de longueur d'attribut.
- 6 – Attribut invalide.

Données :

Ce champ de longueur variable est utilisé pour diagnostiquer la raison de la NOTIFICATION. Le contenu du champ Données dépend du code d'erreur et du sous-code d'erreur.

Noter que la longueur des données peut être déterminée d'après le champ Longueur du message par la formule :

$$\text{Longueur des données} = \text{Longueur du message} - 5$$

La longueur minimale du message NOTIFICATION est de 5 octets (y compris l'en-tête de message).

5. Attributs TRIP

Cette section donne des détails sur la syntaxe et la sémantique de chaque attribut METTREÀJOUR TRIP.

5.1 CheminsRetirés

Obligatoire conditionnel : Faux.

Fanions exigés : Bien connus

Fanions potentiels : Encapsulation d'état de liaison (lors de l'arrosage).

Code de type TRIP : 1

L'attribut CheminsRetirés spécifie un ensemble de chemins qui sont à supprimer du service par le ou les LS receveurs. L'ensemble de chemins PEUT être vide, ce qui est indiqué par un champ de longueur de zéro.

5.1.1 Syntaxe de CheminsRetirés

L'attribut CheminsRetirés code une séquence de chemins dans son champ Valeur. Le format des chemins individuels est donné au paragraphe 5.1.1.1. L'attribut CheminsRetirés fait la liste des chemins individuels à la suite les uns des autres sans bourrage comme le montre la Figure 11. Chaque chemin comporte un champ Longueur de sorte que au sein de l'attribut chaque chemin individuel puisse être délimité.

```
+-----+-----+...
| CheminRetiré1... | CheminRetiré2... |...
+-----+-----+...
```

Figure 11 : Format de CheminsRetirés

5.1.1.1 Format générique de chemin TRIP

Le format générique d'un chemin TRIP est donné à la Figure 12.

```

0                               1                               2                               3
0 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Famille d'adresses           |           Protocole d'application           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Longueur           |           Adresse (variable)           |...
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 12 : Format générique de chemin TRIP

Famille d'adresse :

Le champ Famille d'adresse donne le type d'adresse pour le chemin. Trois familles d'adresse sont définies dans cette Section :

Code	Famille d'adresse
1	Numéros d'acheminement décimaux
2	Numéros d'acheminement pentadécimaux
3	Numéros E.164

Le présent document réserve le code de famille d'adresses 0. Il réserve les codes de famille d'adresses 32 768 à 65 535 pour les applications spécifiques de fabricant (ce sont les codes avec le premier bit de la valeur de code égal à 1). Des familles d'adresses supplémentaires pourront être définies à l'avenir. L'allocation des codes de famille d'adresses est sous le contrôle de l'IANA. Voir à la Section 13 les considérations relatives à l'IANA.

Protocole d'application :

Le protocole d'application indique le protocole pour lequel ce tableau d'acheminement est conservé. Les protocoles d'application actuellement définis sont :

Code	Protocole
1	SIP
2	H.323-H.225.0-Q.931
3	H.323-H.225.0-RAS
4	H.323-H.225.0-Annexe-G

Le présent document réserve le code de protocole d'application 0. Il réserve les codes de protocole d'application 32 768 à 65 535 pour les applications spécifiques de fabricant (ce sont les codes avec le premier bit de la valeur de code égal à 1). Des protocoles d'application supplémentaires pourront être définis à l'avenir. L'allocation des codes de famille d'adresses est sous le contrôle de l'IANA. Voir à la Section 13 les considérations relatives à l'IANA.

Longueur :

C'est la longueur du champ Adresse, en octets.

Adresse :

C'est une adresse (préfixe) du type de famille donné par le champ Famille d'adresse. La longueur en octets de l'adresse est variable et est déterminée par le champ Longueur du chemin.

5.1.1.2 Numéros d'acheminement décimaux

La famille d'adresses Numéros d'acheminement décimaux est un sur-ensemble de tous les numéros E.164, des numéros nationaux, des numéros locaux, et des numéros privés. Elle peut aussi être utilisée pour représenter les numéros d'acheminement décimaux utilisés en conjonction avec la portabilité du numéro dans certains pays/régions. Un ensemble de numéros de téléphone est spécifié par un préfixe de numéro d'acheminement décimal. Les préfixes de numéros d'acheminement décimaux sont représentés par une chaîne de chiffres, chaque chiffre étant codé par sa représentation de caractères ASCII. Cet objet d'acheminement couvre tous les numéros de téléphone commençant par ce préfixe. La syntaxe du préfixe de numéro d'acheminement décimal est :

```
Numéro d'acheminement décimal = *chiffre-décimal
chiffre-décimal = CHIFFRE-DÉCIMAL
CHIFFRE-DÉCIMAL = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"
```

Ce préfixe de numéro d'acheminement DÉCIMAL n'est pas limité en longueur. Ce format est similaire à celui d'un numéro de téléphone mondial tel que défini dans SIP [RFC2543] sans séparateurs visuels et sans le préfixe "+" pour les numéros internationaux. Ce format facilite une comparaison efficace lorsque on utilise TRIP pour acheminer SIP ou H323, dont tous deux utilisent des représentations fondées sur le caractère de numéros de téléphone. La longueur du préfixe est déterminée à partir du champ Longueur du chemin. Le type de numéro d'acheminement décimal (privé, local, national, ou international) peut être déduit des premiers chiffres du préfixe.

5.1.1.3 Numéros d'acheminement pentadécimaux

Cette famille d'adresses est utilisée pour représenter les numéros d'acheminement pentadécimaux utilisés en conjonction avec la portabilité du numéro dans certains pays/régions. Les préfixes de numéros d'acheminement pentadécimaux sont

représentés par une chaîne de chiffres, chaque chiffre étant codé par sa représentation de caractère ASCII. Cet objet d'acheminement couvre tous les numéros d'acheminement commençant par ce préfixe. La syntaxe du préfixe de numéro d'acheminement pentadécimal est :

```
Numéro d'acheminement pentadécimal = *chiffre-pentadecimal*
chiffre-pentadecimal = CHIFFRE-PENTADECIMAL
CHIFFRE-PENTADECIMAL = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"A"|"B"|"C"|"D"|"E"
```

Noter la différence des alphabets entre numéros d'acheminement décimaux et numéros d'acheminement pentadécimaux. Un préfixe de numéro d'acheminement pentadécimal n'est pas limité en longueur.

Noter que la famille d'adresses, qui convient aux numéros d'acheminement d'un pays/région spécifique dépend de l'alphabet utilisé pour les numéros d'acheminement dans ce pays/région. Par exemple, les numéros d'acheminement de l'Amérique du Nord DEVRAIENT utiliser la famille d'adresses des numéros d'acheminement décimaux, parce que leur alphabet est limité aux chiffres de "0" à "9". Un autre exemple, dans la plupart des pays d'Europe, les numéros d'acheminement utilisent l'alphabet de "0" à "9" et de "A" à "E", et donc ces pays DEVRAIENT utiliser la famille d'adresses des numéros d'acheminement pentadécimaux.

5.1.1.4 Numéros E.164

La famille d'adresses des numéros E.164 est dédiée aux numéros E.164 pleinement qualifiés. Un ensemble de numéros de téléphone est spécifié par un préfixe E.164. Les préfixes E.164 sont représentés par une chaîne de chiffres, chaque chiffre est codé par sa représentation en caractères ASCII. Cet objet d'acheminement couvre tous les numéros de téléphone qui commencent par ce préfixe. La syntaxe du préfixe E.164 est :

```
Numéro-E164 = *chiffre-e164
chiffre-e164 = CHIFFRE-E164
CHIFFRE-E164 = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"
```

Ce format facilite l'efficacité de la comparaison lorsque on utilise TRIP pour acheminer SIP ou H323, qui utilisent tous deux les représentations fondées sur le caractère des numéros de téléphone. La longueur de préfixe est déterminée à partir du champ Longueur du chemin.

La famille d'adresses des numéros E.164 et la famille d'adresses des numéros d'acheminement décimaux ont le même alphabet. La famille d'adresses des numéros E.164 DEVRAIT être utilisée chaque fois que possible. La famille d'adresses des numéros d'acheminement décimaux peut être utilisée dans le cas de plans de numérotage privés ou d'applications qui ne désirent pas annoncer des numéros de téléphone complètement développés, pleinement qualifiés. Si les numéros d'acheminement décimaux sont utilisés pour annoncer des préfixes non pleinement qualifiés, les préfixes peuvent devoir être manipulés (par exemple, développés) à la frontière entre les ITAD. Cela ajoute une complexité significative au LS frontière d'ITAD, parce que il va devoir transposer les préfixes du format utilisé dans son propre ITAD au format utilisé dans l'ITAD homologue.

5.2 CheminsAccessibles

Obligatoire conditionnel : Faux.

Fanions exigés : Bien connus.

Fanions potentiels : Encapsulation d'état de liaison (lors de l'arrosage).

Code de type TRIP : 2

L'attribut CheminsAccessibles spécifie un ensemble de chemins qui sont à ajouter au service par le ou les LS receveurs. L'ensemble des chemins PEUT être vide, comme indiqué en réglant le champ Longueur à zéro.

5.2.1 Syntaxe de CheminsAccessibles

L'attribut CheminsAccessibles a la même syntaxe que l'attribut CheminsRetirés. Voir au paragraphe 5.1.1.

5.2.2 Origine de chemin et CheminsAccessibles

Les chemins sont injectés dans TRIP par une méthode qui sort du domaine d'application de la présente spécification. Les méthodes possibles incluent un protocole frontal, un protocole d'acheminement intra-domaine, ou une configuration statique.

5.2.3 Choix de chemin et CheminsAccessibles

Les chemins qui sont dans l'attribut CheminsAccessibles sont nécessaires pour le choix du chemin.

5.2.4 Agrégation et CheminsAccessibles

Pour agréger plusieurs chemins, l'ensemble des CheminsAccessibles à agréger DOIT se combiner pour former un ensemble moins spécifique.

Il n'y a pas de mécanisme au sein de TRIP pour communiquer qu'un préfixe d'adresse particulier n'est pas utilisé et donc que ces adresses pourraient être sautées durant l'agrégation. Les LS PEUVENT utiliser des méthodes qui sortent du domaine d'application de TRIP pour apprendre les préfixes invalides qui peuvent être ignorés durant l'agrégation.

Si un LS annonce un chemin agrégé, il DOIT inclure l'attribut AgrégatAtomique.

5.2.5 Dissémination de chemin et CheminsAccessibles

L'attribut CheminsAccessibles est recalculé à chaque LS sauf lorsque l'arrosage est utilisé (par exemple, au sein d'un domaine). Il est donc possible qu'un LS change le champ Protocole d'application d'un chemin avant d'annoncer ce chemin à un homologue externe.

Si un LS change le protocole d'application d'un chemin qu'il annonce, il DOIT inclure l'attribut CheminConverti dans le message METTREÀJOUR.

5.2.6 Spécificités de l'agrégation pour les numéros d'acheminement décimaux, pentadécimaux et E.164

Un LS qui a des chemins pour tous les numéros valides dans un préfixe spécifique DEVRAIT annoncer ce préfixe comme CheminsAccessibles, même si il y a des préfixes plus spécifiques qui n'existent en fait pas sur le RTPC. Généralement, cela prend dix préfixes d'acheminement décimal/E.164, ou quinze préfixes d'acheminement pentadécimal, de longueur n pour agréger dans un préfixe de longueur n-1. Cependant, si un LS sait qu'un préfixe est dans un préfixe d'acheminement décimal/E.164 invalide, ou un préfixe d'acheminement pentadécimal, le LS PEUT alors agréger en sautant ce préfixe. Par exemple, si il est connu que le préfixe d'acheminement décimal 19191 n'existe pas, un LS peut alors agréger sur 1919 sans 19191. Un préfixe qui représente un ensemble invalide de destinations du RTPC est parfois appelé un "trou noir". La méthode par laquelle un LS prend connaissance des trous noirs sort du domaine d'application de TRIP, mais si un LS a une telle information, il peut l'utiliser lors de l'agrégation.

5.3 ServeurDeProchainBond

Obligatoire conditionnel : Vrai (si l'attribut CheminsAccessibles et/ou CheminsRetirés est présent).

Fanions exigés : Bien connus.

Fanions potentiels : Aucun.

Code de type TRIP : 3.

Étant donné un chemin avec un protocole d'application A et la destination D, le ServeurDeProchainBond indique au prochain bond que les messages du protocole A destinés à D devraient lui être envoyés. Cela peut ou non représenter la destination ultime de ces messages.

5.3.1 Syntaxe de ServeurDeProchainBond

En général, l'adresse du serveur de prochain bond peut être de divers types (nom de domaine, IPv4, IPv6, etc.). L'attribut ServeurDeProchainBond comporte le numéro d'ITAD du serveur de prochain bond, un champ de longueur, et un nom ou adresse de prochain bond.

La syntaxe de ServeurDeProchainBond est donnée à la Figure 13.



Figure 13 : Syntaxe de ServeurDeProchainBond

L'ITAD de prochain bond indique le domaine du prochain bond. Le champ Longueur donne le nombre d'octets dans le champ Serveur, et le champ Serveur contient le nom ou l'adresse du serveur de prochain bond. Le champ Serveur est représenté par une chaîne de caractères ASCII. Il est défini comme suit :

Serveur = hôte [":" accès]

hôte = < Un nom de domaine d'hôte Internet légal ou une adresse IPv4 qui utilise la représentation textuelle définie au paragraphe 2.1 de la [RFC1123] ou une adresse IPv6 qui utilise la représentation textuelle définie au paragraphe .2 de la [RFC2373]. L'adresse IPv6 DOIT être incluse entre les caractères "[" et "]">

accès = *CHIFFRE

Si l'accès est vide, ou n'est pas donné, l'accès par défaut est supposé (par exemple, accès 5060 si le protocole d'application est SIP).

5.3.2 Origine de chemin et ServeurDeProchainBond

Lorsque un LS génère un objet d'acheminement dans TRIP, il DOIT inclure un ServeurDeProchainBond dans son domaine. Le ServeurDeProchainBond pourrait être une adresse de la passerelle de sortie ou d'un mandataire de signalisation.

5.3.3 Choix de chemin et ServeurDeProchainBond

La politique du LS peut préférer certains prochains bonds ou domaines de prochain bond à d'autres.

5.3.4 Agrégation et ServeurDeProchainBond

Lors de l'agrégation de plusieurs objets d'acheminement en un seul objet d'acheminement, un LS DOIT insérer un nouveau serveur de signalisation provenant de l'intérieur de son domaine comme nouveau ServeurDeProchainBond sauf si tous les chemins agrégés ont le même prochain bond.

5.3.5 Dissémination des chemins et ServeurDeProchainBond

Lorsque il propage des objets d'acheminement à des homologues, un LS peut choisir d'insérer un mandataire de signalisation au sein de son domaine comme nouveau prochain bond, ou il peut laisser inchangé le prochain bond. Insérer un nouveau prochain bond causera l'envoi des messages de signalisation à cette adresse, et va permettre un contrôle plus fin sur le chemin de signalisation. Laisser le prochain bond inchangé va donner un chemin de signalisation plus efficace (moins de bonds). C'est une décision locale du LS de décider de propager le ServeurDeProchainBond ou de le changer.

5.4 CheminD'Annonce

Obligatoire conditionnel : Vrai (si l'attribut CheminsAccessibles et/ou CheminsRetirés est présent).

Fanions exigés : Bien connu.

Fanions potentiels : Aucun.

Code de type TRIP : 4.

Cet attribut identifie les ITAD à travers lesquels sont passées les informations d'acheminement portées dans une annonce. L'attribut CheminD'Annonce est analogue à l'attribut AS_PATH dans BGP. Les attributs diffèrent en ce que l'AS_PATH de BGP reflète aussi le chemin vers la destination. Dans TRIP, tous les domaines n'ont pas besoin de modifier le prochain bond, de sorte que CheminD'Annonce peut inclure beaucoup plus de bonds que le chemin réel jusqu'à la destination. L'attribut CheminSuivi (paragraphe 5.5) reflète le chemin de signalisation réel vers la destination.

5.4.1 Syntaxe de CheminD'Annonce

CheminD'Annonce est un attribut de longueur variable qui est composé d'une séquence de segments de chemin d'ITAD. Chaque segment de chemin d'ITAD est représenté par un triplet type-longueur-valeur.

Le type du segment de chemin est un champ d'un octet dont les valeurs suivantes sont définies :

Valeur Type de segment

- 1 AP_SET : ensemble non ordonné des ITAD qu'un chemin a traversé dans le message d'annonce
- 2 AP_SEQUENCE : ensemble ordonné des ITAD qu'un chemin a traversé dans le message d'annonce

La longueur du segment de chemin est un champ d'un octet qui contient le nombre des ITAD dans le champ Valeur du

segment de chemin.

Le champ Valeur du segment de chemin contient un ou plusieurs numéros d'ITAD, chacun codé par un champ de 4 octets. Les numéros d'ITAD identifient de façon univoque un domaine administratif de téléphonie Internet, et doivent être obtenus de l'IANA. Voir à la Section 13 les procédures pour obtenir un numéro d'ITAD de l'IANA.

5.4.2 Origine de chemin et CheminD'Annonce

Lorsque un LS génère un chemin, alors :

- Le LS générateur devra inclure son propre numéro d'ITAD dans l'attribut CheminD'Annonce de toutes les annonces envoyées aux LS localisés dans les ITAD voisins. Dans ce cas, le numéro d'ITAD de l'ITAD du LS générateur sera la seule entrée dans l'attribut CheminD'Annonce.
- Le LS générateur devra inclure un attribut CheminD'Annonce vide dans toutes les annonces envoyées aux LS localisés dans son propre ITAD. Un attribut CheminD'Annonce vide est celui dont le champ longueur contient la valeur zéro.

5.4.3 Choix de chemin et CheminD'Annonce

Le CheminD'Annonce peut être utilisé pour le choix du chemin. Les critères de choix possibles sont le nombre de bonds sur le chemin et la présence ou l'absence de certains ITAD sur le chemin.

Comme exposé à la Section 10, le CheminD'Annonce est utilisé pour empêcher les informations d'acheminement de faire une boucle. Si un LS reçoit un chemin avec son propre ITAD qui figure déjà dans le CheminD'Annonce, le chemin DOIT être éliminé.

5.4.4 Agrégation et CheminD'Annonce

Les règles pour agréger les attributs CheminD'Annonce sont données dans les paragraphes qui suivent, où le terme "chemin" utilisé aux paragraphes 5.4.4.1 et 5.4.4.2 est entendu comme signifiant CheminD'Annonce.

5.4.4.1 Agrégation de chemins avec des chemins identiques

Si tous les chemins à agréger ont des attributs de chemin identiques, le chemin agrégé a le même attribut de chemin que les chemins individuels.

5.4.4.2 Agrégation de chemins avec des chemins différents

Pour les besoins de l'agrégation des attributs de chemin, on modélise chaque ITAD au sein du chemin comme une paire <type, valeur>, où "type" identifie un type du segment de chemin (AP_SEQUENCE ou AP_SET) et "valeur" est le numéro d'ITAD. Deux ITAD sont dits être le même si leurs <type, valeur> correspondants sont les mêmes.

Si les chemins à agréger ont des attributs de chemin différents, alors l'attribut du chemin agrégé devra satisfaire toutes les conditions suivantes :

- Toutes les paires de type AP_SEQUENCE dans le chemin agrégé DOIVENT apparaître dans tous les chemins des chemins à agréger.
- Toutes les paires de type AP_SET dans le chemin agrégé DOIVENT apparaître dans au moins un des chemins de l'ensemble initial (ils peuvent apparaître comme type AP_SET ou AP_SEQUENCE).
- Pour toute paire X du type AP_SEQUENCE qui précède la paire Y dans le chemin agrégé, X précède Y dans chaque chemin de l'ensemble initial qui contient Y, sans considération du type de Y.
- Aucune paire avec la même valeur ne devra apparaître plus d'une fois dans le chemin agrégé, sans considération du type de la paire.

Une mise en œuvre peut choisir tout algorithme qui se conforme à ces règles. Au minimum, une mise en œuvre conforme DOIT être capable d'effectuer l'algorithme suivant qui satisfait à toutes les conditions ci-dessus :

- Déterminer la plus longue séquence de tuplets de début (comme définie ci-dessus) commune à tous les chemins à agréger. Faire de cette séquence la séquence de début du chemin agrégé.
- Régler le type du reste des tuplets provenant des chemins à agréger à AP_SET, et les ajouter au chemin agrégé.
- Si le chemin agrégé a plus d'un tuple avec la même valeur (sans considération du type du tuple) éliminer tous ces tuplets sauf un en supprimant les tuplets du type AP_SET du chemin agrégé.

Une mise en œuvre qui choisit de fournir un algorithme d'agrégation de chemin qui conserve des quantités significatives d'informations de chemin peut souhaiter utiliser la procédure du paragraphe 5.4.4.3.

5.4.4.3 Exemple d'algorithme d'agrégation de chemins

Un exemple d'algorithme pour agréger deux chemins fonctionne comme suit :

- Identifier les ITAD (comme défini au paragraphe 5.4.1) au sein de chaque attribut de chemin qui sont dans le même ordre relatif au sein des deux attributs de chemin. Deux ITAD, X et Y, sont dits être dans le même ordre si X précède Y dans les deux chemins, ou si Y précède X dans les deux chemins.
- Le chemin agrégé consiste en ITAD identifiés dans (a) exactement le même ordre qu'ils apparaissent dans les chemins à agréger. Si deux ITAD consécutifs identifiés en (a) ne se suivent pas immédiatement l'un l'autre dans les deux chemins à agréger, alors les ITAD intercalés (les ITAD qui sont entre les deux ITAD consécutifs qui sont les mêmes) dans les deux attributs sont combinés dans un segment de chemin AP_SET qui consiste en les ITAD intercalés provenant des deux chemins ; ce segment est alors placé entre les deux ITAD consécutifs identifiés en (a) de l'attribut agrégé. Si deux ITAD consécutifs identifiés en (a) se suivent immédiatement l'un l'autre dans un attribut, mais pas dans un autre, alors les ITAD intercalés du dernier sont combinés en un segment de chemin AP_SET ; ce segment est alors placé entre les deux ITAD consécutifs identifiés en (a) du chemin agrégé.

Si par suite de la procédure ci-dessus un certain numéro d'ITAD apparaît plus d'une fois au sein du chemin agrégé, toutes les instances sauf la dernière (l'occurrence la plus à droite) de ce numéro d'ITAD devraient être retirées du chemin agrégé.

5.4.5 Dissémination de chemins et CheminD'Annonce

Lorsque un LS propage un chemin qu'il a appris d'un autre LS, il doit modifier l'attribut CheminD'Annonce du chemin sur la base de la localisation du LS auquel le chemin sera envoyé.

- Lorsque un LS annonce un chemin à un autre LS situé dans son propre ITAD, le LS annonceur NE DOIT PAS modifier l'attribut CheminD'Annonce associé au chemin.
- Lorsque un LS annonce un chemin à un LS situé dans un ITAD voisin, le LS annonceur DOIT alors mettre à jour l'attribut CheminD'Annonce comme suit :
 - * Si le premier segment de chemin du CheminD'Annonce est du type AP_SEQUENCE, le système local devra ajouter en tête son propre numéro d'ITAD comme dernier élément de la séquence (le mettre dans la position la plus à gauche).
 - * Si le premier segment de chemin du CheminD'Annonce est du type AP_SET, le système local devra ajouter en tête un nouveau segment de type AP_SEQUENCE au CheminD'Annonce, incluant son propre numéro d'ITAD dans ce segment.

5.5 CheminSuivi

Obligatoire conditionnel : Vrai (si l'attribut CheminsAccessibles est présent).

Fanions exigés : Bien connu.

Fanions potentiels : Aucun.

Code de type TRIP : 5.

Cet attribut identifie les ITAD à travers lesquels vont passer les messages envoyés en utilisant ce chemin. Les ITAD dans ce chemin sont un sous ensemble de ceux qui sont dans CheminD'Annonce.

5.5.1 Syntaxe de CheminSuivi

La syntaxe de l'attribut CheminSuivi est la même que celle de l'attribut CheminD'Annonce. Voir au paragraphe 5.4.1.

5.5.2 Origine de chemin et CheminSuivi

Lorsque un LS génère un chemin, il DOIT inclure l'attribut CheminSuivi.

- Le LS d'origine devra inclure son propre numéro d'ITAD dans l'attribut CheminSuivi de toutes les annonces envoyées aux LS situés dans les ITAD du voisinage. Dans ce cas, le numéro d'ITAD de l'ITAD du LS d'origine sera la seule entrée dans l'attribut CheminSuivi.
- Le LD d'origine devra inclure un attribut CheminSuivi vide dans toutes les annonces envoyées aux LS situés dans son propre ITAD. Un attribut CheminSuivi vide est celui dont le champ Longueur contient la valeur zéro.

5.5.3 Choix de chemin et CheminSuivi

Le CheminSuivi PEUT être utilisé pour le choix du chemin, et dans la plupart des cas il est préféré au CheminD'Annonce dans ce rôle. Des critères possibles sont le nombre de bonds sur le chemin, et la présence ou l'absence de certains ITAD sur le chemin.

5.5.4 Agrégation de chemins et CheminSuivi

Les règles pour agréger les attributs CheminSuivi sont données aux paragraphes 5.4.4.1 et 5.4.4.2, où le terme "chemin" utilisé aux paragraphes 5.4.4.1 et 5.4.4.2 est entendu comme signifiant CheminSuivi.

5.5.5 Dissémination de chemins et CheminSuivi

Lorsque un LS propage un chemin qu'il a appris d'un autre LS, il modifie l'attribut CheminSuivi du chemin sur la base de la localisation du LS auquel le chemin est envoyé.

- Lorsque un LS annonce un chemin à un autre LS situé dans son propre ITAD, le LS annonceur NE DOIT PAS modifier l'attribut CheminSuivi associé au chemin.
- Si le LS n'a pas changé l'attribut ServeurDeProchainBond, le LS NE DOIT alors PAS changer l'attribut CheminSuivi.
- Autrement, le LS a changé le ServeurDeProchainBond et annonce le chemin à un LS dans un autre ITAD. Le LS annonceur DOIT mettre à jour l'attribut CheminSuivi comme suit :
 - * Si le premier segment de chemin du CheminSuivi est du type AP_SEQUENCE, le système local devra ajouter en tête son propre numéro d'ITAD comme dernier élément de la séquence (le mettre dans la position la plus à gauche).
 - * Si le premier segment de chemin du CheminSuivi est du type AP_SET, le système local devra ajouter en tête un nouveau segment de chemin du type AP_SEQUENCE au CheminSuivi, incluant son propre numéro d'ITAD dans ce segment.

5.6 AgrégatAtomique

Obligatoire conditionnel : Faux.

Fanions exigés : Bien connu.

Fanions potentiels : Aucun.

Code de type TRIP : 6.

L'attribut AgrégatAtomique indique qu'un chemin peut traverser des domaines non énumérés dans CheminSuivi. Si un LS, lorsque un LS homologue lui présente un ensemble de chemins qui se chevauchent, choisit un chemin moins spécifique sans retenir le chemin le plus spécifique, le LS inclut alors l'attribut AgrégatAtomique avec l'objet d'acheminement.

5.6.1 Syntaxe de AgrégatAtomique

Cet attribut est de longueur zéro (0) ; le champ Valeur est vide.

5.6.2 Origine ce chemin et AgrégatAtomique

Les chemins ne sont jamais générés avec l'attribut AgrégatAtomique.

5.6.3 Choix de chemin et AgrégatAtomique

L'attribut AgrégatAtomique peut être utilisé pour le choix de chemin - il indique que le CheminSuivi peut être incomplet.

5.6.4 Agrégation et AgrégatAtomique

Si un des chemins à agréger a l'attribut AgrégatAtomique, l'agrégat résultant DOIT l'avoir aussi.

5.6.5 Dissémination de chemin et AgrégatAtomique

Si un LS, lorsque un LS homologue lui présente un ensemble de chemins qui se chevauchent, choisit le chemin le moins spécifique (voir la Section 1) sans choisir le chemin le plus spécifique, alors le LS DOIT inclure l'attribut AgrégatAtomique avec l'objet d'acheminement (si il n'est pas déjà présent).

Un LS qui reçoit un objet d'acheminement avec un attribut AgrégatAtomique NE DOIT PAS rendre l'ensemble de destinations plus spécifique lorsque il l'annonce aux autres LS, et NE DOIT PAS retirer l'attribut lorsque il propage cet objet à un LS homologue.

5.7 PréférenceLocale

Obligatoire conditionnel : Faux.
Fanions exigés : Bien connu.
Fanions potentiels : Aucun.
Code de type TRIP : 7.

L'attribut PréférenceLocale n'est utilisé qu'en intra domaine ; il indique la préférence du LS local pour l'objet d'acheminement aux autres LS au sein du même domaine. Cet attribut NE DOIT PAS être inclus lors d'une communication avec un LS dans un autre domaine, et il DOIT être inclus sur des liaisons intra domaine.

5.7.1 Syntaxe de PréférenceLocale

L'attribut PréférenceLocale est une valeur numérique non signée de 4 octets. Une valeur supérieure indique une préférence plus élevée.

5.7.2 Origine de chemin et PréférenceLocale

Les chemins NE DOIVENT PAS être générés avec l'attribut PréférenceLocale aux homologues inter domaine. Les chemins vers des homologues intra domaine DOIVENT être générés avec l'attribut PréférenceLocale.

5.7.3 Choix de chemin et PréférenceLocale

L'attribut PréférenceLocale permet à un LS dans un domaine de calculer une préférence pour un chemin, et de communiquer cette préférence aux autres LS au sein du domaine.

5.7.4 Agrégation et PréférenceLocale

L'attribut PréférenceLocale n'est pas affecté par l'agrégation.

5.7.5 Dissémination de chemin et PréférenceLocale

Un LS DOIT inclure l'attribut PréférenceLocale lorsqu'il communique avec des LS homologues au sein de son propre domaine. Un LS NE DOIT PAS inclure l'attribut PréférenceLocale lorsqu'il communique avec des LS dans d'autres domaines. Les attributs PréférenceLocale reçus d'homologues inter domaine DOIVENT être ignorés.

5.8 DécouvMultiSorties

Obligatoire conditionnel : Faux.
Fanions exigés : Bien connu.
Fanions potentiels : Aucun.
Code de type TRIP : 8.

Lorsque deux ITAD sont connectés par plus d'un ensemble d'homologues, l'attribut DécouvMultiSorties peut être utilisé pour spécifier les préférences pour les chemins reçus sur une de ces liaisons plutôt que les chemins reçus sur d'autres liaisons. Le paramètre DécouvMultiSorties n'est utilisé que pour le choix du chemin.

5.8.1 Syntaxe de DécouvMultiSorties

L'attribut DécouvMultiSorties porte une valeur numérique de quatre octets non signée. Une valeur supérieure représente un objet d'acheminement de plus forte préférence.

5.8.2 Origine ce chemin et DécouvMultiSorties

Les chemins générés vers des homologues intra domaine NE DOIVENT PAS être générés avec l'attribut DécouvMultiSorties. Pour générer un chemin pour un homologue inter domaine, l'attribut DécouvMultiSorties peut être inclus.

5.8.3 Choix de chemin et DécouvMultiSorties

L'attribut DécouvMultiSorties est utilisé pour exprimer une préférence lorsque il y a plusieurs liaisons entre deux domaines. Si tous les autres facteurs sont égaux, alors un chemin avec un plus fort attribut DécouvMultiSorties sera préféré à un chemin avec un attribut DécouvMultiSorties plus faible.

5.8.4 Agrégation et DécouvMultiSorties

Les chemins qui ont des paramètres DécouvMultiSorties différents NE DOIVENT PAS être agrégés. Les chemins avec la même valeur dans l'attribut DécouvMultiSorties PEUVENT être agrégés et le même attribut DécouvMultiSorties être rattaché à l'objet agrégé.

5.8.5 Dissémination de chemin et DécouvMultiSorties

Si il est reçu d'un LS homologue dans un autre domaine, un LS PEUT propager l'attribut DécouvMultiSorties aux autres LS au sein de son domaine. L'attribut DécouvMultiSorties NE DOIT PAS être propagé aux LS d'autres domaines.

Un LS peut ajouter l'attribut DécouvMultiSorties lors de la propagation d'objets d'acheminement à un LS dans un autre domaine. L'inclusion de l'attribut DécouvMultiSorties est une question de politique, comme l'est la valeur de l'attribut.

5.9 Communautés

Obligatoire conditionnel : Faux.

Fanions exigés : Pas Bien connu, transitif indépendant.

Fanions potentiels : Aucun.

Code de type TRIP : 9.

Une communauté est un groupe de destinations qui partagent certaines propriétés communes.

L'attribut Communautés est utilisé pour grouper des destinations de telle sorte que la décision d'acheminement puisse se fonder sur l'identité du groupe. L'utilisation de l'attribut Communautés devrait significativement simplifier la distribution des informations d'acheminement en fournissant une unité d'agrégation définie administrativement.

Chaque administrateur d'ITAD peut définir les communautés auxquelles appartient un chemin particulier. Par défaut, tous les chemins appartiennent à la communauté générale de la téléphonie Internet.

Par exemple, l'attribut Communautés pourrait être utilisé pour définir une alliance entre un groupe de fournisseurs de service de téléphonie Internet pour un sous-ensemble spécifique d'informations d'acheminement. Dans ce cas, les membres de cette alliance n'accepteraient que les chemins pour des destinations dans ce groupe et qui sont annoncés par les autres membres de l'alliance. Les autres destinations pourraient être acceptées plus librement. Pour réaliser cela, un membre étiquetterait chaque chemin avec une valeur désignée de l'attribut Communauté avant de le disséminer. Cela libérerait les membres d'une telle alliance de la responsabilité de garder trace de l'identité de tous les autres membres de cette alliance.

Un exemple utilise l'attribut Communautés avec l'agrégation. Il est souvent utile d'annoncer à la fois le chemin agrégé et les composants plus spécifiques du chemin qui ont été utilisés pour former les agrégats. Ces informations sur les composants ne sont utiles que pour les homologues TRIP du voisinage, et peut-être l'ITAD de l'homologue TRIP voisin, de sorte qu'il est souhaitable de filtrer les chemins composants. Cela peut être réalisé en spécifiant une valeur d'attribut Communauté que les homologues du voisinage vont vérifier et sur laquelle ils vont effectuer le filtrage. De cette façon, on peut être sûr que les chemins les plus spécifiques ne seront pas propagés au delà de leur portée désirée.

5.9.1 Syntaxe de Communautés

L'attribut Communautés est de longueur variable. Il consiste en un ensemble de valeurs de 8 octets, dont chacune spécifie une communauté. Les quatre premier octets de la valeur de Communauté sont le numéro d'ITAD de la Communauté et les quatre octets suivants sont l'identifiant de communauté.

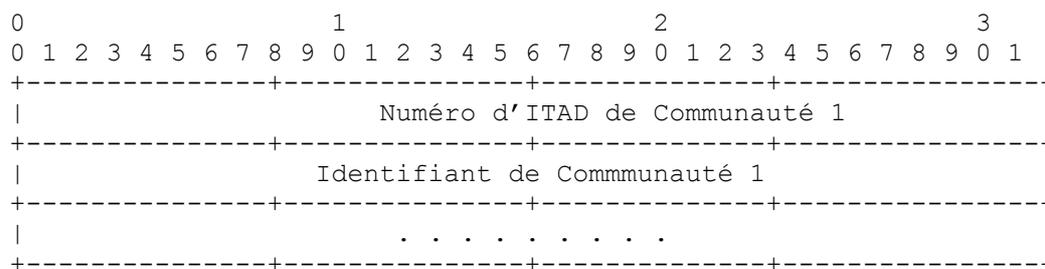


Figure 14 : Syntaxe de Communautés

Pour une allocation administrative, les hypothèses suivantes peuvent être faites :

Les valeurs d'attribut Communauté commençant par un numéro ITAD de Communauté de 0x00000000 sont réservées.

Les communautés suivantes ont une signification mondiale et leur fonctionnement DOIT être mis en œuvre dans tout LS TRIP qui a la capacité de l'attribut Communauté.

- NO_EXPORT (Numéro ITAD Communauté = 0x00000000 et ID de communauté = 0xFFFFF01). Tout chemin reçu avec un attribut Communauté contenant cette valeur DOIT ne pas être annoncé en dehors de l'ITAD TRIP receveur.

Les autres valeurs de communauté DOIVENT être codées en utilisant un numéro d'ITAD dans les quatre octets de poids fort. La sémantique des quatre octets finaux (les octets de l'identifiant de communauté) peuvent être définis par l'ITAD (par exemple, l'ITAD 690 peut définir des identifiants de la communauté de la recherche, de l'éducation, et du commerce qui peuvent être utilisés pour la politique d'acheminement définie par les opérateurs de cet ITAD).

5.9.2 Origine de chemin et Communautés

L'attribut Communautés n'est pas bien connu. Si un chemin a un attribut Communautés qui lui est associé, le LS DOIT inclure cet attribut dans les annonces qu'il génère.

5.9.3 Choix de chemin et Communautés

L'attribut Communautés peut être utilisé pour le choix du chemin. Un chemin qui est membre d'une certaine communauté peut être préféré à un autre chemin qui n'est pas membre de cette communauté. De même, les chemins sans une certaine valeur de communauté peuvent être exclus.

5.9.4 Agrégation et Communautés

Si un ensemble de chemins doit être agrégé et si l'agrégat résultant ne porte pas un attribut AgrégatAtomique, l'agrégat résultant devrait alors avoir un attribut Communautés qui contient l'union des attributs Communauté des chemins agrégés.

5.9.5 Dissémination de chemin et Communautés

Un LS peut manipuler les attributs Communautés avant de disséminer un chemin à un homologue. La manipulation de l'attribut Communautés peut inclure d'ajouter des communautés, de retirer des communautés, d'ajouter (si il n'en existe pas) un attribut Communautés, de supprimer l'attribut Communautés, etc.

5.10 Topologie d'ITAD

Obligatoire conditionnel : Faux.

Fanions exigés : Bien connu, Encapsulation d'état de liaison.

Fanions potentiels : Aucun.

Code de type TRIP : 10.

Au sein d'un ITAD, chaque LS doit savoir l'état des autres LS afin que la défaillance d'un LS puisse être détectée. Pour ce faire, chaque LS annonce sa topologie interne aux autres LS au sein du domaine. Lorsque un LS détecte qu'un autre LS n'est plus actif, les informations générées par ce LS peuvent être supprimées (le Adj-TRIB-In pour cet homologue peut être supprimé). L'attribut Topologie d'ITAD est utilisé pour communiquer cette information aux autres LS au sein du domaine.

Un LS DOIT envoyer une mise à jour de topologie chaque fois qu'il détecte un changement du réglage de son homologue interne. La mise à jour de la topologie peut être envoyée dans un message METTREÀJOUR par lui-même ou elle peut être portée sur un message METTREÀJOUR qui inclut des informations de CheminsAccessibles et/ou CheminsRetirés.

Lorsque un LS reçoit une mise à jour de topologie d'un LS interne, il DOIT recalculer quels LS sont actifs au sein de l'ITAD via un algorithme de connexité sur la topologie.

5.10.1 Syntaxe de topologie d'ITAD

L'attribut Topologie d'ITAD indique les LS avec lesquels le LS est actuellement en relation d'homologue. L'attribut consiste en une liste des identifiants TRIP avec lesquels le LS est actuellement en relations d'homologue ; son format est donné à la Figure 15. Cet attribut DOIT utiliser l'encapsulation d'état de liaison, comme défini au paragraphe 4.3.2.4.

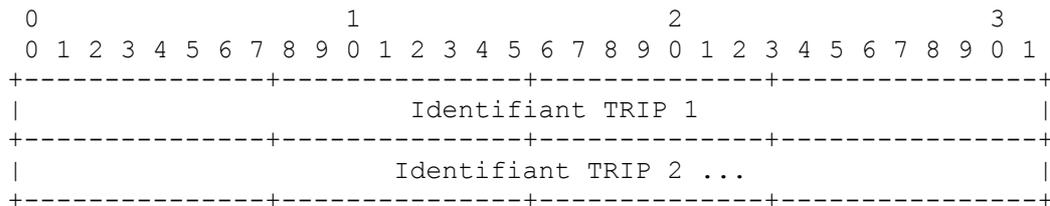


Figure 15 : Syntaxe de topologie d'ITAD

5.10.2 Origine de chemin et topologie d'ITAD

L'attribut Topologie d'ITAD est indépendant de tout chemin dans le METTREÀJOUR. Chaque fois que l'ensemble des homologues internes d'un LS change, il DOIT créer un METTREÀJOUR avec l'attribut Topologie d'ITAD inclus qui fait la liste de l'ensemble actuel des homologues internes. Le LS DOIT inclure cet attribut dans le premier METTREÀJOUR qu'il envoie à un homologue après l'établissement de la session d'échange de communication.

5.10.3 Choix de chemin et topologie d'ITAD

Cet attribut est indépendant de toute information d'acheminement dans le METTREÀJOUR. Quand un LS reçoit un METTREÀJOUR avec un attribut Topologie d'ITAD, il DOIT calculer l'ensemble des LS actuellement actifs dans le domaine en effectuant un essai de connexité sur la topologie de l'ITAD tel que donnée par l'ensemble des attributs Topologie d'ITAD générés. Le LS DOIT purger localement le Adj-TRIB-In de tout LS qui n'est plus actif dans le domaine. Le LS NE DOIT PAS propager ces informations de purge aux autres LS car ils vont prendre une décision similaire.

5.10.4 Agrégation et topologie d'ITAD

Ces informations ne sont pas agrégées.

5.10.5 Dissémination de chemin et topologie d'ITAD

Un LS DOIT ignorer l'attribut si il est reçu d'un homologue dans un autre domaine. Un LS NE DOIT PAS envoyer cet attribut à un homologue inter domaine.

5.11 CheminConverti

Obligatoire conditionnel : Faux.

Fanions exigés : Bien connu.

Fanions potentiels : Aucun.

Code de type TRIP : 12.

L'attribut CheminConverti indique qu'un LS intermédiaire a altéré le chemin en changeant le protocole d'application du chemin. Par exemple, si un LS reçoit un chemin avec le protocole d'application X et change le protocole d'application en Y avant d'annoncer le chemin à un homologue externe, le LS DOIT inclure l'attribut CheminConverti.

L'attribut est une indication que le protocole d'application annoncé ne sera pas utilisé de bout en bout, c'est-à-dire, que les informations annoncées au sujet de ce chemin ne sont pas complètes.

5.11.1 Syntaxe de CheminConverti

Cet attribut à une longueur de zéro (0) ; le champ Valeur est vide.

5.11.2 Origine de chemin et CheminConverti

Les chemins ne sont jamais générés avec l'attribut CheminConverti.

5.11.3 Choix de chemin et CheminConverti

L'attribut CheminConverti peut être utilisé pour le choix de chemin - il indique que les informations d'acheminement annoncées ne sont pas complètes.

5.11.4 Agrégation et CheminConverti

Si un des chemins à agréger a l'attribut CheminConverti, l'agrégat résultant DOIT aussi l'avoir.

5.11.5 Dissémination de chemin et CheminConverti

Si un LS change le protocole d'application d'un chemin avant d'annoncer le chemin à un homologue externe, le LS DOIT inclure l'attribut CheminConverti.

5.12 Considérations pour définir de nouveaux attributs TRIP

Toute proposition de définition de nouveaux attributs TRIP devrait spécifier ce qui suit :

- l'usage de cet attribut,
- les fanions de l'attribut,
- la syntaxe de l'attribut,
- comment fonctionne l'attribut avec la génération de chemins,
- comment fonctionne l'attribut avec l'agrégation de chemin, et
- comment fonctionne l'attribut avec la dissémination de chemin et la portée de l'attribut (par exemple, intra domaine seul comme PréférenceLocale)

L'IANA gèrera l'allocation des codes de type TRIP aux nouveaux attributs.

6. Détection et traitement d'erreur TRIP

La présente section décrit les erreurs à détecter et les actions à entreprendre lors du traitement des messages TRIP.

Lorsque l'une des conditions décrites ici est détectée, un message NOTIFICATION avec les champs indiqués de Code d'erreur, Sous-code d'erreur, et Données DOIT être envoyé, et la connexion TRIP DOIT être close. Si aucun sous-code d'erreur n'est spécifié, un sous-code de zéro DOIT être utilisé.

La phrase "la connexion TRIP est close" signifie que la connexion de protocole de transport a été close et que toutes les ressources pour cette connexion TRIP ont été désallouées. Si la connexion était inter domaines, les entrées de tableau d'acheminement associées à l'homologue distant DOIVENT être marquées comme étant invalides. Les entrées de tableau d'acheminement NE DOIVENT PAS être marquées comme invalides si une session interne d'échange de trafic se termine. Le fait que le chemins ait été marqué comme invalide est passé aux autres homologues TRIP avant que les chemins ne soient supprimés du système.

Sauf spécification explicite, le champ Données du message NOTIFICATION envoyé pour indiquer une erreur DOIT être vide.

6.1 Détection et traitement d'erreur d'en-tête de message

Toutes les erreurs détectées durant le traitement de l'en-tête de message sont indiquées par l'envoi du message NOTIFICATION avec le code d'erreur Erreur d'en-tête de message. Le sous-code d'erreur développe la nature spécifique de l'erreur. Les vérifications d'erreur dans la présente section DOIVENT être effectuées par chaque LS à réception de chaque message.

Si le champ Longueur de l'en-tête de message est de moins de 3 ou de plus de 4096, ou si le champ Longueur d'un message OUVERT fait moins que la longueur minimum de message OUVERT, ou si le champ Longueur d'un message METTREÀJOUR fait moins que la longueur minimum de message METTREÀJOUR, ou si le champ Longueur d'un message GARDERENVIE n'est pas égal à 3, ou si le champ Longueur d'un message NOTIFICATION fait moins que la longueur minimale du message NOTIFICATION, le sous-code d'erreur DOIT être réglé à Mauvaise longueur de message. Le champ Données contient le champ Longueur erroné.

Si le champ Type de l'en-tête de message n'est pas reconnu, le sous-code d'erreur DOIT alors être réglé à "Mauvais type de message". Le champ Données contient le champ Type erroné.

6.2 Détection et traitement d'erreur du message OUVERT

Toutes les erreurs détectées lors du traitement du message OUVERT sont indiquées par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur de message OUVERT". Le sous-code d'erreur développe la nature spécifique de l'erreur. Les vérifications d'erreur de cette section DOIVENT être effectuées par chaque LS à réception de chaque message OUVERT.

Si le numéro de version contenu dans le champ Version du message OUVERT reçu n'est pas accepté, le sous-code d'erreur DOIT alors être réglé à "Numéro de version non accepté". Le champ Données est un entier non signé d'un octet, qui indique le plus grand numéro de version accepté localement, qui est inférieur à la version de la proposition de l'homologue TRIP distant (comme indiqué dans le message OUVERT reçu).

Si le champ ITAD du message OUVERT est inacceptable, le sous-code d'erreur DOIT alors être réglé à "Mauvais ITAD d'homologue". La détermination des numéros d'ITAD acceptables sort du domaine d'application de ce protocole.

Si le champ Temps de garde du message OUVERT est inacceptable, le sous-code d'erreur DOIT alors être réglé à "Temps de garde inacceptable". Une mise en œuvre DOIT rejeter les valeurs de temps de garde d'une ou deux secondes. Une mise en œuvre PEUT rejeter tout temps de garde proposé. Une mise en œuvre qui accepte un temps de garde DOIT utiliser la valeur négociée pour le temps de garde.

Si le champ Identifiant TRIP du message OUVERT n'est pas valide, le sous-code d'erreur DOIT alors être réglé à "Mauvais identifiant TRIP". Un identifiant TRIP est long de quatre octets et peut prendre n'importe quelle valeur. Un LS considère que l'identifiant TRIP est invalide si il a déjà une connexion ouverte avec un autre LS homologue qui a le même ITAD et le même identifiant TRIP.

Deux LS au sein du même ITAD NE DOIVENT PAS avoir de valeurs égales d'identifiant TRIP. Cette restriction ne s'applique pas aux LS dans des ITAD différents car le but est d'identifier de façon univoque un LS en utilisant son identifiant TRIP et son numéro d'ITAD.

Si un des paramètres facultatifs dans le message OUVERT n'est pas reconnu, le sous-code d'erreur DOIT alors être "Paramètres facultatifs non acceptés".

Si les paramètres facultatifs du message OUVERT incluent des informations de capacités avec une capacité non acceptée (que ce soit dans le type de capacité ou la valeur) le sous-code d'erreur DOIT alors être réglé à "Capacité non acceptée" et toutes les capacités non acceptées DOIVENT être énumérées dans le champ Données du message NOTIFICATION.

Si les paramètres facultatifs du message OUVERT incluent des informations de capacités qui ne correspondent pas aux capacités du LS receveur, le sous-code d'erreur DOIT alors être réglé à "Discordance de capacités" et toutes les capacités discordantes DOIVENT être énumérées dans le champ Données du message NOTIFICATION.

6.3 Détection et traitement d'erreur du message METTREÀJOUR

Toutes les erreurs détectées pendant le traitement du message METTREÀJOUR sont indiquées par l'envoi du message NOTIFICATION avec le code d'erreur "Erreur de message METTREÀJOUR". Le sous-code d'erreur précise la nature spécifique de l'erreur. Les vérifications d'erreur de cette section DOIVENT être effectuées par chaque LS à réception de chaque message METTREÀJOUR. Ces vérifications d'erreur DOIVENT survenir avant que les procédures d'arrosage ne soient invoquées avec les homologues internes.

Si un attribut reconnu a des fanions d'attribut qui sont en conflit avec le code de type d'attribut, le sous-code d'erreur DOIT alors être réglé à "Erreur des fanions d'attribut". Le champ Données contient l'attribut erroné (type, longueur et valeur).

Si un attribut reconnu a une longueur d'attribut qui est en conflit avec la longueur attendue (sur la base du code de type de l'attribut) le sous-code d'erreur DOIT alors être réglé à "Erreur de longueur d'attribut". Le champ Données contient l'attribut erroné (type, longueur et valeur).

Si un attribut bien connu obligatoire (c'est-à-dire, un attribut conditionnel obligatoire et les conditions remplies pour l'inclure dans le message METTREÀJOUR) n'est pas présent, le sous-code d'erreur DOIT alors être réglé à "Attribut bien connu obligatoire manquant". Le champ Données contient le code de type d'attribut des attributs bien connus obligatoires conditionnels manquants.

Si un des attributs bien connus n'est pas reconnu, le sous-code d'erreur DOIT alors être réglé à "Attribut bien connu non reconnu". Le champ Données contient l'attribut non reconnu (type, longueur et valeur).

Si un attribut a une valeur syntaxiquement incorrecte, ou une valeur indéfinie, le sous-code d'erreur est alors réglé à "Attribut invalide". Le champ Données contient l'attribut incorrect (type, longueur et valeur). Un tel message NOTIFICATION est envoyé, par exemple, lorsque un attribut ServeurDeProchainBond est reçu avec une adresse invalide.

Les informations portées par l'attribut CheminD'Annonce sont vérifiées à la recherche de boucles d'ITAD. La détection de boucle d'ITAD est faite en examinant le chemin d'annonce complet, et en vérifiant que le numéro d'ITAD de l'ITAD local n'apparaît pas dans le chemin d'annonce. Si le numéro d'ITAD local apparaît dans le CheminD'Annonce, le chemin PEUT alors être mémorisé dans le Adj-TRIB-In. Cependant, sauf si le LS est configuré pour accepter les chemins avec son propre ITAD dans le chemin d'annonces, le chemin NE DOIT PAS être passé au processus de décision TRIP. Le fonctionnement d'un LS qui est configuré à accepter des chemins avec son propre numéro d'ITAD dans le chemin d'annonces sort du domaine d'application du présent document.

Si le message METTREÀJOUR a été reçu d'un homologue interne et si l'attribut CheminsRetirés, CheminsAccessibles, ou Topologie d'ITAD n'a pas le fanion Encapsulation d'état de liaison établi, le sous-code d'erreur est alors réglé à "Attribut invalide" et le champ Données contient l'attribut. De même, l'attribut est invalide si il est reçu d'un homologue externe et si le fanion État de liaison est établi.

Si un attribut apparaît plus d'une fois dans le message METTREÀJOUR, le sous-code d'erreur est alors réglé à "Liste d'attributs mal formée."

6.4 Détection et traitement d'erreur du message NOTIFICATION

Si un homologue envoie un message NOTIFICATION, et si il y a une erreur dans ce message, il n'y a malheureusement aucun moyen de rapporter cette erreur via un autre message NOTIFICATION. Une telle erreur, comme un code d'erreur ou sous-code d'erreur non reconnu, devrait être notée, enregistrée en local, et portée à l'attention de l'administration de l'homologue. Les moyens de le faire sont cependant en dehors du domaine d'application du présent document.

6.5 Traitement de l'erreur Expiration du temporisateur de garde

Si un système ne reçoit pas de messages suivants durant la période spécifiée par le temps de garde négocié, un message NOTIFICATION avec le code d'erreur "Temps de garde expiré" DOIT être envoyé et la connexion TRIP DOIT être close.

6.6 Traitement de l'erreur de l'automate à états finis

Une erreur détectée par l'automate à états finis de TRIP (par exemple, la réception d'un événement inattendu) DOIT avoir pour résultat l'envoi d'un message NOTIFICATION avec le code d'erreur "Erreur d'automate à états finis" et la connexion TRIP DOIT être close.

6.7 Cesser

En l'absence d'erreur fatale (qui sont indiquées dans cette section) un homologue TRIP PEUT choisir à tout moment de fermer sa connexion TRIP en envoyant le message NOTIFICATION avec le code d'erreur "Cesser". Cependant, le message NOTIFICATION Cesser NE DOIT PAS être utilisé lorsque existe une erreur fatale indiquée par cette section.

6.8 Détection de collision de connexion

Si une paire de LS essaye simultanément d'établir une connexion de transport entre eux, deux connexions parallèles entre cette paire de locuteurs peut très bien être formée. On appelle cette situation une collision de connexions. En clair, une de

ces connexions doit être fermée.

Sur la base de la valeur de l'identifiant TRIP, une convention est établie pour détecter quelle connexion TRIP est à préserver lorsque survient une collision. La convention est de comparer les identifiants TRIP des homologues impliqués dans la collision et de ne retenir que la connexion initiée par le LS qui a la plus forte valeur d'identifiant TRIP.

À réception d'un message OUVERT, le LS local DOIT examiner toutes ses connexions qui sont dans l'état OuvertConfirmé. Un LS PEUT aussi examiner les connexions qui sont dans l'état OuvertEnvoyé si il sait l'identifiant TRIP de l'homologue par des moyens qui sont en dehors du protocole. Si parmi ces connexions il y en a une avec un LS distant, dont l'identifiant TRIP est égal à celle qui est dans le message OUVERT, le LS local DOIT alors effectuer la procédure de résolution de collision suivante :

L'identifiant TRIP et l'ITAD du LS local sont comparés à l'identifiant TRIP et à l'ITAD du LS distant (comme spécifié dans le message OUVERT). Les identifiants TRIP sont traités comme des entiers non signés de quatre octets pour la comparaison.

Si la valeur de l'identifiant TRIP local est inférieure à celle du LS distant, ou si les deux identifiants TRIP sont égaux, et si la valeur de l'ITAD du LS local est inférieure à la valeur de l'ITAD du LS distant, alors le LS local DOIT clore la connexion TRIP qui existe déjà (celle qui est déjà dans l'état OuvertConfirmé) et accepter la connexion TRIP initiée par le LS distant :

1. Autrement, le LS local clôt la connexion TRIP nouvellement créée et continue d'utiliser celle existante (celle qui est déjà dans l'état OuvertConfirmé).
2. Si une collision de connexion survient avec une connexion TRIP existante qui est dans l'état Établi, alors le LS DOIT clore sans condition la connexion nouvellement créée. Noter qu'une collision de connexions ne peut pas être détectée avec les connexions dans les états Repos, Connecté, ou Actif.
3. Pour clore la connexion TRIP (qui résulte de la procédure de résolution de collision) un LS DOIT envoyer un message NOTIFICATION avec le code d'erreur "Cesser" et la connexion TRIP DOIT être close.

7. Négociation de version TRIP

Des LS homologues peuvent négocier la version du protocole en faisant plusieurs tentatives d'ouverture d'une connexion TRIP, en commençant par le plus fort numéro de version que chacun prend en charge. Si une tentative d'ouverture échoue avec un code d'erreur "Erreur de message OUVERT" et un sous-code d'erreur "Numéro de version non accepté", le LS a alors à sa disposition le numéro de version qu'il a essayé, le numéro de version que son homologue a essayé, le numéro de version passé par son homologue dans le message NOTIFICATION, et les numéros de version qu'il prend en charge. Si les deux homologues prennent en charge une ou plusieurs versions communes, cela va alors leur permettre de rapidement déterminer la plus forte version commune. Afin de prendre en charge la négociation de la version de TRIP, de futures versions de TRIP devront retenir le format des messages OUVERT et NOTIFICATION.

8. Négociation de capacité TRIP

Un LS PEUT inclure l'option Capacités dans son message OUVERT à un homologue pour indiquer les capacités acceptées par le LS. Un LS qui reçoit un message OUVERT NE DOIT PAS utiliser des capacités qui n'étaient pas incluses dans le message OUVERT de l'homologue lorsque il communique avec cet homologue.

9. Automate à états finis de TRIP

La présente section spécifie le fonctionnement de TRIP en termes d'automate à états finis (FSM, *Finite State Machine*). Ensuite figure un bref résumé et survol des opérations TRIP par état comme déterminé par ce FSM. Une version condensée du FSM TRIP se trouve à l'Appendice 1. Il y a un FSM TRIP par homologue et ces FSM fonctionnent de façon indépendante.

État Repos :

TRIP est au départ dans l'état Repos (*Idle*) chez chaque homologue. Dans cet état, TRIP refuse toutes les connexions entrantes. Aucune ressource n'est allouée à l'homologue. En réponse à l'événement Début (*Start*) (initié par l'un ou l'autre des systèmes ou par l'opérateur) le système local initialise toutes les ressources TRIP, lance le temporisateur RéessaiDeConnexion, initie une connexion de transport avec l'homologue, commence à écouter si une connexion aurait pu être initiée par l'homologue TRIP distant, et change son état en celui de Connecté. La valeur exacte du temporisateur

RéessaiDeConnexion est une affaire locale, mais devrait être suffisamment longue pour permettre l'initialisation de TCP.

Si un LS détecte une erreur, il clôt la connexion de transport et change son état en Repos. La transition à partir de l'état Repos exige de générer l'événement Début. Si un tel événement est généré automatiquement, des erreurs TRIP persistantes peuvent résulter en un flottement persistant du LS. Pour éviter une telle condition, les événements Début NE DOIVENT PAS être générés immédiatement pour un homologue qui est passé précédemment de l'état Repos du fait d'une erreur. Pour un homologue qui est passé précédemment à l'état Repos du fait d'une erreur, l'intervalle de temps entre des événements Début consécutifs, si de tels événements sont générés automatiquement, DOIT augmenter de façon exponentielle. La valeur du temporisateur initial DEVRAIT être de 60 secondes, et l'intervalle DEVRAIT au moins être doublé pour chaque essai consécutif jusqu'à une certaine valeur maximum.

Tout autre événement reçu dans l'état Repos est ignoré.

État Connecter :

Dans cet état, un LS attend qu'une connexion de protocole de transport soit réalisée avec l'homologue, et il écoute les connexions de transport entrantes provenant de l'homologue.

Si la connexion de protocole de transport réussit, le LS local arrête le temporisateur RéessaiDeConnexion, achève l'initialisation, envoie un message OUVERT à son homologue, règle son temporisateur de garde à une grande valeur, et change son état en OuvertEnvoyé. Une valeur de temporisateur de garde de 4 minutes est suggérée.

Si la connexion de protocole de transport échoue (par exemple, à cause de l'expiration de la temporisation de retransmission) le système local redémarre le temporisateur RéessaiDeConnexion, continue d'écouter une connexion qui pourrait être initiée par le LS distant, et change son état en l'état Actif.

En réponse à l'événement expiration du temporisateur RéessaiDeConnexion, le LS local annule toute connexion de transport en instance avec l'homologue, relance le temporisateur RéessaiDeConnexion, initie une connexion de transport avec le LS distant, continue d'écouter si une connexion est initiée par le LS distant, et reste dans l'état Connexion.

Si le LS local détecte qu'un homologue distant essaie d'établir une connexion avec lui et si l'adresse IP de l'homologue n'est pas une adresse attendue, le LS local rejette alors la tentative de connexion et continue d'écouter si il y a une connexion de la part des homologues qu'il attend sans changer d'état.

Si une connexion de protocole de transport entrante réussit, le LS local arrête le temporisateur RéessaiDeConnexion, achève l'initialisation, envoie un message OUVERT à son homologue, règle son temporisateur de garde à une grande valeur, et change son état en OuvertEnvoyé. Une valeur de temporisateur de garde de 4 minutes est suggérée.

L'événement Début est ignoré dans l'état Connexion.

En réponse à tout autre événement (initié par le système ou par l'opérateur) le système local libère toutes les ressources TRIP associées à cette connexion et change son état en Repos.

État Actif:

Dans cet état, un LS écoute si il y a une connexion entrante venant de l'homologue, mais il n'est pas dans le processus d'initialisation d'une connexion avec l'homologue.

Si une connexion de protocole de transport entrante réussit, le LS local arrête le temporisateur RéessaiDeConnexion, achève l'initialisation, envoie un message OUVERT à son homologue, règle son temporisateur de garde à une grande valeur, et change son état en OuvertEnvoyé. Une valeur de temporisateur de garde de 4 minutes est suggérée.

En réponse à l'événement expiration du temporisateur RéessaiDeConnexion, le système local relance le temporisateur RéessaiDeConnexion, initie une connexion de transport avec l'homologue TRIP, continue d'écouter s'il existe une connexion qui pourrait être initiée par l'homologue TRIP distant, et change son état en Connexion.

Si le LS local détecte qu'un homologue distant essaie d'établir une connexion avec lui et si l'adresse IP de l'homologue n'est pas une de celles attendues, le LS local rejette alors la tentative de connexion et continue d'écouter pour une connexion provenant d'homologues espérés sans changer d'état.

L'événement Début est ignoré dans l'état Actif.

En réponse à tout autre événement (initiés par le système ou par l'opérateur) le système local libère toutes les ressources TRIP associées à cette connexion et change son état en Repos.

État OuvertEnvoyé :

Dans cet état, un LS a envoyé un message OUVERT à son homologue et attend un message OUVERT de son homologue. Lorsque un message OUVERT est reçu, la correction de tous les champs est vérifiée. Si la vérification de l'en-tête du message TRIP ou la vérification du message OUVERT détecte une erreur (voir au paragraphe 6.2) ou une collision de connexions (voir au paragraphe 6.8) le système local envoie un message NOTIFICATION et change son état en Repos.

Si il n'y a pas d'erreur dans le message OUVERT, TRIP envoie un message GARDERENVIE et lance un temporisateur GarderEnVie. Le temporisateur de garde, qui était à l'origine réglé à une grande valeur (voir ci-dessus) est remplacé par la valeur négociée de Temps de garde (voir le paragraphe 4.2). Si la valeur négociée de temps de garde est zéro, le temporisateur de temps de garde et le temporisateur GarderEnVie ne sont pas lancés. Si la valeur du champ ITAD est la même que celle du numéro d'ITAD local, la connexion est alors une connexion "interne" ; autrement, elle est "externe" (cela va affecter le traitement de METTREÀJOUR). Finalement, l'état est changé en OuvertConfirmé.

Si le LS local détecte qu'un homologue distant essaye d'établir une connexion avec lui et si l'adresse IP de l'homologue n'est pas une de celles attendues, le LS local rejette alors la tentative de connexion et continue d'écouter s'il existe une connexion provenant des homologues espérés sans changer d'état.

Si une notification de déconnexion est reçue du protocole de transport sous-jacent, le LS local clôt la connexion de transport, relance le temporisateur RéessaiDeConnexion, continue d'écouter pour une connexion qui pourrait être initiée par l'homologue TRIP distant, et passe à l'état Actif.

Si le temporisateur de garde arrive à expiration, le LS local envoie un message NOTIFICATION avec le code d'erreur "temporisateur de garde expiré" et change son état en Repos.

En réponse à l'événement Stop (initié par le système ou par l'opérateur) le LS local envoie un message NOTIFICATION avec le code d'erreur "Cesser" et change son état en Repos.

L'événement Début est ignoré dans l'état OuvertEnvoyé.

En réponse à tout autre événement, le LS local envoie un message NOTIFICATION avec le code d'erreur "Erreur d'automate à états finis" et change son état en Repos.

Chaque fois que TRIP change son état de OuvertEnvoyé à Repos, il ferme la connexion de transport et libère toutes les ressources associées à cette connexion.

État OuvertConfirmé :

Dans cet état, un LS a envoyé un OUVERT à son homologue, reçu un OUVERT de son homologue, et envoyé un GARDERENVIE en réponse au OUVERT. Le LS est maintenant en attente d'un message GARDERENVIE ou NOTIFICATION en réponse à son OUVERT.

Si le LS local reçoit un message GARDERENVIE, il change son état en Établi.

Si le temporisateur de garde expire avant que soit reçu un message GARDERENVIE, le LS local envoie un message NOTIFICATION avec le code d'erreur "Temporisateur de garde expiré" et change son état en Repos.

Si le LS local reçoit un message NOTIFICATION, il change son état en Repos.

Si le temporisateur GarderEnVie expire, le LS local envoie un message GARDERENVIE et il relance son temporisateur GarderEnVie.

Si une notification de déconnexion est reçue du protocole de transport sous-jacent, le LS local clôt la connexion de transport, relance le temporisateur RéessaiDeConnexion, continue d'écouter si une connexion aurait pu être initiée par l'homologue TRIP distant, et passe à l'état Actif.

En réponse à l'événement Stop (initié par le système ou par l'opérateur) le LS local envoie un message NOTIFICATION avec le code d'erreur "Cesser" et change son état en Repos.

L'événement Début est ignoré dans l'état OuvertConfirmé.

En réponse à tout autre événement, le LS local envoie un message NOTIFICATION avec le code d'erreur "Erreur de l'automate à états finis" et change son état en Repos.

Chaque fois que TRIP change son état de OuvertConfirmé à Repos, il clôt la connexion de transport et libère toutes les ressources associées à cette connexion.

État Établi :

Dans l'état Établi, un LS peut échanger des messages `METTREÀJOUR`, `NOTIFICATION`, et `GARDERENVIE` avec son homologue.

Si le temporisateur de garde négocié est de zéro, aucune procédure n'est alors nécessaire pour garder en vie une session d'échange de trafic. Si la valeur de temps de garde négocié est différente de zéro, les procédures de ce paragraphe s'appliquent. Si le temporisateur de garde expire, le LS local envoie un message `NOTIFICATION` avec le code d'erreur "Temporisateur de garde expiré" et change son état en Repos. Si le temporisateur `GarderEnVie` expire, le LS local envoie alors un message `GarderEnVie` et relance le temporisateur `GarderEnVie`. Si le LS local reçoit un message `METTREÀJOUR` ou `GARDERENVIE`, il relance alors son temporisateur de garde. Chaque fois que le LS envoie un message `METTREÀJOUR` ou `GARDERENVIE`, il redémarre son temporisateur `GarderEnVie`.

Si le LS local reçoit un message `NOTIFICATION`, il change son état en Repos.

Si le LS local reçoit un message `METTREÀJOUR` et si la procédure de traitement d'erreur du message `METTREÀJOUR` (voir au paragraphe 6.3) détecte une erreur, le LS local envoie un message `NOTIFICATION` et change son état en Repos.

Si une notification de déconnexion est reçue du protocole de transport sous-jacent, le LS local change son état en Repos.

En réponse à l'événement `Stop` (initié par le système ou par l'opérateur) le LS local envoie un message `NOTIFICATION` avec le code d'erreur "Cesser" et change son état en Repos.

L'événement `Début` est ignoré dans l'état Établi.

En réponse à tout autre événement, le LS local envoie un message `NOTIFICATION` avec un code d'erreur "Erreur de l'automate à états finis" et change son état en Repos.

Chaque fois que TRIP change son état de Établi à Repos, il clôt la connexion de transport et libère toutes les ressources associées à cette connexion. De plus, si l'homologue est externe, le LS supprime tous les chemins déduits de cette connexion.

10. Traitement du message `METTREÀJOUR`

Un message `METTREÀJOUR` ne peut être reçu que dans l'état Établi. Lorsque un message `METTREÀJOUR` est reçu, la validité de chaque champ est vérifiée comme spécifié au paragraphe 6.3. La suite de cette section suppose que le message `METTREÀJOUR` a passé avec succès les procédures de vérification d'erreurs du paragraphe 6.3.

Si le message `METTREÀJOUR` a été reçu d'un homologue interne, les procédures d'arrosage du paragraphe 10.1 DOIVENT être appliqués. Le processus d'arrosage synchronise les `Loc-TRIB` de tous les LS dans le domaine. Certains chemins au sein de `METTREÀJOUR` peuvent être marqués comme périmés ou dupliqués par le processus d'arrosage et sont ignorés durant le reste du traitement `METTREÀJOUR`.

Si le message `METTREÀJOUR` contient des chemins retirés, le chemin correspondant précédemment annoncé devra être retiré du `Adj-TRIB-In`. Ce LS DOIT relancer son processus de décision car le chemin précédemment annoncé n'est plus disponible à l'utilisation.

Si le message `METTREÀJOUR` contient un chemin, alors le chemin DOIT être placé dans l'`Adj-TRIB-In` approprié, et les actions supplémentaires suivantes DOIVENT être entreprises :

1. Si ses destinations sont identiques à celles d'un chemin actuellement mémorisé dans l'`Adj-TRIB-In`, le nouveau chemin DOIT alors remplacer l'ancien chemin dans l'`Adj-TRIB-In`, retirant donc implicitement le chemin plus ancien du service. Le LS DOIT relancer son processus de décision car l'ancien chemin n'est plus disponible à l'utilisation.
2. Si le nouveau chemin est plus spécifique qu'un chemin antérieur contenu dans le `Adj-TRIB-In` et a des attributs identiques, aucune autre action n'est alors nécessaire.
3. Si le nouveau chemin est plus spécifique qu'un chemin antérieur contenu dans le `Adj-TRIB-In` mais n'a pas des attributs identiques, le LS DOIT alors refaire son processus de décision car le chemin plus spécifique a rendu implicitement indisponible à l'utilisation une portion du chemin moins spécifique.
4. Si le nouveau chemin a des destinations qui ne sont pas présentes dans un des chemins actuellement mémorisés dans le `Adj-TRIB-In`, le LS DOIT alors refaire son processus de décision.

5. Si le nouveau chemin est moins spécifique qu'un chemin antérieur contenu dans le Adj-TRIB-In, le LS DOIT lancer son processus de décision sur l'ensemble des destinations qui ne sont décrites que par le chemin moins spécifique.

10.1 Processus d'arrosage

Lorsque un LS reçoit un message METTREÀJOUR d'un homologue interne, le LS arrose des nouvelles informations provenant de ce message tous ses autres homologues internes. L'arrosage est utilisé pour synchroniser efficacement tous les LS au sein d'un domaine sans faire peser de contraintes sur la topologie interne du domaine. Le mécanisme d'arrosage se fonde sur les techniques utilisées dans OSPF [RFC2328] et SCSP [RFC2334]. On peut objecter que le processus d'arrosage de TRIP est en réalité un mécanisme de diffusion contrôlée.

10.1.1 Informations de la base de données

Le LS DOIT conserver le numéro de séquence et l'identifiant TRIP d'origine pour chaque attribut encapsulé d'état de liaison dans un Adj-TRIB-In interne. Ces valeurs sont incluses avec le chemin dans les attributs CheminsAccessibles, CheminsRetirés, et Topologie d'ITAD. L'identifiant TRIP d'origine donne la LS interne qui a généré ce chemin dans l'ITAD, le numéro de séquence donne la version de ce chemin au LS d'origine.

10.1.2 Détermination de la nouveauté

Pour chaque chemin dans le champ CheminsAccessibles ou CheminsRetirés, le LS décide si le chemin est nouveau ou ancien. Cela est déterminé en comparant le numéro de séquence du chemin dans le METTREÀJOUR avec le numéro de séquence du chemin sauvegardé dans le Adj-TRIB-In. Le chemin est nouveau si il n'existe pas dans le Adj-TRIB-In pour le LS d'origine, ou si le chemin existe dans le Adj-TRIB-In mais si le numéro de séquence dans le METTREÀJOUR est supérieur au numéro de séquence sauvegardé dans le Adj-TRIB-In. Noter que la vérification de la nouveauté est indépendamment appliquée à chaque attribut encapsulé d'état de liaison dans le METTREÀJOUR (CheminsRetirés ou CheminsAccessibles ou Topologie d'ITAD).

10.1.3 Arrosage

Chaque chemin dans le champ CheminsAccessibles ou CheminsRetirés qui est déterminé comme étant ancien est ignoré dans la suite du traitement. Si le chemin est déterminé comme nouveau, les actions suivantes surviennent alors.

Si le chemin est en train d'être retiré, le LS DOIT alors arroser le chemin retiré à tous les autres homologues internes, et DOIT marquer le chemin comme retiré. Un LS DOIT conserver les chemins marqués comme retirés dans ses bases de données pendant MaxDélaiPurge secondes.

Si le chemin est en train d'être mis à jour, le LS DOIT alors mettre à jour le chemin dans le Adj-TRIB-In et DOIT en arroser tous les autres homologues internes.

Si ces procédures résultent en des changements au Adj-TRIB-In, le chemin est alors aussi rendu disponible pour le traitement de chemin local comme décrit précédemment à la Section 10.

Pour mettre en œuvre l'arrosage, on recommande ce qui suit. Tous les chemins reçus dans un seul message METTREÀJOUR qui sont déterminés être nouveaux devraient être transmis à tous les autres homologues internes dans un seul message METTREÀJOUR. D'autres variantes de l'arrosage sont possibles, mais le LS local DOIT s'assurer que chaque nouveau chemin (et tous les attributs associés) reçus d'un homologue interne sont transmis à chaque autre homologue interne.

10.1.4 Considérations sur les numéros de séquence

Le numéro de séquence est utilisé pour déterminer quand une version d'un chemin est plus récente qu'une autre. Un plus grand numéro de séquence indique une version plus récente. Le numéro de séquence est alloué par le LS d'origine du chemin dans l'ITAD local. Le numéro de séquence est un entier non signé de quatre octets dans la gamme de 1 à $2^{31}-1$ (MinSequenceNum à MaxSequenceNum). La valeur 0 est réservée. Lorsque un LS génère un chemin (y compris lorsque le LS redémarre/réamorçe) dans son ITAD, il DOIT le générer avec un numéro de séquence de MinSequenceNum. Chaque fois que le chemin est mis à jour au sein de l'ITAD par son origine, le numéro de séquence DOIT être augmenté.

Si jamais il arrive que le numéro de séquence soit MaxSequenceNum-1 et qu'il ait besoin d'être augmenté, le module TRIP du LS DOIT alors être désactivé pendant une durée de DélaiTripDésactivé afin que tous les chemins générés par ce LS avec des numéros de séquence élevés puissent être retirés.

10.1.5 Purge d'un chemin dans l'ITAD

Pour retirer un chemin qu'il a généré au sein de l'ITAD, un LS inclut le chemin dans le champ CheminsRetirés d'un message METTREÀJOUR. Le numéro de séquence DOIT être supérieur à la dernière version valide du chemin. Le LS PEUT choisir d'utiliser un numéro de séquence de MaxSequenceNum lors du retrait de chemins au sein de son ITAD, mais ceci n'est pas obligé.

Après le retrait d'un chemin, un LS DOIT marquer le chemin comme "retiré" dans sa base de données, et conserver le chemin retiré dans sa base de données pendant MaxDélaiPurge secondes. Si le LS a besoin de générer à nouveau un chemin qui a été purgé mais figure encore dans sa base de données, il peut soit générer à nouveau le chemin immédiatement en utilisant un numéro de séquence supérieur à celui utilisé pour le retrait, soit il peut attendre que MaxDélaiPurge secondes se soient écoulées depuis que le chemin a été retiré.

10.1.6 Réception de chemins auto générés

Il est courant qu'un LS reçoive des METTREÀJOUR pour des chemins qu'il a générés au sein de l'ITAD via la procédure d'arrosage. Si le LS reçoit un METTREÀJOUR pour un chemin qu'il a généré et qui est plus récent (qui a un numéro de séquence supérieur) que la version actuelle du LS, des actions particulières doivent être entreprises. Cela devrait être une occurrence relativement rare et elle indique qu'un chemin existe encore au sein de l'ITAD depuis le dernier redémarrage/réamorçage du LS.

Si un LS reçoit une mise à jour de chemin auto générée qui est plus récente que la version actuelle du chemin chez le LS, les actions suivantes DOIVENT être entreprises. Si le LS souhaite toujours annoncer les informations dans le chemin, le LS DOIT alors augmenter le numéro de séquence du chemin à une valeur supérieure à celle reçue dans le METTREÀJOUR et re-générer le chemin. Si le LS ne souhaite pas continuer à annoncer le chemin, il DOIT alors purger le chemin comme décrit au paragraphe 10.1.5.

10.1.7 Suppression des chemins retirés

Un LS DEVRAIT s'assurer que les chemins marqués retirés sont supprimés de la base de données dans les temps après l'expiration de MaxDélaiPurge. Cela pourrait être fait, par exemple, en balayant périodiquement la base de données, et en supprimant les entrées qui ont été retirées il y a plus de MaxDélaiPurge secondes.

10.2 Processus de décision

Le processus de décision choisit les chemins pour les prochaines annonces en appliquant les politiques de la base de données d'informations de politique (PIB, *Policy Information Base*) locale aux chemins mémorisés dans son Adj-TRIBs-In. Le résultat du processus de décision est l'ensemble des chemins qui seront annoncés à tous les homologues ; les chemins choisis seront mémorisés dans le Adj-TRIBs-Out local du LS.

Le processus de sélection est formalisé en définissant une fonction qui prend les attributs d'un certain chemin comme argument et retourne un entier non négatif qui note le degré de préférence pour le chemin. La fonction qui calcule le degré de préférence pour un certain chemin ne devra utiliser comme entrées aucun des éléments suivants : l'existence d'autres chemins, la non existence d'autres chemins, ou les attributs d'autres chemins. La sélection des chemins consiste alors en une application individuelle de la fonction de degré de préférence à chaque chemin praticable, suivi par le choix de celui qui a le plus fort degré de préférence.

Tous les LS internes dans un ITAD DOIVENT faire fonctionner le processus de décision et appliquer les mêmes critères de décision, autrement, il ne sera pas possible de synchroniser leurs Loc-TRIB.

Le processus de décision opère sur des chemins contenus dans chaque Adj-TRIBs-In, et il est chargé de :

- choisir les chemins à annoncer aux homologues internes
- choisir les chemins à annoncer aux homologues externes
- faire l'agrégation de chemins et la réduction des informations de chemins.

Le processus de décision se déroule en trois phases distinctes, déclanchée chacune par un événement différent :

- La phase 1 est chargée de calculer le degré de préférence de chaque chemin reçu d'un homologue externe.
- La phase 2 est invoquée à l'achèvement de la phase 1. Elle est chargée de choisir le meilleur chemin parmi tous ceux disponibles pour chaque destination distincte, et d'installer chaque chemin choisi dans la Loc-TRIB.
- La phase 3 est invoquée après la modification de la Loc-TRIB. Elle est chargée de disséminer les chemins de la Loc-

TRIB à chaque homologue externe, conformément aux politiques contenues dans la PIB. L'agrégation de chemins et la réduction des informations peut facultativement être effectuée dans cette phase.

10.2.1 Phase 1 : Calcul du degré de préférence

La phase 1 de la fonction de décision devra être invoquée chaque fois que le LS local reçoit d'un homologue un message METTREÀJOUR qui annonce un nouveau chemin, un chemin de remplacement, ou un retrait de chemin.

La phase 1 de la fonction de décision est un processus distinct des autres qui s'achève lorsque sa tâche est terminée.

La phase 1 de la fonction de décision devra verrouiller une Adj-TRIB-In avant d'opérer sur tout chemin contenu en son sein, et devra la déverrouiller après avoir opéré sur tous les chemins nouveaux ou de remplacement contenus dedans.

Le LS local DOIT déterminer un degré de préférence pour chaque chemin nouveau ou de remplacement reçu. Si le chemin est appris d'un homologue interne, la valeur de l'attribut PréférenceLocale DOIT être prise comme degré de préférence. Si le chemin est appris d'un homologue externe, alors le degré de préférence DOIT être calculé sur la base des informations de politique préconfigurées et utilisé comme valeur de PréférenceLocale dans toute annonce TRIP intra domaine. La nature exacte de ces informations de politique et les calculs que cela implique sont une affaire locale.

Le résultat du processus de détermination du degré de préférence est la préférence locale d'un chemin. Le LS local calcule la préférence locale des chemins appris des homologues externes ou générés en interne à ce LS. La préférence locale d'un chemin appris d'un homologue interne est incluse dans l'attribut PréférenceLocale associée à ce chemin.

10.2.2 Phase 2 : Choix du chemin

La phase 2 de la fonction de décision devra être invoquée à l'achèvement de la phase 1. La fonction de phase 2 est un processus distinct qui s'achève lorsque il n'a plus rien à faire. La phase 2 comporte deux sous phases : 2a et 2b. La même fonction de choix de chemin est appliquée dans les deux sous phases, mais les entrées dans chaque phase sont différentes. Le processus de la phase 2a DOIT considérer comme entrées tous les chemins externes qui sont présents dans la Adj-TRIBs-In des homologues externes, et tous les chemins locaux. Le résultat de la phase 2a est inséré dans la Ext-TRIB. Le processus de la phase 2b devra être invoqué à l'achèvement de la phase 2a et il DOIT considérer comme entrées tous les chemins qui sont dans la Ext-TRIB et tous les chemins qui sont présents dans la Adj-TRIBs-In des LS internes. Le résultat de la phase 2b est mémorisé dans la Loc-TRIB.

Le déroulement de la phase 2 de la fonction de décision DOIT être bloqué tant que la phase 3 de la fonction de décision est en cours d'exécution. La phase 2 de la fonction DOIT verrouiller toutes les Adj-TRIBs-In et Ext-TRIB avant de commencer à opérer, et elle DOIT les déverrouiller lorsque elle s'achève.

Si le LS détermine que le ServeurDeProchainBond figurant dans un chemin est inaccessible, le chemin PEUT alors être exclu de la phase 2 de la fonction de décision. Le moyen par lequel est réalisée une telle détermination n'est pas décrit ici.

Pour chaque ensemble de destinations pour lesquelles un ou plusieurs chemins existent, la fonction de sélection de chemin du LS local DOIT identifier le chemin qui a :

- le plus haut degré de préférence, ou
- qui est choisi par suite des règles de départage spécifiées au paragraphe 10.2.2.1.

Les chemins retirés DOIVENT être supprimés de Loc-TRIB, Ext-TRIB, et Adj-TRIBs-In.

10.2.2.1 Départage (Phase 2)

Plusieurs chemins pour la même destination qui ont le même degré de préférence peuvent être en entrée à la phase 2 de la fonction de sélection de chemin. Le LS local peut choisir un seul de ces chemins pour l'inclure dans la Ext-TRIB associée (Phase 2a) ou la Loc-TRIB (Phase 2b). Le LS local considère tous les chemins qui ont le même degré de préférence. L'algorithme suivant devra être utilisé pour le départage.

- Si le LS local est configuré pour utiliser l'attribut DécouvMultiSorties pour le départage, et si les chemins candidats reçus du même ITAD du voisinage diffèrent par la valeur de l'attribut DécouvMultiSorties, on choisit alors le chemin qui a la valeur la plus élevée de DécouvMultiSorties.
- Si au moins un des chemins a été généré par un LS interne, choisir le chemin qui a été annoncé par le LS interne qui a le plus faible identifiant TRIP.
- Autrement, choisir le chemin qui a été annoncé par le domaine voisin qui a le plus faible numéro d'ITAD.

10.2.3 Phase 3 : Dissémination de chemin

La phase 3 de la fonction de décision DOIT être invoquée à l'achèvement de la phase 2 si la phase 2 résulte en des changements de la Loc-TRIB ou lorsque une nouvelle session de LS à LS homologue est établie.

La fonction de phase 3 est un processus distinct qui s'achève lorsque tout le travail est fait. La fonction de décision d'acheminement de phase 3 DOIT être bloquée lorsque la fonction de décision de phase 2 est en cours.

Tous les chemins dans la Loc-TRIB devront être traités dans une entrée correspondante dans la Adj-TRIBs-Out associée. Les techniques d'agrégation de chemin et de réduction des informations (voir au paragraphe 10.3.4) PEUVENT facultativement être appliquées.

Lorsque la mise à jour de la Adj-TRIBs-Out est terminée, le LS local DOIT procéder à la mise à jour externe du paragraphe 10.3.2.

10.2.4 Chevauchement de chemins

Lorsque des chemins qui se chevauchent sont présents dans la même Adj-TRIB-In, le chemin le plus spécifique doit avoir la préséance, dans l'ordre, du plus spécifique au moins spécifique.

L'ensemble des destinations décrites par le chevauchement représente une portion du chemin le moins spécifique faisable, mais non actuellement utilisé. Si un chemin plus spécifique est ultérieurement retiré, l'ensemble des destinations décrites par le chemin le plus spécifique sera encore accessible en utilisant le chemin moins spécifique.

Si un LS reçoit des chemins qui se chevauchent, le processus de décision DOIT prendre en compte la sémantique des chemins en recouvrement. En particulier, si un LS accepte le chemin moins spécifique tout en rejetant le chemin le plus spécifique de même homologue, les destinations représentées par le chevauchement peuvent alors n'être plus transmises sur les domaines énumérés dans l'attribut CheminD'Annonce de ce chemin. Donc, un LS a les choix suivants :

1. Installer les deux chemins, le plus spécifique et le moins spécifique,
2. Installer seulement le chemin le plus spécifique,
3. Installer seulement la partie non chevauchante du chemin le moins spécifique (ce qui implique la désagrégation du chemin le moins spécifique)
4. Agréger les deux chemins et installer le chemin agrégé,
5. Installer seulement le chemin le moins spécifique,
6. N'installer aucun des deux chemins.

Si un LS choisit 5), il DEVRAIT alors ajouter l'attribut AgrégatAtomique au chemin. Un chemin qui porte l'attribut AgrégatAtomique NE DOIT PAS être désagrégé. C'est-à-dire que le chemin ne peut pas être rendu plus spécifique. La transmission le long d'un tel chemin ne garantit pas qu'elle traverse seulement les domaines énumérés dans le CheminSuivi du chemin. Si un LS choisit 1), il NE DOIT PAS annoncer alors le chemin moins spécifique sans le chemin plus spécifique.

10.3 Processus de Mise-à-jour-Envoi

Le processus de Mise-à-jour-Envoi est chargé d'annoncer les messages METTREÀJOUR à tous les homologues. Par exemple, il distribue les chemins choisis par le processus de décision aux autres LS qui peuvent être situés soit dans le même ITAD, soit dans un ITAD du voisinage. Les règles de l'échange d'informations entre les LS homologues situés dans des ITAD différents sont données au paragraphe 10.3.2 ; les règles de l'échange d'informations entre des LS homologues situés dans le même ITAD figurent au paragraphe 10.3.1.

Avant de transmettre les chemins aux homologues, un LS DOIT déterminer quels attributs devraient être transmis avec ce chemin. Si un attribut non transitif qui n'est pas bien connu n'est pas reconnu, il est ignoré en silence. Si un attribut transitif dépendant qui n'est pas bien connu n'est pas reconnu, et si l'attribut ServeurDeProchainBond a été changé par le LS, l'attribut non reconnu est ignoré en silence. Si un attribut transitif dépendant qui n'est pas bien connu n'est pas reconnu, et si l'attribut ServeurDeProchainBond n'a pas été modifié par le LS, le bit Partiel dans l'octet des fanions de l'attribut est réglé à 1, et l'attribut est conservé pour la propagation aux autres locuteurs TRIP. De même, si un attribut transitif indépendant qui n'est pas bien connu n'est pas reconnu, le bit Partiel dans l'octet des fanions de l'attribut est réglé à 1, et l'attribut est conservé pour la propagation aux autres locuteurs TRIP. Si un attribut qui n'est pas bien connu est reconnu, et si il a une valeur valide, alors, selon le type de l'attribut non bien connu, il est mis à jour, si nécessaire, pour une propagation possible aux autres locuteurs TRIP.

10.3.1 Mises à jour internes

Le processus de mise à jour interne concerne la distribution des informations d'acheminement aux homologues internes.

Lorsque un LS reçoit un message METTREÀJOUR d'un autre LS TRIP situé dans son propre ITAD, il est arrosé comme décrit au paragraphe 10.1.

Lorsque un LS reçoit un nouveau chemin d'un LS d'un ITAD du voisinage, ou si un chemin local est injecté dans TRIP, le LS détermine les préférences de ce chemin. Si le nouveau chemin a le plus fort degré de préférence pour tous les chemins externes et les chemins locaux pour une certaine destination (ou si le chemin a été choisi via une procédure de départage comme spécifié au paragraphe 10.3.1.1) le LS DOIT insérer ce nouveau chemin dans la base de données Ext-TRIB et le LS DOIT annoncer ce chemin à tous les autres LS qui sont dans son ITAD au moyen d'un message METTREÀJOUR. Le LS DOIT s'annoncer lui-même comme générateur de ce chemin au sein de l'ITAD.

Lorsque un LS reçoit un message METTREÀJOUR avec un attribut CheminsRetirés non vide de la part d'un homologue externe, ou si un chemin local est retiré de TRIP, le LS DOIT retirer de son Adj-TRIB-In tous les chemins dont les destinations étaient portées dans ce champ. Si le chemin retiré avait été précédemment choisi dans Ext-TRIB, le LS DOIT effectuer les étapes supplémentaires suivantes :

- Si un nouveau chemin est choisi pour être annoncé pour toutes ces destinations, le LS DOIT alors insérer le chemin de remplacement dans Ext-TRIB pour remplacer le chemin retiré et l'annoncer à tous les LS internes.
- Si un chemin de remplacement n'est pas disponible pour annonce, le LS DOIT alors inclure les destinations du chemin dans l'attribut CheminsRetirés d'un message METTREÀJOUR, et DOIT envoyer ce message à chaque homologue interne. Le LS DOIT aussi supprimer le chemin retiré de Ext-TRIB.

10.3.1.1 Départage (Chemins reçus d'homologues externes)

Si un LS a des connexions avec plusieurs homologues externes, il y aura plusieurs Adj-TRIBs-In associées à ces homologues. Ces bases de données peuvent contenir plusieurs chemins également préférables pour la même destination, qui ont toutes été annoncées par des homologues externes. Le LS local devra choisir un de ces chemins conformément aux règles suivantes :

- Si le LS est configuré pour utiliser l'attribut DécouvMultiSorties pour faire le départage, et si les chemins candidats diffèrent de la valeur de l'attribut DécouvMultiSorties, choisir alors le chemin qui a la plus forte valeur de DécouvMultiSorties, autrement,
- Choisir le chemin qui a été annoncé par le LS externe qui a le plus faible numéro TRIP.

10.3.2 Mises à jour externes

Le processus de mise à jour externe concerne la distribution des informations d'acheminement aux homologues externes. Au titre de la phase 3 du processus de choix de chemin, le LS a mis à jour son Adj-TRIBs-Out. Tous les chemins nouvellement installés et tous les chemins nouvellement impraticables pour lesquels il n'y a pas de chemin de remplacement DOIVENT être annoncés aux homologues externes au moyen de messages METTREÀJOUR.

Tout chemin dans Loc-TRIB marqué comme retiré DOIT être supprimé. Les changements de destinations accessibles au sein de son propre ITAD DEVRONT aussi être annoncés dans un message METTREÀJOUR.

10.3.3 Contrôle de la redondance du trafic d'acheminement

Le protocole TRIP restreint la quantité de trafic d'acheminement (c'est-à-dire, de messages METTREÀJOUR) afin de limiter à la fois la bande passante de liaison nécessaire pour annoncer les messages METTREÀJOUR et la puissance de traitement nécessaire au processus de décision pour résumer les informations contenues dans les messages METTREÀJOUR.

10.3.3.1 Fréquence des annonces de chemin

Le paramètre MinRouteAdVERTISEMENTInterval (*intervalle minimum d'annonce de chemin*) détermine la quantité minimale de temps qui doit s'écouler entre les annonces de chemins pour une certaine destination pour un seul LS. Cette procédure de limitation du taux d'annonce s'applique destination par destination, bien que la valeur de MinRouteAdVERTISEMENTInterval soit établie sur la base de l'homologue LS.

Deux messages METTREÀJOUR envoyés d'un seul LS qui annonce des chemins praticables à un ensemble commun de destinations reçues d'homologues externes DOIVENT être séparés par au moins MinRouteAdVERTISEMENTInterval. En clair, ceci ne peut être réalisé précisément en tenant des temporisateurs distincts pour chaque ensemble commun de destinations. Cela serait une charge intolérable. Toute technique qui assure que l'intervalle entre deux messages METTREÀJOUR envoyés d'un même LS qui annonce les chemins praticables à un ensemble commun de destinations reçues des homologues

externes sera d'au moins `MinRouteAdvertisementInterval`, et assurera aussi qu'une limite supérieure constante de l'intervalle est acceptable.

Deux messages `METTREÀJOUR`, envoyés d'un même LS à un homologue externe, qui annonce les chemins accessibles pour un ensemble commun de destinations reçues des homologues internes DOIVENT être séparés par au moins `MinRouteAdvertisementInterval`.

Comme une convergence rapide est nécessaire au sein d'un ITAD, cette procédure de limitation de taux ne s'applique pas aux chemins reçus des homologues internes et qui sont diffusées aux autres homologues internes. Pour éviter des trous noirs de longue durée, la procédure ne s'applique pas aux retraits explicites de chemins (c'est-à-dire, aux chemins dont les destinations sont explicitement retirées par les messages `METTREÀJOUR`).

Cette procédure ne limite pas le taux de sélection de chemins, mais seulement le taux d'annonce de chemins. Si de nouveaux chemins sont choisis plusieurs fois tout en attendant l'expiration de `MinRouteAdvertisementInterval`, le dernier chemin choisi devra être annoncé à la fin de `MinRouteAdvertisementInterval`.

10.3.3.2 Fréquence de Origine de chemin

Le paramètre `MinITADOriginationInterval` détermine la quantité minimum de temps qui doit s'écouler entre des annonces successives de messages `METTREÀJOUR` qui rapportent des changements au sein du propre ITAD du LS annonceur.

10.3.3.3 Gigue

Pour minimiser la probabilité que la distribution des messages TRIP par un certain LS contienne des crêtes, la gigue devrait être appliquée aux temporisateurs associés à `MinITADOriginationInterval`, à `GarderEnVie`, et à `MinRouteAdvertisementInterval`. Un LS devra appliquer la même gigue à chacune de ces quantités sans considération des destinations auxquelles les mises à jour sont envoyées ; c'est-à-dire que la gigue ne sera pas appliquée homologue par homologue.

La quantité de gigue à introduire devra être déterminée en multipliant la valeur de base du temporisateur approprié par un facteur aléatoire qui soit uniformément distribué dans la gamme de 0,75 à 1,0.

10.3.4 Organisation efficace des informations d'acheminement

Ayant choisi les informations d'acheminement qu'il va annoncer, un locuteur TRIP peut utiliser des méthodes pour organiser ces informations de manière efficace. Ces méthodes sont exposées dans les paragraphes qui suivent.

10.3.4.1 Réduction d'informations

La réduction d'informations peut impliquer une réduction de la granularité du contrôle de politique – après que les informations sont compressées, les mêmes politiques s'appliqueront à toutes les destinations et tous les chemins dans la classe d'équivalence.

Le processus de décision peut facultativement réduire la quantité d'informations qu'il va placer dans la `Adj-TRIBs-Out` par une des méthodes suivantes :

- `CheminsAccessibles` : Un ensemble de destinations peut normalement être représenté en forme compacte. Par exemple, un ensemble de numéros de téléphone E.164 peut être représenté sous une forme plus compacte en utilisant les préfixes E.164.
- `CheminD'Annonce` : Les informations d'un chemin d'annonce peuvent être représentées comme des `AP_SEQUENCE` ordonnées ou des `AP_SET` non ordonnés. Les `AP_SET` sont utilisés dans l'algorithme d'agrégation de chemin décrit au paragraphe 5.4.4. Ils réduisent la taille des informations de l'`AP_PATH` en faisant la liste de chaque numéro d'ITAD une seule fois, sans considération du nombre de fois qu'ils peuvent apparaître dans plusieurs chemins d'annonce qui ont été agrégés.

Un `AP_SET` implique que les destinations annoncées dans le message `METTREÀJOUR` peuvent être joignables par des chemins qui traversent au moins certains des ITAD constituants. Les `AP_SET` donnent des informations suffisantes pour éviter des chemins en boucle ; cependant leur utilisation peut élaguer des chemins potentiellement praticables, car ces chemins ne sont plus énumérés individuellement comme dans la forme des `AP_SEQUENCE`. En pratique, il est peu probable que cela pose un problème, car une fois qu'un appel est arrivé à la bordure d'un groupe d'ITAD, le LS aura vraisemblablement des informations de chemin plus détaillées et peut distinguer les chemins individuels vers les destinations.

10.3.4.2 Agrégation des informations d'acheminement

L'agrégation est le processus de combinaison des caractéristiques de plusieurs chemins différents d'une façon telle qu'un

seul chemin puisse être annoncé. L'agrégation peut se faire au titre du processus de décision pour réduire la quantité d'informations d'acheminement qui est placée dans la Adj-TRIBs-Out.

L'agrégation réduit la quantité d'informations qu'un LS doit mémoriser et échanger avec les autres LS. Les chemins peuvent être agrégés en appliquant la procédure suivante séparément aux attributs de même type.

Les chemins qui ont les attributs qui suivent ne devront pas être agrégés si les attributs correspondants de chaque chemin ne sont pas identiques : DécouvMultiSorties, ServeurDeProchainBond.

Les attributs qui ont des codes de type différents ne peuvent pas être agrégés. Les attributs de même code de type peuvent être agrégés. Les règles pour agréger chaque attribut DOIVENT être fournies avec la définition de l'attribut. Les règles d'agrégation pour les attributs de base de TRIP, par exemple, CheminsAccessibles et CheminD'Annonce, sont données à la Section 5.

10.4 Critères de choix de chemin

D'une façon générale, les règles supplémentaires pour comparer les chemins parmi plusieurs alternatives sortent du domaine d'application du présent document. Il y a deux exceptions :

- Si l'ITAD local apparaît dans le CheminD'Annonce du nouveau chemin considéré, ce nouveau chemin ne peut alors pas être vu comme meilleur qu'aucun autre chemin. Si un tel chemin est utilisé, il peut en résulter une boucle d'acheminement (voir au paragraphe 6.3).
- Afin de réaliser un fonctionnement réparti réussi, seuls les chemins qui ont une probabilité élevée de stabilité peuvent être choisis. Donc, un ITAD doit éviter d'utiliser des chemins instables, et il ne doit pas faire de changements rapides et spontanés de ses choix de chemin. Quantifier les termes "instable" et "rapide" dans la phrase précédente demandera de l'expérience, mais le principe est clair.

10.5 Génération des choix de chemins

Un LS peut générer des chemins locaux en injectant des informations d'acheminement acquises par d'autres moyens (par exemple, via un protocole d'acheminement intra domaine, ou par configuration manuelle, ou par un mécanisme/protocole d'enregistrement dynamique) dans TRIP. Un LS qui génère des chemins TRIP devra allouer le degré de préférence à ces chemins en les passant à travers le processus de décision (voir au paragraphe 10.2). Pour TRIP, les chemins locaux sont identiques aux chemins externes et sont soumis au même mécanisme de sélection de chemin en deux phases. Un chemin local qui est choisi dans la Ext-TRIB DOIT être annoncé à tous les LS internes. La décision de savoir si il faut distribuer les chemins acquis par d'autres moyens que TRIP au sein d'un ITAD via TRIP ou non dépend de l'environnement au sein de l'ITAD (par exemple, du type de protocole d'acheminement intra domaine) et devrait être contrôlé via la configuration.

11. Transport TRIP

La présente spécification définit l'utilisation de TCP comme couche transport pour TRIP. TRIP utilise l'accès TCP 6069. Le fonctionnement de TRIP sur d'autres protocoles de transport fera l'objet d'études ultérieures.

12. Topologie d'ITAD

Il n'y a pas de restriction quant à la topologie intra-domaine des LS TRIP. Par exemple, les LS dans un ITAD peuvent être configurés en maillage complet, en étoile, ou dans toute autre topologie de connexion. De même, il n'y a pas de restriction sur la topologie des ITAD TRIP. Par exemple, les ITAD peuvent être organisés en topologie plate (maillage ou anneau) ou en hiérarchie à plusieurs niveaux ou en toute autre topologie.

La frontière entre deux ITAD TRIP peut être située soit sur la liaison entre deux LS TRIP, soit peut coïncider avec un LS TRIP. Dans ce dernier cas, le même LS TRIP sera membre de plus d'un ITAD, et il apparaît comme homologue interne aux LS dans chaque ITAD dont il est membre.

13. Considérations relatives à l'IANA

Le présent document crée un nouveau registre IANA pour les paramètres TRIP. Les paramètres TRIP suivants sont inclus dans le registre :

- Capacités TRIP
- Attributs TRIP
- Familles d'adresse TRIP
- Protocoles d'application TRIP
- Numéros d'ITAD TRIP

Les paramètres de protocole sont fréquemment initialisé/remis à 0. Le présent document réserve la valeur 0 de chacun des paramètres TRIP ci-dessus afin de distinguer clairement entre un paramètre non établi et toutes les autres valeurs enregistrées pour ce paramètre.

Les sous-registres pour chacun des paramètres ci-dessus sont exposés dans les paragraphes qui suivent.

13.1 Capacités TRIP

Les demandes d'ajout de capacités TRIP autres que celles définies au paragraphe 4.2.1.1 doivent être soumises à iana@iana.org. Suivant les politiques d'allocation de numéro exposées dans la [RFC2434], les codes de capacités dans la gamme 32768-65535 sont réservés pour utilisation privée (ce sont les codes dont le premier bit de la valeur du code est égal à 1). Le présent document réserve la valeur 0. Les codes de capacité 1 et 2 ont été alloués au paragraphe 4.2.1.1. Les codes de capacité dans la gamme de 2 à 32 767 sont contrôlés par l'IANA, et sont alloués sous la condition de spécification exigée (RFC de l'IETF ou équivalent). La spécification DOIT inclure une description de la capacité, les valeurs possibles qu'elle peut prendre, et ce qui constitue une discordance de capacité.

13.2 Attributs TRIP

Le présent document réserve les codes de type d'attribut 224-255 pour utilisation privée (ce sont les codes dont les trois premiers bits du code sont égaux à 1). Le présent document réserve la valeur 0. Les codes de type d'attribut de 1 à 11 ont déjà été alloués par ce document. Les codes de type d'attribut de 1 à 11 sont définis aux paragraphes 5.1 à 5.11.

Les codes de type d'attribut dans la gamme de 12 à 223 sont contrôlés par l'IANA, et exigent un document de spécification (RFC ou équivalent). La spécification DOIT fournir toutes les informations exigées au paragraphe 5.12 de ce document.

Les demandes d'enregistrement de code de type d'attribut doivent être envoyées à iana@iana.org. En plus de l'exigence d'une spécification, la demande DOIT inclure une indication de qui a le contrôle des modifications de l'attribut et des informations de contact (adresse postale et de messagerie).

13.3 Familles d'adresse de destination

Le présent document réserve la famille d'adresses 0. Les demandes d'ajout de familles d'adresses TRIP autres que celles définies au paragraphe 5.1.1.1 (familles d'adresses 1, 2, et 3) c'est-à-dire, dans la gamme 4-32767, doivent être soumises à iana@iana.org. La demande DOIT inclure une brève description de la famille d'adresses, son alphabet, et les règles et lignes directrices de traitement particulières, comme les lignes directrices pour l'agrégation, s'il en est. Les demandes sont soumises à revue par expert. Le présent document réserve les codes de famille d'adresses 32 768-65 535 pour des applications spécifiques des fabricants.

13.4 Protocoles d'applications TRIP

Le présent document crée un nouveau registre IANA pour les protocoles d'application TRIP. Le présent document réserve le code de protocole d'application 0. Les demandes d'ajout de protocoles d'application TRIP autres que ceux définis au paragraphe 5.1.1.1 (protocoles d'application 1 à 4) c'est-à-dire, dans la gamme 5 à 32 767, doivent être soumises à iana@iana.org. La demande DOIT inclure un bref développement sur le protocole d'application, et une description de la façon dont TRIP peut être utilisé pour annoncer les chemins pour ce protocole. Les demandes sont soumises à revue par expert. Le présent document réserve les codes de protocole d'application 32 768 à 65 535 pour des applications spécifiques de fabricants.

13.5 Numéros d'ITAD

Le présent document réserve le numéro d'ITAD 0. Les numéros d'ITAD dans la gamme 1 à 255 sont destinés à utilisation privée. Les numéros d'ITAD dans la gamme de 256 à (2**32)-1 sont alloués par l'IANA au premier qui les demande. Les demandes de numéros d'ITAD doivent être soumises à iana@iana.org. Les demandes DOIVENT inclure ce qui suit :

- Informations sur l'organisation qui va administrer l'ITAD.

- Informations de contact (adresse postale et de messagerie).

14. Considérations pour la sécurité

Cette section traite de la sécurité entre LS TRIP homologues lorsque TRIP fonctionne sur TCP dans un environnement IP.

Il est clair qu'un mécanisme de sécurité est nécessaire pour empêcher des entités non autorisées d'utiliser le protocole défini dans le présent document pour établir des sessions d'homologues non autorisés avec d'autres LS TRIP ou d'interférer avec des sessions d'homologues autorisés. Le mécanisme de sécurité pour le protocole, lorsque il est transporté sur TCP dans un réseau IP, est IPsec [RFC2401]. IPsec utilise deux protocoles pour fournir la sécurité du trafic : l'en-tête d'authentification (AH, *Authentication Header*) [RFC2402] et l'encapsulation de charge utile de sécurité (ESP, *Encapsulating Security Payload*) [RFC2406].

L'en-tête AH permet l'authentification de l'origine des données, l'intégrité sans connexion et la protection facultative contre la répétition des messages passés entre les LS homologues. L'en-tête ESP fournit l'authentification de l'origine, l'intégrité sans connexion, la protection contre la répétition, et la confidentialité des messages.

Les mises en œuvre du protocole défini dans le présent document qui emploient l'en-tête ESP DEVRONT se conformer à la section 5 de la [RFC2406] qui définit un ensemble minimum d'algorithmes de vérification de l'intégrité et du chiffrement. De même, les mises en œuvre qui emploient l'en-tête AH DEVRONT se conformer à la section 5 de la [RFC2402] qui définit un ensemble minimum d'algorithmes de vérification d'intégrité avec des clés manuelles.

Les mises en œuvre DEVRAIENT utiliser IKE [RFC2409] pour permettre des options de chiffrement plus robustes. Les mises en œuvre qui emploient IKE DEVRAIENT prendre en charge l'authentification avec les signatures RSA et le chiffrement à clé publique RSA.

Une association de sécurité (SA, *Security Association*) [RFC2401] est une "connexion" unidirectionnelle qui permet des services de sécurité pour le trafic qu'elle porte. Les services de sécurité sont donnés à une SA par l'utilisation de AH, ou ESP, mais pas des deux. Deux types de SA sont définis : mode transport et mode tunnel [RFC2401]. Une SA en mode transport est une association de sécurité entre deux hôtes, et elle est appropriée pour protéger la session TRIP entre deux LS homologues.

Appendice 1. Transitions d'état et actions de l'automate TRIP

Le présent Appendice expose les transitions entre les états dans l'automate à états finis (FSM) TRIP en réponse aux événements TRIP. Voici la liste de ces états et événements lorsque la valeur du temps de garde négocié est différente de zéro.

États TRIP :

- 1 - Repos
- 2 - Connecté
- 3 - Actif
- 4 - OuvertEnvoyé
- 5 - OuvertConfirmé
- 6 - Établi

Événements TRIP :

- 1 - Début TRIP
- 2 - Arrêt TRIP
- 3 - Connexion de transport TRIP ouverte
- 4 - Connexion de transport TRIP fermée
- 5 - Échec d'ouverture de connexion de transport TRIP
- 6 - Erreur fatale de transport TRIP
- 7 - Expiration du temporisateur RéessaiDeConnexion
- 8 - Expiration du temporisateur de garde
- 9 - Expiration du temporisateur GarderEnVie
- 10 - Message OUVERT reçu
- 11 - Message GARDERENVIE reçu
- 12 - Message MISEÀJOUR reçu
- 13 - Message NOTIFICATION reçu

Le tableau suivant décrit les transitions d'état du FSM TRIP et les actions déclenchées par ces transitions.

Événement	Actions	Message envoyé	Prochain état
Repos (1)			
1	Initialise les ressources Lance le temporisateur RéessaiDeConnexion Initie une connexion de transport	aucun	2
autres	aucune	aucun	1
Connecté (2)			
1	aucune	aucun	2
3	Achève l'initialisation Libère temporisateur RéessaiDeConnexion	OUVERT	4
5	Relance temporisateur RéessaiDeConnexion	aucun	3
7	Redémarre temporisateur RéessaiDeConnexion Initie une connexion de transport	aucun	2
autres	Libère les ressources	aucun	1
Actif (3)			
1	aucune	aucun	3
3	Achève l'initialisation Libère le temporisateur RéessaiDeConnexion	OUVERT	4
5	Clôt la connexion Redémarre le temporisateur RéessaiDeConnexion		3
7	Redémarre le temporisateur RéessaiDeConnexion Initie une connexion de transport	aucun	2
autres	Libère les ressources	aucun	1
OuvertEnvoyé (4)			
1	aucune	aucun	4
4	Clôt la connexion de transport Redémarre le temporisateur RéessaiDeConnexion	aucun	3

6	Libère les ressources	aucun	1
10	Le processus OUVERT est OK	GARDERENVIE	5
	Le processus OUVERT a échoué	NOTIFICATION	1
autres	Clôt la connexion de transport	NOTIFICATION	1
	Libère les ressources		
OuvvertConfirmé (5)			
1	aucun	aucun	5
4	Libère les ressources	aucun	1
6	Libère les ressources	aucun	1
9	Redémarre le temporisateur GarderEnVie	GARDERENVIE	5
11	Achève l'initialisation	aucun	6
	Redémarre le temporisateur de garde		
13	Clôt la connexion de transport		1
	Libère les ressources		
autres	Clôt la connexion de transport	NOTIFICATION	1
	Libère les ressources		
Établi (6)			
1	aucun	aucun	6
4	Libère les ressources	aucun	1
6	Libère les ressources	aucun	1
9	Redémarre le temporisateur GarderEnVie	GARDERENVIE	6
11	Redémarre le temporisateur de garde	aucun	6
12	Le processus METTREÀJOUR est OK	METTREÀJOUR	6
	Le processus METTREÀJOUR a échoué	NOTIFICATION	1
13	Clôt la connexion de transport		1
	Libère les ressources		
autres	Clôt la connexion de transport	NOTIFICATION	1
	Libère les ressources		

Voici une version condensée du tableau de transition d'états ci-dessus.

Événements	Repos (1)	Connecté (2)	Actif (3)	OuvvertReçu (4)	OuvvertConfirmé (5)	Établi (6)
1	2	2	3	4	5	6
2	1	1	1	1	1	1
3	1	4	4	1	1	1
4	1	1	1	3	1	1
5	1	3	3	1	1	1
6	1	1	1	1	1	1
7	1	2	2	1	1	1
8	1	1	1	1	1	1
9	1	1	1	1	5	6
10	1	1	1	1 ou 5	1	1
11	1	1	1	1	6	6
12	1	1	1	1	1	1 ou 6
13	1	1	1	1	1	1

Appendice 2 Recommandations pour la mise en œuvre

La présente section présente des recommandations de mise en œuvre.

A.2.1 Plusieurs réseaux par message

Le protocole TRIP permet que soient spécifiés dans un seul message plusieurs préfixes d'adresse avec le même chemin d'annonce et le même serveur de prochain bond. Il est très recommandé de faire usage de cette capacité. Avec un préfixe d'adresse par message, il y a une augmentation substantielle de la charge chez le receveur. Non seulement la redondance du système augmente à cause de la réception de multiples messages, mais le travail d'examen du tableau d'acheminement pour les mises à jour des homologues TRIP est aussi à prendre en charge à chaque fois. Une méthode pour construire des messages contenant de nombreux préfixes d'adresse par chemin d'annonce et prochain bond à partir d'un tableau

d'acheminement qui n'est pas organisé par chemin d'annonce est de construire de nombreux messages lors de l'examen du tableau d'acheminement. Lorsque chaque préfixe d'adresse est traité, un message est alloué pour le chemin d'annonce et le prochain bond associés, si il n'existe pas, et le nouveau préfixe d'adresse lui est ajouté. Si un tel message existe, le nouveau préfixe d'adresse lui est juste accolé. Si le message manque d'espace pour contenir le nouveau préfixe d'adresse, il est transmis, un nouveau message est alloué, et le nouveau préfixe d'adresse est inséré dans le nouveau message. Lorsque la totalité du tableau d'acheminement a été examinée, tous les messages alloués sont envoyés et leurs ressources libérées. La compression maximale est réalisée lorsque toutes les destinations couvertes par les préfixes d'adresses partagent le même serveur de prochain bond et les attributs communs, rendant possible l'envoi de nombreux préfixes d'adresse dans un message de 4096 octets.

Lors d'un échange de trafic avec une mise en œuvre de TRIP qui ne compresse pas plusieurs préfixes d'adresse en un message, il peut être nécessaire de prendre des mesures pour réduire la charge résultant de l'arrosage des données reçues lors de l'acquisition d'un homologue ou lorsque survient un changement significatif de la topologie du réseau. Une méthode pour ce faire est de limiter le taux de mise à jour. Cela va éliminer l'examen redondant du tableau d'acheminement pour fournir des mises à jour éclair pour les homologues TRIP. Un inconvénient de cette approche est qu'elle augmente la latence de propagation des informations d'acheminement. En choisissant un intervalle minimum de mise à jour éclair qui ne soit pas très supérieur au temps que prend le traitement des messages multiples, cette latence devrait être minimisée. Une meilleure méthode serait de lire tous les messages reçus avant d'envoyer les mises à jour.

A.2.2 Traitement des messages sur un protocole de flux

TRIP utilise TCP comme mécanisme de transport. Du fait de la nature par flux de TCP, toutes les données d'un message reçu n'arrivent pas nécessairement au même moment. Cela peut rendre difficile de traiter les données comme des messages, en particulier sur des systèmes où il n'est pas possible de déterminer comment beaucoup de données ont été reçues mais pas encore traitées.

Une méthode qui peut être utilisée dans cette situation est d'abord d'essayer de lire juste l'en-tête du message. Pour le type de message GARDERENVIE, c'est un message complet ; pour les autres types de message, l'en-tête devrait d'abord être vérifié, en particulier sa longueur totale. Si toutes les vérifications réussissent, la longueur spécifiée, moins la taille de l'en-tête du message est la quantité de données qui reste à lire. Une mise en œuvre qui voudrait "serrer" le processus d'information d'acheminement tout en essayant de lire ce qu'elle reçoit d'un homologue pourrait installer une mémoire tampon de message (4096 octets) par homologue et la remplir des données lorsqu'elles sont disponibles jusqu'à ce qu'un message complet ait été reçu.

A.2.3 Réduction du flottement de chemin

Pour éviter un flottement de chemin excessif, un LS qui a besoin de retirer une destination et d'envoyer une mise à jour sur un chemin plus spécifique ou moins spécifique DEVRAIT les combiner dans le même message METTREÀJOUR.

A.2.4 Temporisateurs TRIP

TRIP emploie plusieurs temporisateurs : RéessaiDeConnexion, TempsDeGarde, GarderEnVie, MaxDélaiPurge, DélaiTripDésactivé, MinITADOriginationInterval, et MinRouteAdvertisementInterval. La valeur suggérée pour le temporisateur RéessaiDeConnexion est de 120 secondes. La valeur suggérée pour le TempsDeGarde est de 90 secondes. La valeur suggérée pour le temporisateur GarderEnVie est 30 secondes. La valeur suggérée pour le temporisateur MaxDélaiPurge est 10 secondes. La valeur suggérée pour le temporisateur DélaiTripDésactivé est 180 secondes. La valeur suggérée pour MinITADOriginationInterval est 30 secondes. La valeur suggérée pour MinRouteAdvertisementInterval est 30 secondes.

Une mise en œuvre de TRIP DOIT permettre que ces temporisateurs soient configurables.

A.2.5 Tri sur AP_SET

Une autre optimisation utile qui peut être faite pour simplifier cette situation est de trier les numéros d'ITAD trouvés dans un AP_SET. Cette optimisation est entièrement facultative.

Remerciements

Nous souhaitons remercier ici Dave Oran pour la perspicacité de ses commentaires et suggestions.

Références

- [H.323] Union Internationale des Télécommunications Secteur de la normalisation, Recommandation H.323, "Systèmes de communication multi supports fondés sur le paquet", version 3. Genève, Confédération Helvétique, novembre 2000.
- [IS-IS] Projet de norme ISO DP 10589, "Protocole d'échange d'informations d'acheminement intra domaine de système intermédiaire à système intermédiaire à utiliser en conjonction avec le protocole de fourniture de services réseau en mode sans connexion (ISO 8473)", février 1990.
- [RFC1123] R. Braden, éditeur, "Exigences pour les hôtes Internet – [Application et prise en charge](#)", STD 3, octobre 1989.
- [RFC1771] Y. Rekhter, T. Li, "Protocole de routeur frontière v. 4 (BGP-4)", mars 1995. (*Obsolète, voir [RFC4271](#)*) (*D.S.*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2328] J. Moy, "[OSPF version 2](#)", STD 54, avril 1998.
- [RFC2334] J. Luciani et autres, "Protocole de [synchronisation d'antémémoire de serveur](#) (SCSP) - NBMA", avril 1998. (*P.S.*)
- [RFC2373] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", juillet 1998. (*Obsolète, voir [RFC3513](#)*) (*P.S.*)
- [RFC2401] S. Kent et R. Atkinson, "[Architecture de sécurité](#) pour le protocole Internet", novembre 1998. (*Obsolète, voir [RFC4301](#)*)
- [RFC2402] S. Kent et R. Atkinson, "En-tête d'authentification IP", novembre 1998. (*Obsolète, voir [RFC4302](#), [4305](#)*)
- [RFC2406] S. Kent et R. Atkinson, "Encapsulation de [charge utile de sécurité](#) IP (ESP)", novembre 1998. (*Obsolète, voir [RFC4303](#)*)
- [RFC2409] D. Harkins et D. Carrel, "L'échange de clés Internet (IKE)", novembre 1998. (*Obsolète, voir la [RFC4306](#)*)
- [RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre, 1998. (*Rendue obsolète par la RFC 5226*)
- [RFC2543] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP : protocole d'initialisation de session", mars 1999. (*Obsolète, voir [RFC3261](#), [RFC3262](#), [RFC3263](#), [RFC3264](#), [RFC3265](#)*) (*P.S.*)
- [RFC2871] J. Rosenberg, H. Schulzrinne, "[Cadre pour l'acheminement de la téléphonie](#) sur IP", juin 2000. (*Information*)

Notice de propriété intellectuelle

L'IETF ne prend aucune position sur la validité ou la portée d'aucun droit de propriété intellectuelle ou d'autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou non disponible ; pas plus qu'elle ne prétend qu'elle ait fait aucun effort pour identifier de tels droits. Des informations sur les procédures de l'IETF au sujet des droits dans la documentation en cours de normalisation et en rapport avec les normes peuvent être trouvées dans le BCP-11. Des copies des revendications de droits peuvent être disponibles à la publication et toutes les assurances de licences peuvent être rendues disponibles, ou le résultat de tentatives d'obtention d'une licence ou permission générale pour l'utilisation de tels droits de propriété par les mises en œuvre ou utilisateurs de la présente spécification peuvent être obtenus auprès du secrétariat de l'IETF.

L'IETF invite toute partie intéressée à porter son attention sur tous droits de reproduction, brevets ou applications de brevets, ou autres droits de propriété pouvant couvrir cette technologie qui pourraient être nécessaires pour mettre en pratique la présente norme. Prière d'adresser les informations au Directeur Général de l'IETF.

Une revendication de droits de propriété intellectuelle a été notifiée à l'IETF à l'égard de tout ou partie de la spécification contenue dans le présent document. Pour d'autres informations, consulter la liste en ligne des revendications de droits.

Adresses des auteurs

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936
téléphone : 973-952-5000
mél : jdrosen@dynamicsoft.com

Hussein F. Salama
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
téléphone : 408-527-7147
mél : hsalama@cisco.com

Matt Squire
Hatteras Networks
639 Davis Drive
Suite 200
Durham, NC 27713

Déclaration complète de droits de reproduction

Copyright (c) The Internet Society (2002). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de copyright ci-dessus et le présent et paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society, ses successeurs ou ayant droits.

Le présent document et les informations y contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Remerciement

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